Diagonalizing Random Matrices with Integrable Systems

by

Christian W. Pfrang

Dipl.–Math., Freie Universität Berlin, 2006

M.Sc., Brown University, Providence, RI, 2009

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in The Division of Applied Mathematics at Brown University

PROVIDENCE, RHODE ISLAND

May 2011

This dissertation by Christian W. Pfrang is accepted in its present form by The Division of Applied Mathematics as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date_____ _____

Govind Menon, Ph.D., Advisor

Recommended to the Graduate Council

Date_____ _____

Percy Deift, Ph.D., Reader

Date_____ _____

Matthew Harrison, Ph.D., Reader

Approved by the Graduate Council

Date_____ _____

Peter M. Weber, Dean of the Graduate School

Vitae

- **Personal Information:**
  - German Citizen
  - Born 2nd of April 1982 in Schweinfurt, Bavaria, Germany
- **Education:**
  - Brown University, M.S. in Applied Mathematics, May 2009
  - Freie Universität Berlin, Diplom in Mathematics, May 2006
- **Experience:**
  - Summer Associate at Morgan Stanley, 2010
  - Summer Associate at Boston Consulting Group, 2006
- **Awards:**
  - ERP Scholarship of the German Ministry of Economy and Technology
  - Member of the German National Acadamic Foundation
  - China Scholarship of the Alfried Krupp von Bohlen und Halbach–Stiftung
  - Max–Weber Scholarship of the Bavarian Government
- **Publication:**
  - Christian W. Pfrang, Karsten Matthies, *Small planar travelling waves in two–dimensional networks of coupled oscillators*, Dynamical Systems 24:2, 2008, pp. 157–170

Acknowledgements

Für meine Eltern, Gerdi und Otmar

This thesis would have been impossible to write without the support from Maria and my family. I would like to thank them for their love and their constant belief in my abilities.

I am deeply indebted to Govind Menon, who has been a terrific advisor. He put me on the right track after a period of scientific disorientation. Govind's ability to come up with the right questions and his breadth of interests are a continuing source of inspiration to me. I would also like to express my thanks for his, Anjali's and Gayatri's hospitality over the last two years.

I am grateful for Percy Deift's interest in my work. He made it possible for me to take his course on Riemann–Hilbert problems and to attend the Random Matrix workshop at MSRI. He was very available for any questions that I had about his work. His deep mathematical understanding and his sense for the practical applications of complete integrability to numerical analysis problems motivate me to continue to become a better mathematician.

I would also like to thank Matt Harrison for several important comments he made that helped me to improve major aspects of this thesis. With him I would like to thank the Pattern Theory community at Brown, especially Stuart Geman and Basilis Gidas. I think of myself as a former member of the Pattern Theory group

and nothing has shaped my view about good problems more than the interaction with this group.

Abstract of " Diagonalizing Random Matrices with Integrable Systems " by Christian W. Pfrang, Ph.D., Brown University, May 2011

This thesis is concerned with the application of a certain class of eigenvalue algorithms to random matrix initial data. In particular we study numerically the time to first deflation ('runtime') as well as the subdiagonal index at which deflation occurs as random variables driven by random initial data.

For the QR and the Toda algorithm we show numerically that initial data with a Wigner–semicircle spectral limit universally leads to exponential (Gaussian) right tails of the corresponding runtime distributions. Moreover, if the runtime distributions are centered and scaled to mean zero and variance one, the shapes of the resulting distributions are universal for both algorithms. We demonstrate that for non Wigner–type initial data the same scaling behavior does not necessarily exist.

We investigate the dependence of the mean and standard deviations of the runtime distribution for the QR and the Toda on matrix size n and deflation tolerance algorithm and find that the QR runtimes show little dependence on n but an approximately linear dependence on $\log \epsilon$.

In addition to these well known algorithms we describe an algorithm that reliably deflates Hermite–1 initial matrices in the middle. This is interesting in the light of recent divide and conquer algorithms for the eigenvalue problem.

We also present theoretical results on the scale of the minimal subdiagonal entry of Hermite–1 distributed Jacobi matrices and derive the Liouville equation for the evolution of this density under the Toda flow. The very last section mentions a result concerning the behavior of the inverse spectral map for Jacobi matrices near the boundary of the positive orthant of the sphere.

Contents

List of Tables

List of Figures

Chapter One

Introduction

1.1 Motivation and Summary

In this thesis we present, among other empirical results, a study of the distributions of runtimes of certain deterministic eigenvalue algorithms when applied to random initial matrices. Computing eigenvalues is one of the core problems in modern applied mathematics and we currently possess versatile numerical libraries which implement a universe of ideas that serve to address this purpose in a wide range of circumstances [2]. The availability of these practical implementations can mask the fact that the current state of the art in eigenvalue algorithms still leaves room for practical improvements as well as for results that increase our theoretical understanding.

An example for recent progress in the design of eigenvalue algorithms is the creation of randomized algorithms for computing eigenvalues and eigenvectors which are very well adapted to the types of problems that the data mining community is interested in solving [30]. Another example of important work from the recent period is the development of communication minimizing eigenvalue algorithms [4]. The creation of these algorithms is a reaction to the shift in the type of computational bottleneck that modern high performance computing is facing: No longer is the speed with which individual processors in a computing cluster can execute an instruction the constraining factor. CPUs have become so fast that the real limit on computational speed is imposed by the bandwith of the information exchange between the processors of a cluster – in order to get the most out of current hardware, eigenvalue computations have to be set up so that to the largest extent possible processors can act indepently of one another, ie. minimize communication.

On the other hand some fundamental questions about even the oldest eigenvalue algorithms remain: Shifting strategies are used in the practical implementation of the QR algorithm (cf. section 2.4) because they can raise the speed of convergence from linear in the standard QR algorithm to cubic in some shifted variants. Un-

fortunately, the dynamics of these shifting strategies are poorly understood and for the well known ones we know of matrices that lead to slower convergence or no convergence at all. In practice, this problem is countered by patching the latest implementation to deal with the newly discovered special cases [17]. It would be very desirable to get a better understanding of shifting and possibly come up with a universally efficient strategy.

The title of this thesis already hints at three areas of applied mathematics at whose intersection the current work is situated: We will draw on ideas and results from Numerical Linear Algebra, Hamiltonian Dynamics and Random Matrices to elucidate what may be called 'typical' behavior of a certain class of eigenvalue algorithms, which includes the most popular (the QR algorithm), but not its most practical variants (shifted QR algorithms).

The idea of studying the typical behavior of numerical algorithms goes back to the 1970s and and 1980s (see [16] and the references therein). Numerical analysts realized that the algorithms they were using were working remarkably well on 'most' matrix problems they needed to solve. In some sense, they realized, 'typical' problems avoid the complications associated with specific pathological examples they knew would create problems for their algorithms. [16] formalizes these thoughts by noting that the inverse of the condition numbers is a measure for the distance of any matrix from a pathological set and equipping all matrices with some 'natural' probability distribution. He shows that the probability of small neighbourhoods around the pathological set in the inverse condition number 'metric' is very small, which corroborates theoretically the observed wellbehavedness. Other important progress in this vein was made in [20], where the distribution of the condition number was computed for certain plausible distributions. Another approach to understanding this phenomenon is called 'Smoothed Analysis' [48]. Here the basic idea is that if we want to take into account numerical error due to floating point arithmetic, then

we should be computing runtime averages over slight random perturbations of the initial data rather than concern ourselves with worst case runtime analysis. This average runtime can be significantly faster than the worst case runtime and provides a more realistic assessment of the quality of an algorithm.

The concept of combining the study of algorithm complexity with random initial matrices thus has a rather successful history. The obvious problem with the approach is that it is not clear what distributions should be chosen on matrices so that the resulting class of random matrices represents typical matrices encountered in practice. In the current work we will be concerned mostly with Wigner type matrices or GOE and related matrices (like the Hermite–1 ensemble described in [19]), which are very well studied and natural ensembles. For certain application domains, however, other matrix distributions are more appropriate, like the Wishart distribution for covariance matrices. How to generate matrix ensembles with even more general spectral distributions was described for example in [47]. Interesting distributions on Jacobi matrices which generalize the Hermite–1 ensemble are the β–ensembles studied in [19]. Our focus on the classical invariant GOE ensemble as well as Wigner type matrices – which show the same limiting spectral distribution – is due to the fact that both the eigenvalue algorithms we want to study as well as the distribution of certain eigenvalue statistics (both in the bulk and on the edge) admit a completely integrable structure (cf. [15, 31], and section 5.1). The question has been raised in [3] whether eventually one will be able to analyze the interaction between these two Hamiltonian structures in a more detailed fashion.

The realization that there is a completely integrable Hamiltonian dynamical system on Jacobi matrices that is intimately connected with the QR algorithm traces back to [42, 49, 50]. The sequence of papers [13, 14, 15, 43] generalizes this notion to a whole class of completely integrable Hamiltonian systems, some of them defined on full symmetric matrices. Describing the symplectic structure even in the case of

Jacobi matrices is somewhat technical (cf. sections 4.2 and 4.3), but the differential equation associated with a Hamiltonian $H_G : L \mapsto \mathrm{tr}\hat{G}(L)$ through this is symplectic structure is simple and elegant:

$$\begin{cases} \dot{L} = [G(L)_+ - G(L)_+^T, L] \\ L(0) = L_0 \end{cases} \tag{1.1}$$

where $L(t)$ denotes a trajectory of Jacobi (or symmetric) matrices, $\hat{G}$ can be a rather general differentiable function on the real line, $G = \hat{G}'$, $G(L) = QG(\Lambda)Q^T$ for $L = Q\Lambda Q^T$, M_+ is the strictly lower triangular part of the matrix M and the bracket is the standard Lie bracket. The time one map of each of these systems can be written explicitly in terms of the QR algorithm (cf. section 5.1). Each of the systems converges to a diagonal matrix with the eigenvalues on the diagonal [42], so they can be construed as eigenvalue algorithms, which leads to the statement that "the choice of an algorithm is ... equivalently a choice of a vectorfield" [43] ie. a Hamiltonian $\hat{G}$.

Our goal in this thesis is to study empirically through simulation the statistics of key properties of some of the algorithms contained in the family described by (1.1). The two properties we'll study are connected through the concept of deflation: It occurs, when a complete offdiagonal block of the given matrix becomes so small through the action of an eigenvalue algorithm that the matrix can be regarded as 'decoupled' into its (appropriate) upper and lower principal minors without sacrificing much accuracy in the further computation of eigenvalues, which is now conducted on the

two matrices separately:

$$\left(\begin{array}{ccc|ccc} m_{11} & \dots & m_{1k} & & < \epsilon & \\ \vdots & \ddots & \vdots & & & \\ m_{k1} & \dots & m_{kk} & & & \\ \hline & & & m_{k+1,k+1} & \dots & m_{k+1,n} \\ & & & \vdots & \ddots & \vdots \\ < \epsilon & & & m_{n,k+1} & \cdots & m_{nn} \end{array}\right) \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} m_{11} & \dots & m_{1k} \\ \vdots & \ddots & \vdots \\ m_{k1} & \dots & m_{kk} \end{pmatrix}, \begin{pmatrix} m_{k+1,k+1} & \dots & m_{k+1,n} \\ \vdots & \ddots & \vdots \\ m_{n,k+1} & \cdots & m_{nn} \end{pmatrix} \quad (1.2)$$

In practical terms this is done to increase the speed of computation by sacrificing very little accuracy. The speed increases, because the computational procedures required to compute the eigenvalues are carried out on subsequently smaller and smaller matrices. In practice, at least for typical implementations of the (shifted) QR algorithm, matrices are deflated exclusively at the bottom end of the matrix. In this case the appropriate principal minor is either a scalar (the lower right entry of the matrix) and thus itself an approximation to an eigenvalue, or a 2×2–matrix, the eigenvalues of which can be easily computed and serve as approximations to two of the eigenvalues of the larger matrix.

Now after fixing an initial distribution, a Hamiltonian, a 'deflation tolerance' ϵ and a matrix dimension n, the first quantity that we simulate is what we call the runtime, deflation time or hitting time $T_\epsilon(L_0)$, namely the minimal amount of time it takes an algorithm to advance the initial matrix L_0 into a state where it is possible to deflate the matrix. The second quantity is the deflation or hitting index I_ϵ, namely the subdiagonal index at which the matrix deflates.

We will see that, depending on the chosen Hamiltonian/ Algorithm, there seems to

exist a certain scale invariance of the runtime distributions for a given initial ensemble across matrix dimensions and deflation tolerances. If we collect samples of the runtime distributions for a range of matrix size and deflation tolerance combinations and scale the resulting empirical distributions to have mean zero and variance one, then for given algorithm and initial distribution the resulting 'normalized' histograms have the same general shape and in particular, the same tails on the right side.

These normalized histograms are very close to one another across the Wigner–type initial distributions, which leads us to observe of a form of universality, where the universality of the spectral distribution carries over to quantities associated with completely integrable ODEs driven by these matrix ensembles as initial data. We apply a statistical methodology developed in [8] to show that the tails of the QR runtimes coincide across all the Wigner–type initial distributions we study as well as that the tails of the Toda runtimes coincide across all these initial distributions. We observe exponential tails in the former case and Gaussian tails in the latter. We also study these algorithms on non Wigner–type initial data and contrast our results in these cases with the results for the Wigner–type initial data case.

As for the distribution of the deflation indices, we observe that for the types of initial data that we study, depending on the algorithm chosen, varied patterns occur: For the QR algorithm the probability of first deflation occuring at a particular subdiagonal index is concentrated at the lower end of the matrix, rapidly decreasing from the lowest subdiagonal entry. For the Toda algorithm we observe that deflations are highly likely to occur both at the top and the bottom of the matrix, with the probability of first deflation appearing at a given index decreasing rapidly towards the middle of the matrix for Wigner–type initial data.

In terms of practical applications it is interesting to note that we succeeded in devising a Hamiltonian which has a extremely high probability of deflating an initial matrix sampled from one of our initial distributions exactly in the middle. Compare

this to the recent randomized divide and conquer algorithms mentioned in [4].

1.2 Outline of the Thesis

This thesis is organized in eight chapters. In the current section we give the reader an overview of the respective contents of the following seven chapters and outline the narrative of the text. In short, chapters two to five provide background information about the three areas we discussed in section 1.1. This background information consists in most cases of elaborations of known facts and statements, but we include some pertinent results of our own where appropriate.

Chapter six deals with the practicalities of our sampling infrastructure and leads into the main focus of the current thesis in chapter seven. This chapter presents our numerical results and the statistical analysis summarized in the last paragraphs of section 1.1. In chapter eight we formulate some conclusions and possible avenues for further investigation.

Chapter Two – Numerical Linear Algebra This chapter summarizes the ideas and concepts behind the well known QR algorithm and its shifted variants. We explain the QR decomposition and discuss an implementation for symmetric matrices. We then provide a geometric intuition for understanding the QR algorithm through simultaneous iterations by paraphrasizing an argument of Watkins [52]. We prove some basic properties of QR and discuss some practicalities drawn from [17, 26]. Lastly we discuss the shifted QR algorithm.

Chapter Three – Random Matrices Here we introduce the initial ensembles that we use in our study. When discussing the Hermite–1 ensemble on tridiagonal matrices we mention a few properties of Jacobi (ie. tridiagonal symmetric) matrices and we mention the spectral coordinates for these matrices. Next, we briefly discuss the Wigner semicircle law and some important results on universality and give some pointers to literature discussing the connection with complete integrability.
In the second section we state and prove a theorem concerning the minimal subdiagonal entry of Hermite–1 distributed Jacobi matrices. This result is crucial for the design of the sampling procedure later on in the text.

Chapter Four – Hamiltonian Systems We briefly describe the basic abstract setting for Hamiltonian dynamics on symplectic manifolds after which we move on to describe the most important source of symplectic manifold for the current text, namely coadjoint orbits with their Lie–Poisson (also called Kostant–Kirillov) brackets. We specialize to the Jacobi matrices and compute: A coordinate representation of their Kostant–Kirillov brackets, an explicit formula for the Hamiltonian vector-field associated with a given Hamiltonian as well as a coordinate representation of the Poisson bracket associated with the Kostant–Kirillov symplectic structure.
Section four of this chapter recalls the fundamentals of the Toda lattice and Flaschka's variables, a coordinate transform that plays a crucial role for the Toda lattice. Next we investigate the connections between Flaschka's variables and the Kostant–Kirillov symplectic structure on Jacobi matrices: First we realize that the pullback of the standard symplectic structure under Flaschka's variables is a multiple of the Kostant–Kirillov form. Then we compute the pullback of the standard matric under Flaschka's variables. We can compute gradients using this pulled back metric and plug them into the pulled back symplectic structure to obtain a Poisson bracket. It turns out that the Poisson bracket computed in this way is a multiple of the Poisson bracket

computed using the abstract approach.

Chapter Five – Hamiltonian Eigenvalue Algorithms In this chapter we apply the theory from chapter four to completely integrable Hamiltonian systems on Jacobi matrices that are related to eigenvalue algorithms. We recall some basic facts about the Toda algorithm (the fundamental example for these algorithms) and go on to introduce what we have come to call the Deift–Li–Nanda–Tomei (DLNT) famework for eigenvalue algorithms. We prove a proposition about the complete integrability of all the algorithms contained in this framework in the Jacobi matrix case and provide representations of the Toda algorithm, the QR algorithm and the shifted QR algorithm in the DLNT framework.

The second section in this chapter derives an explicit representation of the density of Jacobi matrices transported using the Toda flow with initial Hermite–1 distribution and we give the Liouville equation for these densities as they evolve under the Toda flow. On the way we compute the Jacobian of the time–t map of the Toda flow in several coordinate systems as well as the Jacobian of Flaschka's transformation.

Chapter Six – Data Collection Here we discuss the practicalities of setting up our sampling procedure. We go into details about the numerical procedures we implemented (most interesting is perhaps the Rutishauser–Kahan–Pal–Walker algorithm for reconstructing Jacobi matrices from spectral data) and we discuss some of the software design issues involved in setting up the sampling and the data handling procedures.

Chapter Seven – Numerical Results As mentioned in section 1.1 this chapter is the most important one in the current presentation. In the first section we give a

short description of the data set we generated and discuss how we set up the actual experiments. We next go into the empirical results for the runtime behavior of the QR algorithm for the four 'Wigner type' ensembles (Hermite–1, GOE, Bernoulli, Gaussian) and present evidence for the scaling behavior as well as for the claim that the scalled runtime distributions are insensitive to the choice of initial distribution from these four.

Then we repeat the presentation for the Toda algorithm. We provide similar corroboration of scale invariance and universality, but it becomes immediately clear that the shape of the scaled runtime distributions in the Toda case is very different from the QR case.

The next section discusses the distribution of the deflation positions first in the QR, then in the Toda case. Again we observe distinct differences between the QR and the Toda case but similarities across the initial distributions.

After that we study empirically a Hamiltonian that was found to consistently deflate the matrix in the middle – relevant for modern divide and conquer ideas for eigenvalue computation.

In the last section of this chapter we explain the statistical analysis we carried out on our data. The statistical analysis supports our pictorial evidence for the Gaussian (right) tails of the Toda runtime data as well as the exponential right tails of the QR runtime data.

Chapter Eight – Conclusions and Further Work This concluding chapter summarizes the findings of the current thesis and discusses some ideas for further research. We prove a theorem related to the behavior of the inverse spectral for Jacobi matrices near the boundary and suggest a quantity the study of which could lead to a better understanding of the phenomena we observed numerically.

Chapter Two

Numerical Linear Algebra

This chapter explains ideas from numerical linear algebra pertinent to the current thesis. We discuss the QR decomposition, the QR algorithm with its shifted variants and provide a geometric intuition for the workings of the QR algorithm.

2.1 The QR Decomposition

Matrix decompositions, ie. ways to write a given matrix as a product of matrices belonging to certain classes, play an important role in applied mathematics. A particular matrix decomposition, the QR decomposition, is often used as a building block in the eigenvalue algorithms that we want to study below. This procedure allows a matrix A to be written as the product of an orthogonal matrix Q and an upper triangular matrix R. Since it is relevant to us only in its simplest special case, this is what we'll focus on: The case of symmetric nonsingular matrices.

Lemma 2.1. *Let $A = (a_1, \ldots, a_n) \in \mathbb{R}^{n\times n}$ be a symmetric nonsingular matrix with columns $a_j \in \mathbb{R}^n$. Then there exist unique matrices Q and R with the property that Q is orthogonal and R is upper triangular with strictly positive diagonal entries such that $A = QR$.*

Proof. We recall that the Gram–Schmidt method orthonormalizes a set of $k \leq n$ linearly independent vectors in $\mathbb{R}^n$. Existence of the QR decomposition follows from

applying the Gram–Schmidt procedure to the vectors a_j: Iteratively we define:

$$\tilde{q}_k = a_k - \sum_{m=1}^{n-1} \langle q_m, a_k \rangle q_m \tag{2.1}$$

$$\hat{q}_k = \frac{\tilde{q}_k}{\|\tilde{q}_k\|} \tag{2.2}$$

If we augment this procedure by setting

$$\hat{r}_{kk} = \langle a_k, \hat{q}_k \rangle \tag{2.3}$$

$$r_{kk} = |\hat{r}_{kk}| \tag{2.4}$$

$$q_k = \operatorname{sign}(\hat{r}_{kk}) \cdot \hat{q}_k \tag{2.5}$$

Then by non–degeneracy and normalization, the family $Q = (q_1, \ldots, q_n)$ is an orthonormal basis of $\mathbb{R}^n$ and a_m is a linear combination of $a_m = r_{1m}q_1 + \cdots + r_{mm}q_m$ of $q_1, \ldots, q_m$ with the property that $r_{mm} > 0$. This means that we can write $A = QR$, where R is upper triangular and strictly positive on the diagonal as required.
Uniqueness follows inductively from the requirements on A, Q and R. In particular $r_{11} = \|a_1\| > 0$ and $q_1 = a_1/\|a_1\|$ serves as the base of the induction. Now assume that $(a_1, \ldots, a_m) = Q_m R_m$ uniquely, where $Q_m \in \mathbb{R}^{n\times m}$ with orthonormal columns and $R_m \in \mathbb{R}^{m\times m}$ upper triangular with strictly positive entries. Uniqueness of the decomposition up to this step enforces the direction of q_{m+1} to be the same as the direction of the vector $a_k - \sum_{j=1}^{k-1} \langle a_k, q_j \rangle q_j$ (cf. (2.1)). In order to enforce the other constraints the equations (2.2) to (2.5) have to be satisfied. This proves the claim. □

From a practical point of view the Gram–Schmidt procedure is not very relevant, because it is unstable [26]. There exist a number of stable and efficient procedures to compute the QR factorization of a matrix [17, 26]. For the sake of completeness

we explain how to compute the QR decomposition using Householder reflections (cf. [26, p.224]). A Householder reflection matrix is a matrix of the form $H = \mathrm{id} - 2\frac{vv^T}{\|v\|^2}$, where $v \in \mathbb{R}^n \neq 0$. These matrices have the following properties:

1. Householder matrices are symmetric, because $V = vv^T$ is: $V_{ij} = v_i v_j$.

2. Householder matrices are orthogonal: $HH^T = \mathrm{id}$.

$$HH^T = \left(\mathrm{id} - 2\frac{vv^T}{\|v\|^2}\right)\left(\mathrm{id} - 2\frac{vv^T}{\|v\|^2}\right)^T = \mathrm{id} - 4\frac{vv^T}{\|v\|^2} + 4\|v\|^{-4}vv^Tvv^T = \mathrm{id}$$

3. Geometrically, applying a Householder matrix to a vector x, is equivalent to reflecting x through the hyperplane which is orthogonal to the vector v. This is because by the first two steps H is an involution and because every vector orthogonal to v is mapped to itself under H.

In algorithm 1 the idea is to go through the columns of the input matrix A and find successive hyperplanes such that the part below the diagonal of a given column if reflected through the hyperplane has zeros everywhere below the diagonal. As a first step, for a nonzero vector $x \in \mathbb{R}^k$ we identify a vector $v(x)$ such that $H(v) := \mathrm{id} - 2\frac{vv^T}{\|v\|^2}$ has the property that $H(v(x))x = e_1\|x\|$, where $e_1 \in \mathbb{R}^k$ is the unit vector in $\mathbb{R}^k$ with all entries except the first one equal to zero. This is accomplished by setting:

$$\tilde{v} = x - \|x\|e_1 \tag{2.6}$$

$$v(x) = \frac{\tilde{v}}{\|\tilde{v}\|} =: \mathrm{house}(x) \tag{2.7}$$

Clearly,

$$H(v(x))x = \left[\mathrm{id} - 2\frac{(x - \|x\|e_1)(x - \|x\|e_1)^T}{\|x - \|x\|e_1\|^2}\right] x =$$
$$= x - 2\frac{(x - \|x\|e_1)}{\|x - \|x\|e_1\|^2} \cdot (\|x\|^2 - |x_1|\|x\|) = \|x\|e_1$$

We use this insight to create a sequence Q_j of structured and orthogonal matrices

Algorithm 1 `QRDecomp` – computes QR decomposition

Require: $A \in \mathbb{R}^{n\times n}$, $A^T = A$
1: $R_0 \leftarrow A, Q \leftarrow \mathrm{id}_{n\times n}$
2: **for** j=1,...,n **do**
3: $v \leftarrow \mathrm{house}(R_{j-1}(j:n, j)) \in \mathbb{R}^{n-j+1}$
4: $Q_j \leftarrow \begin{bmatrix} \mathrm{id}_{(j-1)\times(j-1)} & 0 \\ 0 & H(v) \end{bmatrix}$
5: $R_j \leftarrow Q_j R_{j-1}$
6: $Q \leftarrow Q_j Q$
7: **end for**
8: $R = R_n$
9: **return** Q, R

$Q_n \cdots Q_1 A = R$, so that we can choose $Q = Q_1^T \cdots Q_n^T$. For a practical implementation of this algorithm additional steps are taken: Firstly, one needs to ensure numerical stability of the subtraction 2.6. Secondly, one uses the special structure of Q_j to speed up the necessary matrix multiplications [26].

2.2 Power Iterations for Matrix Diagonalization

As a first foray into eigenvalue algorithms we present a simple idea called power iteration and a generalization called simultaneous iteration (also called orthogonal iteration) [26, 52]. The importance of this technique lies in the fact that it provides a geometrical basis for the understanding of the well known QR algorithm. This geo-

metric intuition will also be useful for understanding other, more general, eigenvalue algorithms encountered later on. This short section will be somewhat impressionistic to provide intuition.

Simultaneous iteration is a power iteration in the sense that powers of the matrix A to be diagonalized are successively applied to a subspace S. The simplest instance of a power iteration method computes an eigenvector of A belonging to the eigenvalue with largest absolute value. In particular, assume that A has eigenvalues $|\lambda_1| > \cdots > |\lambda_n|$ with corresponding (orthonormalized) eigenvectors $v_1, \ldots, v_n$. Choose any (eg. random) vector $u_0 := \alpha_1 v_1 + \cdots + \alpha_n v_n$ with $\alpha_1 \neq 0$ and alternate the instructions

$$\tilde{u}_{n+1} = Au_n \tag{2.8}$$

$$u_{n+1} = \frac{\tilde{u}_{n+1}}{\|\tilde{u}_{n+1}\|} \tag{2.9}$$

Then

$$\|v_1 - u_k\| = \left\| v_1 - \frac{a_1\lambda_1^k v_1 + \cdots + a_n\lambda_n^k v_n}{\|a_1\lambda_1^k v_1 + \cdots + a_n\lambda_n^k v_n\|} \right\| \leq C \cdot \left|\frac{\lambda_2}{\lambda_1}\right|^n \tag{2.10}$$

The next step in this hierarchy of algorithms is called subspace iteration and generalizes the basic power iteration. We use the notation of [52] and define the subspaces $T := \text{span}(v_1, \ldots, v_m)$, $U := \text{span}(v_{m+1}, \ldots, v_n)$. Subspace iteration now starts out with a subspace S in generic position such that $S \cap U = \{0\}$ and computes the subspace $A^m S$ (for example by applying powers of A to a basis of S). Based on the reasoning from 2.10 the following theorem, proven in [44], is plausible:

Theorem 2.2. *Let T, U and S be as above. For two subspaces X, and Y define the*

metric

$$d(X,Y) := \sup_{x\in X, \|x\|=1} \inf_{y\in Y} \|x-y\| \tag{2.11}$$

Then

$$d(A^n S, T) \leq C \left| \frac{\lambda_{m+1}}{\lambda_m} \right|^n \tag{2.12}$$

Note that $A^m S$ converges to an invariant subspace of A, which means that for an orthonormal basis $P = [P_1, P_2]$ such that P_1 is a basis of $A^m S$ we should expect that in

$$P^T A P = \begin{bmatrix} P_1^T A P_1 & P_1^T A P_2 \\ P_2^T A P_1 & P_2^T A P_2 \end{bmatrix} \tag{2.13}$$

the blocks $P_1^T A P_2 = P_2^T A P_1$ are small. One practical way of realizing subspace iteration is given by simultaneous iteration. Assume that $S = \text{span}(q_1^0, \ldots, q_m^0)$ as above. Then alternate the following instructions:

1. Let $\tilde{q}_j^{k+1} := A q_j^k$
2. Orthonormalize $\tilde{q}_1^{k+1}, \ldots, \tilde{q}_m^{k+1}$ from left to right to obtain $q_1^{k+1}, \ldots, q_m^{k+1}$, an orthonormal basis for $A^{k+1} S$.

Observe that orthonormalizing in this way from left to right conserves the subspaces $\text{span}(A q_1^k, \ldots, A q_l^k)$ for all $1 \leq l \leq n$. This means that by following these instructions not only do we converge to an invariant subspace of dimension k, but we actually obtain invariant subspaces of dimensions $1, 2, \ldots, k-1$. Let us start out with $S = \text{span}(q_1^0, \ldots, q_n^0)$ a basis of the full space $\mathbb{R}^n$. By orthonormality and the subspace

iteration property that $\text{span}(q_1^n, \ldots, q_l^n)$ converges to $T_l = \text{span}(v_1, \ldots, v_l)$ we obtain that $q_j^n \to v_j$ and, hence, for $Q_k := (q_1^k, \ldots, q_n^k)$ we have that $Q_n^T A Q_n \to \Lambda$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$.

2.3 The QR algorithm for Matrix Diagonalization

Over the years variants of the QR algorithm have become the most widely used strategy to determine the eigenvalues and eigenvectors of small to moderately sized matrices. Here we will focus exclusively on the symmetric eigenvalue problem as described for example in [26]. The algorithm was discovered independently in [24, 34] and in its most basic form calls for iteratively reversing the subsequent QR decompositions of a given matrix:

$$A_0 = Q_1 R_1 \tag{2.14}$$

$$A_n = R_n Q_n = Q_{n+1} R_{n+1} \tag{2.15}$$

Equation 2.15 is called the QR step. Before we delve into establishing finer properties of the QR algorithm, let us argue that the QR algorithm is 'neither more nor less than a clever implementation of simultaneous iteration' [52]: At the core of this claim is the idea that emerged in lemma 2.1, that the QR decomposition is a matrix version of the Gram–Schmidt orthogonalization procedure. Let $\hat{Q}_0$ be the identity matrix (ie. a basis matrix of $\mathbb{R}^n$). Phrasing simultaneous iteration in terms of the QR decomposition then implies alternating $D_{m+1} = A\hat{Q}_m$ and $D_{m+1} = \hat{Q}_{m+1} R_{m+1}$. D_{m+1} corresponds to the result of the first step of simultaneous iteration, $\hat{Q}_{m+1}$ corresponds to the result of the second step of simultaneous iteration.

Lemma 2.3. *Define A, D_m, $\hat{Q}_m$ and R_m as above and denote $A_m := \hat{Q}_m^T A \hat{Q}_m$.*

Then

$$A_m = R_m Q_m = Q_{m+1} R_{m+1} \tag{2.16}$$

$$\hat{Q}_m = Q_1 \cdots Q_m \tag{2.17}$$

for a sequence of orthogonal matrices Q_m.

Proof. We proceed by induction. Since $\hat{Q}_0 = \mathrm{id}$, we have $A_0 = A = D_1 = \hat{Q}_1 R_1$ and $A_1 = \hat{Q}_1^T A \hat{Q}_1 = R_1 \hat{Q}_1$. Thus we can set $Q_1 = \hat{Q}_1$, which shows the basis of the induction.

For the induction step, assume that $A_m = R_m Q_m$ and that $\hat{Q}_m = Q_1 \cdots Q_m$. Then by definition let $D_{m+1} = A\hat{Q}_m$ with QR decomposition $\hat{Q}_{m+1} R_{m+1}$ and denote the QR decomposition of A_m by $A_m = Q_{m+1} \tilde{R}_{m+1}$. By definition of A_m we have that $A_m = \hat{Q}_m^T A \hat{Q}_m$, so that $\hat{Q}_m A_m = A\hat{Q}_m$ which implies $D_{m+1} = A\hat{Q}_m = \hat{Q}_m A_m = \hat{Q}_m Q_{m+1} \tilde{R}_{m+1}$. Using uniqueness of the QR decomposition we obtain $R_{m+1} = \tilde{R}_{m+1}$ and $\hat{Q}_{m+1} = Q_1 \cdots Q_{m+1}$ as necessary to prove the claim. □

Thus, the QR algorithm is an implementation of simultaneous iteration, where the matrices D_m are never explicitly computed. We have argued informally in section 2.2 that $\hat{Q}_m$ will converge to the eigenvector matrix with eigenvectors sorted by the moduli of their eigenvalues and consequently $A_m = \hat{Q}_m^T A \hat{Q}_m$ will converge to the eigenvalue matrix Λ. It is this sequence of approximations to the eigenvalue matrix which the QR algorithm deals with explicitly.

The reason for the popularity of this iterative method rests in the following three observations:

1. The QR step preserves Jacobi (ie. tridiagonal symmetric) matrices.

2. Any symmetric matrix can be deterministically reduced to a Jacobi (ie. tridiagonal symmetric) matrix with the same spectrum.

3. In all variants of the QR algorithm the QR step for Jacobi matrices can be implemented to run in linear time.

We prove the first observation in the following lemma:

Lemma 2.4. *The QR step preserves Jacobi matrices.*

Proof. It is sufficient to show that the QR step preserves both symmetric and Hessenberg matrices, because Jacobi matrices are symmetric Hessenberg matrices.
Symmetry: Recall that the sequence of matrices A_m generated by the QR algorithm can be interpreted as a sequence of basis transformations of the initial matrix A: $A_m = \hat{Q}_m^T A \hat{Q}_m$ in the notation of lemma 2.3. Clearly $A_m^T = \hat{Q}_m^T A^T \hat{Q}_m = \hat{Q}_m^T A \hat{Q}_m = A_m$.
Hessenberg: Derive from equation (2.15) that $A_m = R_m A_{m-1} R_m^{-1}$ holds. The upper triangular matrices form a group, but in our case only the existence of an inverse upper triangular matrix matters: Using Cramer's rule we have that $(A^{-1})_{ij} = \frac{\det A_{ji}}{\det A}$ where A_{ji} is the matrix arising from A by deleting the j–th row and the i–th column. Let $i > j$. It's easy to see that $\det A_{ji} = 0$ as required. As the last ingredient, we need to demonstrate that Hessenberg matrices are conserved under left and right multiplication of upper triangular matrices. Multiplication of a Hessenberg matrix H with an upper triangular matrix U from the right implies that the j–th column of the result HU is a linear combination of the first j columns of H – this preserves the Hessenberg property. Multiplication from the left means that the j–th row of UH is a combination of the last j columns of H – this also preserves the Hessenberg property. □

The Jacobi reduction procedure for symmetric matrices mentioned above is very similar to the Householder reduction strategy for the QR decomposition which was presented in section 2.1. There are two crucial differences: Firstly, only the sub-diagonal parts of the columns are reflected onto their first components. Secondly, the original matrix is multiplied with the Householder reflection matrices on both sides rather than just the left side. What makes this algorithm work is the following (block) matrix identity:

$$\begin{aligned} Z(b) \underbrace{\left(\begin{array}{cc|c} & & 0_{(k-1)\times(n-k)} \\ & T_{k\times k} & b^T \\ \hline 0_{(n-k)\times(k-1)} & b & M_{n-k} \end{array}\right)}_{=:X} Z^T(b) \\ = \left(\begin{array}{cc|c} & & 0_{(k-1)\times(n-k)} \\ & T_{k\times k} & \|b\|e_1^T \\ \hline 0_{(n-k)\times(k-1)} & \|b\|e_1 & H(b)M_{n-k}H(b) \end{array}\right) = \\ \underbrace{\left(\begin{array}{cc|c} & & 0_{k\times(n-k-1)} \\ & T_{(k+1)\times(k+1)} & \tilde{b}^T \\ \hline 0_{(n-k-1)\times(k)} & \tilde{b} & M_{n-k-1} \end{array}\right)}_{=:Y} \end{aligned} \tag{2.18}$$

$$Z(b) = \left(\begin{array}{c|c} \mathrm{id}_{k\times k} & 0_{k\times(n-k)} \\ \hline 0_{(n-k)\times k} & H(b) \end{array}\right) \tag{2.19}$$

Here, $T_{k\times k}$ is a Jacobi matrix of size $k \times k$, M_{n-k} is a (full) symmetric matrix of size $(n-k) \times (n-k)$, $0_{m\times n}$ is a block of zeros of size $m \times n$ and $b \in \mathbb{R}^{n-k}$. The index k ranges from $1 \leq k \leq n$. The significance of equation (2.18) lies in the fact that the structured matrix X is multiplied from both sides with a highly structured block matrix Z which increases the size of the Jacobi block in X by one and decrease the size of the full symmetric block by one. Algorithm 2 details this procedure.

Algorithm 2 `HouseholderJacobi` – Reduce symmetric to Jacobi matrices

Require: $A \in \mathbb{R}^{n\times n}$, $A^T = A$

1: $A_0 \leftarrow A$
2: **for** $i = 1, \ldots, n-2$ **do**
3: $\quad b \leftarrow A_{i-1}(i+1:n, i)$
4: $\quad Z_i \leftarrow Z(b)$
5: $\quad A_i \leftarrow Z_i A_{i-1} Z_i$
6: **end for**
7: **return** A_{n-2}

Direct application of the QR step from (2.15) is referred to as the unshifted QR algorithm for reasons that will become clearer in the next section. We have promised that it is possible to devise a fast procedure to compute the QR step for a Jacobi matrix. The first step in designing this procedure is to recall the 'implicit Q theorem' from [26], which we've simplified here to make it exactly fit our application:

Theorem 2.5. *Let $Q = (q_1, \ldots, q_n)$ and $V = (v_1, \ldots, v_n)$ denote orthogonal matrices such that both $Q^T AQ = H$ and $V^T AV = G$ are unreduced Jacobi matrices (ie. the subdiagonal contains only strictly non zero entries). If $v_1 = q_1$, then $v_i = \pm q_i$ for $i = 1, \ldots, n$ and $|h_{i,i-1}| = |g_{i,i-1}|$*

Proof. Define the orthogonal matrix $W = (w_1, \ldots, w_n) = V^T Q$. Clearly $GW = V^T AQ = WH$ and because of $v_1 = q_1$ we have that $w_1 = e_1$. For $j \geq 2$ we have that

$$h_{j,j-1} w_j = G w_{j-1} - h_{j-2,j-1} w_{j-2} - h_{j-1,j-1} w_{j-1} \tag{2.20}$$

This makes the matrix W upper triangular, which together with orthogonality implies that $W = \mathrm{diag}(\pm 1, \ldots, \pm 1)$. The claim follows. □

Before detailing the algorithm, we introduce Givens rotations

$$G_{ij}(\theta) = G_{ij}(\cos\theta, \sin\theta) = \mathrm{id} - (1-\cos\theta)E_{ii} - (1-\cos\theta)E_{jj} + \sin\theta E_{ij} - \sin\theta E_{ji} \tag{2.21}$$

where E_{mn} is a matrix which is zero everywhere except for a one in the m–th row and the n–th column. Recall that in order for

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix}^T \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} = \begin{pmatrix} \sqrt{a^2+b^2} \\ 0 \end{pmatrix} \tag{2.22}$$

to hold and the matrix containing c and s still to be a rotation, we have to set $c(a,b) = a/\sqrt{a^2+b^2}$ and $s(a,b) = -b/\sqrt{a^2+b^2}$. Now assume that we would like to compute the QR step T_1 of the unreduced Jacobi matrix $T_0 = QR$, ie. $T_1 = RQ = Q^T T_0 Q$. By properties of the QR decomposition it follows directly that $T_0 e_1 = QRe_1 = \|t_0^1\| q_1$, where t_0^1 denotes the first column of T_0. Thus, $q_1 = t_0^1/\|t_0^1\|$. Let $J_1 = G_{12}(c(a_1,b_1), s(a_1,b_1))$. Then by definition of Givens rotations, the first column of J_1 is exactly equal to the first column of Q. Multiplying T_0 with J_1 left and right creates a 'bulge', which means that this procedure 'nearly' preserves the tridiagonal structure:

$$J_1^T T_0 J_1 = \begin{pmatrix} x & x & + & 0 & 0 & 0 & \cdots \\ x & x & x & 0 & 0 & 0 & \\ + & x & x & x & 0 & 0 & \\ 0 & 0 & x & x & x & 0 & \\ 0 & 0 & 0 & x & x & x & \\ 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots \end{pmatrix} \tag{2.23}$$

Here the +–sign indicates where the tridiagonal structure is destroyed. In order to

'fix' this situation we engage in what's called 'bulge chasing' – again using Givens rotations: We successively push the bulges towards the lower end of the matrix by multiplying from left and right with Givens rotations of appropriate angles. For example, in (2.23) we would choose rotations that eliminate the +, but in doing so they would introduce a new bulge in the next rows and columns from the current bulge. Consider algorithm 3: It is clear by design that A_{n-1} is a Jacobi matrix, but how can we be sure that it is actually related to the QR step $\tilde{A} = RQ$ of $A = QR$? This is where the 'implicit Q theorem' comes in. Clearly $\tilde{A} = Q^T A Q$ is Jacobi by lemma 2.4. The matrix A_{n-1} returned by algorithm 3 is also Jacobi and it's also conjugate to A, because

$$A_{n-1} = (J_1 \cdots J_{n-1})^T A J_1 \cdots J_{n-1} \tag{2.24}$$

Note that, by design of the Givens matrices J_j for $J \geq 2$ we have $J_j e_1 = e_1$, so that $V e_1 := J_1 \cdots J_{n-1} e_1 = J_1 e_1 = Q e_1$, so that the first column of V is the same as the first column of Q. Theorem 2.5 now implies that $\hat{A}$ and A_{n-1} are 'essentially' the same, ie. the same up to signs of the off diagonal terms

Algorithm 3 `QRStep` – Efficient QR Step for Jacobi matrices

Require: $A \in \mathbb{R}^{n \times n}$, Jacobi
1: $J_1 = G_{12}(c(a_1, b_1), s(a_1, b_1))$
2: $A_1 = J_1^T A J_1$
3: **for** $j = 2, \ldots, n-1$ **do**
4: $\quad a \leftarrow A_{j-1}(j, j-1))$
5: $\quad b \leftarrow A_{j-1}(j+1, j-1)$
6: $\quad J_j = G_{j,j+1}(c(a, b), s(a, b))$
7: $\quad A_j = J_j^T A_{j-1} J_j$
8: **end for**
9: **return** A_{n-1}

In section 2.2 we have already argued intuitively that simultaneous iteration converges linearly, the speed depending on ratios of neighbouring eigenvalues. We cite [26, pp. 330–339] where it is shown that the QR algorithm in this basic form exhibits

linear convergence speed as well:

Theorem 2.6. *Let $T_0 \in \mathbb{R}^{n\times n}$ be a Jacobi matrix with spectral decomposition $T_0 = Q^T \Lambda Q$, where $\Lambda = diag[\lambda_1, \cdots, \lambda_n]$ so that $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$. Let T_n denote the n^{th} iterate of T in the QR algorithm and let*

$$Q_k, R_k \leftarrow \texttt{QRDecomp}(T_k) \tag{2.25}$$

$$\lambda_i^{(k)} := r_{k,ii} \tag{2.26}$$

$$\hat{Q}_k := Q_0 \cdots Q_k =: \left[\hat{q}_1^{(k)}, \ldots, \hat{q}_n^{(k)}\right] \tag{2.27}$$

Then the following estimates are satisfied:

$$\left|\lambda_i - \lambda_i^{(k)}\right| \leq C_1 \cdot \left|\frac{\lambda_{i+1}}{\lambda_i}\right|^k \tag{2.28}$$

$$dist\left(span\{\hat{q}_1^{(k)}, \ldots, \hat{q}_i^{(k)}\}, span\{q_1, \ldots, q_i\}\right) \leq C_2 \left|\frac{\lambda_{i+1}}{\lambda_i}\right|^k \tag{2.29}$$

so that in particular, $\hat{Q}_k \overset{n\to\infty}{\longrightarrow} Q$, $R_k \overset{n\to\infty}{\longrightarrow} \Lambda$ as well as $\hat{Q}_k^T T_0 \hat{Q}_k \overset{n\to\infty}{\longrightarrow} \Lambda$.

2.4 The Shifted QR Algorithm

Consider the inequalities (2.28) and (2.29) from theorem 2.6. The eigenvalues of a matrix scale with the matrix itself:

$$\det(T_0 - \lambda \mathrm{id}) = \alpha^{-n} \cdot \det(\alpha T_0 - \alpha\lambda \mathrm{id}) \tag{2.30}$$

so that λ is an eigenvalue of T_0 if and only if $\alpha\lambda$ is an eigenvalue of αT_0. It is thus clear that the estimates (2.28) and (2.29) of the speed of convergence for the QR procedure remain unaffected by scaling the initial matrix T_0.

On the other hand, shifting all eigenvalues along the real axis can significantly increase the speed of convergence, because if $\mu \approx \lambda_{k+1}$, then

$$\left|\frac{\lambda_{k+1}}{\lambda_k}\right| \ll \left|\frac{\lambda_{k+1}-\mu}{\lambda_k-\mu}\right| \tag{2.31}$$

The theoretical usefulness of this idea is illustrated by the following proposition (cf. [26], pp. 353):

Proposition 2.7. *Let μ be an eigenvalue of an $n \times n$ unreduced Jacobi matrix T. If $\bar{T} := RQ + \mu\mathrm{id}$, where $T - \mu\mathrm{id} = QR$, then $\bar{b}_{n-1} = 0$ and $\bar{a}_n = \mu$, where $\bar{b}_j$ denote the subdiagonal and $\bar{a}_j$ denote the diagonal entries of $\bar{T}$.*

Proof. Clearly the first $n-1$ columns of T are linearly independent, regardless of the choice of μ, because each pair of columns contains at least a row where one of them is zero and the other one is non zero. This implies that $r_{ii} \neq 0$ for $1 \leq i \leq n-1$ (compare algorithm `QR`). Since $T - \mu\mathrm{id}$ is singular by assumption we have:

$$0 = \det(T - \mu\mathrm{id}) = \det(QR) = \det R = r_{11} \cdots r_{nn} \tag{2.32}$$

which implies $r_{nn} = 0$. This implies that

$$\bar{b}_{n-1} = \sum_{j=1}^{n} r_{n,j} \cdot q_{j,n-1} = 0 \tag{2.33}$$

$$\bar{a}_n = \sum_{j=1}^{n} r_{n,j} \cdot q_{j,nn} + \mu = \mu \tag{2.34}$$

□

Note how the exact shift allows for deflating the exact eigenspace corresponding to the eigenvalue μ in the last row and column of the matrix $\bar{T}$. Two ideas come

with this realization:

1. We can shift adaptively, ie choose an improved approximation μ to an eigenvalue at each QR step.

2. We should always try to approximate the eigenvalue corresponding to the last row and column of T. Otherwise it follows from proposition 2.7 that the corresponding eigenspace will continually change, rather than converge.

Algorithmically, this leads to the shifted QR algorithm:

$$A_n - \mu_n \mathrm{id} = Q_n R_n \tag{2.35}$$

$$A_{n+1} = R_n Q_n + \mu \mathrm{id} \tag{2.36}$$

Before we discuss the speed increase that is gained from a shifting strategy, let us first note that shifting is compatible with the logic behind the efficient QR step for Jacobi matrices discussed in algorithm 3. It follows from (2.35) and (2.36) that

$$A_{n+1} = Q_n^T (A_n - \mu \mathrm{id}) Q_n + \mu \mathrm{id} = Q_n^T A_n Q_n \tag{2.37}$$

So that again the sequence A_n is a sequence of conjugate matrices. Assume we want to apply one shifted QR step to the matrix A. We compute $A - \mu \mathrm{id} = QR$ and by the same argument as the one given in section 2.3 it follows that the first column of Q is equal to the first column of the Givens matrix

$$J_1 := G_{12}(c(a_1 - \mu, b_1), s(a_1 - \mu, b_1)) \tag{2.38}$$

Now the exact same arguments about bulge chasing apply to yield the result that the matrix returned from algorithm 3 with J_1 chosen as in 2.38 is 'essentially the same'

as the one returned from the QR step $Q^T AQ$. Thus shifted QR can be implemented for Jacobi matrices with no extra cost as compared to the standard QR algorithm. Time tested choices for the shifts μ_n are the so called Francis ([26], p. 385) and Wilkinson ([26], p.418) shifts. In many cases these shifting strategies lead to cubic convergence to zero of the lowest subdiagonal entry. However, examples have been discovered that exhibit failures of cubic converge:

1. The Francis shift can fail to converge at all for certain matrices [5, 17]
2. The Wilkinson shift has been shown to be only quadratically convergent for certain small sets of matrices [37]

In light of these concerns it is not surprising that "it is still an open problem to find a shift strategy that guarantees fast convergence for all matrices [17]."

Chapter Three

Random Matrices

The goal of this chapter is to introduce the initial distributions that were used in our numerical simulations along with some properties of their spectra. We also recall some well known facts about Jacobi matrices. The last section of this chapter is devoted to a theorem regarding the scale of the minimum subdiagonal entry of Hermite–1 distributed Jacobi matrices.

3.1 Background on Random Matrices

Our goal is to study aspects of the behavior of eigenvalue algorithms when the initial matrix is assumed to be chosen from some distribution. In the following we will introduce the distributions that we are using in this thesis and elucidate some of their properties.

3.1.1 Symmetric Matrix Distributions

The Gaussian Orthogonal Ensemble is a widely used matrix model in Random Matrix Theory (RMT) (cf. [11, 39]). This distribution on symmetric, $\mathbb{R}$–valued matrices is completely characterized by the requirements that:

1. The distribution is invariant under conjugation with orthogonal matrices U:

$$\mathbb{P}_{\beta=1}(U^T M U)\mathrm{d}M = \mathbb{P}_{\beta=1}(M)\mathrm{d}M \tag{3.1}$$

2. The entries m_{ij} for $1 \leq j \leq i \leq n$ are statistically independent of each other.

In particular, these two requirements are sufficient to derive that for $Q(x) = -ax^2 + bx + c$ ($a, b, c \in \mathbb{R}$ and $a > 0$), the density of $\mathbb{P}_{\beta=1}$ is given by:

$$P(M) = \frac{1}{Z} \exp\{\mathrm{tr} Q(M)\} \tag{3.2}$$

where $Q(M)$ is to be read as an expression in the Functional Calculus. A proof of this statement can be found on pp. 43 in [39]. The standard Gaussian Orthogonal Ensemble GOE_n is defined by setting $Q(x) = -\frac{x^2}{2}$. We thus obtain:

$$P(M) \propto \exp\left\{-\frac{1}{2}\mathrm{tr}M^2\right\} = \exp\left\{-\frac{1}{2}\sum_{i,k=1}^{n} m_{ik}^2\right\} = \\ = \exp\left\{-\frac{1}{2}\sum_{i=1}^{n} m_{ii}^2 - \sum_{\substack{1\le i\le n \\ j<i}}^{n} m_{ij}^2\right\} \tag{3.3}$$

so that

$$P(M) = \prod_{i=1}^{n}\left[\frac{\exp\{-\frac{m_{ii}^2}{2}}{\sqrt{2\pi}}\right] \cdot \prod_{\substack{1\le i\le n \\ j<i}}^{n}\left[\frac{\exp\left\{-\frac{m_{ij}^2}{2\cdot\frac{1}{2}}\right\}}{\sqrt{\pi}}\right] \tag{3.4}$$

Let $\eta_{\mu,\sigma^2} \sim \mathcal{N}(\mu, \sigma^2)$. Then we can represent GOE_n schematically by

$$M \sim \begin{pmatrix} \eta_{0,1} & * & \cdots & * \\ \eta_{0,\frac{1}{2}} & \eta_{0,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ \eta_{0,\frac{1}{2}} & \cdots & \eta_{0,\frac{1}{2}} & \eta_{0,1} \end{pmatrix} \tag{3.5}$$

Wigner Matrices are probably the simplest symmetric ensemble imaginable. We will use this term to denote symmetric matrices with iid standard Gaussian entries:

$$M \sim \begin{pmatrix} \eta_{0,1} & * & \cdots & * \\ \eta_{0,1} & \eta_{0,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ \eta_{0,1} & \cdots & \eta_{0,1} & \eta_{0,1} \end{pmatrix} \tag{3.6}$$

sometimes the term is used more generally as for example in [29]

The Bernoulli Ensemble is a symmetric Matrix ensemble similar to the Wigner ensemble, where the individual matrix entries are iid Bernoulli with values ± 1 with equal probability $\frac{1}{2}$.

3.1.2 Distributions on Tridiagonal Matrices

The Hermite–1 Ensemble We have already alluded to the importance of Jacobi matrices for the practical implementation of eigenvalue algorithms in section 2.3 (cf. algorithm 2. It is therefore natural to experiment with distributions on the Jacobi matrices as initial data. One natural distribution to consider is the so called Hermite–1 distribution that was introduced in [19]. It is a distribution on Jacobi matrices where all entries on the diagonal and the subdiagonal are mutually independent and the diagonal entries are distributed normally, the subdiagonal entries are χ–distributed with decreasing degrees of freedom. Schematically we can represent the

1–Hermite distribution by

$$M \sim \frac{1}{\sqrt{2}} \begin{pmatrix} \eta_{0,2} & * & \cdots & \cdots & * \\ \chi_{n-1} & \eta_{0,2} & \ddots & \ddots & \vdots \\ 0 & \chi_{n-2} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & * \\ 0 & \cdots & 0 & \chi_1 & \eta_{0,2} \end{pmatrix} \tag{3.7}$$

where χ_k denotes the χ distribution with k degrees of freedom. If $a = (a_1, \ldots, a_n)$ denotes the diagonal entries of M and $b = (b_1, \ldots, b_{n-1})$ denotes the subdiagonal entries of M, then we can write the Hermite–1 distribution in formulas by:

$$f_1(a,b)dadb = \underbrace{\frac{2^{2n-1}}{(2\pi)^{n/2} \cdot \prod_{j=1}^{n} \Gamma(j/2)}}_{=:c_{a,b}} \cdot \prod_{j=1}^{n-1} b_j^{n-j-1} e^{-b_j^2} \cdot \prod_{k=1}^{n} e^{-a_j^2/2} =$$

$$= c_{a,b} \cdot e^{-\mathrm{tr}M^2/2} \cdot \prod_{j=1}^{n-1} b_j^{n-j-1} \tag{3.8}$$

The constant $c_{a,b}$ is the product of the normalization constants of all the χ and normal random variables, where we have incorporated the scaling $\frac{1}{\sqrt{2}}$. Before delving into finer details of the Hermite–1 distribution of Jacobi matrices we mention their well known spectral representation, which goes back at least to Stieltjes [41]. It is used extensively in [15, 19, 41]. In the following theorem we combine formulations from [15, 19]:

Theorem 3.1. *Let T be a Jacobi matrix with diagonal entries $a_1, \ldots, a_n$ and off-diagonal entries $b_1, \ldots, b_{n-1}$, such that $b_i > 0$ for all i. Let $T = Q\Lambda Q^T$, $q = (|q_{11}|, \ldots, |q_{1n}|)$ as well as $\Lambda = diag(\lambda)$, $\lambda = (\lambda_1 < \cdots < \lambda_n)$ then the so called*

spectral map

$$S : (a_1, \ldots, a_n, b_1, \ldots, b_{n-1}) \mapsto (\lambda, q) \tag{3.9}$$

is a diffeomorphism on the set of Jacobi matrices with strictly positive entries on the offdiagonal. Its Jacobian is given by:

$$Jac\ S = \frac{\prod_{j=1}^{n} q_j}{\prod_{i=1}^{n-1} b_i} = \frac{\prod_{i=1}^{n-1} b_i^{i-1}}{\Delta(\lambda)} \tag{3.10}$$

where $\Delta(\lambda)$ *is the Vandermonde determinant* $\Delta(\lambda) = \prod_{i<j} |\lambda_i - \lambda_j|$

As Deift says in [3] it is out of serendipity that the GOE turns out to be eminently compatible with the well known tridiagonalization procedure for symmetric matrices using Householder Reflections. This connection was observed before in [19] (cf. theorem 2.1). We collect the following theorem mainly from [19]:

Theorem 3.2. *Let* $M \sim GOE_n$ *and let* $T = \texttt{HouseholderJacobi}(M)$ *(cf. algorithm 2). Then*

1. *T is distributed as a random variable drawn from the 1–Hermite ensemble.*
2. *The distribution of eigenvalues of T is the same as the distribution of eigenvalues of M.*
3. *Let $M = U\Lambda U^T$ and $T = V\Lambda V^T$. Then $U^T e_1 = V^T e_1$, ie the Householder tridiagonalization procedure leaves the first row of the eigenvector matrix unchanged.*
4. *Let $M \sim GOE_n$ and $M = Q\Lambda Q^T$ with the convention that the first row q of Q is positive. Then the distribution of q is uniform on the positive orthant of the sphere S^{n-1}.*

Proof. 1. Let $M_1 \sim \mathrm{GOE}_n$. The `HouseholderJacobi` procedure produces a sequence of matrices $M_1, M_1, \ldots, M_{n-1}$ with the property that

$$M_{k+1} = G_k M_k G_k^T \tag{3.11}$$

$$G_k = \begin{pmatrix} \mathrm{id}_k & 0 \\ 0 & Q_k \end{pmatrix} \tag{3.12}$$

where id_k is a $k \times k$–identity matrix and Q_k is a $(n-k) \times (n-k)$ orthogonal matrix. Q_k is chosen such that the vector $v_k := (m^{(k)}_{k+1,k}, \ldots, m^{(k)}_{n,k})^T \in \mathbb{R}^{n-k}$ gets mapped to $Q_k v_k = (\|v_k\|, 0, \ldots, 0)^T \in \mathbb{R}^{n-k}$, where $m^{(k)}_{ij}$ are the entries of the matrix M_k. Recall that the χ distribution with n degrees of freedom is defined as the distribution of the norm of a random n dimensional Gaussian vector. Note that any block around the diagonal of a GOE matrix is again a GOE matrix and recall that the GOE is invariant under orthogonal transformations. Then we conclude using induction that M_k has the structure

$$M_k = \left(\begin{array}{c|c} T_k & X_k^T \\ \hline X_k & \tilde{M}_k \end{array}\right) \tag{3.13}$$

$$X_k = \begin{pmatrix} 0 & \chi_{n-k+1} \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{(n-k+1)\times(k-1)} \tag{3.14}$$

where T_k is a $(k-1) \times (k-1)$–dimensional Jacobi matrix whose diagonal is distributed like a $k-1$–dimensional standard Gaussian vector with covariance matrix equal to the identity and whose subdiagonal is distributed like $(\chi_{n-1}, \ldots, \chi_{n-k+2})$. The block $\tilde{M}_k$ is GOE_{n-k+1}. The claim follows by letting k range from 1 to $n-1$.

2. This is true, because each step in the `HouseholderJacobi` procedure is a conjugation by an orthogonal matrix and thus leaves the eigenvalues untouched.

3. Note that $V^T e_1 = U^T G_1^T \cdots G_{n-1}^T$. Because of the fact that the first rows of all of the G_1 have the form $(1, 0, \ldots, 0)$ we have that $U^T G_1^T \cdots G_{n-1}^T e_1 = U^T e_1$, this proves the claim.

4. This claim is a special case of theorem 2.12 in [19]: We need to transform the Hermite–1 distribution on (a, b)–space into a distribution on (q, λ)–space:

$$f_1(a,b)dadb = c_{a,b} \cdot e^{-\mathrm{tr}M^2/2} \cdot \prod_{j=1}^{n-1} b_j^{n-j-1} dadb =$$
$$= c_{a,b} \cdot e^{-\mathrm{tr}\Lambda^2/2} \cdot \prod_{j=1}^{n-1} b_j^{n-j-1} \cdot \frac{\prod_{i=1}^{n-1} b_i}{\prod_{j=1}^{n} q_j} d\lambda dq = c_{a,b} \cdot e^{-\mathrm{tr}\Lambda^2/2} \cdot \Delta(\lambda) d\lambda dq$$
$$= \left(s_n^{-1} dq\right) \cdot \left(c_H^1 e^{-\mathrm{tr}\Lambda^2/2} \cdot \Delta(\lambda) d\lambda\right) \quad (3.15)$$

Here we have defined

$$c_H^1 = (2\pi)^{-n/2} \prod j = 1^n \frac{\Gamma(3/2)}{\Gamma(1 + j/2)} \quad (3.16)$$

$$s_n^{-1} = n! \cdot \frac{2^{2n-1}\Gamma(n/2)}{\pi^{n/2}} \quad (3.17)$$

It is a simple exercise to see that $c_{a,b} = c_H^1 \cdot s_n^{-1}$. Since s_n is the surface area of the positive orthant of the $n-1$–dimensional sphere, the claim follows.

□

Theorem 3.2 states that the distribution of the random variable $q := U^T e_1$ is the uniform distribution on the positive quadrant of the sphere $\mathbb{S}_{n-1}$. Consider the map $\Psi : (q_1, \ldots, q_n) \mapsto (q_1^2, \ldots, q_n^2)$. This is a bijective map taking this quadrant into the $n-1$–dimensional probability Simplex Σ_n. We compute the pushforward under this map of the uniform distribution on the positive quadrant of $\mathbb{S}_{n-1}$.

Proposition 3.3. *Let μ denote the uniform probability measure on the positive quad-*

rant on $\mathbb{S}_{n-1}$. *Then the pushforward measure* $\Psi^*\mu$ *onto the simplex* Σ_n *is the Dirichlet distribution* $Dir(\frac{1}{2}, \ldots, \frac{1}{2})$

For a proof we ask to consult the appendix. Note that the uniform measure on the complex sphere is pushed forward to the uniform measure on Σ_n by the same "squaring map." This result is stated on p. 115 in the famous chapter $3\frac{1}{2}$ in Gromov's [28].

Uniform Doubly Stochastic Jacobi Matrices These matrices were introduced and studied recently in [18]. They provide an example of initial data whose spectral limit is not the Wigner semicircle law, as opposed to all the ensembles studied so far. When generating these initial matrices we use the Gibbs sampler described in [18] to generate matrices of the form:

$$\begin{pmatrix} 1-c_1 & c_1 & 0\cdots & 0 \\ c_1 & 1-c_2-c_1 & \ddots & \vdots \\ 0 & \ddots & \ddots & c_{n-1} \\ 0 & \cdots & c_{n-1} & 1-c_{n-1} \end{pmatrix} \tag{3.18}$$

where $(c_1, \ldots, c_{n-1})$ is distributed (approximately) uniformly on the polytope $\{c_i \geq 0, c_i + c_{i+1} \leq 1\}$.

Jacobi Matrices with Uniform Eigenvalues To generate matrices of size $n \times n$ we sample $\lambda_i \sim$ iid Unif$(-2\sqrt{n}, 2\sqrt{n})$ as well as q uniformly on the positive orthant of the sphere S^{n-1}. We use the inverse spectral map S^{-1} to obtain a Jacobi matrix T.

3.1.3 Spectra of large Random Matrices

Theorem 2.6, equation (5.13), equation (5.26) in theorem 5.1 and equation (5.39) all clearly demonstrate the importance of the eigenvalues in determining the behavior of the classical QR algorithm as well as the general DLNT algorithms as described in theorem 5.1.

Understanding the distributional aspects of the eigenvalues for elements of the initial matrix ensembles described in sections 3.1.1 and 3.1.2 is thus a task of prime importance. This subject has, after initial results by Wigner [54], become a large field and we'll describe in the following a few of the results pertaining to our study. Let

$$\sigma(dx) := \frac{1}{2\pi}\sqrt{4-x^2}\mathbb{1}_{[-2,2]}(x)dx \tag{3.19}$$

Wigners Semicircle Law

The modern result most similar in spirit to Wigners original theorem can be found for example on p.16 of [29]:

Theorem 3.4. *Consider a real or complex self adjoint $N \times N$–matrix A^N with independent lower triangular entries. Assume that*

$$\begin{cases} \mathbb{E}A_{ij}^N = 0,\ 1 \le i \le j \le N \\ \lim_{N\to\infty} \frac{1}{N^2} \sum_{1\le i,j\le N} \left| N \cdot \mathbb{E}\left|A_{ij}^N\right|^2 - 1 \right| = 0 \end{cases} \tag{3.20}$$

and, additionally, that for all $k \in \mathbb{N}$:

$$B_k := \sup_{N\in\mathbb{N}} \sup_{i,j\in\{1,\dots,N\}} \mathbb{E}\left|\sqrt{N}A_{ij}^N\right|^k < \infty \tag{3.21}$$

Then

$$\lim_{N\to\infty} \frac{1}{N} tr\left[(A^N)^k\right] = \begin{cases} 0 & \text{if } k \text{ is odd} \\ C_{\frac{k}{2}} & \text{otherwise} \end{cases} \tag{3.22}$$

where C_n is the n^{th} Catalan number $C_k = \binom{2k}{k}/(k+1)$ and the limit holds both in expectation and almost surely.

From this theorem the limit of the spectral density L_{A^N} of random matrices A^N

$$L_{A^N} := \frac{1}{N} \sum_{j=1}^{N} \delta_{\lambda_j(A^N)} \tag{3.23}$$

where λ_j are the eigenvalues of A^N is quickly derived, because the moments of the semicircle law σ are related to the Catalan numbers:

Theorem 3.5. *Under the assumptions of theorem 3.4, if f is any bounded, continuous function defined on the reals, then almost surely:*

$$\int f(x) L_{A^N}(dx) \overset{N\to\infty}{\longrightarrow} \int f(x)\sigma(dx) \tag{3.24}$$

In other words: $L_{A^N} \overset{a.s.}{\Rightarrow} \sigma$.

Remarks:

- Theorem 3.4 is proved using an elegant combinatorial argument whereby the highest order contributions to $\mathrm{tr}A^k$ are seen to come from products of entries of A whose indices form trees on the vertex set $\{1, \ldots, N\}$. The Catalan number C_k is exactly the number of rooted oriented trees with k edges.

- Theorem 3.5 follows from theorem 3.4 because the $2n^{\text{th}}$ moment of σ is exactly C_n.

- Note that both theorems apply to the full symmetric matrix ensembles which we have called Wigner Matrices and the Bernoulli Ensemble in section 3.1.1 if we scale each matrix with the factor $1/\sqrt{n}$. This is the so called Wigner scaling.

The standard GOE

Because of the Normality of the entries and the imposed symmetry of the distribution, it is possible to compute the eigenvalue density for the GOE_n exactly. Doing this involves deriving an explicit expression for the Jacobian of the spectral map $A \mapsto (\Lambda, Q)$ where $A = Q^T \Lambda Q$ (for details, see [11, pp. 19], [23, pp. 5]). Mimicking the notation in [9, chapter 5], the eigenvalue distribution for GOE_n–matrices is given by:

$$P^{(n)}_{\beta=1}(\mathrm{d}\lambda) = \frac{1}{n!\hat{Z}_n} \cdot e^{-\frac{1}{2}\sum_{j=1}^{n}\lambda_n^2} \prod_{1\leq i<j\leq n} |\lambda_i - \lambda_j| \, \mathrm{d}\lambda \tag{3.25}$$

Here the factorial divisor $n!$ is introduced because we don't assume the eigenvalues to be ordered [11, p. 33].

In this case the convergence of the spectral measure to the semicircle law can be proved by establishing a large deviations principle as is done in [6] (for the GUE a similar approach is followed in [9, chapter 6]). Note, however, that the variance of the overwhelming majority (all off–diagonal entries) of the GOE_n entries (3.5) is exactly half of the variance of the entries of the Wigner matrices (3.6). Thus we end up with the following convergence theorem:

Theorem 3.6. *Let L_N be the empirical spectral measure of a Wigner scaled matrix $M/\sqrt{N}$, where $M \sim GOE_N$. Then*

$$L_N(x)\,dx \xRightarrow{a.s.} \sigma(\sqrt{2}x)\,dx \tag{3.26}$$

Remarks

- This means that for large N the rescaled eigenvalues will be concentrated around the interval $[-\sqrt{2}, \sqrt{2}]$.
- Theorem 3.6 is one of the main results in [6].

It turns out that for the Gaussian ensembles much finer properties of the spectrum can be computed explicitly and their limits evaluated than just the average number of eigenvalues in an interval, divided by the size of the matrix (the spectral measure L_N gives us exactly this information). The reason for this lies in two facts:

1. For functions f of the form

$$f(M) = \det(\mathrm{id} + g(M)) = \prod_{j=1}^{N} [1 + g(\lambda_j)] \tag{3.27}$$

the expected value $\mathbb{E}_{\beta=1,N} f(M)$ has an explicit formula [11, p. 85]:

$$\mathbb{E}_{\beta=1,N} f(M) = \sqrt{\det_2 \left(\mathrm{id}_{[L^2(\mathbb{R})]^2} + K_{N,1}\chi_g\right)} \tag{3.28}$$

where $\det_2$ is defined in [11, p. 83], the correlation kernel $K_{N,1}$ is independent of g and the operator $\chi_g : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ is defined by: $\chi_g : f \mapsto g \cdot f$.

2. Interesting quantities about the distribution of eigenvalues can be cast into the form f [11, pp. 86]:

   - Gap probabilities: Let $\Omega \subset \mathbb{R}$, then

   $$\mathbb{P}_{\beta=1,N}\left(\text{No eigenvalue in } \Omega\right) = \mathbb{E}_{\beta=1,N} f(M) \tag{3.29}$$

   for $g = -\mathbb{1}_{\Omega}$.

   - Occupational probabilities (level spacing distributions): This is a generalization of the gap probabilities. It turns out that for $\{\Omega_j\}_{1\leq j\leq k}$ a set of k disjoint subsets of $\mathbb{R}$

   $$\begin{aligned}&\mathbb{P}_{\beta=1,N}\left(\text{Exactly } n_j \text{ eigenvalues in } \Omega_j : 1 \leq j \leq k\right) = \\ &\quad = \frac{1}{n_1! \cdots n_k!} \frac{\partial^{n_1+\cdots+n_k}}{\partial_{\eta_1}^{n_1} \cdots \partial_{\eta_k}^{n_k}}\bigg|_{\eta_i=-1} \left[\det_2\left(\mathrm{id}_{[L^2(\mathbb{R})]^2} + K_{N,1}\sum_{j=1}^{k}\eta_j \mathbb{1}_{\Omega_j}\right)\right]^{\frac{1}{2}}\end{aligned} \tag{3.30}$$

Note that

$$\begin{aligned}&\frac{1}{N}\mathbb{E}_{\beta=1,N}\{\#\text{ eigenvalues in A}\} = L_N(A) = \\ &\qquad \frac{1}{N}\sum_{j=1}^{N} j \cdot \mathbb{P}_{\beta=1,N}\left(\text{Exactly } j \text{ eigenvalues in } A : 1 \leq j \leq k\right)\end{aligned} \tag{3.31}$$

There is a second class of quantities for which asymptotics can be computed using explicit formulae [11, p. 111]. These are the so called correlation functions:

$$R_n(x_1, \ldots, x_n) := \frac{N!}{(N-n)!} \int \cdots \int P_{\beta=1}^{(N)}(x_1, \ldots, x_N) \mathrm{d}x_{n+1} \cdots \mathrm{d}x_N \tag{3.32}$$

Note that

$$\begin{aligned}\int_A R_1(x)\mathrm{d}x &= \int_{\mathbb{R}} \mathbb{1}_A(x)R_1(x)\mathrm{d}x = \\ &= \frac{N!}{(N-1)!}\int_{\mathbb{R}^N} \mathbb{1}_A(x_1)P^{(N)}_{\beta=1}(x_1,\ldots,x_N)\mathrm{d}x_1\cdots\mathrm{d}x_N = \\ &= N\int_{\mathbb{R}^N} \mathbb{1}_A(x_1)P^{(N)}_{\beta=1}(x_1,\ldots,x_N)\mathrm{d}x_1\cdots\mathrm{d}x_N = \\ &= \int_{\mathbb{R}^N} \sum_{j=1}^{N} \mathbb{1}_A(x_j)P^{(N)}_{\beta=1}(x_1,\ldots,x_N)\mathrm{d}x_1\cdots\mathrm{d}x_N = \\ &= \mathbb{E}_{\beta=1,N}\{\#\text{ eigenvalues in A}\} \quad (3.33)\end{aligned}$$

It is well known [11, p. 111], that actually

$$R_k(a_1,\ldots,a_k) = \mathrm{Pf}\left(K^{\#}_{N,1}(a_i,a_j)\right)_{1\le i,j\le k} \quad (3.34)$$

where $K^{\#}_{N,1}$ is strongly related to $K_{N,1}$ appearing in (3.28). So that the limiting behavior of L_N (its convergence to σ) can be construed as one facet of the limiting behavior of the correlation kernel $K_{N,1}$.

Remarks on Universality

The question of universality for various eigenvalue statistics, that is the independence of the limiting behavior of these statistics of the details of the distributions of the matrix entries, apart maybe from their first few moments is a very important issue and progress has been made in recent times in several directions:

1. [51] gives an overview of the proof of the so called Circular Law conjecture, which establishes the analog of the Wigner semicircle law for general complex matrices with iid centered entries of unit variance.

2. Universality for the finer spectral properties of the Gaussian Ensembles usually means establishing that within a symmetry class (ie orthogonal, unitary, symplectic) the specific asymptotic behavior for a spectral quantity is independent of the second order polynomial Q in (3.2). The most famous result in this direction is the so called sine kernel limit [9, theorem 8.16, p. 253] in the bulk scaling of GUE:

$$\lim_{n\to\infty} \frac{1}{\mathcal{K}_{n,2}(x,x)} \mathcal{K}_{n,2}\left(\frac{\xi}{\mathcal{K}_{n,2}(x,x)} + x, \frac{\eta}{\mathcal{K}_{n,2}(x,x)} + x, \right) = \\ = \frac{\sin \pi(\xi - \eta)}{\xi - \eta} \quad (3.35)$$

For the so called edge scaling convergence to the Airy Kernel

$$K_{\text{Airy}} = \frac{\text{Ai}(\xi)\text{Ai}'(\eta) - \text{Ai}'(\xi)\text{Ai}(\eta)}{\xi - \eta} \quad (3.36)$$

implies universality [10, theorem 1.1]

3.2 Minimum Subdiagonal Entry of Hermite–1 Jacobi Matrices

In the introductory section we already outlined the goal of this thesis: Eventually we want to conduct numerical experiments regarding the amount of time it takes for a diagonalization procedure to reach a state where a complete off diagonal block is very small. In the case of Jacobi matrices this translates to a state where the minimum of the absolute values of the subdiagonal entries is small (cf. section 6.2.1).
Clearly the minimum absolute value of the subdiagonal entries of the initial matrix distribution is a random variable. If the scale of this random variable were to become

more and more concentrated around zero as the number of dimensions of the initial matrix increases, we could run into a problem: In this case any sampling procedure would come up with many very small deflation times, because our initial distribution would already put a lot of mass on 'near deflated' matrices – it would hardly be necessary to run the diagonalization procedure anymore. In the following proposition we establish that the minimum subdiagonal entry has a fixed scale as $n \to \infty$. This means that if we choose our deflation tolerance small enough we will not run into the problem that was just described.

Proposition 3.7. *Consider $\chi_1, \ldots, \chi_n$ a sequence of n independent random variables, where χ_j is distributed as a χ distribution with j degrees of freedom. Then for $r > 0$ small enough we have that:*

$$\lim_{n\to\infty} \mathbb{P}\left(\min_{1\leq j\leq n} \chi_j \geq r\right) = p(r) > 0 \tag{3.37}$$

Proof. By independence of the individual variables we have that:

$$\mathbb{P}\left(\min_{1\leq j\leq n} \chi_j \geq r\right) = \prod_{k=1}^{n} \int_r^\infty f_k(s)\mathrm{d}s \tag{3.38}$$

where the χ_k–density f_k is given by:

$$f_k(x) = D_k^{-1}\gamma_k(x) \tag{3.39}$$

$$D_k = 2^{k/2-1}\Gamma\left(\frac{k}{2}\right) \tag{3.40}$$

$$\gamma_k(x) = \begin{cases} x^{k-1}e^{-x^2/2} & k \geq 1 \\ 0 & k < 1 \end{cases} \tag{3.41}$$

Thus, by analyticity of (the antiderivative $\hat{f}_k(x) = 1 - \mathbb{P}(\chi_k \geq x)$ of) f_k:

$$\mathbb{P}\left(\min_{1\leq j\leq n} \chi_j \geq r\right) = \prod_{k=1}^{n}\left[1 - \hat{f}_k(r)\right] = \prod_{k=1}^{n}\left[1 - \sum_{j=0}^{\infty}\frac{r^j}{j!}\hat{f}_k^{(j)}(0)\right] =$$
$$= \prod_{k=1}^{n}\left[1 - \sum_{j=1}^{\infty}\frac{r^j}{j!}f_k^{(j-1)}(0)\right] = \prod_{k=1}^{n}\left[1 - D_k^{-1}\sum_{j=1}^{\infty}\frac{r^j}{j!}\gamma_k^{(j-1)}(0)\right] \quad (3.42)$$

The first goal of this proof is to bring (3.42) into a more manageable form. In particular, what can we say about the derivatives $\gamma_k^{(j)}(0)$? We notice immediately the following recursive structure:

$$\gamma_k^{(1)} = (k-1)\gamma_{k-1} - \gamma_{k+1} \quad (3.43)$$

$$\gamma_k^{(2)} = (k-1)(k-2)\gamma_{k-2} - (2k-1)\gamma_k + \gamma_{k+2} \quad (3.44)$$

Induction allows us to show that:

$$\gamma_k^{(j)}(x) = \sum_{m=0}^{j} C_m^{(j,k)}\gamma_{k-j+2m}(x) \quad (3.45)$$

Here, because $\gamma_k^{(0)} = C_0^{(0,k)}\gamma_k = \gamma_k$, we derive

$$C_m^{0,k} = \begin{cases} 1 & m = 0 \\ 0 & m \neq 0 \end{cases} \quad (3.46)$$

take the x–derivative on both sides of the equation (3.45) and use (3.43) to yield:

$$\gamma_k^{(j+1)} = \sum_{m=0}^{j} C_m^{(j,k)}\gamma'_{k-j+2m} =$$
$$= \sum_{m=0}^{j} C_m^{(j,k)} \cdot [(k-j+2m-1)\gamma_{k-j+2m-1} - \gamma_{k-j+2m+1}] \quad (3.47)$$

Now, because of

$$k - j + 2m - 1 = k - (j + 1) + 2m = k - j + 2(m - 1) + 1 \tag{3.48}$$

(3.47) means that

$$\gamma_k^{(j+1)} = \sum_{m=0}^{j+1} \underbrace{\left[(k - (j + 1) + 2m) \cdot C_m^{(j,k)} - C_{m-1}^{(j,k)}\right]}_{=:C_m^{(j+1,k)}} \cdot \gamma_{k-(j+1)+2m} \tag{3.49}$$

This shows that for $k, j \geq 1$ the following recursion holds for the coefficients:

$$C_m^{(j,k)} = \\ = \begin{cases} -C_{m-1}^{(j-1,k)} + (k - j + 2m) C_m^{(j-1,k)} & 1 \leq k - j + 2m \leq k + j \\ 0 & 0 \leq 2m < j - k + 1 \text{ or } m > j \end{cases} \tag{3.50}$$

Equation (3.41) demonstrates that $\gamma_k(0) \equiv 0$ for all $k > 1$ and $\gamma_1(0) = 1$. This allows us to conclude that

$$\gamma_k^{(j)}(0) = \begin{cases} 0 & j - k + 1 < 0 \\ 0 & j - k + 1 > 0 \text{ and odd} \\ C_{\frac{1}{2}(j-k+1)}^{(j,k)} & j - k + 1 \geq 0 \text{ and even} \end{cases} \tag{3.51}$$

The explanation goes as follows: The sum in (3.41) evaluated at zero has contributions only from the term $\gamma_1(0)$. We separate three cases:

1. The first case in (3.51) covers the first few derivatives $0 \leq j < k - 1$, where the term γ_1 can not show up in the expansion (3.45).

2. The second case in (3.51) covers the cases where γ_1 can not show up in the expansion (3.45), because the "starting point" $k - j$ of the expansion is even.

3. The third case in (3.51) covers the only situation where $\gamma_k^{(j)}$ can actually be different from zero, ie when γ_1 is contained in the sum (3.45). For that to happen, the derivative j has to be high enough that the "starting point" $k-j$ of the sum lies at or to the left of 1. In addition $k-j$ has to be odd, because otherwise we could not reach 1 by adding m multiples of 2 to $k-j$. If all this is satisfied, summand $m_1(j,k)$ which contains γ_1 as a factor is given by:

$$m_1(j,k) = \frac{j-k+1}{2} \tag{3.52}$$

Summarizing, (3.42) leads to:

$$\mathbb{P}\left(\min_{1\le j\le n}\chi_j \ge r\right) = \prod_{k=1}^{n}\left[1 - D_k^{-1}\sum_{\alpha=0}^{\infty}\frac{C_\alpha^{(k-1+2\alpha,k)}}{(k+2\alpha)!}r^{k+2\alpha}\right] = \\ = \prod_{k=1}^{n}\left[1 - \frac{r^k}{D_k}\sum_{\alpha=0}^{\infty}\frac{C_\alpha^{(k-1+2\alpha,k)}}{(k+2\alpha)!}\cdot r^{2\alpha}\right] \tag{3.53}$$

We can now turn to the second part of the proof: Deriving growth estimates for the terms

$$\kappa(k,\alpha) := \frac{|C_\alpha^{(k-1+2\alpha,k)}|}{(k+2\alpha)!} \tag{3.54}$$

Therefore, note that $C_0^{(k-1,k)} = (k-1)!$, so that $\kappa(k,0) = 1/k$. Moreover it is true that

$$|C_{\alpha+1}^{(k-1+2(\alpha+1),k)}| \le (k+\alpha-1)! + |C_\alpha^{(k-1+2\alpha,k)}|\cdot\sum_{j=0}^{k-2+\alpha}\frac{(j+1)!}{j!} = \\ = (k+\alpha-1)! + |C_\alpha^{(k-1+2\alpha,k)}|\cdot\sum_{j=1}^{k-1+\alpha} j = \\ = (k+\alpha-1)! + |C_\alpha^{(k-1+2\alpha,k)}|\cdot\frac{(k-1+\alpha)(k+\alpha)}{2} \tag{3.55}$$

as can be read off of figure ... Overall we have the estimate:

$$\kappa(k,\alpha) \leq (5/8)^{\alpha},\ \alpha \geq 0,\ k \geq 1 \tag{3.56}$$

which we establish by induction:

1. Consider the base case $\alpha = 0$. Clearly $\kappa(k,0) = 1/k \leq 1 = (5/8)^0$ holds for all $k \geq 1$.

2. Consider $\alpha = 1$. Compare with figure ... to obtain

$$|C_1^{(k+1,k)}| = k! + (k-1)! \cdot \sum_{j=0}^{k-2} \frac{(j+1)!}{j!} = \\ = k! + (k-1)! \cdot \frac{k(k-1)}{2} = \frac{(k+1)!}{2} \tag{3.57}$$

   Consequently $\kappa(k,1) = [2(k+2)]^{-1} \leq 1/6 \leq (5/8)^3$

3. Now assume that (3.56) holds, let $\alpha \geq 1$ and consider

$$\kappa(k,\alpha+1) = \frac{C_{\alpha+1}^{(k+2\alpha+1,k)}}{(k+2\alpha+2)!} \leq \\ \leq \frac{(k+\alpha-1)!}{(k+2\alpha+2)!} + \frac{(k-1+\alpha)(k+\alpha)|C_{\alpha}^{(k-1+2\alpha,k)}|}{2(k+2\alpha+2)!} \leq \\ \leq \frac{1}{(k+\alpha)^{\alpha+3}} + \frac{|C_{\alpha}^{(k-1+2\alpha,k)}|}{(k+2\alpha)!} \cdot \frac{(k-1+\alpha)(k+\alpha)}{2(k+2\alpha+1)(k+2\alpha+2)} \leq \\ \leq 2^{-\alpha-3} + \frac{1}{2}(5/8)^{\alpha} \leq (5/8)^{\alpha}\left(\frac{1}{2} + \frac{1}{8}\right) \leq (5/8)^{\alpha+1} \tag{3.58}$$

   This proves the claim.

Going back to equation (3.53) the estimate (3.56) implies:

$$\mathbb{P}\left(\min_{j\leq 1\leq n}\chi_j \geq r\right) = \prod_{k=1}^{n}\left[1-\frac{r^k}{D_k}\sum_{\alpha=0}^{\infty}\kappa(k,\alpha)r^{2\alpha}\right] \geq$$
$$\geq \prod_{k=1}^{n}\left[1-\frac{r^k}{D_k}\sum_{\alpha=0}^{\infty}(5/8)^{\alpha}r^{2\alpha}\right] =: S_n(r) \quad (3.59)$$

In the third and last part of the proof we will show that for small enough r, $S(r) := \lim_{n\to\infty} S_n(r)$ exists and is greater than zero. The idea is to rewrite:

$$S_n(r) = 1 - \sum_{k=1}^{\infty} A_k^{(n)} r^k \quad (3.60)$$

The following statements follow immediately from (3.59):

1. $A_1^{(n)} \equiv D_1^{-1} > 0$ for all n.
2. $A_m^{(n)} \equiv A_m^{(m)}$ for all $n \geq k+1$. This is because all the monomials showing up in the factors of $S_n(r)$ with $k > m$ are of degree at least r^{m+1}, so that their coefficients can't contribute to the coefficient A_m^n of the monomial r^m.

We will now estimate the growth of $|A_k^{(k)}|$. Firstly, note that $D_k^{-1} \leq (5/8)^k$ for $k \geq 8$. Thus, for $c_1, \ldots$ with $c_n = 1$ for $n \geq 8$ we have that

$$S_n(r) \geq \prod_{k=1}^{n}\left[1-\sum_{j=k}^{\infty}(5/8)^{j/2}r^j\right] = \prod_{k=1}^{n}\left[1-(1-5r/8)^{-1}\left(\sqrt{\frac{5}{8}}r\right)^k\right] \geq$$
$$\geq \prod_{k=1}^{n}\left[1-\left(\sqrt{\frac{5}{8}}r\right)^k\right] = \exp\left\{\sum_{k=1}^{n}\log\left[1-\left(\sqrt{\frac{5}{8}}r\right)^k\right]\right\} \geq$$
$$\geq \exp\left\{-C\cdot\sum_{k=1}^{n}\left(\sqrt{\frac{5}{8}}r\right)^k\right\} \quad (3.61)$$

The last term in this expression clearly converges for r small enough as $n \to \infty$, call

this limit $s(r) > 0$. Note that $S_n(r)$ is a monotonously decreasing sequence in n, which is bounded from below by $s(r)$. This means that $S(r) = \lim_{n\to\infty} S_n(r)$ exists by standard Real Analysis and is strictly larger than zero.

The fact that $\mathbb{P}\left(\min_{1\leq j\leq n} \chi_j \geq r\right)$ converges follows, because $\min_{1\leq j\leq n} \chi_j$ is a monotonously decreasing sequence of random variables, so that the sequence of probabilities is itself monotonously decreasing in n. On the other hand we have that

$$\mathbb{P}\left(\min_{1\leq j\leq n} \chi_j \geq r\right) \geq S(r) \tag{3.62}$$

This proves the claim. □

We make the following remarks:

- This proof is inspired by the limit law for the maximum of iid random variables of regular variation by Fisher and Gnedenko as presented on pp. 277 in [21]
- The proposition shows that the scale of the minimum of a sequence $(\chi_j)_{1\leq j\leq n}$ of χ_j–distributed random variables stabilizes as $n \to \infty$. This makes intuitive sense, as the mass of the χ_j–distribution keeps shifting to the right as j becomes larger.

Chapter Four

Hamiltonian Systems

This chapter contains background information about Hamiltonian systems and symplectic manifolds. In particular we discuss properties of Jacobi matrices as coadjoint orbits. We also highlight connections to Flaschka's variables.

4.1 General Setting

In this section we briefly outline the abstract setting for the systems we investigate. General references for this chapter are [12, 42]. In our discussion we will assume that the reader is comfortable with the notions of elementary differential geometry like manifolds, tangent spaces and differential forms.
Hamiltonian systems are defined on symplectic manifolds, ie. a $2n$–dimensional manifolds equipped with a symplectic structure.

Definition 4.1. *A symplectic structure ω on an even dimensional manifold M is a 2–form with the following properties.*

1. *ω is closed, ie. $d\omega = 0$.*
2. *ω is nowhere degenerate, ie. for every $x \in M$ and $X \in T_xM$ with $X \neq 0$ there is a vector $Y \in T_xM$ with $\omega_x(X, Y) \neq 0$.*

If these requirements are satisfied we call the pair (M, ω) a symplectic manifold.

Recall, that a 2–form in differential geometry is antisymmetric: $\omega_x(u, v) = -\omega(v, u)$.
Functions $H : M \to \mathbb{R}$ can be used to generate vector fields on M using the symplectic structure ω. Clearly at any point $x \in M$, the differential dH_x is a 1–form, ie.

$T_xM \ni v \mapsto dH_x(v) \in \mathbb{R}$ is a linear map. The following lemma tells us how to turn this 1–form into a vectorfield:

Lemma 4.2. *Let (M, ω) denote a symplectic manifold and let α denote a 1–form on M. Then there exists a vector field X_α such that $\alpha = v \mapsto \omega(X_\alpha, \cdot)$.*

Proof. The map $\Psi : v \mapsto \omega_x(v, \cdot)$ is an isomorphism from T_xM to $(T_xM)^*$, because T_xM it is finite dimensional and Ψ is linear and bijective. The first two properties are clear, the third property follows, because

1. It is injective: Assume that $\Psi(v) = \Psi(u)$ with $u \neq v$. Then $0 = \Psi(u - v) = \omega_x(u - v, \cdot)$, which contradicts nondegeneracy.
2. Injectivity implies that the dimension of the kernel of Ψ is zero, so that by the rank–nullity–theorem from Linear Algebra, the map is surjective (cf. [36, p. 20]).

The claim follows by setting $v_x := \Psi^{-1}(\alpha_x)$. Thus for every point $m \in M$ let $X_\alpha(m) := v_m$. This proves the claim. □

In the following we will denote the vectorfield generated from the Hamiltonian function H using lemma (4.1) by v_H. Lemma 4.1 demonstrates the role the non degeneracy condition for ω plays.
The next notion we introduce here is the notion of a Poisson bracket on a symplectic manifold. For two Hamiltonian functions H and K we define their Poisson bracket at $x \in M$ by:

$$\{H, K\}(x) = \omega_x(v_H, v_K) \tag{4.1}$$

Note that by definition:

$$\{H, K\}(x) = dH_x(v_K) = v_K(H) \tag{4.2}$$

ie. $\{H, K\}(x)$ equals the derivative of H in the direction of $v_K(x)$. Consider the commutator $[\cdot, \cdot]$ of Hamiltonian vector fields:

$$[v_H, v_K](L) = v_H(v_K(L)) - v_K(v_H(L)) \tag{4.3}$$

which is the difference between the mixed directional derivatives of L. In order to elucidate the role which the closedness property of ω plays for the theory, we prove the following lemma (cf. [42, p. 76])

Lemma 4.3. *For a nondegenerate 2–form on M the following conditions are equivalent: $d\omega = 0 \Leftrightarrow [v_F, v_G] = v_{\{F,G\}} \Leftrightarrow$ The Jacobi identity*

$$\{\{H, K\}, L\} + \{\{K, L\}, H\} + \{\{L, H\}, K\} = 0 \tag{4.4}$$

holds

Proof. Note that by a well known formula from exterior calculus:

$$\begin{aligned} d\omega(v_H, v_K, v_L) = v_H(\omega(v_K, v_L)) - v_K(\omega(v_H, v_L) + v_L(\omega(v_H, v_K)) - \\ - \omega([v_H, v_K], v_L) + \omega([v_H, v_L], v_K) - \omega([v_K, v_L], v_H) = \\ = v_H(\omega(v_K, v_L)) + v_K(\omega(v_L, v_H) + v_L(\omega(v_H, v_K)) - \\ - \omega([v_H, v_K], v_L) - \omega([v_L, v_H], v_K) - \omega([v_K, v_L], v_H) \end{aligned} \tag{4.5}$$

Note also that from equation 4.2

$$\begin{aligned}\omega([v_H, v_K], v_L) &= -dL([v_H, v_K]) = -[v_H, v_K](L) = \\ &= -v_H(v_K(L)) + v_K(v_H(L)) = -v_H(\{L, K\}) + v_K(\{L, H\}) = \\ &= -\{\{L, K\}, H\} + \{\{L, H\}, K\} \end{aligned} \tag{4.6}$$

Putting everything together we obtain that

$$d\omega(v_H, v_K, v_L) = 3\left[\{\{H, K\}, L\} + \{\{K, L\}, H\} + \{\{L, H\}, K\}\right] \tag{4.7}$$

On the other hand, from the computation in 4.6 it is clear that

$$\begin{aligned}[v_H, v_K](L) &= -\{\{K, L\}, H\} - \{\{L, H\}, K\} = \\ &= -\{\{K, L\}, H\} - \{\{L, H\}, K\} + \{L, \{H, K\}\} + v_{\{H,K\}}(L) = \\ &= -3d\omega(v_H, v_K, v_L) + v_{\{H,K\}}(L) \end{aligned} \tag{4.8}$$

This proves the claim. □

Let us summarize:

1. We have set the stage for the abstract consideration of Hamiltonian ODEs on symplectic manifolds. In particular: Let (M, ω) denote a symplectic manifold, $H : M \to \mathbb{R}$ a smooth function (the 'Hamiltonian'). Then we can consider the ODE

$$\dot{x} = v_H(x) \tag{4.9}$$

where v_H is the vectorfield defined using the symplectic structure ω.

2. We have introduced the commutator of vectorfields and the Poisson bracket of Hamiltonian functions and proved some elementary connections between the two.

We are still missing an example of these ideas in action. For all the examples that we'll be considering only one symplectic manifold will be playing a role. This symplectic manifold falls into a class of standard examples of symplectic manifolds generated from coadjoint orbits of groups on their dual Lie Algebras. The following section develops abstractly the example of the coadjoint orbits with their natural symplectic structure, the Kostant–Kirillov bracket.

4.2 Coadjoint Orbits as Symplectic Manifolds

A group that is also a differentiable manifold is called a Lie Group. Let $(G, \circ)$ a Lie group. The tangent space $\mathfrak{g}$ of T_eG at the group identity e equipped with the Lie bracket $[\cdot,\cdot]_G$ corresponding to the group G (to be defined below in equation (4.14)) is called the group's Lie algebra. Now define the left and right actions of the group on itself as usual:

$$\mathcal{L}_A B = A \circ B \tag{4.10}$$

$$\mathcal{R}_A B = B \circ A \tag{4.11}$$

Let $\mathfrak{L}(X)$ denote the set of linear maps from X to X. The adjoint representation $\mathrm{Ad} : G \to \mathfrak{L}(\mathfrak{g})$ of G on $\mathfrak{g}$ is defined by

$$\mathrm{Ad}A(\xi) = \frac{d}{dt} A \circ e^{t\xi} \circ A^{-1}\Big|_{t=0} = \mathrm{d}\mathcal{L}_A \mathrm{d}\mathcal{R}_{A^{-1}}\xi \tag{4.12}$$

and there is an infinitesimal representation $\mathrm{ad} : \mathfrak{g} \to \mathfrak{L}(\mathfrak{g})$ of $\mathfrak{g}$ on $\mathfrak{g}$ of the form

$$\mathrm{ad}\xi(\eta) := \frac{\mathrm{d}}{\mathrm{d}t}\left[\mathrm{Ad}e^{t\xi}\right](\eta)\Big|_{t=0} \tag{4.13}$$

This infinitesimal adjoint representation is what's denoted with $[\cdot,\cdot]_G$. More succinctly, for $\xi, \eta \in \mathfrak{g}$ we define

$$[\xi,\eta]_G := \mathrm{ad}\xi(\eta) \in \mathfrak{g} \tag{4.14}$$

Lemma 4.4. *The bracket $[\cdot,\cdot]_G$ is bilinear, antisymmetric and satisfies the Jacobi identity*

$$[[\xi,\eta]_G,\zeta]_G + [[\eta,\zeta]_G,\xi]_G + [[\zeta,\xi]_G,\eta]_G = 0 \tag{4.15}$$

Proof. We prove the three properties one after the other:

1. *Bilinearity* follows from the following computation:

$$\begin{aligned}\mathrm{ad}(a\xi_1 + b\xi_2)(\eta) = \frac{d}{ds}\Big|_{s=0}\frac{d}{dt}\Big|_{t=0} e^{s(a\xi_1+b\xi_2)}e^{t\eta}e^{-s(a\xi_1+b\xi_2)} = \\ = (a\xi_1+b\xi_2)\eta - \eta(a\xi_1+b\xi_2) = a\mathrm{ad}\xi_1(\eta) + b\mathrm{ad}\xi_2(\eta)\end{aligned} \tag{4.16}$$

   Linearity for the second argument follows similarly.

2. *Antisymmetry* follows from a computation similar to the one in (4.16).

3. *Jacobi identity*: Based on equation (4.16) we conclude:

$$[[\xi,\eta]_G,\zeta]_G = \xi\eta\zeta - \eta\xi\zeta - \zeta\xi\eta + \zeta\eta\xi \tag{4.17}$$

Now expand the other two brackets in (4.15) and collect terms to yield the desired result.

□

Mediated by the duality pairing $\langle \cdot, \cdot \rangle : \mathfrak{g}^* \times \mathfrak{g} \to \mathbb{R}$ we define the coadjoint representations $Ad' : G \to \mathfrak{L}(\mathfrak{g}^*)$ and its infinitesimal version $\mathrm{ad}' : \mathfrak{g} \to \mathfrak{L}(\mathfrak{g}^*)$ by adjointness:

$$\langle \mathrm{Ad}' A(\alpha), \xi \rangle = \left\langle \alpha, \mathrm{Ad} A^{-1}(\xi) \right\rangle$$
$$\langle \mathrm{ad}' \xi(\alpha), \eta \rangle = \langle \alpha, -\mathrm{ad}\xi(\eta) \rangle$$

Note – in spite of this definition – that by linearity:

$$\left\langle \left.\frac{d}{dt}\right|_{t=0} \mathrm{Ad}' e^{t\xi}(\alpha), \eta \right\rangle = \left.\frac{d}{dt}\right|_{t=0} \left\langle \alpha, \mathrm{Ad} e^{-t\xi}(\eta) \right\rangle =$$
$$= \left\langle \alpha, \left.\frac{d}{dt}\right|_{t=0} \mathrm{Ad} e^{-t\xi}(\eta) \right\rangle = \langle \alpha, -\mathrm{ad}\xi(\eta) \rangle = \langle \mathrm{ad}' \xi(\alpha), \eta \rangle \quad (4.18)$$

In this sense the infitesimal coadjoint representation is the derivative of the coadjoint representation. Coadjoint orbits are defined on the dual $\mathfrak{g}^*$ of the Lie algebra as follows:

$$O_\alpha = \{ \mathrm{Ad}' A(\alpha) : A \in G \} \subset \mathfrak{g}^* \quad (4.19)$$

What can we say about the tangent space at $\beta = \mathrm{Ad}' A(\alpha) \in \mathfrak{g}^*$ for some $A \in G$ to O_α? Clearly all trajectories through β in O_α have the form $\mathrm{Ad}' e^{t\xi}(\beta)$ for some $\xi \in \mathfrak{g}$.

The tangent space is thus given by all vectors of the form:

$$\left.\frac{d}{dt}\right|_{t=0} \mathrm{Ad}' e^{t\xi}(\beta) \overset{(4.18)}{=} \mathrm{ad}'\xi(\beta) \tag{4.20}$$

Summarizing we find that at any given element of O_α the tangent space is equal to

$$T_\beta O_\alpha = \{(\beta, \mathrm{ad}'_\xi \beta) : \xi \in \mathfrak{g}\}$$

that is to say that the map $\xi \mapsto (\beta, \mathrm{ad}'_\xi \beta)$ is surjective from $\mathfrak{g}$ to $T_\beta O_\alpha$. We can now define a skew symmetric bilinear form, the Kostant–Kirillov bracket, ω which makes (O_α, ω) a symplectic manifold:

$$\omega_\beta \left((\beta, \mathrm{ad}'_\xi \beta), (\beta, \mathrm{ad}'_\eta \beta)\right) := \langle \beta, [\xi, \eta]_G \rangle \tag{4.21}$$

From the definition it is clear that ω is bilinear and skew symmetric. It remains to be shown that it is well defined (due to the possibility of non–injectivity) and that it is non–degenerate.

Lemma 4.5. *The form ω defined in equation (4.21) is a symplectic form on O_α.*

Proof. We check that ω is well defined, non–degenerate and closed.

1. ω is well defined: Let $(\beta, u), (\beta, v) \in T_\beta O_\alpha$ and assume that $v = \mathrm{ad}'_\xi \beta = \mathrm{ad}'_{\tilde{\xi}} \beta$ and $u = \mathrm{ad}'_\eta \beta = \mathrm{ad}'_{\tilde{\eta}} \beta$. We have to show that $\langle \beta, [\xi, \eta] \rangle = \langle \beta, [\tilde{\xi}, \tilde{\eta}] \rangle$. By linearity it is clear that $\xi - \tilde{\xi}$ and $\eta - \tilde{\eta}$ are contained in $\mathrm{Ker}(\mathrm{ad}'_\cdot \beta)$.
   For $\zeta \in \mathrm{Ker}(\mathrm{ad}'_\cdot \beta)$ we have that for all $\theta \in \mathfrak{g}$: $\langle \mathrm{ad}'_\zeta \beta, \theta \rangle = \langle \beta, [\zeta, \theta] \rangle = 0$. It remains to do the following computation:

$$\langle \beta, [\xi, \eta] \rangle = \langle \beta, [\tilde{\xi} + (\xi - \tilde{\xi}), \tilde{\eta} + (\eta - \tilde{\eta})] \rangle = \langle \beta, [\tilde{\xi}, \tilde{\eta}] \rangle$$

where the last equation holds due to bilinearity.

2. ω is non–degenerate: We have to show that if $\omega_\beta(u,v)=0$ for all $(\beta,v)\in T_\beta O_\alpha$, then $u=0$. Assume $u=\mathrm{ad}'_\xi\beta$ and $v=\mathrm{ad}'_\eta\beta$, then our assumption translates to $\langle\beta,[\xi,\eta]\rangle=0$ for all η, but this (see above) means that $\xi\in\mathrm{Ker}(\mathrm{ad}'\beta)$ which proves the claim.

3. ω is closed: Fix $\xi,\eta\in\mathfrak{g}$ and consider the function

$$H:\beta\mapsto\omega_\beta((\beta,\mathrm{ad}'_\xi(\beta),(\beta,\mathrm{ad}'_\eta(\beta))=\langle\beta,[\xi,\eta]_G\rangle \tag{4.22}$$

Then by linearity:

$$\begin{aligned}\langle dH_\beta,\mathrm{ad}'\zeta(\beta)\rangle=\langle\mathrm{ad}'\xi(\beta),[\xi,\eta]_G\rangle=\langle\beta,-[\zeta,[\xi,\eta]_G]_G\rangle=\\ =\langle\beta,[[\xi,\eta]_G,\zeta]_G\rangle=\omega_\beta((\beta,\mathrm{ad}'[\xi,\eta]_G(\beta),(\beta,\mathrm{ad}'\zeta(\beta))\end{aligned} \tag{4.23}$$

This computation implies that for the Hamiltonian H from equation 4.22 we have $v_H(\beta)=\mathrm{ad}'[\xi,\eta]_G(\beta)$. Now let us try to show the Jacobi identity (4.4), which is equivalent to closedness of ω by lemma 4.3. Choose three Hamiltonian functions F_i, $1\le i\le 3$, and choose ξ_i such that $v_{F_i}(\beta)=\mathrm{ad}'\xi_i(\beta)$. Then

$$\{F_1,F_2\}=\omega_\beta(v_{F_1},v_{F_2})=\langle\beta,[\xi_1,\xi_2]_G\rangle \tag{4.24}$$

$$\{\{F_1,F_2\},F_3\}=\omega_\beta(\mathrm{ad}'[\xi_1,\xi_2]_G(\beta),\mathrm{ad}'\xi_3(\beta))=\langle\beta,[[\xi_1,\xi_2]_G,\xi_3]_G\rangle \tag{4.25}$$

Now the Jacobi identity follows from the Jacobi identity for $[\cdot,\cdot]_G$ in lemma 4.4.

□

Again, let us summarize: We defined the coadjoint orbit O_α, a subset of the dual $\mathfrak{g}^*$ of the Lie algebra $\mathfrak{g}$, and we have defined a natural symplectic form ω (called the Kostant–Kirillov bracket) on the manifold O_α. Hamiltonian vectorfields on O_α are defined as usual: Assume that we're given a Hamiltonian function H defined on O_α. $\mathrm{d}H$ is a one form on O_α and the Hamiltonian vectorfield v_H is given by

$$\langle \mathrm{d}H, u \rangle = \omega_\beta(v_H, u) \tag{4.26}$$

then we can consider flows given by by $\dot{\alpha} = v_H(\alpha)$.
In this section we introduced a class of symplectic manifolds arising from equipping coadjoint Lie Group orbits with a natural symplectic structure, the Kostant–Kirillov bracket. The next section discusses an actual hands on example for a symplectic manifold that is a coadjoint Lie Group orbit: The zero trace Jacobi Matrices.

4.3 Jacobi Matrices and their Bracket

Eventually we will use the Kostant–Kirillov form to define Hamiltonian vectorfields on trace free Jacobi matrices, because the Hamiltonian flows arising in this way all have connections to eigenvalue algorithms. The realization that trace free Jacobi matrices are a coadjoint orbit of the lower triangular matrices is due independently to [1, 33] In this section we will use this realization and explicitly compute the forms and brackets introduced in section 4.2. In particular we will:

1. Compute a coordinate representation of the Kostant–Kirillov bracket associated with this coadjoint orbit (cf. lemma 4.6).

2. For a Hamiltonian on the coadjoint orbit, compute its Hamiltonian vectorfield

(cf. lemma 4.7).

3. Compute a coordinate representation of the Poisson bracket corresponding to the Kostant–Kirillov bracket (cf. lemma 4.9).

To accomplish the above tasks, we have to first specialize the ideas in section 4.2 to the case of real matrix groups. In particular here $(G, \circ)$ denotes a matrix group, its Lie algebra $\mathfrak{g}$ is endowed with the usual Lie bracket of matrices: $[\xi, \eta] = \xi\eta - \eta\xi$ and we introduce an inner product $(\xi, \eta)_{\mathfrak{g}\times\mathfrak{g}} = \mathrm{tr}(\xi\eta)$ (the same inner product can be used on $\mathfrak{g}^*$). This inner product will allow us to identify $\mathfrak{g}^*$ with some vector space of matrices.

Let L denote the group of lower triangular matrices with positive diagonal entries and let $\mathfrak{l}$ denote its Lie algebra (all lower triangular matrices). Using the inner product $(\cdot,\cdot)_{\mathfrak{l}\times\mathfrak{l}}$ we identify $\mathfrak{l}^*$ with the space of symmetric matrices $\mathfrak{s}$. We will compute all the representations defined in 4.2: Clearly $\mathcal{L}_A B = AB$, $\mathcal{R}_A B = BA$. Since we're concerned with linear spaces, the tangent space $T_A L$ at any matrix $A \in L$ is the same so that $TL = L \times \mathfrak{l}$. The differentials of left and right multiplication are thus given by:

$$\mathrm{d}\mathcal{L}_A : TL \to TL \tag{4.27}$$

$$\mathrm{d}\mathcal{L}_A : (B, \xi) \to (AB, A\xi) \tag{4.28}$$

$$\mathrm{d}\mathcal{R}_A : TL \to TL \tag{4.29}$$

$$\mathrm{d}\mathcal{R}_A : (B, \xi) \to (BA, \xi A) \tag{4.30}$$

This implies that

$$\mathrm{Ad} : L \to \mathfrak{L}(\mathfrak{l}) \tag{4.31}$$

$$\mathrm{Ad}_A : \xi \mapsto A\xi A^{-1} \tag{4.32}$$

$$\mathrm{ad} : \mathfrak{l} \to \mathfrak{L}(\mathfrak{l}) \tag{4.33}$$

$$\mathrm{ad}_\xi : \eta \mapsto \left.\frac{\mathrm{d}}{\mathrm{d}t} e^{t\xi}\eta e^{-t\xi}\right|_{t=0} = [\xi, \eta] \tag{4.34}$$

The coadjoint representations are defined by duality. We have $\mathrm{Ad}' : L \to \mathfrak{L}(\mathfrak{l}^*)$, i.e. $\mathrm{Ad}'_A\alpha : \mathfrak{l} \to \mathbb{R}$, namely:

$$\langle \mathrm{Ad}'_A\alpha, \xi\rangle = \langle \alpha, \mathrm{Ad}_{A^{-1}}\xi\rangle \tag{4.35}$$

after having identified $\mathfrak{l}^* \simeq \mathfrak{s}$ we can assume that α is a symmetric matrix, so that in particular:

$$\langle \mathrm{Ad}'_A\alpha, \xi\rangle = \mathrm{tr}\left(\alpha A^{-1}\xi A\right) = \mathrm{tr}\left(A\alpha A^{-1}\xi\right) = \mathrm{tr}\left(\left(A\alpha A^{-1}\right)\xi\right) \tag{4.36}$$

We define the following projection:

$$\mathrm{pr}_{\mathfrak{s}} M := M_+ + M_+^T + M_0 \tag{4.37}$$

which symmetrizes any matrix M by reflecting the strictly upper diagonal part along the diagonal. Then, because ξ is lower triangular we have that

$$\langle \mathrm{Ad}'_A\alpha, \xi\rangle = \mathrm{tr}\left(\left(\mathrm{pr}_{\mathfrak{s}} A\alpha A^{-1}\right)\xi\right) \tag{4.38}$$

and, hence:

$$\mathrm{Ad}'_A\alpha = \mathrm{pr}_{\mathfrak{s}} A\alpha A^{-1} \in \mathfrak{s} \simeq \mathfrak{l}^* \tag{4.39}$$

With regards to the infinitesimal representation we have $\mathrm{ad}' : \mathfrak{l} \to \mathfrak{L}(\mathfrak{l}^*)$, i.e. $\mathrm{ad}'_\xi\alpha : \mathfrak{l} \to \mathbb{R}$. In particular:

$$\langle \mathrm{ad}'_\xi\alpha, \eta\rangle = \langle \alpha, -\mathrm{ad}_\xi\eta\rangle = \langle \alpha, -[\xi,\eta]\rangle \tag{4.40}$$

Again we can assume that α is a symmetric matrix, so that

$$\begin{aligned}\langle \mathrm{ad}'_\xi\alpha, \eta\rangle = \langle \alpha, -\mathrm{ad}_\xi\eta\rangle = -\langle \alpha, \mathrm{ad}_\xi\eta\rangle = -\langle \mathrm{ad}_\xi\alpha, \eta\rangle = \\ = \langle -\mathrm{ad}_\xi\alpha, \eta\rangle = \langle [\alpha,\xi], \eta\rangle = \langle \mathrm{pr}_{\mathfrak{s}}[\alpha,\xi], \eta\rangle\end{aligned} \tag{4.41}$$

As above we convince ourselves that

$$\mathrm{ad}'_\xi\alpha = \mathrm{pr}_{\mathfrak{s}}[\alpha,\xi] \in \mathfrak{s} \simeq \mathfrak{l}^* \tag{4.42}$$

Let $\mathfrak{j} := O_\alpha \subset \mathfrak{s}$ denote the coadjoint orbit of the trace free Jacobi matrices. For $\beta \in \mathfrak{j}$ we want to determine an explicit form for the Kostant–Kirillov from ω_β. In particular, let $u, v \in T_\beta\mathfrak{j}(= \mathfrak{j})$, then we know that $u = \mathrm{ad}'_\xi\beta$, $v = \mathrm{ad}'_\eta\beta$ for matrices $\xi, \eta \in \mathfrak{l}$. By definition:

$$\omega_\beta(u,v) = \omega_\beta(\mathrm{ad}'_\xi\beta, \mathrm{ad}'_\eta\beta) = \mathrm{tr}\beta[\xi,\eta] \tag{4.43}$$

So that an explicit formula for the Kostant–Kirillov form can be obtained, if we're able to invert the mapping $\xi \mapsto \mathrm{ad}'_\xi\beta$. This we will do now. First of all we have to evaluate the expression $\mathrm{pr}_{\mathfrak{s}}[\alpha,\xi]$ for a trace free Jacobi matrix α and a lower triangular matrix ξ. Clearly the relevant entries are the diagonal and the super

diagonal of $[\alpha, \xi]$, since the projection equates the superdiagonal to the subdiagonal. Because of well definedness of the Kostant–Kirillov form any $\xi \in \mathfrak{l}$ will do the job, so in order to solve the ensuing linear equations we impose the artificial conditions that ξ be trace free and zero except for the diagonal and the first sub diagonal. We compute

$$\begin{aligned} [\alpha, \xi]_{ii} &= (\alpha\xi^T)_{ii} - (\xi^T\alpha)_{ii} = \\ &= \alpha_{i+1,i}\xi_{i+1,i} - \alpha_{i,i-1}\xi_{i,i-1} \qquad (4.44) \\ [\alpha, \xi]_{i,i+1} &= (\alpha\xi)_{i,i+1} - (\xi\alpha)_{i,i+1} = \\ &= \alpha_{i,i,i+1}(\xi_{i+1,i+1} - \xi_{ii}) \qquad (4.45) \end{aligned}$$

For any trace free Jacobi matrix

$$u = \begin{pmatrix} u_{11} & u_{21} & 0 & \cdots & 0 \\ u_{21} & u_{11} & u_{31} & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & u_{n-1,n-2} & u_{n-1,n-1} & u_{n,n-1} \\ 0 & \cdots & 0 & u_{n,n-1} & u_{nn} \end{pmatrix} \qquad (4.46)$$

Let $u_a = (u_{11}, \ldots, u_{nn})$, $u_b = (u_{21}, \ldots, u_{n,n-1})$ (same notation for lower triangular matrices of width one!) and define the following $(n-1) \times n$–matrix

$$A(\alpha) := \begin{pmatrix} \alpha_{21} & -\alpha_{21} & 0 & \cdots & 0 \\ 0 & \alpha_{32} & -\alpha_{32} & \cdots & 0 \\ 0 & \cdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \alpha_{n,n-1} & -\alpha_{n,n-1} \end{pmatrix} \qquad (4.47)$$

Then finding ξ such that $u = \mathrm{ad}'_\xi \alpha$ is the same as finding ξ such that

$$\begin{pmatrix} u_a \\ u_b \end{pmatrix} = \begin{pmatrix} 0 & A^T(\alpha) \\ -A(\alpha) & 0 \end{pmatrix} \begin{pmatrix} \xi_a \\ \xi_b \end{pmatrix} \tag{4.48}$$

Just by naively solving $u_a = A^T(\alpha)\xi_b$ we obtain:

$$\xi_{l+1,l} = \frac{1}{\alpha_{l+1,l}} \sum_{j=1}^{l} u_{jj} \tag{4.49}$$

for $1 \leq l \leq n-1$. This solution is compatible with $\xi_{n,n-1} = -\alpha_{n,n-1}^{-1} u_{nn}$, because u is assumed trace free. On the other hand the equation $u_b = -A(\alpha)\xi_a$ implies that

$$\frac{u_{l+1,l}}{\alpha_{l+1,l}} = \xi_{l+1,l+1} - \xi_{ll} \tag{4.50}$$

and hence:

$$\xi_{ll} = \xi_{11} + \sum_{j=1}^{l-1} \frac{u_{j+1,j}}{\alpha_{j+1,j}} \tag{4.51}$$

Since we're looking for ξ with $\mathrm{tr}\xi = 0$ we have

$$0 = \sum_{k=1}^{n} \left(\xi_{11} + \sum_{j=1}^{k-1} \frac{u_{j+1,j}}{\alpha_{j+1,j}} \right) = n\xi_{11} + \sum_{j=1}^{n-1} \frac{n-j}{\alpha_{j+1,j}} \cdot u_{j+1,j} \tag{4.52}$$

From which we obtain

$$\xi_{11} = -\frac{1}{n} \sum_{j=1}^{n-1} \frac{n-j}{\alpha_{j+1,j}} \cdot u_{j+1,j} \tag{4.53}$$

$$\begin{aligned} \xi_{ll} &= -\frac{1}{n} \sum_{j=1}^{n-1} \frac{n-j}{\alpha_{j+1,j}} \cdot u_{j+1,j} + \sum_{j=1}^{l-1} \frac{u_{j+1,j}}{\alpha_{j+1,j}} = \\ &= -\sum_{j=l}^{n-1} \frac{n-j}{n\alpha_{j+1,j}} u_{j+1,j} + \sum_{j=1}^{l-1} \frac{j}{n\alpha_{j+1,j}} \cdot u_{j+1,j} \end{aligned} \tag{4.54}$$

Now define the $n \times (n-1)$ matrix $B_n(\alpha)$ as follows:

$$B_n(\alpha) = \hat{B}_n D_{\alpha_b^{-1}} \tag{4.55}$$

$$(\hat{B}_{n,ij})_{\substack{1\le i\le n\\ 1\le j\le n-1}} = \begin{cases} \frac{n-j}{n} & , i \le j \\ -\frac{j}{n} & , i > j \end{cases}, \quad D_{\alpha_b^{-1}} = \mathrm{diag}(\alpha_{21}^{-1}, \ldots, \alpha_{n,n-1}^{-1}) \tag{4.56}$$

Then $\xi_a = -B_n(\alpha)u_b$ and $\xi_b = B_n^T(\alpha)u_a$ (because trace free u allows adding constants to all elements of a column of $B_n(\alpha)$). One sees this is true by checking that $A(\alpha)B_n(\alpha) = \mathrm{id}_{n-1}$. Let $C_n(\alpha) = A(\alpha)B_n(\alpha)$, an $(n-1)\times(n-1)$–matrix, then:

$$c_{n,ij} = \sum_{k=1}^{n} a_{ik}(\alpha)\cdot b_{n,kj}(\alpha) = \begin{cases} \alpha_{i+1,i}\left(\frac{n-j}{\alpha_{j+1,j}n} - \frac{n-j}{\alpha_{j+1,j}n}\right) = 0 & \text{if } i < j \\ \alpha_{i+1,i}\left(\frac{n-i}{\alpha_{i+1,i}n} - \frac{-i}{\alpha_{i+1,i}n}\right) = 1 & \text{if } i = j \\ \alpha_{i+1,i}\left(\frac{-j}{\alpha_{j+1,j}n} - \frac{-j}{\alpha_{j+1,j}n}\right) = 0 & \text{if } i > j \end{cases} \tag{4.57}$$

On the other hand let's evaluate $D_n(\alpha) = A^T(\alpha)\cdot B_n^T(\alpha)$, an $n\times n$ matrix:

$$d_{n,ij} = \sum_{k=1}^{n} a_{ik}^T(\alpha)\cdot b_{n,kj}^T(\alpha) = -\alpha_{i,i-1}\cdot b_{n,(j,i-1)} + \alpha_{i+1,i}\cdot b_{n,ji}$$

$$\begin{cases} \alpha_{21}\cdot\frac{n-1}{\alpha_{21}n} = \frac{n-1}{n} & \text{if } i=j=1 \\ -\frac{\alpha_{21}}{\alpha_{21}n} = -\frac{1}{n} & \text{if } i=1, j>1 \\ -\alpha_{i,i-1}\cdot\frac{n-i+1}{\alpha_{i,i-1}n} + \alpha_{i+1,i}\cdot\frac{n-i}{\alpha_{i+1,i}n} = -\frac{1}{n} & \text{if } 1<i<n, i>j \\ -\alpha_{i,i-1}\cdot\frac{-i+1}{\alpha_{i,i-1}n} + \alpha_{i+1,i}\cdot\frac{n-i}{\alpha_{i+1,i}n} = \frac{n-1}{n} & \text{if } 1<i<n, i=j \\ -\alpha_{i,i-1}\cdot\frac{-i+1}{\alpha_{i,i-1}n} + \alpha_{i+1,i}\cdot\frac{-i}{\alpha_{i+1,i}n} = -\frac{1}{n} & \text{if } 1<i<n, i<j \\ -\alpha_{n-1,n}\frac{1}{\alpha_{n-1,n}n} = -\frac{1}{n} & \text{if } i=n, j<n \\ \alpha_{n-1,n}\frac{n-1}{\alpha_{n-1,n}n} = \frac{n-1}{n} & \text{if } i=j=n \end{cases} \tag{4.58}$$

In short, $D_n(\alpha) = D_n$ is a matrix of the form:

$$D_n = \begin{pmatrix} \frac{n-1}{n} & -\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & \frac{n-1}{n} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ -\frac{1}{n} & \cdots & -\frac{1}{n} & \frac{n-1}{n} \end{pmatrix} \tag{4.59}$$

Because of the fact that u is trace free we have that $u_a = D_n u_a$ which settles the issue. We have thus proved the lemma

Lemma 4.6. *Let u be a trace free Jacobi matrix whose diagonal is denoted u_a and whose off diagonal is denoted u_b. Then by setting $\xi_b := B_n^T(\alpha)u_a$ and $\xi_a := -B_n(\alpha)u_b$, where $B_n(\alpha)$ is the matrix defined in equation (4.55) we can form a Jacobi matrix ξ with diagonal ξ_a and off diagonal ξ_b. This matrix satisfies:*

$$ad'\xi(\alpha) = u \tag{4.60}$$

If we define the matrix $M_n(\alpha)$:

$$M_n(\alpha) = \begin{pmatrix} 0 & -B_n(\alpha) \\ B_n^T(\alpha) & 0 \end{pmatrix} \tag{4.61}$$

then we have found, for given u and v in the tangent space of the coadjoint orbit, elements $(\xi_a, \xi_b)^T = M_n(\alpha)(u_a, u_b)^T$ and $(\eta_a, \eta_b)^T = M_n(\alpha)v$ such that $\mathrm{ad}'_\xi\alpha = u$ and $\mathrm{ad}'_\eta\alpha = v$.

It is now a simple computational task to find out an explicit formulation of the Kostant–Kirillov form:

$$\omega_\alpha(u, v) = \omega_\alpha(\mathrm{ad}'_\xi\alpha, \mathrm{ad}'_\eta\alpha) = \mathrm{tr}\alpha[\xi, \eta] \tag{4.62}$$

We first expand the trace to obtain

$$\mathrm{tr}\alpha[\xi,\eta] = \sum_{j=1}^{n} (\alpha[\xi,\eta])_{ii} \tag{4.63}$$

where

$$(\alpha[\xi,\eta])_{ii} = \alpha_{i,i-1}[\xi,\eta]_{i-1,i} + \alpha_{i,i}[\xi,\eta]_{ii} + \alpha_{i,i+1}[\xi,\eta]_{i+1,i} = \\ = \alpha_{i,i}[\xi,\eta]_{ii} + \alpha_{i,i+1}[\xi,\eta]_{i+1,i} \tag{4.64}$$

and

$$\begin{aligned} [\xi,\eta]_{i+1,i} &= (\xi\eta)_{i+1,i} - (\eta\xi)_{i+1,i} = \\ &= \xi_{i+1,i}\eta_{ii} + \xi_{i+1,i+1}\eta_{i+1,i} - \eta_{i+1,i}\xi_{ii} - \eta_{i+1,i+1}\xi_{i+1,i} = \\ &= \xi_{i+1,i}\left(\eta_{ii} - \eta_{i+1,i+1}\right) - \eta_{i+1,i}\left(\xi_{ii} - \xi_{i+1,i+1}\right) \qquad (4.65) \\ [\xi,\eta]_{i,i} &= (\xi\eta)_{ii} - (\eta\xi)_{ii} = \\ &= \xi_{i,i-1}\eta_{i-1,i} + \xi_{i,i}\eta_{i,i} - \eta_{i,i-1}\xi_{i-1,i} - \eta_{i,i}\xi_{i,i} = 0 \qquad (4.66) \end{aligned}$$

Based on our definition of ξ and η we obtain

$$[\xi,\eta]_{i+1,i} = -\frac{1}{\alpha_{i+1,i}^2}\left(v_{i+1,i}\sum_{l=1}^{i} u_{ll} - u_{i+1,i}\sum_{l=1}^{i} v_{ll}\right) \tag{4.67}$$

So that altogether we get:

$$\begin{aligned}\mathrm{tr}\alpha[\xi,\eta] &= -\sum_{j=1}^{n}\frac{1}{\alpha_{j+1,j}}\left(v_{j+1,j}\sum_{l=1}^{j}u_{ll}-u_{i+1,i}\sum_{l=1}^{j}v_{ll}\right)= \\ &= -\left\{\sum_{j=1}^{n-1}\sum_{l=1}^{j}\frac{u_{j+1,j}}{\alpha_{j+1,j}}v_{ll}-\sum_{j=1}^{n-1}\sum_{l=1}^{j}\frac{v_{j+1,j}}{\alpha_{j+1,j}}u_{ll}\right\}= \\ &= \left\{\sum_{j=1}^{n-1}\sum_{l=j+1}^{n}\frac{u_{j+1,j}}{\alpha_{j+1,j}}v_{ll}-\sum_{j=1}^{n-1}\sum_{l=j+1}^{n}\frac{v_{j+1,j}}{\alpha_{j+1,j}}u_{ll}\right\} \quad (4.68)\end{aligned}$$

or, more concisely:

Lemma 4.7. *For two matrices u, v their Kostant–Kirillov bracket is given as*

$$\omega_\alpha(u,v) = u_b^T K_n(\alpha) v_a - v_b^T K_n(\alpha) u_a \quad (4.69)$$

where the $(n-1)\times n$–matrix $K_n(\alpha)$ is given by

$$k_{n,ij}(\alpha) = \begin{cases} 0 & \text{if } i \geq j \\ \alpha_{i+1,i}^{-1} & \text{if } i < j \end{cases} \quad (4.70)$$

The Hamiltonian vectorfield v_H is contained in the tangent space $T\mathbf{j} = \mathbf{j}$. We will now show that

$$\langle dH, \mathrm{ad}'_\eta\beta\rangle = \mathrm{tr}\beta[\partial H,\eta] = \omega_\beta(\mathrm{ad}'_{\mathrm{pr}_\mathfrak{l}\partial H}\beta, \mathrm{ad}'_\eta\beta) \quad (4.71)$$

Where ∂H denotes the matrix

$$(\partial H)_{ij} = \frac{\partial H}{\partial \xi_{ij}}(\xi),\ |i-j|\leq 1 \quad (4.72)$$

and

$$\mathrm{pr}_{\mathfrak{l}} L = L_- + L_0 + L_+^T \in \mathfrak{l} \tag{4.73}$$

This implies that $v_H(\beta) = \mathrm{ad}'_{\mathrm{pr}_{\mathfrak{l}}\partial H(\beta)}\beta$. First of all, how do we evaluate $\langle dH, u\rangle$ for $u = \mathrm{ad}'_\eta\beta$? Recall for intuition that $\mathfrak{j}$ and hence tangent and cotangent space at any $\beta \in \mathfrak{j}$ are $2(n-1)$–dimensional if we're talking about $n \times n$ matrix groups. This is because of the trace zero condition. Let

$$\mathcal{M}\left(u_{11}, \ldots, u_{n-1,n-1}, u_{21}, \ldots, u_{n,n-1}\right) = \begin{pmatrix} u_{11} & u_{21} & 0 & \cdots & 0 \\ u_{21} & u_{22} & u_{23} & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & u_{n-1,n-2} & u_{n-1,n-1} & u_{n,n-1} \\ 0 & \cdots & 0 & u_{n,n-1} & -\sum_{j=1}^{n-1} u_{jj} \end{pmatrix} \tag{4.74}$$

Then the infinitesimal change in H along the element u of the tangent space of $\mathfrak{j}$ at β is surely exactly the infinitesimal change of $H \circ \mathcal{M}$ along $\tilde{u}$ at $\tilde{\beta}$ where

$$\tilde{u} := (u_{11}, \ldots, u_{n-1,n-1}, u_{21}, \ldots, u_{n,n-1}) \tag{4.75}$$

$$\tilde{\beta} := (\beta_{11}, \ldots, \beta_{n-1,n-1}, \beta_{21}, \ldots, \beta_{n,n-1}) \tag{4.76}$$

The formal expression for this reads

$$\langle dH_\beta, u\rangle_{\mathfrak{s}^* \times \mathfrak{s}} = \langle d[H \circ \mathcal{M}]_{\tilde{\beta}}, \tilde{u}\rangle_{\mathbb{R}^{2(n-1),*} \times \mathbb{R}^{2(n-1)}} = \\ = \sum_{j=1}^{n-1} \frac{\partial H \circ \mathcal{M}}{\partial \tilde{\beta}_{jj}} \cdot \tilde{u}_{jj} + \sum_{j=2}^{n} \frac{\partial H \circ \mathcal{M}}{\partial \tilde{\beta}_{j,j-1}} \cdot \tilde{u}_{j,j-1} \tag{4.77}$$

Note that for $i = 1, \ldots, n-1$:

$$\frac{\partial H \circ \mathcal{M}}{\partial \tilde{\beta}_{ii}} = \frac{\partial H}{\partial \beta_{ii}} - \frac{\partial H}{\partial \beta_{nn}} \tag{4.78}$$

$$\frac{\partial H \circ \mathcal{M}}{\partial \tilde{\beta}_{i+1,i}} = 2\frac{\partial H}{\partial \beta_{i+1,i}} \tag{4.79}$$

and that $u_{nn} = -\sum_{j=1}^{n-1} \tilde{u}_{jj}$. This implies:

$$\langle dH_\beta, u\rangle_{\mathfrak{s}^* \times \mathfrak{s}} = \sum_{j=1}^{n-1} \left(\frac{\partial H}{\partial \beta_{jj}} - \frac{\partial H}{\partial \beta_{nn}} \right) \cdot \tilde{u}_{jj} + \sum_{j=2}^{n} \frac{\partial H}{\partial \beta_{j,j-1}} \cdot \tilde{u}_{j,j-1} =$$

$$= \sum_{j=1}^{n} \frac{\partial H}{\partial \beta_{jj}} \cdot u_{jj} + 2\sum_{j=2}^{n} \frac{\partial H}{\partial \beta_{j,j-1}} \cdot u_{j,j-1} = \operatorname{tr}\left(\partial H u\right) \tag{4.80}$$

Clearly ∂H is a Jacobi matrix (not necessarily trace free!). We will now show that

$$\langle dH_\beta, \operatorname{ad}'_\eta \beta\rangle_{\mathfrak{s}^* \times \mathfrak{s}} = \operatorname{tr}\left(\partial H \operatorname{ad}'_\eta \beta\right) = \operatorname{tr}\left(\beta[\operatorname{pr}_{\mathfrak{l}} \partial H, \eta]\right) \tag{4.81}$$

This directly implies the claim that $v_H \beta = \operatorname{ad}'_{\operatorname{pr}_{\mathfrak{l}} \partial H(\beta)} \beta$. (note that this expression fits nicely into our discussion so far: According to our definition $\operatorname{pr}_{\mathfrak{l}} \partial H \in \mathfrak{l}$ as required by the infinitesimal coadjoint representation). To proceed we compute

$$\langle dH_\beta, \operatorname{ad}'_\eta \beta\rangle_{\mathfrak{s}^* \times \mathfrak{s}} = \operatorname{tr}\left(\partial H \operatorname{ad}'_\eta \beta\right) = \operatorname{tr}\left(\partial H \operatorname{pr}_{\mathfrak{s}}[\beta, \eta]\right) \tag{4.82}$$

We need to evaluate the diagonal terms m_{ii} $(1 \leq i \leq n)$ of the matrix $M := \partial H \operatorname{pr}_{\mathfrak{s}}[\beta, \eta]$:

$$m_{ii} = (\partial H)_{i,i-1}(\operatorname{pr}_{\mathfrak{s}}[\beta, \eta])_{i-1,i} + (\partial H)_{ii}(\operatorname{pr}_{\mathfrak{s}}[\beta, \eta])_{ii} + (\partial H)_{i,i+1}(\operatorname{pr}_{\mathfrak{s}}[\beta, \eta])_{i+1,i} =$$

$$= (\partial H)_{i+1,i}[\beta, \eta]_{i-1,i} + (\partial H)_{ii}[\beta, \eta]_{ii} + (\partial H)_{i+1,i}[\beta, \eta]_{i,i+1} \tag{4.83}$$

Evaluating the bracket yields

$$\begin{aligned}
[\beta,\eta]_{ii} &= (\beta\eta)_{ii} - (\eta\beta)_{ii} = \\
&= \beta_{i,i+1}\eta_{i+1,i} - \beta_{i-1,i}\eta_{i,i-1} \qquad (4.84)\\
[\beta,\eta]_{i,i+1} &= (\beta\eta)_{i,i+1} - (\eta\beta)_{i,i+1} = \\
&= \beta_{i,i+1}\left(\eta_{i+1,i+1} - \eta_{ii}\right) \qquad (4.85)
\end{aligned}$$

So that overall

$$\begin{aligned}
\mathrm{tr}\left(\partial H \mathrm{pr}_{\mathfrak{s}}[\beta,\eta]\right) = \sum_{j=1}^{n} m_{jj} = \\
&= -\sum_{i=1}^{n}(\partial H)_{i-1,i}\beta_{i,i-1}\left(\eta_{i-1,i-1} - \eta_{i,i}\right) + \\
&\quad -\sum_{i=1}^{n}(\partial H)_{ii}\left[\beta_{i-1,i}\eta_{i,i-1} - \beta_{i,i+1}\eta_{i+1,i}\right] + \\
&\quad -\sum_{i=1}^{n}(\partial H)_{i+1,i}\beta_{i+1,i}\left(\eta_{ii} - \eta_{i+1,i+1}\right) = \\
&= -\sum_{i=2}^{n}(\partial H)_{i,i-1}\beta_{i,i-1}\left(\eta_{i-1,i-1} - \eta_{i,i}\right) + \\
&\quad -\sum_{i=1}^{n}(\partial H)_{ii}\left[\beta_{i-1,i}\eta_{i,i-1} - \beta_{i,i+1}\eta_{i+1,i}\right] + \\
&\quad -\sum_{i=1}^{n}(\partial H)_{i+1,i}\beta_{i+1,i}\left(\eta_{ii} - \eta_{i+1,i+1}\right) = \\
&= -\sum_{i=1}^{n}(\partial H)_{ii}\left[\beta_{i-1,i}\eta_{i,i-1} - \beta_{i,i+1}\eta_{i+1,i}\right] + \\
&\qquad -2\sum_{i=1}^{n-1}(\partial H)_{i+1,i}\beta_{i+1,i}\left(\eta_{ii} - \eta_{i+1,i+1}\right) \qquad (4.86)
\end{aligned}$$

where in all expressions like this we assume that terms which by their indices would lie outside the matrix are equal to zero. We have to compare this with the expression

that we obtain for $\mathrm{tr}\,(\beta[\mathrm{pr}_{\mathfrak{l}}\partial H,\eta])$. Here $l_{ii}:=(\beta[\mathrm{pr}_{\mathfrak{l}}\partial H,\eta])_{ii}$. Thus:

$$l_{ii}=\beta_{i-1,i}[\mathrm{pr}_{\mathfrak{l}}\partial H,\eta]_{i-1,i}+\beta_{ii}[\mathrm{pr}_{\mathfrak{l}}\partial H,\eta]_{ii}+\beta_{i+1,i}[\mathrm{pr}_{\mathfrak{l}}\partial H,\eta]_{i,i+1}$$

Now we evaluate the bracket expressions:

$$\begin{aligned}
[\mathrm{pr}_{\mathfrak{l}}\partial H,\eta]_{i-1,i}&=(\mathrm{pr}_{\mathfrak{l}}\partial H\eta)_{i-1,i}-(\eta\mathrm{pr}_{\mathfrak{l}}\partial H)_{i-1,i}=\\
&=(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i-1,i-2}\eta_{i-2,i}+(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i-1,i-1}\eta_{i-1,i}-\\
&-\eta_{i-1,i-2}(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i-2,i}-\eta_{i-1,i-1}(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i-1,i}=\\
&=(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i-1,i-1}\eta_{i-1,i}-(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i-1,i}\eta_{i-1,i-1}=0 \qquad (4.87)\\
[\mathrm{pr}_{\mathfrak{l}}\partial H,\eta]_{ii}&=(\mathrm{pr}_{\mathfrak{l}}\partial H\eta)_{ii}-(\eta\mathrm{pr}_{\mathfrak{l}}\partial H)_{ii}\\
&=(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i,i-1}\eta_{i-1,i}+(\mathrm{pr}_{\mathfrak{l}}\partial H)_{ii}\eta_{ii}-\\
&-\eta_{i,i-1}(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i-1,i}-\eta_{ii}(\mathrm{pr}_{\mathfrak{l}}\partial H)_{ii}=\\
&=(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i-1,i-1}\eta_{i-1,i}-(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i-1,i}\eta_{i-1,i-1}=0 \qquad (4.88)\\
[\mathrm{pr}_{\mathfrak{l}}\partial H,\eta]_{i+1,i}&=(\mathrm{pr}_{\mathfrak{l}}\partial H\eta)_{i+1,i}-(\eta\mathrm{pr}_{\mathfrak{l}}\partial H)_{i+1,i}\\
&=(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i+1,i}\eta_{ii}+(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i+1,i+1}\eta_{i+1,i}-\\
&-\eta_{i+1,i}(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i,i}-\eta_{i+1,i+1}(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i+1,i}=\\
&=(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i+1,i}\left[\eta_{ii}-\eta_{i+1,i+1}\right]-\\
&-\eta_{i+1,i}\left[(\mathrm{pr}_{\mathfrak{l}}\partial H)_{ii}-(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i+1,i+1}\right] \qquad (4.89)
\end{aligned}$$

After noticing that due to symmetry we have $(\mathrm{pr}_{\mathfrak{l}}\partial H)_{i+1,i}=2(\partial H)_{i,i+1}$ we combine everything and obtain:

$$\begin{aligned}
&\mathrm{tr}\,(\beta[\mathrm{pr}_{\mathfrak{l}}\partial H,\eta])=\sum_{j=1}^{n}l_{jj}=\\
&=\sum_{i=1}^{n}\beta_{i+1,i}\left\{(\partial H)_{i+1,i}\left[\eta_{ii}-\eta_{i+1,i+1}\right]-\eta_{i+1,i}((\partial H)_{ii}-(\partial H)_{i+1,i+1})\right\}=
\end{aligned}$$

$$= -\sum_{i=1}^{n}(\partial H)_{ii}\beta_{i+1,i}\eta_{i+1,i} + \sum_{i=1}^{n}(\partial H)_{i+1,i+1}\beta_{i+1,i}\eta_{i+1,i} +$$
$$+ 2\sum_{i=1}^{n}(\partial H)_{i+1,i}\beta_{i+1,i}\left[\eta_{ii} - \eta_{i+1,i+1}\right] =$$
$$= -\sum_{i=1}^{n}(\partial H)_{ii}\beta_{i+1,i}\eta_{i+1,i} + \sum_{i=2}^{n}(\partial H)_{ii}\beta_{i,i-1}\eta_{i,i-1} +$$
$$+ 2\sum_{i=1}^{n}(\partial H)_{i+1,i}\beta_{i+1,i}\left[\eta_{ii} - \eta_{i+1,i+1}\right] =$$
$$= -\sum_{i=1}^{n}(\partial H)_{ii}\beta_{i+1,i}\eta_{i+1,i} + \sum_{i=1}^{n}(\partial H)_{ii}\beta_{i,i-1}\eta_{i,i-1} +$$
$$+ 2\sum_{i=1}^{n}(\partial H)_{i+1,i}\beta_{i+1,i}\left[\eta_{ii} - \eta_{i+1,i+1}\right] =$$
$$= \sum_{i=1}^{n}(\partial H)_{ii}\left[\beta_{i,i-1}\eta_{i,i-1} - \beta_{i+1,i}\eta_{i+1,i}\right] +$$
$$+ 2\sum_{i=1}^{n}(\partial H)_{i+1,i}\beta_{i+1,i}\left[\eta_{ii} - \eta_{i+1,i+1}\right] =$$
$$= -\mathrm{tr}(\partial H^T \mathrm{pr}_{\mathfrak{s}}[\beta, \eta^T]) \quad (4.90)$$

Thus we have proved:

Lemma 4.8. *For a Hamiltonian function H on the coadjoint orbit of trace free Jacobi matrices, the Hamiltonian vectorfield v_H is given by $v_H = -ad'_{pr_{\mathfrak{l}}\partial H(\beta)}(\beta)$*

As always in Hamiltonian systems we want to define a Poisson bracket on our symplectic manifold $(\mathfrak{j}, \omega)$. The standard definition is

$$\{H, G\}(\beta) = \omega_\beta(v_H, v_G) = \mathrm{tr}\beta^T[\mathrm{pr}_{\mathfrak{l}}\partial H, \mathrm{pr}_{\mathfrak{l}}\partial G] \quad (4.91)$$

Evaluating the last expression we obtain:

$$\{H, G\}(\beta) = \sum_{j=1}^{n}\Big\{\beta_{j,j+1}\big[2(\partial H)_{j+1,j}((\partial G)_{jj} - (\partial G)_{j+1,j+1}) -$$

$$(\partial G)_{j+1,j}((\partial H)_{jj} - (\partial H)_{j+1,j+1})\big]\Big\} =$$
$$= 2\sum_{j=1}^{n} \beta_{j,j+1}(\partial H)_{j+1,j}(\partial G)_{jj} - 2\sum_{j=1}^{n} \beta_{j,j+1}(\partial H)_{j+1,j}(\partial G)_{j+1,j+1} -$$
$$- 2\sum_{j=1}^{n} \beta_{j,j+1}(\partial G)_{j+1,j}(\partial H)_{jj} + 2\sum_{j=1}^{n} \beta_{j,j+1}(\partial G)_{j+1,j}(\partial H)_{j+1,j+1} =$$
$$= 2\sum_{j=1}^{n} \beta_{j,j+1}(\partial H)_{j+1,j}(\partial G)_{jj} - 2\sum_{j=2}^{n} \beta_{j-1,j}(\partial H)_{j,j-1}(\partial G)_{j,j} -$$
$$- 2\left(\sum_{j=1}^{n} \beta_{j,j+1}(\partial G)_{j+1,j}(\partial H)_{jj} - \sum_{j=2}^{n} \beta_{j-1,j}(\partial G)_{j,j-1}(\partial H)_{jj}\right) =$$
$$= 2\sum_{j=1}^{n} \left(\beta_{j,j+1}(\partial H)_{j+1,j} - \beta_{j-1,j}(\partial H)_{j,j-1}\right)(\partial G)_{jj} -$$
$$- 2\sum_{j=1}^{n} \left(\beta_{j,j+1}(\partial G)_{j+1,j} - \beta_{j-1,j}(\partial G)_{j,j-1}\right)(\partial H)_{jj} \quad (4.92)$$

written in more familiar notation this becomes:

Lemma 4.9. *The Kostant–Kirillov-Poisson bracket for two Hamiltonian functions on the Jacobi matrices is given by*

$$\{H, G\}(\beta) = 2\sum_{j=1}^{n} \left(\beta_{j+1,j}\frac{\partial H}{\partial \alpha_{j+1,j}} - \beta_{j,j-1}\frac{\partial H}{\partial \alpha_{j,j-1}}\right)\frac{\partial G}{\partial \alpha_{jj}} -$$
$$- \left(\beta_{j+1,j}\frac{\partial G}{\partial \alpha_{j+1,j}} - \beta_{j,j-1}\frac{\partial G}{\partial \alpha_{j,j-1}}\right)\frac{\partial H}{\partial \alpha_{jj}} \quad (4.93)$$

4.4 The Toda Lattice and Flaschka's Variables

The Toda Lattice is a Hamiltonian system on $\mathbb{R}^{2n}$ equipped with the standard symplectic form $\omega = \sum_{i=1}^{n} \mathrm{d}x_i \wedge \mathrm{d}y_i$. It is defined as the flow generated by the following

Toda Hamiltonian (cf. [12, 41, 42]):

$$H_T = \frac{1}{2}\sum_{i=1}^{n} y_i^2 + \sum_{i=1}^{n} e^{x_i - x_{i+1}} \tag{4.94}$$

The resulting equations, called the Toda lattice equations, are of the form:

$$\ddot{x}_i = e^{x_{i-1}-x_i} - e^{x_i - x_{i+1}},\ 1 \le i \le n \tag{4.95}$$

where $x^{x_0 - x_1} = e^{x_n - x_{n+1}=0}$. Setting $y_i := \dot{x}_i$ we note that in addition to conserving H_T

$$\frac{d}{dt}H(x(t), y(t)) = \sum_{j=1}^{n} \frac{\partial H_T}{\partial x_i}\dot{x}_i + \sum_{j=1}^{n} \frac{\partial H_T}{\partial y_i}\dot{y}_i = 0 \tag{4.96}$$

the Toda flow also conserves total momentum $\sum_{i=1}^{n} y_i$

$$\frac{d}{dt}\sum_{i=1}^{n} y_i = 0 \tag{4.97}$$

The following coordinates are called Flaschka's variables after their discoverer [22]

$$a_k = -y_k/2, 1 \le k \le n \tag{4.98}$$

$$b_k = \frac{1}{2}e^{\frac{1}{2}(x_k - x_{k+1})},\ 1 \le k \le n-1 \tag{4.99}$$

Now compute:

$$\dot{a}_k = -\frac{1}{2}\dot{y}_k \overset{(4.95)}{=} 2(b_k^2 - b_{k-1}^2) \tag{4.100}$$

$$\dot{b}_k = \frac{1}{4}(\dot{x}_k - \dot{x}_{k+1}) \cdot e^{\frac{1}{2}(x_k - x_{k+1})} = b_k(a_{k+1} - a_k) \tag{4.101}$$

If we set the matrix L equal to

$$L_0 = \begin{pmatrix} a_1 & b_1 & 0 & \cdots & 0 \\ b_1 & a_2 & b_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & b_{n-2} & a_{n-1} & b_{n-1} \\ 0 & \cdots & 0 & b_{n-1} & a_n \end{pmatrix} \tag{4.102}$$

then the relations (4.100), (4.101) translate into the matrix ODE

$$\dot{L} = [L_+ - L_+^T, L] \tag{4.103}$$

where L_+ refers to the strictly upper triangular part of L. We will now proceed to show that equation (4.103) is a Hamiltonian equation on the Jacobi matrices, viewed as a coadjoint orbit of the lower triangular matrices and equipped with the corresponding Kostant–Kirillov form from equation (4.69). Let $H(L) := \frac{1}{2}\mathrm{tr}L^2$, where L is a trace free Jacobi matrix. In lemma 4.8 we established that the associated Kostant–Kirillov vectorfield is given by:

$$v_H(L) = -\mathrm{ad}'_{\mathrm{pr}_{\mathfrak{l}}\partial H}(L) = -\mathrm{pr}_{\mathfrak{s}}[L, (\mathrm{pr}_{\mathfrak{l}}\partial H(L))] \tag{4.104}$$

Clearly $\partial H(L) = L$. We will now show that in fact:

$$-\mathrm{pr}_{\mathfrak{s}}[L, (\mathrm{pr}_{\mathfrak{l}}L)^T] = [\mathrm{pr}_{\mathfrak{k}}L, L] \tag{4.105}$$

where the skew symmetric projection $\mathrm{pr}_{\mathfrak{k}}$ is defined by:

$$\mathrm{pr}_{\mathfrak{k}}L = L_+ - L_+^T \tag{4.106}$$

Because of the fact that $\mathrm{pr}_{\mathfrak{l}} + \mathrm{pr}_{\mathfrak{k}} = \mathrm{id}$ we can conclude

$$-\mathrm{pr}_{\mathfrak{s}}[L, (\mathrm{pr}_{\mathfrak{l}} L)^T] = -\mathrm{pr}_{\mathfrak{s}}[L, ([\mathrm{pr}_{\mathfrak{l}} + \mathrm{pr}_{\mathfrak{k}} - \mathrm{pr}_{\mathfrak{k}}]L)^T] = -\mathrm{pr}_{\mathfrak{s}}[L, \mathrm{pr}_{\mathfrak{k}} L] = [\mathrm{pr}_{\mathfrak{k}} L, L] \tag{4.107}$$

So that equation (4.103) is exactly the Hamiltonian system arising from H under the Kostant–Kirillov symplectic structure on the Jacobi matrices.

4.5 Flaschka's Variables and the Kostant–Kirillov Bracket

In this section we investigate some connections between Flaschka's coordinate transform (4.98), (4.99) and the Kostant–Kirillov symplectic structure on the Jacobi matrices. We will present the following results:

1. The pullback under Flaschka's variables of the standard symplectic structure on $\mathbb{R}^{2n}$ gives rise to a symplectic structure on the symmetric trace free Jacobi matrices of dimension $n \times n$ which is a multiple of the Kostant–Kirillov form (cf. 4.119).

2. The pullback of the standard metric on $\mathbb{R}^{2n}$ gives a metric structure on $\mathfrak{j}$ and in particular allows us to compute gradients of functions on $\mathfrak{j}$. It will turn out that these gradients are compatible with the pulled back symplectic form in the sense that they yield a multiple of the Poisson bracket coming from the abstract framework (cf. equation (4.133)).

Denote the Flaschka transformation from (4.98), (4.99) by $F : \mathbb{R}^{2n} \to \mathbb{R}^{2n-1}$, $(a, b) = F(x, y)$. Note that the none of the Toda Hamiltonian (4.94), the Toda equations (4.95) and Flaschka's variables (4.99) are influenced by parallel translations of the form $x_j \mapsto x_j + s$, for $1 \leq j \leq n$. This and the conservation of momentum allows us to fix the center of mass and the momentum at zero and restrict domain of F to the set $\mathcal{K} = \{\sum_j x_j = \sum_j y_j = 0\}$. The range of values (a, b) which is of interest is likewise of dimension $2n - 2$, because of the restriction that a sum up to one. With these restrictions F is invertible and all the computations in this chapter make sense. Let M_n denote the following matrix:

$$M_n = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & -1 & 0 \\ 0 & \cdots & 0 & 1 & -1 \end{pmatrix}$$

Then

$$M_n x = \begin{pmatrix} 0 \\ 2\ln(2b_1) \\ \vdots \\ 2\ln(2b_{n-1}) \end{pmatrix} \tag{4.108}$$

and the inverse of F is given by

$$F^{\text{inv}}(a,b) : \mathbb{R}^{2n-1} \to \mathcal{K} \tag{4.109}$$

$$F^{\text{inv}} : (a,b) \mapsto \begin{pmatrix} M^{\text{inv}}_{n,-1} & 0 \\ 0 & id_n \end{pmatrix} g(a,b) = (x,y)^T \tag{4.110}$$

where $M^{\text{inv}}_{n,-1}$ is a matrix of size $n \times (n-1)$ which results from M^{inv}_n by deleting the first column. The function g is defined by:

$$g_j(a,b) = 2\ln(2b_j), j = 1, \ldots, n-1 \tag{4.111}$$

$$g_j(a,b) = -2a_{j-n-1}, j = n, \ldots, 2n-1 \tag{4.112}$$

We have assembled nearly enough information to compute the pull back under F (same as pushforward under F^{inv}) of the standard symplectic form $\omega = \sum_{j=1}^{n} dx_j \wedge dy_j$ (restricted to $\mathcal{K}$). For the following computation, choose u, v to be elements of the tangent space of the Jacobi matrix $\alpha = \mathcal{M}(a,b)$ (cf. equation (4.74)). This means that the diagonals u_a and v_a sum to zero.

$$\begin{aligned} [[F^*\omega](a,b)]\,(u,v) &= \left[[F^{\text{inv}}_*\omega](a,b)\right](u,v) = \\ &= \left[\omega(F^{\text{inv}}(a,b))\right](D_{(a,b)}F^{\text{inv}}u, D_{(a,b)}F^{\text{inv}}v) = \\ &\qquad \left[\sum_{j=1}^{n} dx_j \wedge dy_j\right](D_{(a,b)}F^{\text{inv}}u, D_{(a,b)}F^{\text{inv}}v) \end{aligned} \tag{4.113}$$

The missing item is thus $D_{(a,b)}F^{\text{inv}}$. It turns out that there's an explicit representation of the inverse of M_n, so that we have

$$D_{(a,b)}F^{\text{inv}} = \begin{pmatrix} 0 & M^{\text{inv}}_{n,-1} \\ id_n & 0 \end{pmatrix} \cdot \text{diag}(-2, \ldots, -2, 2b_1^{-1}, \ldots, 2b_{n-1}^{-1}) \tag{4.114}$$

where, finally,

$$(M^{\text{inv}}_{n,-1})_{ij} = \begin{cases} \frac{n-j}{n} & \text{if } j \geq i \\ -\frac{j}{n} & \text{if } j < i \end{cases} \tag{4.115}$$

(recall that $M^{\text{inv}}_{n,-1}$ is a $n \times (n-1)$ matrix.) This is exactly the matrix $\hat{B}_n$ arising in equation (4.56). Define $R := M^{\text{inv}_{n,-1}} \cdot \text{diag}(b_1^{-1}, \ldots, 2b_{n-1}^{-1})$. Now, we obtain:

$$[[F^*\omega](a,b)]\,(u,v) = \left[[F^{\text{inv}}_*\omega](a,b)\right]\,(u,v) = -4\left(u_b^T R^T v_a - v_b^T R^T u_a\right) \tag{4.116}$$

Based on the fact that the trace of the 'tangent matrices' is zero, we have that the sum of the entries in the a-part of u and v is zero. This means that adding the same constant to all entries in a column of a matrix R does not change the value of $R^T v_a$. In particular, if we subtract $\frac{-j}{a_j n}$ from the j^{th} column of R we obtain the matrix

$$\hat{M}^{\text{inv}}_{n,-1} = \begin{pmatrix} b_1^{-1} & b_2^{-1} & \cdots & b_{n-1}^{-1} \\ 0 & b_2^{-1} & \cdots & b_{n-1}^{-1} \\ 0 & \ddots & \cdots & \vdots \\ 0 & \cdots & 0 & b_{n-1}^{-1} \\ 0 & \cdots & \cdots & 0 \end{pmatrix} \tag{4.117}$$

Let $\alpha = \mathcal{M}(a,b)$ (cf. (4.74), but abuse the notation by setting the diagonal of α equal to a). Then because of $\sum_{j=1}^{l} u_{a,j} = -\sum_{j=l+1}^{n} u_{a,j}$ we have that -$\hat{M}^{\text{inv},T}_{n,-1} u_a = K_n(\alpha) u_a$, where $K_n(\alpha)$ was defined in (4.69). Overall this means:

$$[[F^*\omega](a,b)]\,(u,v) = 4\left(u_b^T K_n(\alpha) v_a - v_b^T K_n(\alpha) u_a\right) \tag{4.118}$$

More concisely:

$$[[F^*\omega](a,b)]\,(u,v) = 4 \cdot \omega_{\mathcal{M}(a,b)}(\mathcal{M}(u), \mathcal{M}(v)) \tag{4.119}$$

On the phase space (x, y) the standard Euclidean metric structure is given. The metric in (a, b)–coordinates is computed as:

$$\begin{aligned} g_{(a,b)}((u_a, u_b), (v_a, v_b)) = \\ = g_{F^{\text{inv}}(a,b)}(D_{(a,b)}F^{\text{inv}}(u_a, u_b), D_{(a,b)}F^{\text{inv}}(v_a, v_b)) = \\ = (u_\alpha, v_\alpha)^T (D_{(a,b)}F^{\text{inv}} D_{(a,b)})^T F^{\text{inv}}(v_\alpha, v_\beta) \end{aligned} \tag{4.120}$$

We evaluate the matrix product in the middle as:

$$\begin{aligned} \left[(D_{(a,b)}F^{\text{inv}})^T D_{(a,b)}F^{\text{inv}}\right]_{ij} = 4 \cdot \text{diag}\left(1, \ldots, 1, b_1^{-1}, \ldots, b_{n-1}^{-1}\right) \cdot \begin{pmatrix} \text{id}_n & 0 \\ 0 & Q_n \end{pmatrix} \cdot \\ \cdot \text{diag}\left(1, \ldots, 1, b_1^{-1}, \ldots, b_{n-1}^{-1}\right) \end{aligned} \tag{4.121}$$

where Q_n is an $(n-1) \times (n-1)$–dimensional matrix of the form:

$$Q_{n,ij} = \begin{cases} \frac{i}{n} \cdot (n-j) & i \leq j \\ \frac{j}{n} \cdot (n-i) & i > j \end{cases} \tag{4.122}$$

It turns out that the inverse of Q_n is related to the matrix of the second finite difference:

$$Q_n^{\text{inv}} = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \tag{4.123}$$

Thus the gradient of any function H defined on (a, b) is defined by

$$\frac{d}{dt} H((a_0, b_0) + t \cdot (v_a, v_b)) = g_{(\alpha_0,\beta_0)}(\nabla_{(\alpha_0,\beta_0)} H, (v_\alpha, v_\beta)) \tag{4.124}$$

but clearly:

$$\frac{d}{dt} H((a_0, b_0) + t \cdot (v_a, v_b)) = \left\langle \frac{\partial H}{\partial a}, v_a \right\rangle + \left\langle \frac{\partial H}{\partial b}, v_b \right\rangle \tag{4.125}$$

and by our above computation in 4.120:

$$g_{(a,b)}(\nabla_{(a,b)} H, (v_a, v_b)) = \\ 4 \left\{ [\nabla_{\alpha,\beta} H]^T_{i=1,\ldots,n} \, v_a + [\nabla_{\alpha,\beta} H]^T_{i=n+1,\ldots,2n-1} \, D_{\alpha^{-1}} Q_n D_{b^{-1}} v_b \right\} \tag{4.126}$$

where $D_{b^{-1}} = \text{diag}(b_1^{-1}, \ldots, b_n^{-1})$. Taking together equations (4.125) and (4.126) we obtain

$$\frac{1}{4} \cdot \frac{\partial H}{\partial b} = [\nabla_{\alpha,\beta} H]^T_{i=n+1,\ldots,2n-1} \, D_{\alpha^{-1}} Q_n D_{\alpha^{-1}} \tag{4.127}$$

Let's denote $\nabla_a H = [\nabla_{\alpha,\beta} H]^T_{i=1,\ldots,n}$, and $\nabla_b H = [\nabla_{a,b} H]^T_{i=n+1,\ldots,2n-1}$. By trace freeness the sum of the components of v_a is zero. For $x \in \mathbb{R}^n$

$$\mathrm{pr}_{\mathrm{tr}=0} x = x - \frac{1}{n}\left(\sum_{j=1}^{n} x_j\right)(1,\ldots,1)^T \tag{4.128}$$

we have that

$$\nabla_b H = \frac{1}{4} D_b Q_n^{-1} D_b \left(\frac{\partial H}{\partial b}\right)^T \tag{4.129}$$

$$\nabla_a H = \frac{1}{4} \cdot \mathrm{pr}_{\mathrm{tr}=0} \left(\frac{\partial H}{\partial a}\right)^T \tag{4.130}$$

Let us now consider $[F^*\omega]_{\alpha,\beta}(\nabla_{\alpha,\beta} H, \nabla_{\alpha,\beta} G)$. This is the Kostant–Kirillov-Poisson bracket defined using the gradients of the Hamiltonians, where the gradient is with respect to the pushed forward metric:

$$\begin{aligned}[F^*\omega]_{\alpha,\beta}(\nabla_{\alpha,\beta} H, \nabla_{\alpha,\beta} G) = 4\left(\nabla_b H^T K_n(\alpha) \nabla_a G - \nabla_b G^T K_n(\alpha) \nabla_a H\right) = \\ \frac{1}{4}\left(\frac{\partial H}{\partial b} D_b Q_n^{-1} D_b K_n(\alpha) \mathrm{pr}_{\mathrm{tr}=0} \frac{\partial G^T}{\partial a} - \frac{\partial G}{\partial b} D_b Q_n^{-1} D_b K_n(\alpha) \mathrm{pr}_{\mathrm{tr}=0} \frac{\partial G^T}{\partial a}\right)\end{aligned} \tag{4.131}$$

where $\alpha = \mathcal{M}(a,b)$ and $K_n(\alpha) D_{b^{-1}} K_n$ is defined in (4.69). We have set K_n to equal an $(n-1) \times n$–matrix containing ones at positions strictly above the diagonal and zeros elsewhere. Thus:

$$[F^*\omega]_{a,b}(\nabla_{a,b} H, \nabla_{a,b} G) = \frac{1}{4}\left(\frac{\partial H}{\partial b} D_\alpha Q_n^{-1} K_n \mathrm{pr}_{\mathrm{tr}=0} \frac{\partial G^T}{\partial a} - \frac{\partial G}{\partial b} D_b Q_n^{-1} K_n \mathrm{pr}_{\mathrm{tr}=0} \frac{\partial G^T}{\partial a}\right) \tag{4.132}$$

Evaluating $D_a Q_n^{-1} K_n P$ yields exactly the $(n-1) \times n$–matrix $-A(\alpha)$ from equation (4.47), so that

$$[F^* \omega]_{\alpha,\beta}(\nabla_{\alpha,\beta} H, \nabla_{\alpha,\beta} G) = \frac{1}{4} \left(\frac{\partial H}{\partial a} A(a) \frac{\partial G^T}{\partial b} - \frac{\partial G}{\partial a} A(a) \frac{\partial G^T}{\partial b} \right) \tag{4.133}$$

Compare this result with equation (4.93) to see that the Poisson structure derived using pulling the standard symplectic and the metric structures is a multiple of the Kostant–Kirillov-Poisson bracket.

Chapter Five

Hamiltonian Eigenvalue Algorithms

In the following we apply the theory of Hamiltonian systems on Jacobi matrices to establish a link between eigenvalue algorithms and the Toda lattice. We will also introduce the Deift–Li–Nanda–Tomei framework for eigenvalue algorithms and discuss how standard algorithms fit into the framework. In the last section we will derive the Liouville equation for the Hermite–1 density under the Toda flow.

5.1 The Connection between Eigenvalue Algorithms and Completely Integrable Systems

A lot of the insight in this chapter goes back to the work in [15, 41, 43, 49, 50]. First of all, let us recall equation (4.103), the Toda equation in its matrix form:

$$\dot{L} = [L_+ - L_+^T, L] \tag{5.1}$$

This is a classical case of a Lax pair equation. It is impossible for someone in Hamiltonian systems to see this and not to think of [35] and suspect that the matrix L undergoes an isospectral deformation as time passes. In fact it is said in [42] that the motivation in [22] for trying to write the Toda system in the form (5.1) was based on [35]. The following lemma and proof follows the discussion in [43]

Lemma 5.1. *Let $L(0)$ be a Jacobi matrix. Then the Toda flow (5.1) preserves the spectrum. The spectrum of $L(t)$ is the same for all t.*

Proof. Recall that $\mathrm{pr}_{\mathfrak{k}} L = L_+ - L_+^T$ from equation (4.106) and denote $B(t) = \mathrm{pr}_{\mathfrak{k}} L(t)$.

Let

$$\dot{U} = BU \tag{5.2}$$

and $U(0) := \mathrm{id}$. Then

$$(UU^T)^{\cdot} = BUU^T - UU^T B = 0 \tag{5.3}$$

So that $U(t)$ is a family of orthogonal matrices. Now let $H(t) := U(t)L(0)U^T(t)$. Then we have that $H(0) = L(0)$ and

$$\dot{H} = BUL_0U^T - UL_0U^TB = [B, H] \tag{5.4}$$

So that $H(t) = L(t)$ for all t. We have thus proved that $L(t)$ is a conjugation of L_0 with the orthogonal matrix $U(t)$, ie. the eigenvalues are preserved. □

We have established that the Toda matrix equation is a Hamiltonian equation in section 4.4. The fact that the Toda flow conserves the spectrum of the initial matrix means that the eigenvalues are n integrals for a matrix of size n. Recall that the flow is defined on a $2(n-1)$–dimensional submanifold, because it keeps the trace of L constant at zero. As a consequence of the trace freeness the eigenvalues add up to zero, too. This means that the first $n-1$ eigenvalues are conserved quantities for a flow on $2(n-1)$ dimensions, which leads us to suspect that the Toda flow is actually completely integrable. That this is in fact the case follows from the following lemma [42, p. 219]:

Lemma 5.2. *Let $\{\cdot,\cdot\}_{KK}$ denote the Kostant–Kirillov-Poisson bracket on Jacobi matrices defined in (4.93) and let $\lambda_i(t)$ denote the eigenvalues of a Jacobi Matrix $L(t)$ evolving under the Toda flow. The eigenvalues λ_i Poisson commute, ie. $\{\lambda_i, \lambda_j\}_{KK} =$*

0.

Proof. As a first step, we need to compute the derivatives of λ_i with respect to the matrix entries. For notational convenience, consider two different eigenvalues λ and μ along with their corresponding (orthonormal) eigenvectors ϕ and ψ, so that $(L-\lambda)\psi = (L-\mu)\phi = 0$. We indicate differentiation with respect to a matrix entry by $'$. Then clearly:

$$\begin{aligned}(L'-\lambda')\phi + (L-\lambda)\phi' &= 0\\ \langle (L'-\lambda')\phi, \phi\rangle + \langle (L-\lambda)\phi', \phi\rangle &= \langle (L'-\lambda')\phi, \phi\rangle + \langle \phi', (L-\lambda)\phi\rangle = \\ &= \langle (L'-\lambda')\phi, \phi\rangle = 0 \end{aligned} \tag{5.5}$$

Thus, by normalization : $\lambda' = \langle L'\phi, \phi\rangle$. More explicitly:

$$\frac{\partial \lambda}{\partial a_i} = \phi_i^2,\ 1 \le i \le n \tag{5.6}$$

$$\frac{\partial \lambda}{\partial b_i} = 2\phi_j\phi_{j+1}, 1 \le i \le n-1 \tag{5.7}$$

Now we can compute $\{\lambda, \mu\}_{\mathrm{KK}}$:

$$\begin{aligned}\{\lambda,\mu\}_{\mathrm{KK}}(a,b) &= 2\sum_{j=1}^{n}\left(b_j\frac{\partial\lambda}{\partial b_j} - b_{j-1}\frac{\partial\lambda}{\partial b_{j-1}}\right)\frac{\partial\mu}{\partial a_j} - \left(b_j\frac{\partial\mu}{\partial b_j} - b_{j-1}\frac{\partial\mu}{\partial b_{j-1}}\right)\frac{\partial\lambda}{\partial a_j} = \\ &= 4\sum_{j=1}^{n}\left(b_j\phi_j\phi_{j+1} - b_{j-1}\phi_{j-1}\phi_j\right)\psi_j^2 - \left(b_j\psi_j\psi_{j+1} - b_{j-1}\psi_{j-1}\psi_j\right))\phi_j^2 = \\ &\qquad 4\sum_{j=1}^{n}\phi_j\psi_j^2\left(b_j\phi_{j+1} - b_{j-1}\phi_{j-1}\right) - \psi_j\phi_j^2\left(b_j\psi_{j+1} - b_{j-1}\psi_{j-1}\right)) \end{aligned} \tag{5.8}$$

Setting $W_j := b_j[\phi_{j+1}\psi_j - \phi_j\psi_{j+1}]$, $j = 0, \ldots, n$ we have

$$W_j - W_{j-1} = (\lambda - \mu)\phi_j - \psi_j \tag{5.9}$$

$$W_j^2 - W_{j-1}^2 = (\lambda - \mu)\left[b_j\phi_{j+1}\phi_j - a_{j-1}\phi_{j-1}\phi_j\right]\psi_j^2 - \tag{5.10}$$

$$-(\lambda - \mu)\left[b_j\psi_{j+1}\psi_j - a_{j-1}\psi_{j-1}\psi_j\right]\phi_j^2 \tag{5.11}$$

So that overall $\{\lambda, \mu\}_{\mathrm{KK}} = 0$. □

A typical feature of completely integrable Hamiltonian systems is the existence of a coordinate transform under which the system can be solved explicitly. In the current case this transform is given by the spectral map described in a section 3.1.2. Let $L = Q\Lambda Q^T$ and $\Lambda = \mathrm{diag}(\lambda)$, $q = |Q^T e_1|$, ie. the absolute value of the entries in the first row of Q, then the set of Jacobi matrices is parametrized by $L \mapsto (q, \lambda)$. We have demonstrated above that $\dot{\lambda} = 0$, now we turn to finding an equation for $\dot{q}$. We proceed using the method outlined in [43]): Let $L_0 := U_0\Lambda U_0^T$, then $L(t) = U(t)U_0\Lambda U_0^T U^T(t)$, where U was defined in equation (5.2), and $q = Q^T e_1 =$. Now,

$$\begin{aligned}\frac{dq}{dt} &= U_0^T \frac{d}{dt}[U^T(t)e_1] = U_0^T L_0 U^T(t)e_1 - a_1(t)U_0^T U^T(t)e_1 = \\ &= \Lambda U_0^T U^T(t)e_1 - a_1(t)U_0^T U^T(t)e_1 = \Lambda q - (\Lambda q, q)q\end{aligned} \tag{5.12}$$

Then it turns out that:

$$\begin{cases} \dot{\lambda} = 0 \\ \dot{q} = \Lambda q - (q, \Lambda q)q \\ q(t) = \frac{e^{\Lambda t}q_0}{\|e^{\Lambda t}q_0\|} \end{cases} \tag{5.13}$$

To summarize: The spectral map $L \mapsto (\lambda, q)$ takes a matrix L in the phase space to its "action–angle" variables. λ corresponds to the actions, q corresponds to the

angles. Altough the Toda lattice and this result in particular can be generalized to other matrix manifolds (generalized Toda lattice, see eg [13]), the elegance of this result in the case of Jacobi matrices is due to the fact that Jacobi matrices are fully determined by their eigenvalues and the first row of their matrix of eigenvectors ([9, 15, 53]).

From equation (4.101) it is clear that the flow preserves off diagonal zeros, which means that it preserves the signs of the offdiagonal terms, too. On the other hand, the proof that for any Jacobi initial matrix the off diagonal terms tend to zero goes back at least to [41]. And a concise proof of this fact can be found in [43]. The authors of [15, 50] were able to show exponential decay of the off diagonal entries with decay rate $\mu := \min_{1\leq k\leq n-1} \lambda_k - \lambda_{k-1}$:

$$|a_k(t) - \lambda_k| \leq ce^{-2\mu t} \tag{5.14}$$

$$b_k^2(t) \leq ce^{-2\mu t} \tag{5.15}$$

We thus have a matrix flow that conserves eigenvalues and sends the off diagonal terms to zero. As a result, we find that the flow sends each Jacobi initial matrix L_0 to a diagonal matrix containing the eigenvalues of L_0 on the diagonal. In fact, the limiting matrix $L(t) \overset{t\to\infty}{\longleftarrow} \Lambda$ actually 'sorts' the eigenvalues of L_0 in the sense that $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ where $\lambda_1 > \cdots > \lambda_n$ (cf. [7, 50]. These results imply that the Toda flow can be used to solve eigenvalue problems: Feed any initial Jacobi matrix into a numerical solver for the Toda equation (4.103) solve until deflation or diagonalization have occured. This and related strategies to compute eigenvalues are referred to as the Toda algorithm later on.

As opposed to the QR algorithm, the Toda algorithm is completely insensitive to

shifting. This is easy to see in (5.13), because

$$q(t) = \frac{e^{\Lambda t} q_0}{\|e^{\Lambda t} q_0\|} = \frac{e^{(\Lambda - \mu \mathrm{id})t} q_0}{\|e^{(\Lambda - \mu \mathrm{id})t} q_0\|} \tag{5.16}$$

We have seen that the first order convergence behavior of the QR algorithm is invariant under scaling the initial matrix. For the Toda flow, scaling the matrix corresponds to changing time in the ODE (4.103). In particular let $\tilde{t} = ct$, Then $\mathrm{d}\tilde{t} = c\mathrm{d}t$, ie. $\mathrm{d}/\mathrm{d}t = c\mathrm{d}/\mathrm{d}\tilde{t}$, so that

$$\frac{\mathrm{d}L}{c\mathrm{d}t} = \frac{\mathrm{d}L}{\mathrm{d}\tilde{t}} \tag{5.17}$$

and hence:

$$\frac{\mathrm{d}L}{c^2\mathrm{d}t} = \frac{\mathrm{d}L/c}{c\mathrm{d}t} = \frac{\mathrm{d}L/c}{\mathrm{d}\tilde{t}} \tag{5.18}$$

$$\frac{\mathrm{d}L}{c^2\mathrm{d}t} = \left[(L/c)_+ - (L/c)_+^T, \frac{L}{c}\right] \tag{5.19}$$

Thus, setting $\tilde{L} := L/c$ we obtain

$$\frac{\mathrm{d}\tilde{L}}{\mathrm{d}\tilde{t}} = [\tilde{L}_+ - \tilde{L}_+^T, \tilde{L}] \tag{5.20}$$

Therefore, the well known Wigner scaling of an $n \times n$–matrix, ie $\tilde{L} := L/\sqrt{n}$ corresponds to decreasing the speed of (4.103) by a factor of $\sqrt{n}$. It has been said before, that the key to making this diagonalization method practical might be a good timestepping procedure [15, 43].

Another point worth emphasizing is the tight connection between the Toda algorithm and the QR algorithm for matrix diagonalization:

Lemma 5.3. *Let L_0 be a Jacobi matrix and define $M_0 = \exp\{L_0\}$. Then $\exp L(n) = M_n$ where M_n is the n–th step in the QR algorithm with initial matrix M_0.*

Proof. We define the matrices V and R by setting $\exp tL_0 = V(t)R(t)$. We first show that $U = V^T$ where U was defined in equation (5.2). Clearly:

$$\frac{d}{dt}U^T(t) = -U^T B = -U^T[L_+ - L_+^T] = \\ = U^T[L_- - L_-^T] = U^T\left[L - L_0 - L_+ - L_-^T\right] = U^T L + U^T R_1 \quad (5.21)$$

where R_1 is the upper triangular matrix $R_1 = -L_0 - L_+ - L_-^T$. Since $U^T L = U^T U L_0 U^T$ we obtain: $dU^T/dt = L_0 U^T + U^T R_1$. Variation of constants shows that $U^T(t) = \exp\{tL_0\}R_2$, where R_2 is the upper triangular matrix with positive entries solving $\dot{R}_2 = R_1 R_2$. Set $R = R_2^{-1}$ and use uniqueness of the factorization to obtain: $\exp\{tL_0\} = U^T(t)R(t)$ as claimed.

Now we prove the lemma using induction: Let $\exp\{L_0\} = Q_0 R_0$. Then $R_0 Q_0 = Q_0^T \exp\{L_0\}Q_0$. By the above discussion, $Q_0 = U^T(1)$, so that we have proved $R_0 Q_0 = \exp\{U(1)L_0 U(1)^T\} = \exp\{L(1)\}$. Now assume the result holds true for m. Clearly

$$M_{m+1} = Q_m^T M_m Q_m = Q_m^T \exp\{L(m)\}Q_m \quad (5.22)$$

Now think of $L(m)$ as the new initial condition for the Toda flow, then the result follows from the same reasoning as above. □

It turns out that the Toda flow in matrix form (4.103) is just one instance of a whole class of matrix ODEs all of which lead to diagonalization procedures, or, in other words, the solutions of which preserve eigenvalues and converge to diagonal matrices [15, 43]. We call this class of matrix ODEs the Deift–Li–Nanda–Tomei (DLNT) framework. The following results are quoted from [43], their proofs aren't much different from the proofs in the ordinary Toda case:

Theorem 5.1. *Let $L_0 \in GL(\mathbb{R}, n)$ and choose G analytic on the spectrum of L_0. Consider:*

$$\begin{cases} \dot{L} = [pr_{\mathfrak{k}} G(L), L] \\ L(0) = L_0 \end{cases} \tag{5.23}$$

where $G(L) := QG(\Lambda)Q^T$, where Q and Λ are an eigenvalue decomposition of L and $pr_{\mathfrak{k}}$ is the skew symmetric projection. Then

1. *$L(t)$ has the same eigenvalues as L_0.*
2. *$L(t) = Q^T(t)L_0Q(t)$ where*

$$\exp\{t \cdot G(L_0)\} = Q(t)R(t) \tag{5.24}$$

   is the unique QR decomposition (cf. section 2.1).

3. *For all $n \in \mathbb{N}_0$*

$$\exp\{G(L(n))\} = M_n \tag{5.25}$$

   where M_k is the n^{th} step in the unshifted QR algorithm (cf. algorithm 3) applied to $M_0 = exp\{G(L_0)\}$.

4. *If L_0 is Hermitian, G injective and real on the spectrum of L_0, then $L_\infty = \lim_{t\to\infty} L(t)$ exists and equals a diagonal matrix consisting of the eigenvalues of L_0.*

5. *Assume that L_0 is Jacobi. Then in spectral variables $L_0 \mapsto (\lambda, q)$, where $L_0 =$*

$Q^T \Lambda Q$, $\Lambda = diag(\lambda)$, $q = Q^T e_1$, we have that

$$\begin{cases} \dot{\lambda} = 0 \\ \dot{q} = g(\Lambda)q - (q, g(\Lambda)q)q \\ q(t) = \frac{\exp\{tG(\Lambda)\}q_0}{\|\exp\{tG(\Lambda)\}q_0\|} \end{cases} \tag{5.26}$$

Several remarks are in order:

- The injectivity needed in the theorem is extremely weak. We really only need that for $i \neq j$: $g(\lambda_i) \neq g(\lambda_j)$ (cf. [43], p.320).
- Paraphrasing a statement on p. 318 in[43] the result suggests a framework for generating eigenvalue algorithms. Every function g corresponds to an algorithm, or in Nanda's own words: "... the choice of an algorithm is ... a choice of a vectorfield on the set of all symmetric matrices having the same eigenvalues."
- The Toda as well as the unshifted QR algorithm fit into this framework. This will be discussed in the closing paragraphs of this section, after the following result.

It turns out that all of the flows on Jacobi matrices generated using the DLNT framework are completely integrable and commute with one another. This result is simple, but seems not to have been stated elsewhere in the literature:

Proposition 5.4. *Let $L_0 \in \mathcal{J}$, the space of zero trace Jacobi matrices equipped with the Kostant–Kirillov–form ω_{KK}. Let G be a function defined, real, injective and analytic on the spectrum of L_0 and denote an antiderivative of G by $\hat{G}$. Then the*

flow on $(\mathcal{J}, \omega_{KK})$ *generated by*

$$H_G(L) := tr\hat{G}(L) \tag{5.27}$$

flow described by the ODE (5.23). This flow is completely integrable and Poisson–commutes under the Kostant–Kirillov–bracket $\{.,.\}_{KK}$ *with the flow generated by* H_F*, if* F *denotes a function with the same properties as* G*.*

Proof. A result that can be found in [12, 49] tells us that in the phase space $(\mathcal{J}, \omega_{\mathrm{KK}})$ the explicit ODE associated with any Hamiltonian H has the form:

$$\begin{cases} \dot{L} = [\mathrm{pr}_{\mathfrak{k}} \partial H(L), L] \\ L(0) = L_0 \end{cases} \tag{5.28}$$

where $\mathrm{pr}_{\mathfrak{k}} L = L_+ - L_+^T$ is the skew–symmetric projection, ∂H is the matrix of entrywise derivatives of the matrix function H:

$$(\partial H)_{ij}(L) = \frac{\partial H}{\partial_{l_{ij}}}(L),\ 1 \leq i, j \leq n,\ |i-j| \leq 1 \tag{5.29}$$

It is sufficient for our purposes to show that for Jacobi matrices T

$$\frac{\partial H_g}{\partial t_{i,i-1}}(T) = [g(T)]_{i,i-1},\ 1 \leq i \leq n-1 \tag{5.30}$$

but

$$\frac{\partial H_g}{\partial l_{ij}}(L) = [g(L)]_{ij},\ 1 \leq i, j \leq n \tag{5.31}$$

holds true for general symmetric matrices L and implies (5.30). Since L is symmetric there exists a diagonalizing orthonormal matrix Q of eigenvectors: $L = Q^T \Lambda Q$ where

$\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. Now consider:

$$\frac{\partial H_g}{\partial l_{ij}} = \frac{\partial}{\partial l_{ij}} \mathrm{tr}\hat{g}(L) = \frac{\partial}{\partial l_{ij}} \sum_{k=1}^{n} \hat{g}(\lambda_k) = \sum_{k}^{n} g(\lambda_k) \frac{\partial \lambda_k}{\partial l_{ij}} =$$

$$= \sum_{k}^{n} g(\lambda_k) q_{ki} q_{kj} = [Q^T g(\Lambda) Q]_{ij} = [g(L)]_{ij} \quad (5.32)$$

This proves (5.31) and therefore also the first statement of the theorem. The fourth equality holds, because the eigenvalue equation $Lq_k = \lambda_k q_k$ ($q_k := Q[:, k]$) implies the following standard computation (eg [26], p. 318) in perturbation theory: Let $\dot{} = \frac{\partial}{\partial l_{ij}}$, then

$$\dot{L}q_k + L\dot{q}_k = \dot{\lambda}_k q_k + \lambda_k \dot{q}_k \quad (5.33)$$

multiplying on both sides with q_k^T from the left yields:

$$\frac{\partial \lambda_k}{\partial l_{ij}} = q_k^T E_{ij} q_k = q_{ki} q_{kj} \quad (5.34)$$

Clearly the λ_i are conserved quantities and in lemma 5.2 we proved that they commute in the Kostant–Kirillov–Poisson–bracket on $\mathcal{J}$, ie. $\{\lambda_i, \lambda_k\}_{\mathrm{KK}} = 0$. Poisson commutativity of integrals defined on the phase space is independent of the Hamiltonian that preserves them. This means in particular that the flow generated by H_G for any G is completely integrable.

Lastly we note that for G, F satisfying the assumptions of the theorem we have:

$$\{H_G, H_F\}_{\mathrm{KK}} = \sum_{i,j=1}^{n} \{\hat{G}(\lambda_i), \hat{F}(\lambda_j)\}_{\mathrm{KK}} =$$

$$\overset{[42]}{=} \frac{1}{4} \sum_{i,j,k=1}^{n} \left[\left(a_k G(\lambda_i) \frac{\partial \lambda_i}{\partial a_k} - a_{k-1} G(\lambda_i) \frac{\partial \lambda_i}{\partial a_{k-1}} \right) F(\lambda_j) \frac{\partial \lambda_j}{\partial b_k} - \right.$$

$$\left. - \left(a_k F(\lambda_j) \frac{\partial \lambda_j}{\partial a_k} - a_{k-1} F(\lambda_j) \frac{\partial \lambda_j}{\partial a_{k-1}} \right) G(\lambda_i) \frac{\partial \lambda_i}{\partial b_k} \right] =$$

$$= \sum_{i,j=1}^{n} g(\lambda_i) h(\lambda_j) \cdot \{\lambda_i, \lambda_j\}_{\mathrm{KK}} = 0 \quad (5.35)$$

This proves that the family of algorithms derived from the DLNT framework is a family of commuting completely integrable flows on $\mathcal{J}$. □

In the last paragraphs of this section we explain how typical eigenvalue algorithms fit into the DLNT framework.

The Toda algorithm in the DLNT framework. We mentioned this before (cf. section 4.4), but for clarity we repeat that setting $G = \mathrm{id}$ in (5.26) gives exactly the solution (5.13) of the Toda system in spectral coordinates (λ, q). In the spirit of proposition 5.4 we set $\hat{G}(x) := \frac{1}{2}x^2$ and obtain:

$$H_G(L) = \frac{1}{2}\mathrm{tr}L^2 = H_T(L) \quad (5.36)$$

Denote the solution to the ODE (4.103) with initial condition $L(0) = L_0$ by $L(t)$, then using (5.25) we obtain

$$\exp\{L(n)\} = M_n \quad (5.37)$$

where M_n is the n^{th} step of the QR algorithm applied to the initial matrix $M_0 = \exp\{L_0\}$. This will be useful in a later section when we will design algorithms to sample the run times of the Toda algorithm.

The unshifted QR algorithm in the DLNT framework. Assume that L_0 is a strictly positive definite Jacobi matrix. The Nanda framework allows us to choose a G such that integer time evaluations of the flow $L(t)$ on $(\mathcal{J}, \omega_{\text{KK}})$ generated by H_G satisfy:

$$L(n) = M_n \tag{5.38}$$

where M_n is the n^{th} step of the unshifted QR algorithm applied to the initial matrix $M_0 = L(0) = L_0$. I know of two ways to prove that $G(x) = \log(x)$ is the correct choice in the Nanda framework:

1. The simplest way is to just read it off equation (5.25).

2. The other, more interesting, way is to show that in spectral coordinates the action of the QR algorithm is given by:

$$\begin{cases} \lambda(m) = \lambda_0 \\ q_m = \frac{\Lambda^m q_0}{\|\Lambda^m q_0\|} \end{cases} \tag{5.39}$$

   Compare this with (5.26) to see that any flow which has the desired property (5.38) has to have $g = \log(x)$.

   One approach to proving (5.39) is using explicit matrix arithmetic. This is done in [43]. Another, somewhat more abstract, approach is found on p. 445 in [53]: Here QR is viewed as an instance of the simultaneous iteration algorithm and

a version of the spectral theorem for self adjoint operators with cyclic vectors is invoked.

Now we choose $\hat{G}(x) = x\log(x) - x$ and define:

$$H_{\mathrm{QR}}(L) := H_G = \mathrm{tr}(L\log(L) - L) \tag{5.40}$$

Propostion 5.4 tells us now that the ODE corresponding to H_{QR} is given by:

$$\dot{L} = [\mathrm{pr}_{\mathfrak{k}}\log(L), L] \tag{5.41}$$

this is what [15, 43] present.

The shifted QR algorithm in the DLNT framework. First of all we determine the impact of a shifted QR Step on the spectral representation. This computation is very similar to the one in [43] for the unshifted `QRStep`. Let

$$T - \mu = QR \tag{5.42}$$

$$\hat{T} = RQ + \mu \tag{5.43}$$

The eigenvalues of T stay unaffected by this step, because:

$$\hat{T} = Q^T(T - \mu)Q + \mu = Q^T T Q \tag{5.44}$$

The impact on the first row of the eigenvector matrix $q = Q^T e_1$ is given by:

$$\hat{q} = \frac{(\Lambda - \mu)q}{\|(\Lambda - \mu)q\|} \tag{5.45}$$

This can be derived from $\hat{T} = Q^T T Q =: Q^T U \Lambda U^T Q$, ie $\hat{q} = \hat{U}^T e_1 = U^T Q e_1$. It follows from the equivalence of the QR method with the Gram–Schmidt process (see 2.1) that

$$q = Qe_1 = \frac{(T-\mu)e_1}{\|(T-\mu)e_1\|} \tag{5.46}$$

which implies:

$$\hat{q} = U^T Q e_1 = \frac{U^T(T-\mu)e_1}{\|(T-\mu)e_1\|} = \frac{U^T(T-\mu)e_1}{\|U^T(T-\mu)e_1\|} = \frac{(\Lambda-\mu)U^T e_1}{\|(\Lambda-\mu)U^T e_1\|} = \frac{(\Lambda-\mu)q}{\|(\Lambda-\mu)q\|} \tag{5.47}$$

as required. Now assume that T_n (with spectral representation (λ, q_n)) is the n^{th} step in the shifted QR algorithm starting from T_0 where the shift μ_{n-1} corresponds to the `QRStep` from T_{n-1} to T_n. Then it turns out that

$$q_m = \frac{(\Lambda-\mu_{m-1})\cdots(\Lambda-\mu_0)q_0}{\|(\Lambda-\mu_{m-1})\cdots(\Lambda-\mu_0)q_0\|} \tag{5.48}$$

- In the case that $\mu_k \equiv \mu$, set $g(x) = \log(x-\mu)$, $\hat{g}(x) = (x-\mu)\log(x-\mu) - x$, $H_g(L) := \mathrm{tr}\hat{g}(L)$. And the flow belonging to this Hamiltonian interpolates exactly the shifted QR algorithm.

- In the case that $\mu_k = \mu(L(k))$ we can define:

$$g_t(x) = \log(x - \mu(L(\lfloor t \rfloor)) \tag{5.49}$$

  and

$$H_{g_t}(L) = \mathrm{tr}\Big((L - \mu(L(\lfloor t \rfloor)) \cdot \log(L - \mu(L(\lfloor t \rfloor)) - L\Big) \tag{5.50}$$

this amounts to an equation where the right hand side is discontinuous at every integer time. More exactly, the Hamiltonian function changes at each integer time k to a Hamiltonian depending on $L(k-1)$.

5.2 Liouville Equations for the Toda Flow with Hermite–1 initial data

We compute the Liouville equations using explicit expressions for the distribution of the matrices at time t, when transported with the Toda flow. Before we can obtain this explicit expression, it is necessary to evaluate Jacobians of certain coordinate transformations. Our computations are similar to the approach in [19] in that we express the Jacobians as ratios of densities. For example if $y = \psi(x)$ then

$$f(x)dx = [\text{Jac } \psi^{-1}](y) f\left(\psi^{-1}(y)\right) dy = g(y)dy \tag{5.51}$$

and in particular:

$$[\text{Jac } \psi^{-1}] = \frac{g(y)}{f(x)} \tag{5.52}$$

For notational convenience, let $\Lambda := \text{diag}(\lambda)$ in the whole section.

Theorem 5.5. *Consider the solution Ψ_t to the Toda flow in the spectral coordinates (q, λ):*

$$(\tilde{q}, \tilde{\lambda}) := \Psi_t(q, \lambda) := \left(\frac{e^{t\Lambda}q}{\|e^{t\Lambda}q\|}, \lambda\right) \tag{5.53}$$

Then the Jacobian of the inverse of this transformation is given by:

$$\text{Jac } \Psi_t^{-1}(q, \lambda) = \frac{\exp\left[-t \cdot \text{tr } \Lambda\right]}{\|\exp[-t \cdot \Lambda]q\|} \tag{5.54}$$

We will prove this using a 'true random matrix calculation' in the following way: Choose a distribution on $S^n \times \mathbb{R}^n$ that we can express in terms of (q, λ) and whose pushforward under Ψ_t can be easily computed and expressed in $(\tilde{q}, \tilde{\lambda})$. In the following we denote by $\Delta(\lambda)$ the Vandermonde determinant defined in theorem 3.1 of section 3.1.2.

Proof. The distribution of choice is given by:

$$\mu := f(q, \lambda) dq d\lambda = Z^{-1} \cdot \Delta(\lambda) e^{-\frac{1}{2}\mathrm{tr}\Lambda^2} dq d\lambda \tag{5.55}$$

Here, we reuse $c_{a,b} := c_H^1 \cdot s_n^{-1}$ defined in [19] and reused in section 3.1.2, where c_H^1 is the normalization constant of the Hermite–1 ensemble (cf. equation (3.16)) and s_n is the surface area of the sphere in the positive orthant (cf. equation 3.17). We want to compute the pushforward of this measure under Ψ_t. This task is simplified by taking into account certain symmetry considerations: $\Psi_t \circ \mu$ is computed by considering the measure

$$\mu_1 = f_1(x, \lambda) dx d\lambda = \underbrace{n!(2\pi)^{-n} \prod_{j=1}^{n} \frac{\Gamma(3/2)}{\Gamma(1 + j/2)}}_{=:Z_1^{-1}} \Delta(\lambda) e^{-\|x\|^2/2 - \|\lambda\|^2/2} d\lambda dx \tag{5.56}$$

on $\mathbb{R}^{2n}$. Now, push μ_1 forward using the map $G_t(x, \lambda) = (e^{t\Lambda} x, \lambda) = (\hat{x}, \hat{\lambda})$, this is easy, because:

$$f_2(\hat{x}, \hat{\lambda}) d\hat{x} d\hat{\lambda} = \frac{e^{-t\mathrm{tr}\hat{\Lambda}}}{Z_1^{-1}} \Delta(\hat{\lambda}) e^{-\|\hat{\lambda}\|^2/2} e^{-\|\exp[-t\hat{\Lambda}]\hat{x}\|^2/2} d\hat{\lambda} d\hat{x} = f_1(x, \lambda) dx d\lambda \tag{5.57}$$

Now transform $\hat{x}$ to polar coordinates:

$$f_2(\hat{x},\hat{\lambda})d\hat{x}d\hat{\lambda} = Z_1^{-1}e^{-t\mathrm{tr}\Lambda}\Delta(\hat{\lambda})r^{n-1}\cdot j(\theta) \\ e^{-\|\hat{\lambda}\|^2/2}\exp\left\{-\frac{r^2}{2}\sum_{j=1}^{n}e^{-2\lambda_j t}\tilde{q}_j^2(\theta_1,\ldots,\theta_n)\right\}drd\theta d\hat{\lambda} \quad (5.58)$$

and integrate out the radial component:

$$\int_0^\infty \exp\left\{-\frac{\alpha r^2}{2}\right\}r^{n-1}dr \overset{s=r\sqrt{\alpha}}{=} \frac{}{\sqrt{\alpha^n}}\int_0^\infty \exp\left\{-\frac{s^2}{2}\right\}s^{n-1}ds = \frac{2^{\frac{n}{2}}\Gamma(\frac{n}{2})}{\alpha^{\frac{n}{2}}} \quad (5.59)$$

Thus we obtain a measure ν on $\mathbb{R}^n \times S^n$:

$$\nu = g(\tilde{q},\tilde{\lambda})d\tilde{q}d\tilde{\lambda} = \frac{e^{-t\mathrm{tr}\tilde{\Lambda}}\Delta(\tilde{\lambda})}{Z\|\exp[-t\tilde{\Lambda}]\tilde{q}\|^n}e^{-\|\hat{\lambda}\|^2/2}d\tilde{q}d\tilde{\lambda} \quad (5.60)$$

This proves:

$$[\mathrm{Jac}\ \Psi_t^{-1}](\tilde{\lambda},\tilde{q}) = \frac{e^{-t\mathrm{tr}\tilde{\Lambda}}}{\|\exp[-t\tilde{\Lambda}]\tilde{q}\|^n} \quad (5.61)$$

along with

$$[\mathrm{Jac}\ \Psi_t](\lambda,q) = [\mathrm{Jac}\ \Psi_t^{-1}(\tilde{\lambda},\tilde{q})]^{-1} = \frac{\|\exp\{-t\tilde{\Lambda}\}\tilde{q}\|^n}{e^{-t\mathrm{tr}\tilde{\Lambda}}} = \quad (5.62)$$

$$= \frac{e^{t\mathrm{tr}\Lambda}}{\|\exp\{t\Lambda\}q\|^n} \quad (5.63)$$

□

In the following we will compute explicitly the Jacobian of the time–t map of the Toda flow in (a,b) coordinates:

Theorem 5.6. *Let Φ_t denote the time–t map of the Toda flow in (a,b)–coordinates.*

Then

$$Jac\ \Phi_t = \frac{\prod_{j=1}^{n-1} b_j^{j-1}}{\prod_{j=1}^{n-1} \tilde{b}_j^{j-1}} \cdot \frac{e^{ttrL}}{\| \exp\{tL\} e_1 \|^n} = \frac{\prod_{j=1}^{n} \tilde{b}_i}{\prod_{j=1}^{n} b_i} \tag{5.64}$$

and

$$Jac\ \Phi_t^{-1} = \prod_{j=1}^{j-1} \frac{\tilde{b}^{j-1}}{b^{j-1}} \cdot \frac{e^{-t \cdot tr\tilde{L}}}{\| \exp\{-t\tilde{L}\} e_1 \|^n} = \frac{\prod_{j=1}^{n} b_i}{\prod_{j=1}^{n} \tilde{b}_i} \tag{5.65}$$

The idea of the proof is again to write the same distribution in terms of (a, b) and in terms of $(\tilde{a}, \tilde{b})$ and then divide the two densities to obtain the Jacobian. We'll use the 1–Hermite ensemble for this purpose.

Let S denote the spectral map $L \mapsto S(L) = (q, \lambda)$ (cf. section 3.1.2, theorem 3.1) and let Φ_t denote the time t map of the toda flow in (a, b) coordinates. To clarify the notation used below, consider the following commuting diagram:

$$\begin{array}{ccc} (a,b) & \overset{S}{\rightarrow} & (q,\lambda) \\ \Phi_t \downarrow & & \Psi_t \downarrow \\ (\tilde{a},\tilde{b}) & \overset{S}{\rightarrow} & (\tilde{q},\tilde{\lambda}) \end{array}$$

Then

$$\frac{1}{Z} e^{\frac{1}{2}\mathrm{tr}L^2} \cdot \prod_{j=1}^{n-1} b_j^{j-1} da db = \frac{\Delta(\Lambda)}{Z} e^{\frac{1}{2}\mathrm{tr}\Lambda^2} d\lambda dq = \tag{5.66}$$

$$= \frac{\Delta(\tilde{\Lambda})}{Z} \cdot \frac{e^{-t \cdot tr\tilde{\Lambda}} e^{\frac{1}{2}\mathrm{tr}\tilde{\Lambda}^2}}{\| \exp\{-t\tilde{\Lambda}\} \tilde{q} \|^n} d\tilde{\lambda} d\tilde{q} = \tag{5.67}$$

$$= \frac{1}{Z} e^{\frac{1}{2}\mathrm{tr}\tilde{L}^2} \cdot \prod_{j=1}^{n-1} \tilde{b}_j^{j-1} \cdot \frac{e^{-t \cdot \mathrm{tr}\tilde{L}}}{\| \exp\{-t\tilde{L}\} e_1 \|^n} d\tilde{a} d\tilde{b} \tag{5.68}$$

Altogether this implies that

$$\text{Jac } \Phi_t^{-1} = \prod_{j=1}^{j-1} \frac{\tilde{b}^{j-1}}{b^{j-1}} \cdot \frac{e^{-t\cdot \text{tr}\tilde{L}}}{\| \exp\{-t\tilde{L}\} e_1\|^n} \tag{5.69}$$

Proof. From the commuting diagram we obtain:

$$(\tilde{a}, \tilde{b}) = \Phi_t(a,b) = \left[S^{-1} \circ \Psi_t \circ S\right](a,b) \tag{5.70}$$

so that

$$\begin{aligned} \text{Jac } \Phi_t(a,b) = [\text{Jac } S^{-1}](\tilde{q},\tilde{\lambda})) \cdot [\text{Jac } \Psi_t](q,\lambda) \cdot [\text{Jac } S](a,b) = \\ = \frac{\prod_{j=1}^{n-1} \tilde{b}_i}{\prod_{j=1}^{n-1} \tilde{q}_i} \cdot \frac{e^{t\text{tr}\Lambda}}{\| \exp\{t\Lambda\} q\|^n} \cdot \frac{\prod_{j=1}^{n} q_i}{\prod_{j=1}^{n-1} b_i} = \\ = \frac{\Delta(\Lambda)}{\prod_{j=1}^{n-1} \tilde{b}_j^{j-1}} \cdot \frac{e^{t\text{tr}\Lambda}}{\| \exp\{t\Lambda\} q\|^n} \cdot \frac{\prod_{j=1}^{n-1} b_j^{j-1}}{\Delta(\Lambda)} = \\ = \frac{\prod_{j=1}^{n-1} b_j^{j-1}}{\prod_{j=1}^{n-1} \tilde{b}_j^{j-1}} \cdot \frac{e^{t\text{tr}\Lambda}}{\| \exp\{t\Lambda\} q\|^n} = \frac{\prod_{j=1}^{n-1} b_j^{j-1}}{\prod_{j=1}^{n-1} \tilde{b}_j^{j-1}} \cdot \frac{e^{t\text{tr}L}}{\| \exp\{tL\} e_1\|^n} \end{aligned} \tag{5.71}$$

For the Jacobian of the inverse we thus obtain:

$$\text{Jac } \Phi_t^{-1} = \frac{\prod_{j=1}^{n-1} \tilde{b}_j^{j-1}}{\prod_{j=1}^{n-1} b_j^{j-1}} \cdot \frac{e^{-t\text{tr}\tilde{L}}}{\| \exp\{-t\tilde{L}\} e_1\|^n} \tag{5.72}$$

Just as above. Now note that also

$$\text{Jac } \Phi_t(a,b) = [\text{Jac } S^{-1}](\tilde{q},\tilde{\lambda})) \cdot [\text{Jac } \Psi_t](q,\lambda) \cdot [\text{Jac } S](a,b) = \tag{5.73}$$

$$= \frac{\prod_{j=1}^{n-1} \tilde{b}_i}{\prod_{j=1}^{n-1} \tilde{q}_i} \cdot \frac{e^{t\text{tr}\Lambda}}{\| \exp\{t\Lambda\} q\|^n} \cdot \frac{\prod_{j=1}^{n} q_i}{\prod_{j=1}^{n-1} b_i} = \frac{\prod_{j=1}^{n-1} \tilde{b}_i}{\prod_{j=1}^{n-1} b_i} \tag{5.74}$$

This proves the claim. □

Next we derive the Jacobian of Flaschka's transformation, which underlies a lot of the theory of the Toda lattice (cf. section 4.4). In particular, let

$$\begin{cases} a_j = -\frac{y_j}{2}, & 1 \le j \le n \\ b_j = \frac{1}{2}\exp\left\{\frac{x_j - x_{j+1}}{2}\right\}, & 1 \le j \le n-1 \end{cases} \tag{5.75}$$

Then $F = (x, y) \mapsto (a, b) : \mathbb{R}^{2n-1} \to \mathbb{R}^{2n-1}$ and we're interested in evaluating Jac $F = \left|\frac{\partial(a,b)}{\partial(x,y)}\right|$. In the same way as above, we can derive an expression for this Jacobian if we can find a distribution that can be explicitly written in terms of both (a, b) and (x, y) variables and then dividing one by the other.

Theorem 5.7. *With the notation from above we have:*

$$Jac\ F = 2^{2n-1} \cdot \prod_{j=1}^{n-1} b_j^{-1} \tag{5.76}$$

Proof. We consider the following distribution on the space of tridiagonal symmetric matrices:

$$\begin{pmatrix} \mathcal{N}(0, \frac{1}{4}) & * & \cdots & \cdots & * \\ \log\mathcal{N}(-\ln 2, \frac{1}{4}) & \mathcal{N}(0, \frac{1}{4}) & \ddots & \ddots & \vdots \\ 0 & \log\mathcal{N}(-\ln 2, \frac{1}{4}) & \mathcal{N}(0, \frac{1}{4}) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & * \\ 0 & \cdots & 0 & \log\mathcal{N}(-\ln 2, \frac{1}{4}) & \mathcal{N}(0, \frac{1}{4}) \end{pmatrix} \tag{5.77}$$

with density:

$$f(a, b) = \prod_{j=1}^{n-1} \frac{e^{-\frac{1}{2}(2\ln 2b_j)^2}}{b_j\sqrt{\pi/2}} \cdot \prod_{j=1}^{n} \frac{e^{-2a_j^2}}{\sqrt{\pi/2}} \tag{5.78}$$

Using the constraint $\sum_{j=1}^{N} x_j \equiv c$ we can construct an inverse for Flaschka' transfor-

mation. It turns out that in particular:

$$x = \frac{c}{n} \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} + A \begin{pmatrix} 2 \ln 2b_1 \\ \vdots \\ 2 \ln 2b_{n-1} \end{pmatrix} \in \mathbb{R}^n \tag{5.79}$$

where $A \in \mathbb{R}^{n \times (n-1)}$. Also, observe that $y_j = -2a_j \sim \mathcal{N}(0,1)$, $x_k - x_{k+1} = 2 \ln 2b_k \sim \mathcal{N}(0,1)$ and that

$$A_{ij} = \begin{cases} \frac{n-j}{n}, & j \geq i \\ -\frac{j}{n}, & j < i \end{cases} \tag{5.80}$$

Note that the image of A is perpendicular to $(1, \ldots, 1)^T \in \mathbb{R}^n$ and that for

$$S = \begin{pmatrix} 1 & -1 & 0 & \cdots \\ \vdots & \ddots & \ddots & \vdots \\ 1 & 0 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & \\ \vdots & , \tilde{S} \\ 1 & \end{pmatrix} \tag{5.81}$$

we have $-A^T \tilde{S} = \mathrm{id}_{n-1}$. Now choose the columns of S as a new basis of $\mathbb{R}^n$: $x = \frac{c}{n} \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} + \tilde{S}\tilde{x}$. Then it emerges that

$$\tilde{x} = S^{-1}x, \; S^{-1} = \begin{pmatrix} \frac{1}{n} & \cdots & \frac{1}{n} \\ & -A^T & \end{pmatrix} \tag{5.82}$$

The covariance matrix of $\tilde{x}$ is then given by

$$\mathrm{Cov}(\tilde{x}) = \begin{pmatrix} \frac{1}{n^2} & 0 \\ 0 & A^T A A^T A \end{pmatrix} \tag{5.83}$$

Let $\Sigma := A^T A A^T A$. In the new basis given by the columns of $\tilde{S}$, the subdiagonal distribution thus turns out to be given by the following density:

$$f(\tilde{x})d\tilde{x} = \frac{n}{\sqrt{(2\pi)^n|\Sigma|}} \exp\left\{-\frac{\tilde{x}_1^2}{2n^{-2}} - \frac{1}{2}(\tilde{x}_2,\ldots,\tilde{x}_n)\Sigma^{-1}(\tilde{x}_2,\ldots,\tilde{x}_n)^T\right\} d\tilde{x}$$
$$= \frac{\det(S^{-1})\cdot n}{\sqrt{(2\pi)^n|\Sigma|}} \exp\left\{-\frac{\left(\sum_{j=1}^n x_j\right)^2}{2} - \frac{1}{2}x^T A\Sigma^{-1}A^T x\right\} dx \quad (5.84)$$

Note that

$$-A^T\tilde{S} = \mathrm{id} = A^T A(A^T A)^{-1} \quad (5.85)$$

so that $\tilde{S} = A(A^T A)^{-1}$ and hence $A\Sigma^{-1}A^T = \tilde{S}\tilde{S}^T$. We can thus express the density in terms of (x, y) as follows:

$$g(x,y) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} \exp\left\{-\frac{c^2}{2} - \frac{1}{2}x^T\tilde{S}\tilde{S}^T x\right\} \cdot \prod_{j=1}^n \frac{e^{-\frac{y_j^2}{2}}}{\sqrt{2\pi}} \quad (5.86)$$

and, finally, the Jacobian Jac F as:

$$\mathrm{Jac}\ F^{-1} = \frac{f(a,b,c)}{g(x,y)} =$$
$$= \frac{\frac{e^{-\frac{c^2}{2}}}{\sqrt{2\pi}}\left(\frac{2}{\pi}\right)^{\frac{2n-1}{2}} \cdot \prod_{j=1}^{n-1} b_j^{-1} \cdot \exp\left\{-\frac{1}{2}\left[c^2 + \sum_{j=1}^{n-1}(2\ln 2b_j)^2 + \sum_{j=1}^n a_j^2/\frac{1}{4}\right]\right\}}{(2\pi)^{-n}\exp\left\{-\frac{1}{2}\left[c^2 + \sum_{j=1}^{n-1}(x_{j+1}-x_j)^2 + \sum_{j=1}^n y_j^2\right]\right\}} =$$
$$= 2^{2n-1}\prod_{j=1}^{n-1} b_j^{-1} \quad (5.87)$$

□

We can use the techniques and results mentioned above to show:

Theorem 5.8. *In (x, y) coordinates the Jacobian of the time t map of the Toda flow*

Γ_t *is given by:*

$$Jac\ \Gamma_t = \frac{\prod_{j=1}^{n-1} b_j^j}{\prod_{j=1}^{n-1} \tilde{b}_j^j} \cdot \frac{e^{ttrL}}{\| \exp\{tL\} e_1 \|^n} \tag{5.88}$$

Proof.

$$\begin{aligned} \text{Jac } \Gamma_t = \text{Jac } F^{-1}(\tilde{a}, \tilde{b}) \cdot \text{Jac } \Phi_t(a,b) \cdot \text{Jac } F(x,y) = \\ = \frac{2^{2n-1}}{\prod_{j=1}^{n-1} \tilde{b}_j} \frac{\prod_{j=1}^{n-1} b_j^{j-1}}{\prod_{j=1}^{n-1} \tilde{b}_j^{j-1}} \frac{e^{t\text{tr}L}}{\| \exp\{tL\} e_1 \|^n} \frac{\prod_{j=1}^{n-1} b_j}{2^{2n-1}} \end{aligned} \tag{5.89}$$

□

Theorem 5.9. *The distribution of the matrix entries at time T is given by*

$$f_T(a,b) = c_{a,b} e^{-trL^2} \prod_{j=1}^{n-1} b_j^{j-1} \frac{e^{-ttrL}}{\| \exp\{-tL\} e_1 \|^n} \tag{5.90}$$

Proof. By Dumitriu and Edelman we have that

$$f_0(a,b) = c_{a,b} e^{-trL_0^2} \prod_{j=1}^{n-1} b_{j,0}^{j-1} \tag{5.91}$$

Clearly,

$$\begin{aligned} f_T(\tilde{a}, \tilde{b}) = \text{Jac } \Phi_T^{-1} \cdot f_0(\Phi_T^{-1}(\tilde{a}, \tilde{b})) = \\ = \frac{\prod_{j=1}^{n-1} \tilde{b}_j^{j-1}}{\prod_{j=1}^{n-1} b_j^{j-1}} \cdot \frac{e^{-t\text{tr}\tilde{L}}}{\| \exp\{-t\tilde{L}\} e_1 \|^n} \cdot c_{a,b} e^{\frac{1}{2}\text{tr}L_0^2} \cdot \prod_{j=1}^{n-1} b_j^{j-1} = \\ = c_{a,b} \prod_{j=1}^{n-1} \tilde{b}_j^{j-1} \frac{e^{\frac{1}{2}\text{tr}\tilde{L}^2 - t\text{tr}\tilde{L}}}{\| \exp\{-t\tilde{L}\} e_1 \|^n} \end{aligned} \tag{5.92}$$

□

Theorem 5.10. *Let f_T denote the distribution of the matrix elements at time T*

when transported by the Toda flow, where the initial distribution f_0 is the Hermite–1 distribution. Then the following PDE holds:

$$\frac{\partial f_t}{\partial t}(a,b) = \left(-trL + \frac{n\langle Le^{-tL}e_1, e^{-tL}e_1\rangle}{\|e^{-tL}e_1\|^2}\right) f_t + \sum_{j=2}^{n-1} b_j(a_{j+1}-a_j)\cdot\frac{\partial f_t}{\partial b_j} \tag{5.93}$$

Proof. Consider

$$\begin{aligned}\frac{\partial f_t}{\partial t}(a,b) &= c_{a,b}e^{\frac{1}{2}\mathrm{tr}L_0^2}\frac{e^{-t\mathrm{tr}L}}{\|e^{-tL}e_1\|^n}\prod_{k=1}^{n} b_k^{k-1}\sum_{j=2}^{n-1}(j-1)b_j^{-1}\dot{b}_j + \\ &+ c_{a,b}e^{\frac{1}{2}\mathrm{tr}L_0^2}\prod_{j=1}^{n-1} b_j^{j-1}\left\{-\mathrm{tr}L\cdot\frac{e^{-t\mathrm{tr}L}}{\|e^{-tL}e_1\|^n} + e^{-t\mathrm{tr}L}\frac{n\langle Le^{-tL}e_1, e^{-tL}e_1\rangle}{\|e^{-tL}e_1\|^{n+2}}\right\} = \\ &\left(-\mathrm{tr}L + \frac{n\langle Le^{-tL}e_1, e^{-tL}e_1\rangle}{\|e^{-tL}e_1\|^2}\right) f_t + \sum_{j=2}^{n-1}(j-1)\frac{f_t}{b_j}\dot{b}_j = \\ &\left(-\mathrm{tr}L + \frac{n\langle Le^{-tL}e_1, e^{-tL}e_1\rangle}{\|e^{-tL}e_1\|^2}\right) f_t + \sum_{j=2}^{n-1} b_j(a_{j+1}-a_j)\cdot\frac{\partial f_t}{\partial b_j}\end{aligned} \tag{5.94}$$

□

Chapter Six

Data Collection

The first section of this chapter explains which numerical procedures were used in our simulations. We discuss the Rutishauser–Kahan–Pal–Walker procedure for the reconstruction of a tridiagonal matrix from its spectral data in detail. We then concern ourselves with some of the software design issues that came up during the creation of the sampling infrastructure.

6.1 Numerical Algorithms

As described in the introductory section we are interested in computing the time it takes one of the diagonalization algorithms to advance a random initial condition to a matrix which is small enough on the off diagonal to allow deflation. In order to do this, it is not enough to choose an algorithm, we also have to choose an implementation. In the following we'll discuss implementation ideas after discussing three algorithmic building blocks that we require for these implementations.

6.1.1 Required Procedures

Spectral Reconstruction The explicit solution of the Toda lattice and related flows in spectral coordinates raises the practical question as to how to reconstruct a tridiagonal matrix from its spectral data numerically.

The first idea one might have is to reconstruct the matrix of eigenvectors from the eigenvalues and its first row using standard Lanczos procedures (ie the well known three term recurrence relation for orthogonal polynomials). Unfortunately, most implementations of this idea turns out to be manifestly unstable.

An elegant way of getting around this instability issue is described in [27], where the

Rutishauser–Kahan–Pal–Walker (`RKPW`) algorithm is developed. The name of the algorithm comes from its similarity to the efficient QR step using Givens rotations presented in algorithm 3 (cf. section 2.3), which is called the Kahan–Pal–Walker algorithm by some authors [27, 45]. First, recall the Definition of Givens rotations from 2.21:

$$G_{ij}(\theta) = G_{ij}(\cos\theta, \sin\theta) = \mathrm{id} - (1-\cos\theta)E_{ii} - (1-\cos\theta)E_{jj} + \sin\theta E_{ij} - \sin\theta E_{ji} \tag{6.1}$$

where $E_{ij} = E^n_{ij}$ is an $n \times n$–dimensional matrix with all entries zero apart from the one at row i, column j, which is one. Also assume $i < j$ along and define

$$c(x,y) = \frac{x}{\sqrt{x^2+y^2}} \tag{6.2}$$

$$s(x,y) = -\frac{y}{\sqrt{x^2+y^2}} \tag{6.3}$$

with slight abuse of notation we will then speak of $G_{ij}(x,y) := G_{ij}(c(x,y), s(x,y))$. Consider a matrix of the form

$$M = \begin{pmatrix} \alpha_0 & \beta_0 & 0^* & \delta \\ \beta_0 & \alpha_1 & \beta_1 e_1^* & \delta' \\ 0 & \beta_1 e_1 & T & 0 \\ \delta & \delta' & 0 & \lambda \end{pmatrix} \tag{6.4}$$

where T is a $k \times k$–Jacobi matrix, e_1 is a vector in $\mathbb{R}^k$ with all zeros except for the first entry which is equal to one and the blocks denoted by 0 are of the appropriate size and contain only zeros. All other symbols in (6.4) are numbers. The algorithm

uses the fact that for an $n \times n$ matrix of this form (6.4) the following relations hold:

$$\tilde{M} := G_{2,n}^T(\beta_0, \delta) M G_{2,n}(\beta_0, \delta) \tag{6.5}$$

$$\tilde{M} = \begin{pmatrix} \alpha_0 & \tilde{\beta}_0 & 0^* & 0 \\ \tilde{\beta}_0 & \tilde{\alpha}_1 & \tilde{\beta}_1 e_1^* & \tilde{\delta} \\ 0 & \tilde{\beta}_1 e_1 & T & \tilde{\delta}' e_1 \\ 0 & \tilde{\delta} & \tilde{\delta}' e_1^* & \tilde{\lambda} \end{pmatrix} \tag{6.6}$$

where

$$\tilde{\beta}_0 = \sqrt{\beta_0^2 + \delta^2} \tag{6.7}$$

$$\tilde{\beta}_1 = c(\beta_0, \delta)\beta_1 \tag{6.8}$$

$$\tilde{\delta}' = s(x, y)\beta_1 \tag{6.9}$$

and

$$\begin{pmatrix} \tilde{\alpha}_1 & \tilde{\delta} \\ \tilde{\delta} & \tilde{\lambda} \end{pmatrix} = G_{12}^T(\beta_0, \delta) \begin{pmatrix} \alpha_1 & \delta \\ \delta & \lambda \end{pmatrix} G_{12}(\beta_0, \delta) \tag{6.10}$$

so that

$$\tilde{\alpha}_1 = c^2\alpha_1 + s^2\lambda \tag{6.11}$$

$$\tilde{\delta} = cs(\alpha_1 - \lambda) \tag{6.12}$$

$$\tilde{\lambda} = s^2\alpha_1 + c^2\lambda \tag{6.13}$$

Note that this way, the upper principial minor $\tilde{M}^{(n-1)}$ is a Jacobi matrix. Intuitively it should be clear that iterative application of this strategy to the initial matrix

$$L_0 = \begin{pmatrix} 1 & q^T \\ q & \Lambda \end{pmatrix} \tag{6.14}$$

yields the tridiagonalization

$$\begin{pmatrix} 1 & 1 \\ 1 & T \end{pmatrix} \tag{6.15}$$

where T denotes the required Jacobi Matrix. We formalize the procedure in algorithm 4 and establish its effectiveness in lemma 6.1.

Algorithm 4 `RKPW` – Reconstruct Jacobi Matrix from Spectral Data

Require: $q \in S^{n-1}$, $\Lambda = \mathrm{diag}(\lambda)$

1: $L \leftarrow \begin{pmatrix} 1 & q^T \\ q & \Lambda \end{pmatrix}$
2: **for** $j = 3$ to $n+1$ **do**
3: **for** $i = 2$ to $j-1$ **do**
4: $L \leftarrow G^T_{i,j}(l_{i,i-1}, l_{j,i-1}) L G_{i,j}(l_{i,i-1}, l_{j,i-1})$
5: **end for**
6: **end for**
7: $\begin{pmatrix} 1 & e_1^T \\ e_1 & T \end{pmatrix} \leftarrow L$
8: **return** T

Lemma 6.1. *Let T denote a Jacobi matrix with strictly positive off diagonal entries and let $T = Q\Lambda Q^T$ and $q := |Q^T e_1|$ where the absolute value is taken componentwise. Then applying algorithm 4 to the initial matrix of the form (6.14) yields as a result the matrix (6.15).*

Proof. The above discussion about turing M into $\tilde{M}$ implies that in algorithm 4 in the j–th outer loop, right after executing the i–th inner loop the L has the following

structure:

$$L = \left(\begin{array}{c|ccc|c} 1 & \left(\sum_{k=1}^{i-1} q_k^2\right)^{\frac{1}{2}} & 0 \cdots \cdots & 0 & q_j \cdots q_n \\ \left(\sum_{k=1}^{i-1} q_k^2\right)^{\frac{1}{2}} & & & \vdots & \\ 0 & & & 0 & \\ \vdots & & T_{ij} & * & 0 \\ \vdots & & & \vdots & \\ 0 & \cdots & 0 * \cdots & * & \\ \hline q_j & & & \lambda_j & \\ \vdots & & 0 & \ddots & \\ q_n & & & & \lambda_n \end{array}\right) \tag{6.16}$$

Here the zeros shown in the first and last columns of the upper left block are meant to cover the first $i-1$ entries and T_{ij} is a $(j-2)\times(j-2)$–dimensional matrix. This means that ultimately, when the algorithm has finished, the matrix has the predicted structure, because the lower right entries will conspire with the Jacobi matrix $T_{n,n+1}$ to form an $n \times n$–dimensional Jacobi matrix S.

Lastly, we have to prove that $S = Q\Lambda Q^T$ with $Q^T e_1 = q$. Then we can use the implicit Q theorem 2.5 to conclude that $S = T$. By construction the product of the Givens matrices employed in algorithm 4 has the structure

$$\begin{pmatrix} 1 & 0 \\ 0 & U \end{pmatrix} \tag{6.17}$$

where Q is an orthogonal matrix (because Givens rotations are orthogonal and orthogonality is preserved under multiplication). Now again by construction $S = U^T\Lambda U$ and $U^T q = e_1$, which implies $q = Ue_1 = Q^T e_1$. The claim follows. □

The stability of this algorithm is benign because of the fact that only rotations

are used.

General QR Step We need to invoke a procedure which yields the QR decomposition of a given matrix and then evaluates the matrix multiplication RQ. An algorithm that accomplishes the first task can be found in algorithm 1. The time complexity of these operations under no special assumptions regarding the initial matrix is given by $\mathcal{O}(n^3)$.

Fast QR Step for Jacobi Matrices For Jacobi matrices we do not need to evaluate the QR decomposition to compute the QR step. Instead we can use algorithm 3 to compute one step in the QR iteration directly. It can be implemented to run in $\mathcal{O}(n)$.

6.1.2 Sampling hitting times for general symmetric matrices

ODE Recall from section 5.1 that any instance of a diagonalization algorithm from the DLNT framework consists of a flow on tridiagonal matrices, determined by some Hamiltonian $\hat{G}$. The resulting vector field is in particular autonomous, so that by elementary ODE theory L_{t+s}, started at L_0 is the same as L_t, started at L_s with initial value L_0. Suppose we are given a procedure `RHS` which evaluates the right hand side of the basic DLNT equation along with a generic `ODESolver`, which requires an initial value, a final time and a step size and `RHS` to return an approximation of the value of the solution of the flow at the final time, then the obvious way to generate the desired hitting time data is using algorithm 5.

In the general case, this approach is rather time consuming if we want it to yield good results, because of two reasons: Firstly, in order for the generic ODE solver to

Algorithm 5 `DLNT-ODE` – Computes hitting time data for DLNT Flows

Require: $A \in \mathbb{R}^{n\times n}$, $A^T = A$, $\epsilon > 0$, T, h, G

```
X ← A, n ← 0
while Condition(X) > ε do
    X ← ODESolver(X, T, h, RHS(G))
    n ← n + 1
end while
return n · T
```

perform accurately, the step size should be rather small. Secondly, computing the right hand side of the DLNT equation is costly in the case of general matrices and general Hamiltonians $\hat{g}$. Not only do we have to multiply full matrices to evaluate the Lax pair, we also have to compute the full eigendecomposition, apply $G = \hat{G}'$ to the eigenvalues and reverse the spectral transformation. In standard implementation this will cost a number of operations cubic in the size of the matrix.

Note, however that solving the Toda ODE can be accomplished much cheaper, because the right hand side of the Toda equation has a simple and explicit structure: For the Toda matrix ODE, each call to `RHS` only requires a number of operations on the order of n, the matrix size. In contrast, computing the evolution of the QR flow using this approach seems much more challenging.

Recursive The recursive approach to generating the hitting data relies on the fact that the time one map for equations from the DLNT framework can always be written in terms of the QR step. Here, the procedure `QRStep` denotes a generic implementation of the map $M = QR \mapsto \tilde{M} = RQ$.

To start, suppose that a routine `Eigh` is available which computes an eigenvalue decomposition along with the eigenvector matrix. Then one way of computing the time one iterates of the DLNT flows is given by algorithm 6. This routine could seem less efficient than necessary, since for example the eigenvalues Λ are preserved and

Algorithm 6 `DLNT-Recursive` – Computes hitting time data for DLNT Flows

Require: $A \in \mathbb{R}^{n\times n}$, $A^T = A$, $\epsilon > 0$, $g(\lambda)$

```
X ← A, n ← 0
while Condition(X) > ε do
    [Q, Λ] ← Eigh(X)
    Y ← Q e^{g(Λ)} Q^T
    Y ← QRStep(Y) {Full Matrix!}
    [Q, Λ] ← Eigh(Y)
    X ← Q g^{-1}[ln(Λ))] Q^T
    n ← n + 1
end while
return n
```

shouldn't have to be recomputed at each step. Computing the eigenvalues, however adds little in terms of complexity since we have to compute the full eigenvectors at any step. Also, note that for specific instances of the DLNT framework, the procedure simplifies a lot.

Suppose that routines `Expm` and `Logm` are available to compute the matrix exponentials and logarithms (in the functional calculus) [40]. Then computing the Toda evaluation is simple (see algorithm 7).

The classical QR algorithm is just the simplest possible instance in this series, since

Algorithm 7 `Toda-Recursive` – Computes hitting time data for Toda Flow

Require: $A \in \mathbb{R}^{n\times n}$, $A^T = A$, $\epsilon > 0$, $g(\lambda)$

```
X ← A, n ← 0
while Condition(X) > ε do
    Y ← Expm(g(X))
    Y ← QRStep(Y) {Full Matrix!}
    X ← Logm(g(Y))
    n ← n + 1
end while
return n
```

$G = \ln$ – consider algorithm 8.

One immediately apparent drawback of this idea as compared to the `DLNT-ODE` scheme is that it only allows for integer evaluations of the Toda algorithms, whereas the `TodaODE` method allows us to choose the evaluation interval T of any size.

Algorithm 8 `QR-Recursive` – Computes hitting time data for QR Flow

Require: $A \in \mathbb{R}^{n\times n}$, $A^T = A$, $\epsilon > 0$, $g(\lambda)$

1: $X \leftarrow A$, $n \leftarrow 0$
2: **while** `Condition`$(X) > \epsilon$ **do**
3: $\quad X \leftarrow$ `QRStep`(X)
4: $\quad n \leftarrow n + 1$
5: **end while**
6: **return** n

Note that even for the simplified cases each loop iteration consists of a number of rather complex matrix operations, including a full symmetric `QRStep` which in general has a cost of $\mathcal{O}(n^3)$ (as do typical matrix exponential and logarithm subroutines). A great advantage is, however, that the evaluations are exact up to the accuracy properties of the borrowed `Expm` and `Logm` routines.

In practice it turns out that this approach yields results faster than the `TodaODE` approach. Care must be taken because the `QRStep` method becomes problematic for very stiff matrices which are likely to result either from unscaled ensembles or from ensembles that yield highly asymmetric eigenvalue distributions when the convergence to the Wigner law hasn't yet set in. For the ensembles that we are studying this problem does not appear.

6.1.3 Sampling hitting times for Jacobi Matrices

Recursive In the case of Jacobi Matrices, the QR algorithm can be sped up in a rather standard way that brings down the cost of one call to `QRStep` to linear in the size of the matrix (cf. algorithm 3). This trick along with the shifting idea is the basis of the success of the QR algorithm. Since the change only concerns the `QRStep` procedure we don't give pseudo code for this case.

What we call the recursive implementation of the Toda algorithm can be written in the same way as the recursive implementation of the Toda algorithm for full sym-

metric matrices (algorithm 7). Unfortunately, this implementation can not take advantage of the special sparse structure of the Jacobi matrices, because this structure is destroyed by the matrix exponential function: In general the result of applying the matrix exponential to a Jacobi matrix is a full symmetric matrix, so that we now have to apply a full symmetric QR step. Taking the matrix logarithm returns a Jacobi matrix.

Spectral In the Jacobi and more general cases, the dynamics of the DLNT family of equations can be explicitly solved in the spectral variables (cf. theorem 5.1). This explicit representation is particularly simple in the case of the Jacobi matrices. It is possible to use this representation to generate samples of hitting times, if we have procedures to map back and forth between the tridiagonal matrix T and its spectral representation (λ, q), where $T = Q\Lambda Q^T$ and $q = Q^T e_1$.

The problematic direction here is going from (q, λ) to T, because the standard Lanczos procedure one would try to use is manifestly unstable. We have already discussed the `RKPW` algorithm with has more favorable stability properties. Unfortunately, even this improved reconstruction procedure is unstable for points on trajectories after long times. We give pseudo code for the general case of an equation from the DLNT framework.

Algorithm 9 `TodaSpectral` – Computes hitting time data for Toda Flow

Require: $A \in \mathbb{R}^{n\times n}$, $A^T = A$, $\epsilon > 0$, h

$X \leftarrow A$, $t \leftarrow 0$
$Q, \Lambda \leftarrow$ `SpectralDecomp`(A)
$p = q = Q[0, :]$
while `Condition`$(X) > \epsilon$ **do**
  $p = e^{t\cdot G(\Lambda)}q / \|e^{t\cdot G(\Lambda)}q\|$
  $X \leftarrow$ `RKPW`(Λ, p)
  $t \leftarrow t + h$
end while
return t

Table 6.1: Overview of sensible implementation alternatives for various diagonalization procedures on various types of matrices

	Matrix Type	
Diagonalization Algorithm	Jacobi	General Symmetric
Toda	Recursive, Spectral, ODE	Recursive, ODE
QR	Optimized Recursive, Spectral	Recursive, ODE
General DLNT	Recursive, Spectral	Recursive, ODE

6.2 Implementation Issues

In order to run the hitting time simulations we implemented the numerical procedures described in section 6.1 for use on a computing cluster. All the algorithms were implemented in Python except for the RKPW reconstruction scheme which was implemented in C. Disctributed computing functionality was obtained using the Python module mpi4py.

The code that was used to generate the numerical results provides the following functionality:

1. There are classes for general symmetric and Jacobi matrices providing efficient implementations for typically encountered matrix operations.

2. There are classes that allow for sampling from the initial distributions described above.

3. There are classes implementing the algorithmic solution procedures for the DLNT eigenvalue algorithms.

When running the code to generate runtime simulations in addition to supplying a list of algorithms and initial distributions one has to input a list of (small) values that serve as required tolerance for the stopping condition. The code will then take care to sample a prespecified number of runtimes for each given tolerance (to save time we follow each trajectory until the smallest tolerance is undercut by a complete subdiagonal block and save the first time that each of the tolerances is undercut) and algorithm/ initial distribution pair.
The results are then written to text files and the user can determine the granularity of the information to be saved. As a last step the textfiles can be parsed and saved to a database for further processing.

6.2.1 Matrix Classes

The idea behing the design of the matrix classes is that different types of matrices have different requirements on the implementation of the basic which we need as building blocks for the run time simulation. As a result we provided a 'Matrix' interface which provides the general types of operations we expect to exist for all matrix types. The two matrix types that we implement are the class 'JacobianMatrix' and 'FullSymmetricMatrix' (cf. figure 6.1). We now describe some noteworthy differences between the implementations of the inherited methods in the two matrix types:

1. The data in `JacobiMatrix` is stored in two arrays, one containing the diagonal elements, the other one containing the subdiagonal elements. For the `FullSymmetricMatrix` type we decide against parsimony of memory for convenience of use by storing the data as a full `numpy array`. Since we are dealing

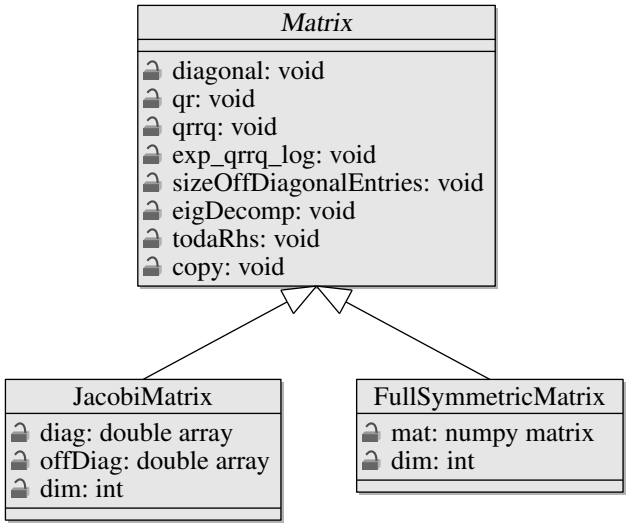

Figure 6.1: Inheritance relations between the matrix classes

with symmetric matrices, this requires twice as much memory as necessary, but the increase in speed and convenience when using readymade functions is considerable.

2. The method `qrrq()` is implemented very differently in the two types. For `JacobiMatrix` we use the optimized QR step that was presented in algorithm 3. This algorithm was implented in C for efficiency purposes and was compiled as a Python module so that it can be called from Python code. For `FullSymmetricMatrix` the QR step was implemented as a (slow and) traditional QR decomposition (called from an existing numerical library) in combination with a matrix multiplication.

3. The Toda timestep `exp_qrrq_log()` is implemented similarly in both cases, because matrix exponentiation destroys the Jacobi structure. The only difference is that before the actual computation the Jacobi matrix needs to be converted into a full matrix.

4. A very important procedure for the empirical methodology is the function

`sizeOffDiagonalEntries`. In the introductory section it was laid out that we are interested in running the diagonalization algorithms until an off diagonal block becomes small. This procedure is where the 'smallness' is measured. For the type `JacobiMatrix` implementation is straightforward: we just return the absolute values of the subdiagonal terms. For the type `FullSymmetricMatrix` the implementation is a bit more complicated and to ensure efficiency we employ a dynamic programming scheme: Consider a full symmetric matrix A of size $n \times n$. Let m_{pq} denote the maximum matrix entry by its absolute value in the (lower left) block $\{a_{ij} : p \leq i \leq n, 1 \leq j \leq q\}$. Then we have the three term recurrence:

$$m_{pq} = \max\{m_{p-1,q}, m_{p,q-1}, |a_{pq}|\} \tag{6.18}$$

where we set $m_{pq} = 0$ if the corresponding block would be empty.

6.2.2 Initial Data Classes

For our purposes the initial data for the diagonalization procedures is sampled from some random matrix ensemble. In this section we describe the sampling procedures for the ensembles we used in our simulations and how they were implemented.

Since we want to specify initial ensembles for the two different matrix types from the section 6.2.1 (`JacobiMatrix` and `FullSymmetricMatrix`) it makes sense to define an abstract class `Sampler` which outlines the functionality required from the actual initial data samplers, irrespective of the matrix types they return. As seen in figure 6.2 this basic functionality is rather simple. The only thing that might be unexpected is the existence of the `scalingFactor` attribute. This is a result of the necessity to simulate scaled versions of the initial ensembles. The particular scaling employed

here will be the Wigner scaling (each matrix entry is divided by the square root of the matrix dimension), so that if the initializer `__init__` receives a `true` in the `WignerScaling` argument, then the `scalingFactor` is set equal to the square root of `dim`, otherwise it is set equal to one. In order to achieve the desired scaling before returning the sample matrix from `_sample` we always divide each entry by the scaling factor. We now give some detail about the sampling procedures:

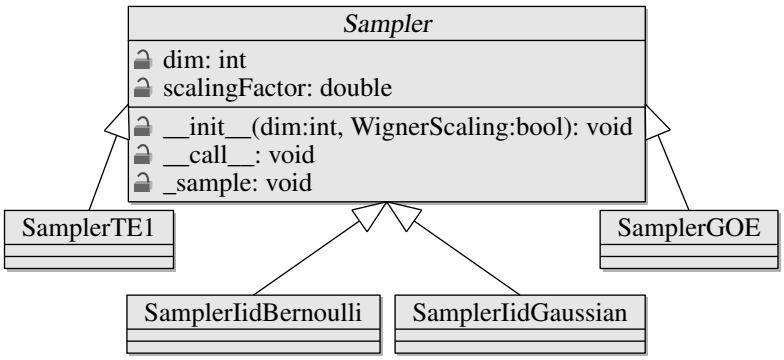

Figure 6.2: Inheritance relations between the initial data classes

1. The classes `SamplerIidBernoulli` and `SamplerIidGaussian` both return instances of `FullSymmetricMatrix`. Here, the upper triangular entries (ie. diagonal and strictly upper triangular part) are sampled iid from either the Bernoulli or the Gaussian distribution with mean zero and variance one. The strictly upper triangular part is then reflected down to obtain the lower triangular part.

2. The class `SamplerGOE` returns a `FullSymmetricMatrix` instance where the entries are sampled from the Gaussian orthogonal ensemble described in section 3.1.1 and then multiply by the square root of two in order to have all off diagonal variances equal to one.

3. The initial data class called `SamplerTE1` returns a random Jacobi matrix sam-

pled from the Hermite–1 ensemble described in [19], multiplied by a factor of $\sqrt{2}$ in order to keep the correspondence with the GOE sampler via Householder transformations in place. In particular it returns an instance of the class `JacobiMatrix`.

4. In addition to the above Wigner–type initial matrix classes we have designed classes that return samples from the uniform doubly stochastic Jacobi ensemble (`SamplerUniformTriDiagonal`) as well as the Jacobi matrices with uniform eigenvalue distribution (`SamplerUniformEig`). For simplicity these aren't shown in figure 6.2.

6.2.3 Algorithms Classes

In order to facilitate better usability we decided to use the 'Factory' design pattern [25] to implement access to the individual diagonalization algorithms. In this approach a class `DiagonalizationAlgoFactory` is defined which contains a static method `createAlgo` that takes a string argument (the name of an algorithm) and returns an instance of a class which implements this diagonalization algorithm. Since all the algorithms are required to possess the same basic functionality we again chose to implement the algorithms as derived classes from a common interface `DiagonalizationAlgo`. This interface possesses several attributes, like `tolList` and `scaledTolList`, which store the tolerance levels up to which the run times will be computed. The attribute `scaledTolList` is equal to `tolList` after division by the `scalingFactor` (which can be equal to either one or the square root of the matrix dimension, depending on whether or not we were using the Wigner scaling). The type of attribute `mat` can be either `JacobiMatrix` or `FullSymmetricMatrix` depending on the initial data used for the current algorithm. This attribute stores the current

'state' of the algorithm, ie. the current iteration in the QR or Toda algorithm. The fact that the state of the algorithm can be of different types is important, because it allows us to implement the eigenvalue algorithms in terms of elementary procedures which are optimized for the current matrix type. For example, the `_step()` method in the `QRAlgo` class would be implemented as the `qrrq()` method which the `mat` instance inherits from the class `Matrix` that was described in section 6.2.1. Depending on the actual type that `mat` has, an appropriate implementation is chosen: For `FullSymmetricMatrix` this is the standard and slow $\mathcal{O}(n^3)$ implementation, for `JacobiMatrix` this is the optimized $\mathcal{O}(n)$ version.

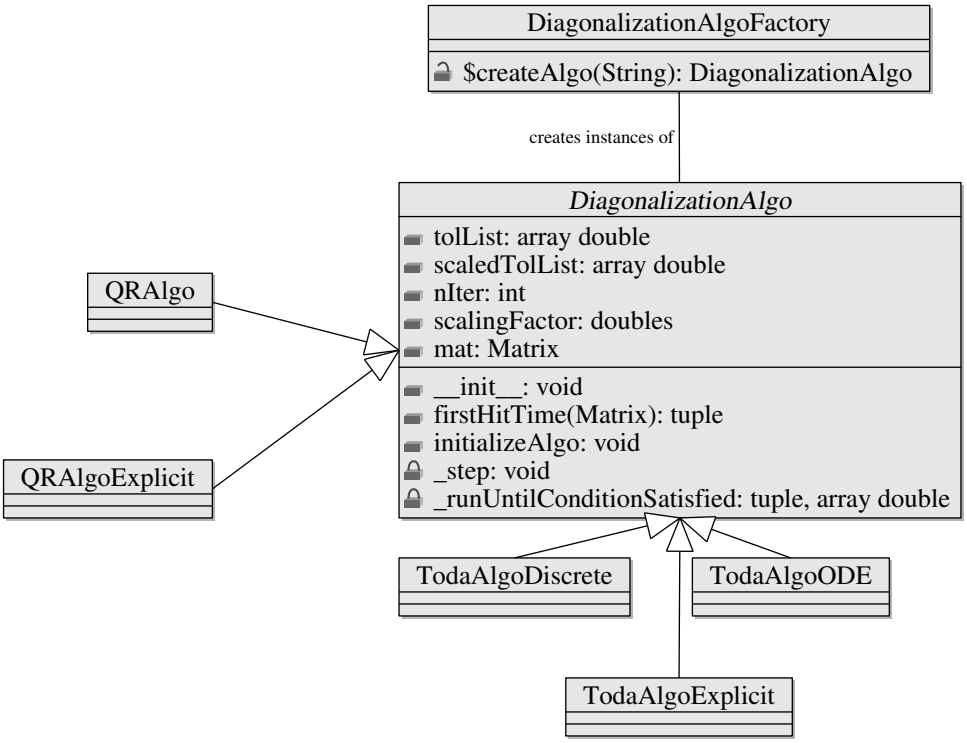

Figure 6.3: Inheritance relations between the algorithm classes

The remaining classes implement the numerical algorithms from section 6.1. The exact correspondence can be derived from the table 6.2. In this table, the nomenclature for the numerical algorithms is the same as the one in 6.1.

Table 6.2: Correspondence between implemented classes and diagonalization algorithm implementations

	Numerical Algorithm Correspondence	
Algorithm Class	Jacobi	General Symmetric
`QRAlgo`	Optimized Recursive	Recursive
`QRAlgoExplicit`	Spectral	(not implemented)
`TodaDiscrete`	Recursive	Recursive
`TodaExplicit`	Spectral	(not implemented)
`TodaODE`	ODE	ODE

6.2.4 Data Generation and Handling

In this section we will make some remarks about how the code is used to generated and handle the data after it has been generated. In particular we will talk about how the database is organised that contains the simulation results and we will sketch how functions access this database for purposes of plotting and analyzing the results.

Data Generation Before starting the simulation the following parameters are supplied: A list of tolerances as well as lists of matrix dimensions and algorithm types, the initial distribution, whether or not Wigner Scaling is to be applied, certain output options, a stopping condition (eg. stop when first block has become small), the overall number of samples as well as the number `samplesPerBatch` of samples to be generated during one 'run' of one individual processor.

For each individual processor an object of type `PerformanceSampler` is created which controls the simulation procedure. At the beginning each `PerformanceSampler` object is initialized with the list of tolerances. Then each processor requests a matrix size along with an algorithm type and samples `samplesPerBatch` run times by running the algorithm until the stopping condition is satisfied. The processors then write the results of the current 'run' into a textfile, request a new matrix size and

algorithm type from the master processor and the process starts from the beginning.

Data Handling After the simulation has finished, the simulation results are contained in text files which will now be read into a database powered by sqlalchemy, a popular Python module. The database contains two tables, one table called `Experiments` which contains information about which specific experiments were conducted (eg. dimension, current tolerance, initial distribution, which algorithm was used) each experiment is also associated with an id. The other table is called `HittingData` and contains specific information about the sampling results: The complete initial and final matrix, the subdiagonal index at which the specified stopping condition was satisfied as well as the value of the final matrix at that subdiagonal index and – most importantly – the time it took until the stopping condition was achieved. The rows of the `HittingData` table are connected to the `Experiments` table, because `HittingData` contains a column which references the corresponding experiment's id. All necessary analysis can now be carried out using the data stored in the database.

Chapter Seven

Numerical Results

This chapter contains the core results of this thesis as we describe in detail the experiments we conducted and the data that we obtained from it. We present empirical runtime distributions for the QR and the Toda algorithms as well as empirical deflation position distributions for these algorithms. We observe that initial data from the Wigner universality class shows similar runtime and deflation position behavior for both algorithms. Other types of initial data can, however, lead to runtime and deflation position distributions that have very different properties. We provide a that demonstrates that on Wigner–type initial data the QR algorithm leads to exponential right tails and the Toda algorithm leads to Gaussian right tails.

In addition we present an algorithm which reliably deflates Hermite–1 matrices in the middle. We could thus use it recursively in a divide and conquer type eigenvalue algorithm.

7.1 Description of the Experiments and the Dataset

In this chapter we summarize our findings on two important statistics associated with the behavior of certain eigenvalue algorithms. In the introductory chapter we already described these quantities:

- The runtime T_ϵ: For a given initial matrix, and given tolerance ϵ, how many iterations (equivalently, how long in terms of the time parameter of the associated ODE) does it take until the matrix iterate deflates at the prescribed tolerance?

- The deflation index I_ϵ: For given initial matrix and tolerance ϵ, at which

subdiagonal index does the first deflation occur?

Standard results from perturbation theory [55] imply that for symmetric matrices A, every eigenvalue $\lambda(\epsilon)$ of a perturbed matrix $A + \epsilon M$ lies in at least one of the discs described by:

$$|\lambda_i - \lambda(\epsilon)| < \epsilon \|M\|_2 \tag{7.1}$$

When deflating Jacobi matrices, the corresponding perturbation matrix has the structure:

$$M = \begin{pmatrix} 0 & E_{1k}^T \\ E_{1k} & 0 \end{pmatrix} \tag{7.2}$$

where $E_{1k} = \mathbb{R}^{(n-k)\times k}$ has all zero entries except for a one in the upper right corner. Clearly, $\|M\|_2 = 1$ in this case. For the deflation of full symmetric matrices, the perturbation matrix has the structure:

$$M = \begin{pmatrix} 0 & M_{21}^T \\ M_{21} & 0 \end{pmatrix} \tag{7.3}$$

where again $M \in \mathbb{R}^{(n-k)\times k}$, but now all entries of M satisfy $|m_{ij}| \leq 1$. We can estimate the norm of M as follows: Let $f \in \mathbb{R}^n$ and denote $f = (f_1, f_2)^T \in \mathbb{R}^k \times \mathbb{R}^{n-k}$. Then

$$Mf = \sum_{j=1}^{k} \langle b^{(j)}, f_2 \rangle e_j + \sum_{j=1}^{k} \langle e_j, f_1 \rangle b^{(j)} \tag{7.4}$$

where $b^{(j)}$ denotes the j–th column of B and $e_j \in \mathbb{R}^n$ is a vector with all zeros except a one as the j–th coefficient. Now estimate the norm of Mf:

$$\|Mf\|^2 = \sum_{j=1}^{k} \langle b^{(j)}, f_2 \rangle^2 + \left\| \sum_{j=1}^{k} \langle e_j, f_1 \rangle b^{(j)} \right\|^2 \tag{7.5}$$

For the first sum we have:

$$\sum_{j=1}^{k} \langle b^{(j)}, f_2 \rangle^2 \leq \epsilon^2 k(n-k)\|f_2\|^2 \tag{7.6}$$

for the second sum we obtain:

$$\begin{aligned}
\left\| \sum_{j=1}^{k} \langle e_j, f_1 \rangle b^{(j)} \right\|^2 &= \sum_{i=1}^{n-k} \left(\sum_{j=1}^{k} f_{1j} b_i^{(j)} \right)^2 = \\
&= \sum_{i=1}^{n-k} \left(\sum_{j=1}^{k} (f_{1j} b_i^{(j)})^2 + 2 \sum_{1 \leq l < m \leq k} f_{1l} b_i^{(l)} f_{1m} b_i^{(m)} \right) \leq \\
&\leq \epsilon^2 (n-k)\|f_1\|^2 + \sum_{i=1}^{n-k} \sum_{1 \leq l < m \leq k} \left[f_{1l}^2 (b_i^{(l)})^2 + f_{1m}^2 (b_i^{(m)})^2 \right] \leq \\
&\leq \epsilon^2 (n-k)\|f_1\|^2 + \epsilon^2 (n-k)(k-1)\|f_1\|^2 = \epsilon^2 k(n-k)\|f_1\|^2
\end{aligned} \tag{7.7}$$

so that overall $\|Mf\|^2 \leq \epsilon^2 k(n-k)\|f\|$ and hence $\|M\| \leq \epsilon\sqrt{k(n-k)}$. According to this reasoning we define the criterion for deflation separately for Jacobi and full matrices. In words, deflation occurs for Jacobi matrices when the first subdiagonal entry becomes less than the prescribed tolerance, whereas for full matrices deflation occurs when for the first time the maximum entry of a complete subdiagonal block, weighted by the square root of the size of this block, becomes smaller than the prescribed tolerance. To be precise, let

$$\hat{\epsilon}_k = b_k \tag{7.8}$$

in the case of Jacobi matrices and

$$\hat{\epsilon}_k = \sqrt{k(n-k)} \max_{\substack{k<i\leq n \\ 1\leq j\leq k}} |m_{ij}| \tag{7.9}$$

Then we define the runtime by

$$T_\epsilon = \inf \{t \,|\exists 1 \leq k < n : \hat{\epsilon}_k(L(t)) < \epsilon\} \tag{7.10}$$

as well as the deflation position by:

$$I_\epsilon = \arg \min_{1\leq i<n} \hat{\epsilon}_k(L(T_\epsilon)) \tag{7.11}$$

Stripping away the distributed computing issues mentioned in section 6.2.4, the algorithm 10 describes succinctly the experiments that were conducted. In words, for each matrix size in `dimList` N samples are drawn from the initial ensemble `MatDist`. We then iterate using the algorithm under consideration (expressed here in the `Step` procedure) until we undercut each of the tolerances in `tolList`. Whenever we have first undercut a certain tolerance, we save the time and the index it happened at in a text file. The procedure `SizeOffDiagonalBlocks` refers to the way that the maximum entry in the off diagonal entries is computed. The procedure is different for full symmetric matrices than for Jacobi matrices (for a description, refer to section 6.2.1). Summarizing, for each (ϵ_m, n_l)–pair we generate N samples. We use, however, one initial matrix sample of size n_i to generate n deflation time and deflation index samples (ϵ_r, n_i), where $1 \leq r \leq n$. For an overview of the numer of samples that we include in our study, refer to tables 7.1 – 7.4. It is apparent from the above paragraph that for collecting the sample statistics for the Toda and QR algorithm we need to specify a `Step` procedure in both cases. In the case of the QR algorithm there clearly is a natural choice, namely the algorithm 3 from section

Table 7.1: Summary of the number of samples based on which the empirical results for the QR algorithm are based.

	Initial Distribution			
Matrix Size	Hermite–1	GOE	iid Gaussian	iid Bernoulli
10	5000	5000	5000	5000
30	5000	5000	5000	5000
50	5000	5000	5000	5000
70	5000	5000	5000	5000
90	5000	5000	5000	5000
110	5000	5000	5000	5000
130	5000	5000	5000	5000
150	5000	5000	5000	5000
170	5000	5000	5000	5000
190	5000	5000	5000	5000

Table 7.2: Summary of the number of samples based on which the empirical results for the Toda algorithm are based.

	Initial Distribution			
Matrix Size	Hermite–1	GOE	iid Gaussian	iid Bernoulli
10	5000	4500	4500	3500
30	5000	3500	4000	5000
50	5000	2000	2000	4000
70	5000	3500	3500	5000
90	5000	3500	3500	5000
110	5000	5000	1500	5000
130	5000	4500	5000	5000
150	5000	2500	2500	5000
170	5000	2500	2500	2500
190	5000	5000	1500	1500

Table 7.3: Summary of the number of samples on which the empirical results for the QR algorithm are based.

	Initial Distribution	
Matrix Size	Uniform Jacobi	Jacobi Unif. Eig.
10	5000	5000
30	5000	5000
50	5000	5000
70	5000	5000
90	5000	5000
110	5000	5000
130	5000	5000
150	5000	5000
170	5000	5000
190	5000	5000

Table 7.4: Summary of the number of samples based on which the empirical results for the Toda algorithm are based.

	Initial Distribution	
Matrix Size	Uniform Jacobi	Jacobi Unif. Eig.
10	5000	4000
30	5000	7000
50	5000	10000
70	4000	10000
90	2500	10000
110	5000	10000
130	5000	10000
150	4500	6500
170	3000	1500
190	1500	

Algorithm 10 `DataGeneration` – samples the required quantities

Require: `tolList` $= [\epsilon_1 > \cdots > \epsilon_n]$, `dimList` $= [n_1, \ldots, n_K]$, Initial Distribution `MatDist`, number of samples N, Algorithm `Step`, step size h.

1: **for** $i = 1, \ldots, K$ **do**
2: **for** $j = 1, \ldots, N$ **do**
3: $t \leftarrow 0$, $r \leftarrow 1$
4: $M_t \sim$ `MatDist`(n_i)
5: **repeat**
6: $\mathbb{R}^{n_i - 1} \ni v_t \leftarrow$ `SizeOffDiagonalBlocks`(M_t)
7: $\hat{v}_t \leftarrow \min_{1 \le l \le n_i - 1} v_t(l)$
8: $\hat{l}_t := \arg\min_{1 \le l \le n_i - 1} v_t(l)$
9: **if** $\hat{v}_t < \epsilon_r$ **then**
10: Save to file: $T_{\epsilon_r} = t$ and $I_{\epsilon_r} = \hat{l}_t$
11: $r \leftarrow r + 1$
12: **end if**
13: $\tilde{M} \leftarrow$ `Step`(M_t)
14: $t \leftarrow t + h$
15: $M_t \leftarrow \tilde{M}$
16: **until** $r > n$
17: **end for**
18: **end for**

2.3 in the case of Jacobi matrices or the standard, unoptimized version for the case of full symmetric matrices from equation (2.15) in the same section. These are the implementations referred to as 'recursive' in table 6.1.

Choosing the implementation strategy for the Toda algorithm is not as straightforward, because the numerical properties of the alternatives aren't equally well studied. We see from the table 6.1 that there are three implementation paradigms that play a role for the Toda algorithm in the case of Jacobi matrices and two paradigms in the case of full symmetric matrices. Our strategy to validate the choice of an implementation paradigm is as follows:

- Choose an initial matrix
- Compute the Toda evolution of this initial matrix with each of the applicable implementation paradigms mentioned in table 6.1.

- The crucial quantity for the determination of our sample statistics T_ϵ and I_ϵ is the minimum entry of the off diagonal block sizes (called $\hat{v}_t$ in algorithm 10). We compare the evolution of relative differences

$$\frac{|\hat{v}_t^{(a)} - \hat{v}_t^{(b)}|}{|\hat{v}_t^{(a)}|} \tag{7.12}$$

(a and b are meant to index the implementation paradigms). This allows us to come up with 'trusted intervals' in which these relative errors are small.

We conducted simulation studies using this idea for many matrix dimensions using Wigner scaled initial data. A typical result of these simulation is found in figure (7.1): For matrices sampled from the Wigner–scaled initial distributions the difference between the recursive and the ODE implementation of the Toda algorithms stays very small for a large enough region to include all the runtime samples we obtain below. Since the recursive implementation tends to be faster than the ODE implementation, we will use it as the `Step` procedure in algorithm 10.

As a last remark in this section we would like to stress that the numerical experiments are reported with 'unscaled' initial data, in the sense that all initial matrices are sampled directly from the chosen initial ensembles and we apply no scaling (for example of Wigner type) to them. We have found, however, that the implementation paradigms for the Toda algorithm work much better with Wigner–scaled initial data and we believe this to be due to the fact that the difference between the minimal and the maximal eigenvalues in the ensembles we consider scales with $\sqrt{n}$, where n is the dimension of the matrix. In order to understand this note that, in contrast to the `TodaODE` implementation paradigm, both `TodaDiscrete` and `TodaExplicit` rely on the computation of matrix exponentials of the initial matrix:

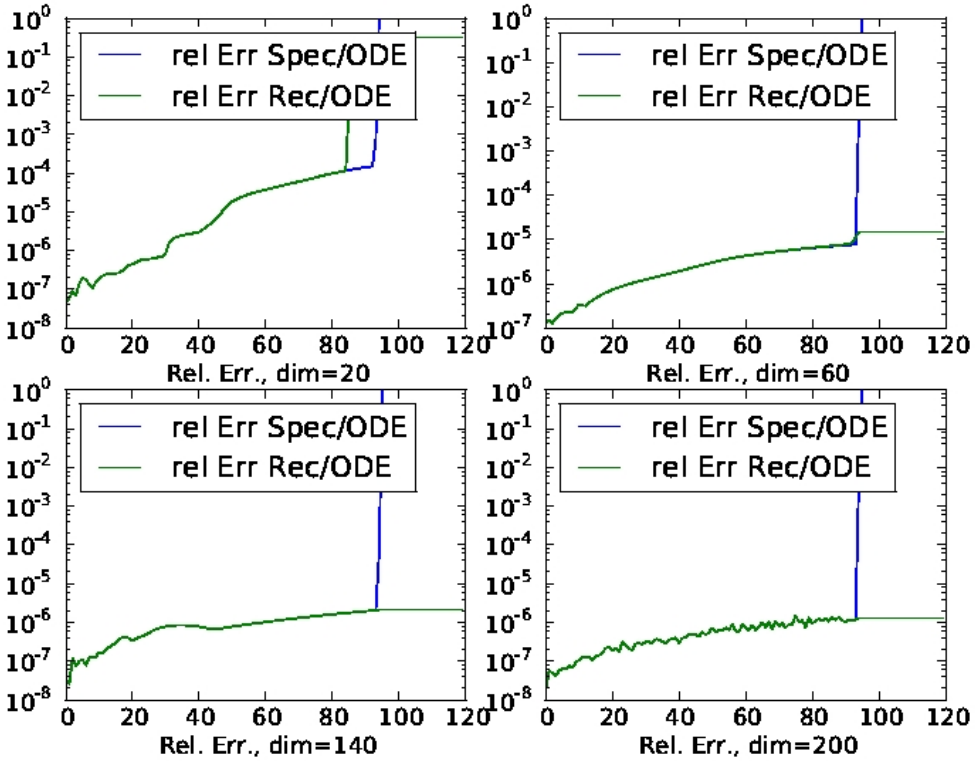

Figure 7.1: Typical result of comparing the relative difference between the Toda trajectory computed using three different algorithms for different matrix sizes. The x–axis corresponds to time, the y–axis presents the relative error in a log scale. The recursive implementation of the Toda algorithm's 'trusted region' is large enough to include all the runtime samples we obtain below.

1. `TodaDiscrete` starts with the initial matrix and then iterates the computations of a matrix exponential, a QR step and a matrix logarithm

2. `TodaExplicit` (for Jacobi matrices) computes $q(t) = \exp\{\Lambda t\}q_0/\|\exp\{\Lambda t\}q_0\|$, the first row of the eigenvector matrix and then reconstructs the whole eigenvector matrix from this row.

Due to the $\sqrt{n}$–scaling in both cases we are dealing with rather 'stiff' expressions for higher dimensions. In the `TodaDiscrete` implementation the QR step leads to a sequence of matrices with (theoretically) constant eigenvalues, where the ratio of

the largest to the smallest eigenvalue is of the order:

$$\frac{\lambda_{\max}}{\lambda_{\min}} \approx \frac{e^{2\sqrt{n}}}{e^{-2\sqrt{n}}} = e^{4\sqrt{n}} \tag{7.13}$$

In practice, the QR step introduces numerical error and subsequent matrices of the QR algorithm may not actually have the exact same eigenvalues. This is problematic in particular if the numerical error leads to replacing a very small, but positive, eigenvalue with a negative eigenvalue. Now taking the matrix logarithm leads to complex eigenvalues and the scheme is not very useful anymore for computing the size of the off diagonal terms.

In the case of the `TodaExplicit` paradigm the unscaled initial data will lead to very fast convergence to zero of some of the entries of q. For example, the ratio of the entries q_1 to q_n where q_1 is the first entry of the eigenvector belonging to λ_1 and q_n is the first entry of the eigenvector belonging to λ_n is of the order

$$\frac{q_1(t)}{q_n(t)} \approx C \cdot \frac{e^{-2\sqrt{n}t}}{e^{2\sqrt{n}}} = e^{-4\sqrt{n}t} \tag{7.14}$$

This will introduce numerical error in the spectral reconstruction procedure (cf. algorithm 4) and thus again cause problems for the accurate computation of the off diagonal terms.

Summarizing the dilemma: We aim to present the numerical results for unscaled initial data, but our fast implementation strategies (`TodaDiscrete` and `TodaExplicit`) both lead to inaccurate results in this regime due to the tendency for large differences between the maximal and the minimal eigenvalues of the initial matrix. For scaled initial data these large differences will tend not to arise, because of the Wigner semicircle law which implies $\lambda_{\min} \to -2$ and $\lambda_{\max} \to 2$, ie. the differences are bounded. Our remedy for this situation is thus to compute the evolution of (Wigner) scaled initial matrices and use equation (5.20) and the discussion surrounding it to derive

the runtime of an unscaled matrix: Clearly equation (5.20) implies $\tilde{L}(\tilde{t}) = L(ct)/c$. It thus follows:

$$T_\epsilon(L_0) = \frac{1}{c} T_{\epsilon/c}(L_0/c) \tag{7.15}$$

Setting $c := \sqrt{n}$ this is exactly the equation necessary to determine the runtime distribution for unscaled initial data from the Wigner scaled initial data.

7.2 Empirical Results on the Runtime Behavior

In this section we discuss some properties of the empirical runtime distributions for different initial value distributions and two different eigenvalue algorithms. In the following sections we present and describe several different types of figures for each algorithm:

- Type 1: For a given i, we present the empirical hit time histograms $\hat{T}_i(10^{-8}, n)$ where n ranges through all its allowed values. The rationale for choosing this smallest tolerance value is that we expect that disturbances arising from the minimal offdiagonal value of the initial matrix are mitigated.

- Type 2: For a given initial distribution and a given dimension, we present the runtime distribution corresponding to varying values of ϵ. Clearly choosing smaller ϵ should cause the algorithm to run 'longer' in some sense.

- Type 3: For a given i we normalize all the empiricial hit time histograms $\hat{T}_i(\epsilon, n)$ where normalization means shifting and scaling the empirical histograms so that they have mean zero and variance one. These normalized histograms are then plotted together in one figure.

- Type 4 and 5: The next two types plot empirical means and standard deviations of the histograms for all value of ϵ and n in the range of our simulations. In these figures we always combine the results for Wigner–type initial data in one figure and for non Wigner–type initial data in another figure.

- Type 6: When appropriate we pick a 'base' initial distribution (GOE) and choose an (n, ϵ)–regime for which we normalize the runtime data (ie. center to mean zero and variance one) for all the initial distributions. We then produce a QQ (quantile–quantile) plot of the normalized runtimes for 'base' distribution initial data against another initial distribution. If the resulting plot does not deviate far from the identity line we conclude that the empirical distributions are close to one another.

- Type 7: In the last type of figure we combine the runtime histograms for several initial distributions. We center and scale the histogram to have mean zero and variance one and then extract an (x, y) pair from each histogram bucket by setting x equal to the midpoint of the bucket and y equal to the height of the bucket. We will combine all the runtime distributions from Wigner–type initial data as well as the runtime distributions from a base initial distributions of Wigner–type (eg GOE) with those of non Wigner–type initial data.

7.2.1 Runtime Behavior for the QR algorithm

Here we present figures associated with properties of T_ϵ, the time to first deflation for the QR algorithm. The initial distributions of Wigner–type we are using are the Hermite–1 distribution, the GOE, the Wigner and the Matrix Bernoulli–distribution. We will compare these results to initial distributions not of Wigner type, namely uniform doubly stochastic Jacobi matrix ensemble and the Jacobi en-

semble with uniform eigenvalues. In all of the full symmetric Wigner–type matrices the distributions of the upper triangular entries are iid and chosen so that their mean equals zero and variance equals one. The tolerance ϵ was set successively to equal $\epsilon = 10^{-k}$, where $k = 2, 4, 6, 8$ and $n = 10, 30, \cdots, 190$. Subsequently we sampled empirical hit time distributions $\hat{T}_i(\epsilon, n)$ where the index $1 \leq i \leq 6$ represents the initial distribution, n represents matrix size in addition to the deflation tolerance ϵ. The figures of type 1 for this case are 7.2 to 7.7. For all the initial distributions, both Wigner and non Wigner–type, the behavior is very similar: The shapes, scales and locations of the histograms for all the matrix dimensions are roughly the same with a mode around $t = 20$ and apparent exponential decay in the right tail. The roughness of the histograms presented in 7.7 can be explained by the relatively small number of samples that was used to generate this figure. Fixing an initial matrix dimension and comparing the runtime distributions for several tolerances, we obtain the figures 7.8 to 7.13 (type 2). In all of these figures we can see that for smaller values of ϵ the empirical runtime distributions 'diffuse' to the right. Note that in all cases, especially in 7.12 the histogram for the largest deflation tolerance $\epsilon = .01$ shows a high concentration of near 'immediate' deflations. The fact that this effect subsides for smaller ϵ is – at least for Hermite–1 initial data – related to the result in section 3.2. Even more interestingly, the shape of the histograms (unimodal, fast decay on the left tail, seemingly exponential decay on the right tail) for fixed initial data appears to be largely independent of the tolerance in addition to size of the matrices that were studied. This claim is substantiated by our figures of type three (7.14–7.19). Direct comparison of these figures also leads to the observation that the shape of the renormalized histograms is rather robust with respect to the underlying initial distribution.

The figures 7.20 to 7.23 combine information about the means and standard deviations of the QR runtime distributions for all Wigner–type (7.20, 7.22) and non

Table 7.5: Regression Parameters for the Means of the QR runtime distributions

Initial Distribution	a_0	a_1	a_2	P–value a_1
Hermite–1	.802338	-.004554	-1.042907	$< 2 \cdot 10^{-16}$
Full Sym. Wigner type	1.96824	0.0004690	-1.0263649	0.0095
Uniform doubly stoch. Jacobi	0.7648330	-0.0072921	-1.0916354	$4.16 \cdot 10^{-8}$
Jacobi Uniform eig.	1.844126	-0.003467	-1.276037	0.0256

Wigner–type (7.21, 7.23) initial data which were removed in the figures of type three. As expected, we can see that in both cases these statistics are barely influenced by matrix size, but they seem to increase linearly in the negative logarithm of the deflation tolerance ϵ. Note that the QR runtime means and variances for the full matrix Wigner–type initial are more or less identical, while for the Hermite–1 initial data runtimes both statistics have a slightly larger value. To support the notion of small dependence on n and linear dependence on $\log \epsilon$ we fit a simple linear regression of the form

$$\mu_{n,\epsilon} \approx a_0 + a_1 n + a_2 \log \epsilon \tag{7.16}$$

$$\sigma_{n,\epsilon} \approx b_0 + b_1 n + b_2 \log \epsilon \tag{7.17}$$

The dots in figures 7.20 to 7.23 correspond to the predicted values of means and variances based on this regression and it is apparent that these values are predicted rather well by our model (R^2 near one). Refer to table 7.5 for the regression parameters of equation (7.16) and to table 7.6 for the parameters of equation (7.17). All the computations were done using the generalized linear model functionality from [46]. Since for the full matrix Wigner data – at least visually – the means and variances do not depend on the dimension of the matrix ensemble, we have included the p–values for the t–test of the hypothesis that the coefficient corresponding to the dimension is zero. for a summary of the regression parameters.

Table 7.6: Regression Parameters for the Standard deviations of the QR runtime distributions

Initial Distribution	b_0	b_1	b_2	P–value b_1
Hermite–1	0.442622	-.003329	-.617517	$8 \cdot 10^{-15}$
Full Sym. Wigner type	1.2799509	0.0005311	-0.5854859	0.0118
Uniform doubly stoch. Jacobi	1.066713	-0.007584	-0.658920	0.000353
Jacobi Uniform eig.	2.0044243	-0.0026034	-0.7961700	0.000185

To facilitate direct comparison of the rescaled runtime histograms (7.14–7.19) we present the QQ plots (type 6) of figures 7.24 to 7.28. Here, we pick specific (n, ϵ)–combinations and plot the centered data points from GOE data against the centered data points of all the other initial distributions. We get very good overlap for all Wigner–type initial distributions, whereas the two non Wigner–type initial distributions seem to lead to somewhat heavier tails than the GOE runtime distribution.

Lastly we consider the combined histograms plots of type 7, namely figures 7.29 to 7.31 Recall that these figures were generated by combining every single normalized empirical histogram we sampled, ie. all initial distributions all matrix dimensions and deflation tolerances. Each bucket of each histogram was used to generate an (x, y)–tupel, where the x–value corresponds to the midpoint of the bucket boundaries and the y–value corresponds to the bucket weight. In these figures the y–axis is set to a log scale. Consider first the combined histograms of the runtime samples for all the Wigner–type initial data (figure 7.29): Given the wide range of tolerances, matrix dimensions and initial conditions we see a good overlap of all the normalized histograms for the empirical runtime distributions Wigner–type initial data. Note also, that a gamma distribution with parameters $k = 2$ and $\theta = 1$ was centered and scaled to obtain mean zero and variance one and plotted into the same figure. We can see that the decay of the normalized histograms fits very well with the decay of the rescaled gamma distribution. The figure 7.30 combines the normalized histograms of the GOE runtimes with the normalized runtimes generated using uniform doubly

stochastic Jacobi matrices. Just as in figure 7.31 (normalized GOE runtimes against normalized Jacobi uniform eigenvalues runtimes) the overlap is worse than when we were comparing exclusively Wigner–type initial data.

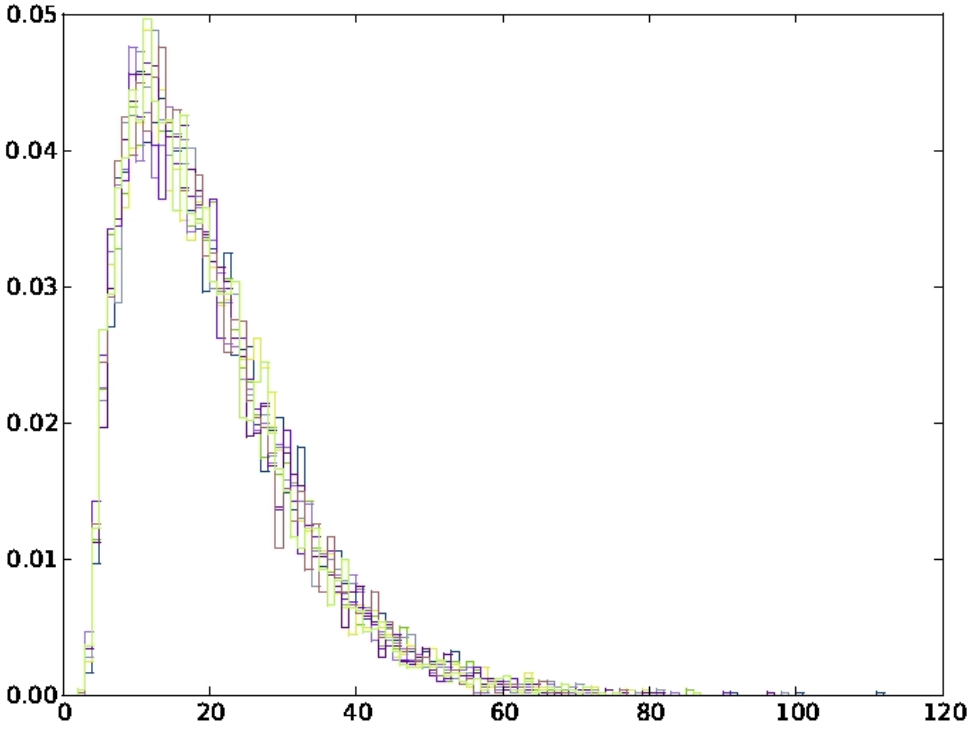

Figure 7.2: Empirical hit time distributions of QR algorithm for Hermite–1 initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 1).

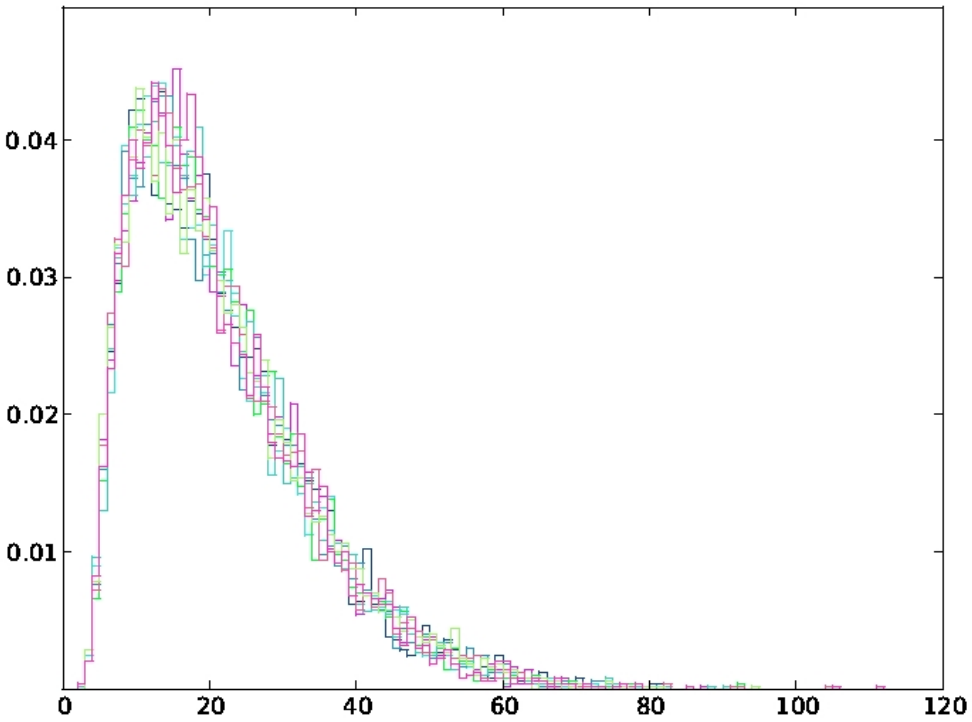

Figure 7.3: Empirical hit time distributions of QR algorithm for GOE initial data with $\epsilon = 10^{-8}$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 1).

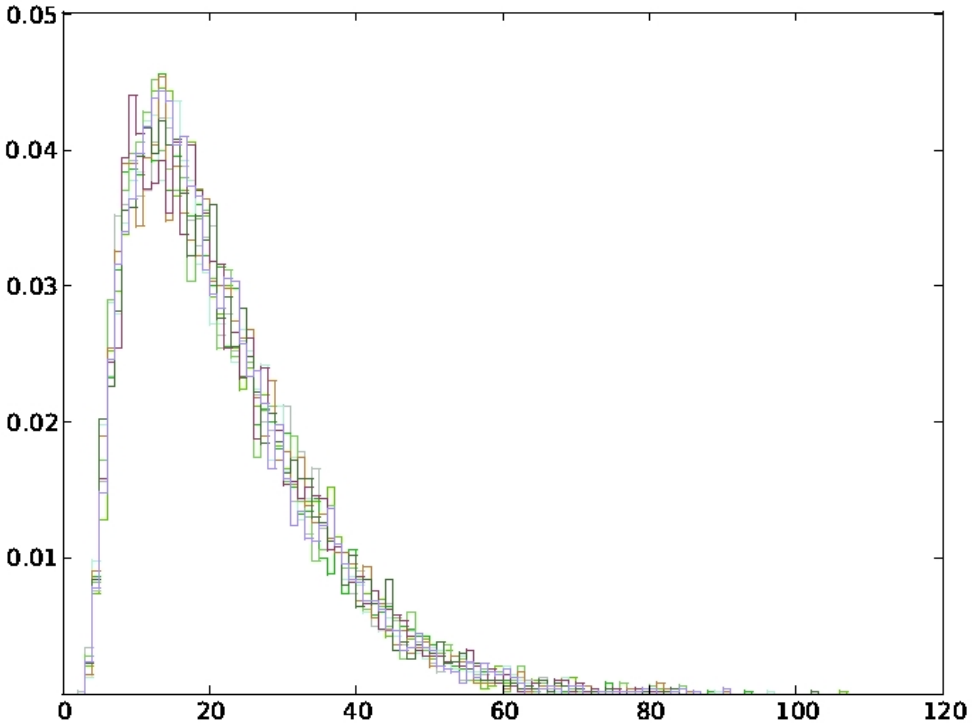

Figure 7.4: Empirical hit time distributions of QR algorithm for Wigner initial data with $\epsilon = 10^{-8}$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 1).

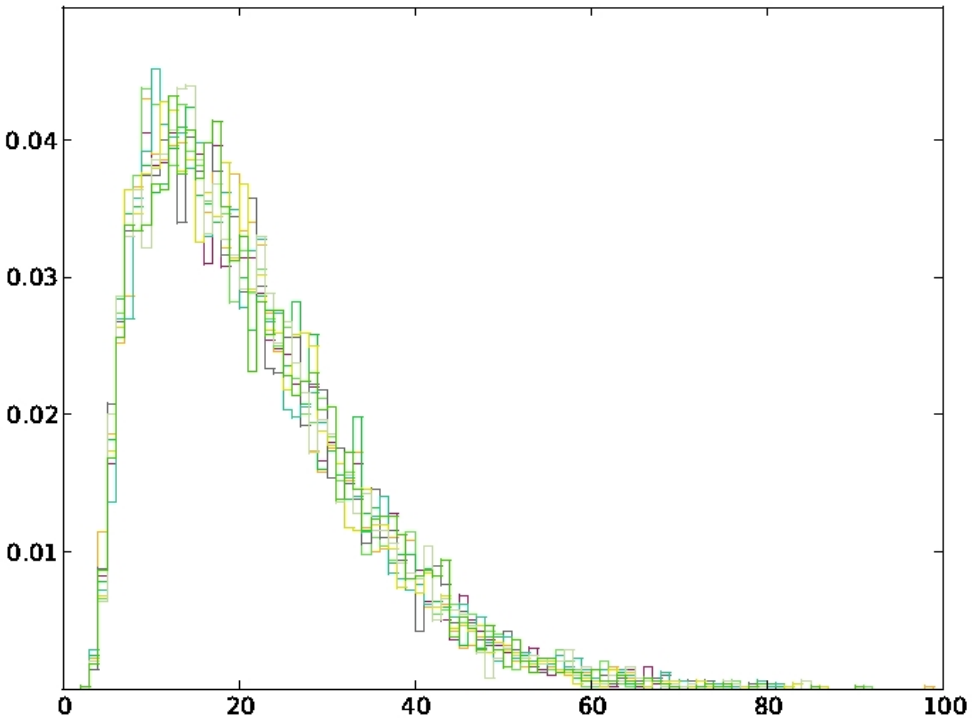

Figure 7.5: Empirical hit time distributions of QR algorithm for Bernoulli matrix initial data with $\epsilon = 10^{-8}$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 1).

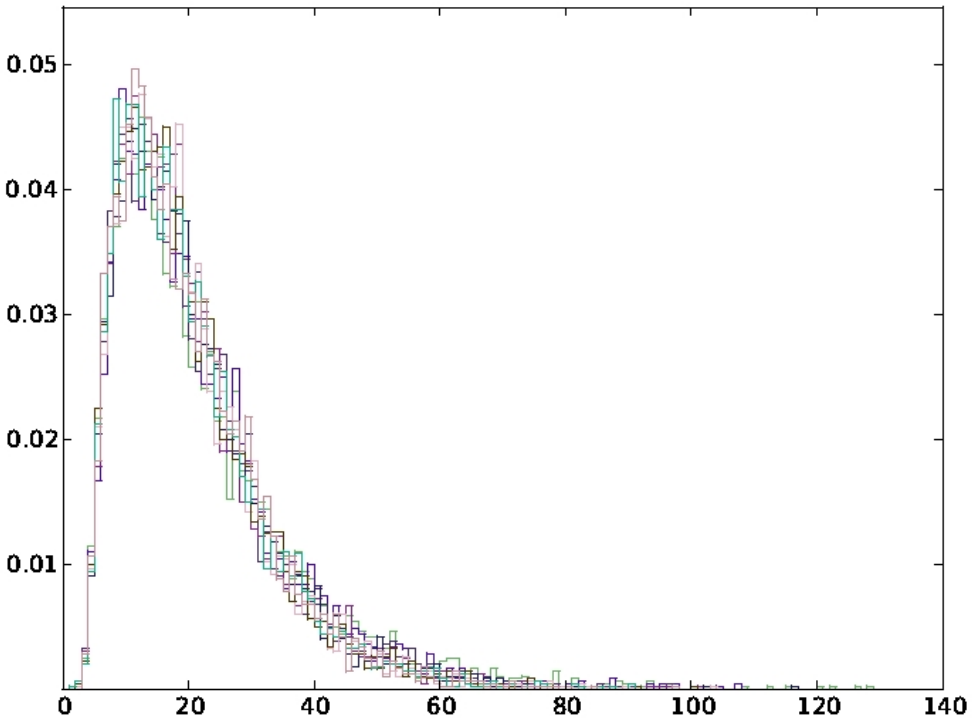

Figure 7.6: Empirical hit time distributions of QR algorithm for uniform doubly stochastic Jacobi matrix initial data with $\epsilon = 10^{-8}$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 1).

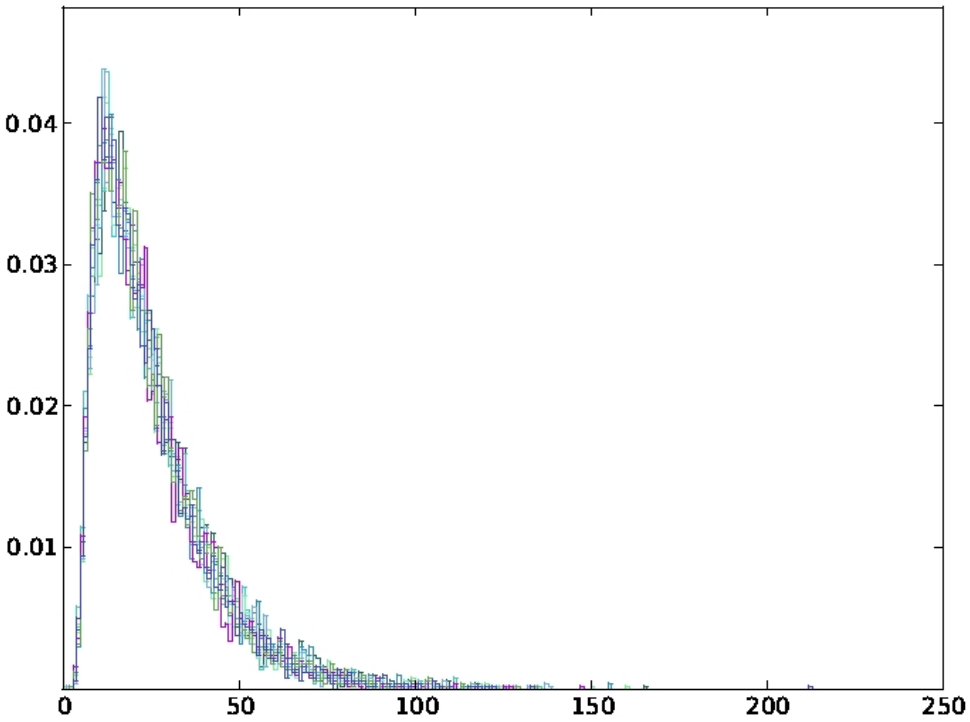

Figure 7.7: Empirical hit time distributions of QR algorithm for Jacobi uniform eigenvalue initial data with $\epsilon = 10^{-8}$ and matrix dimension ranging through $10, 30, \ldots, 130$ (type 1).

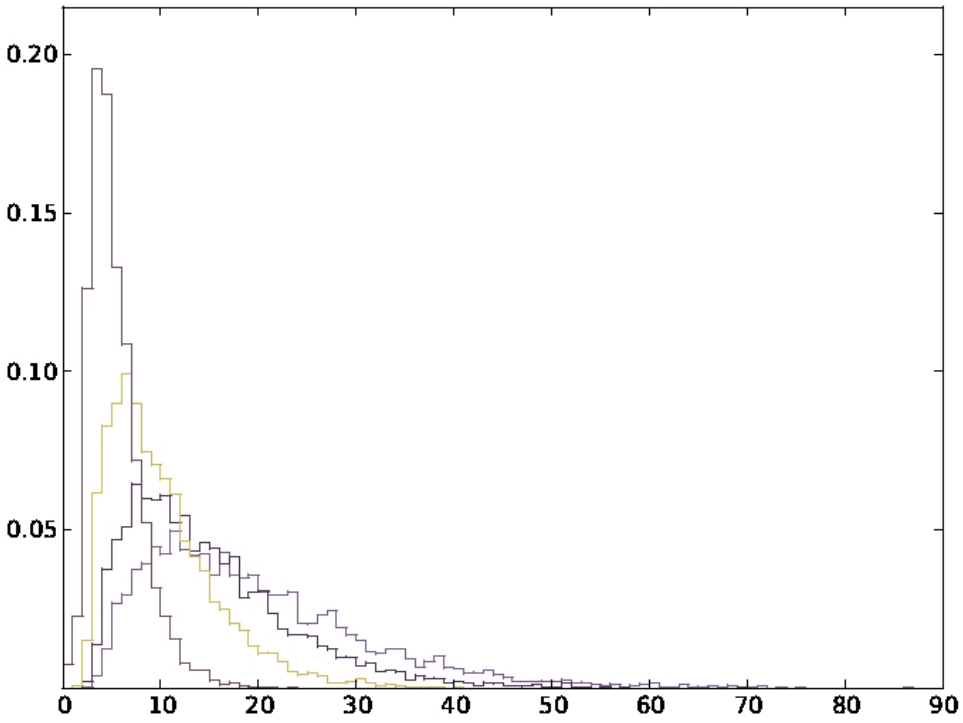

Figure 7.8: Empirical hit time distributions of QR algorithm for Hermite–1 initial data with $n = 190$ and deflation tolerance ranging through $\epsilon = 10^{-k}$ where $k = 2, 4, 6, 8$ (type 2). Smaller ϵ corresponds to heavier tail.

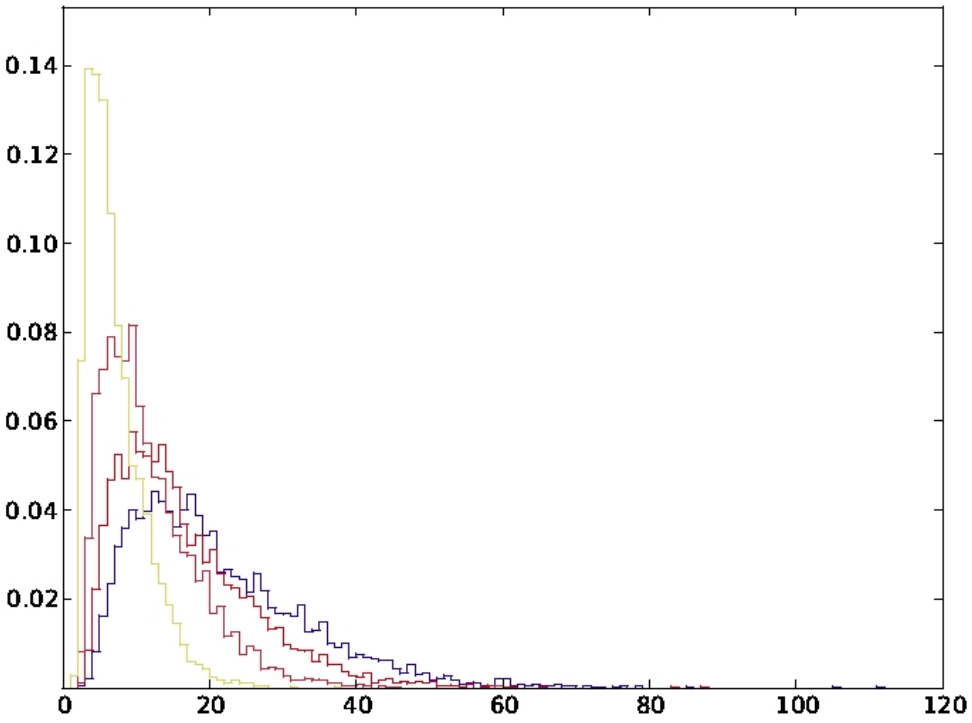

Figure 7.9: Empirical hit time distributions of QR algorithm for GOE initial data with $n = 190$ and deflation tolerance ranging through $\epsilon = 10^{-k}$ where $k = 2, 4, 6, 8$ (type 2). Smaller ϵ corresponds to heavier tail.

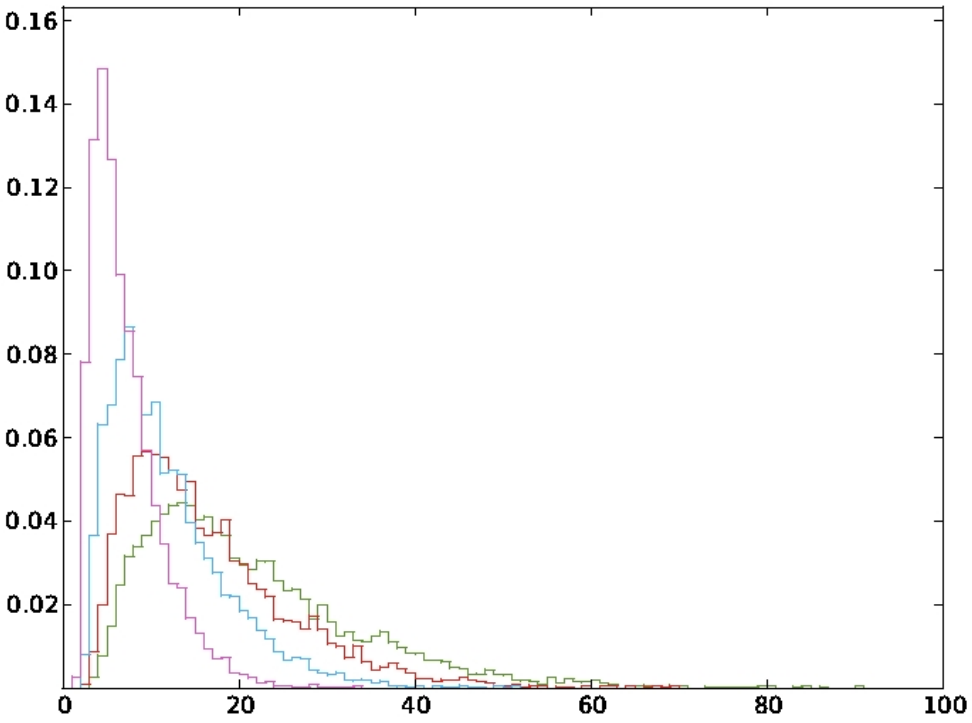

Figure 7.10: Empirical hit time distributions of QR algorithm for Wigner initial data with $n = 190$ and deflation tolerance ranging through $\epsilon = 10^{-k}$ where $k = 2, 4, 6, 8$ (type 2). Smaller ϵ corresponds to heavier tail.

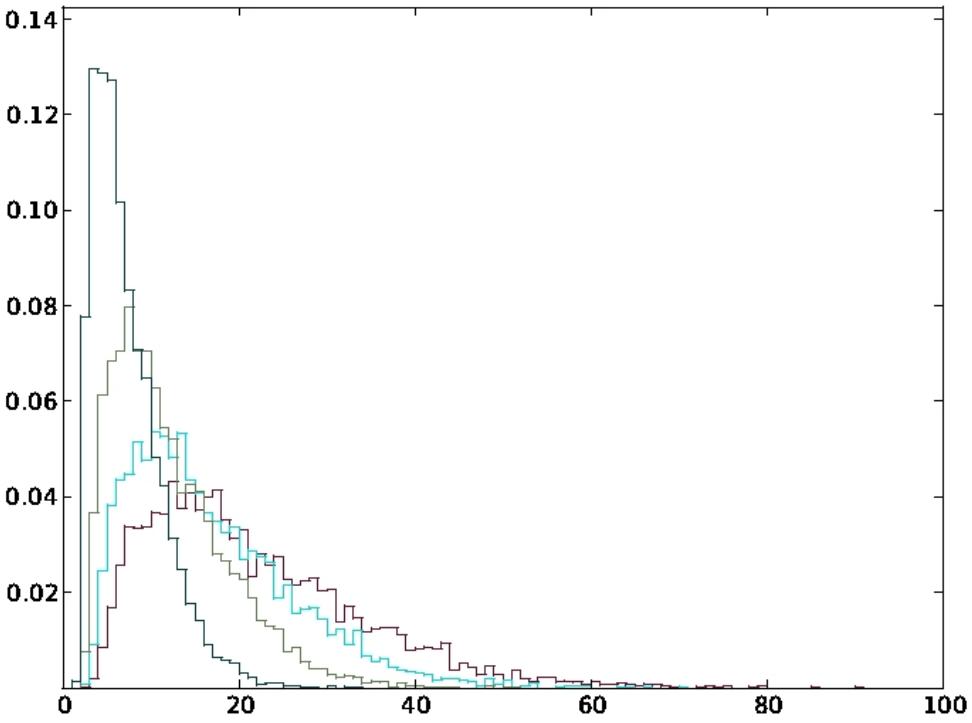

Figure 7.11: Empirical hit time distributions of QR algorithm for Bernoulli matrix initial data with $n = 190$ and deflation tolerance ranging through $\epsilon = 10^{-k}$ where $k = 2, 4, 6, 8$ (type 2). Smaller ϵ corresponds to heavier tail.

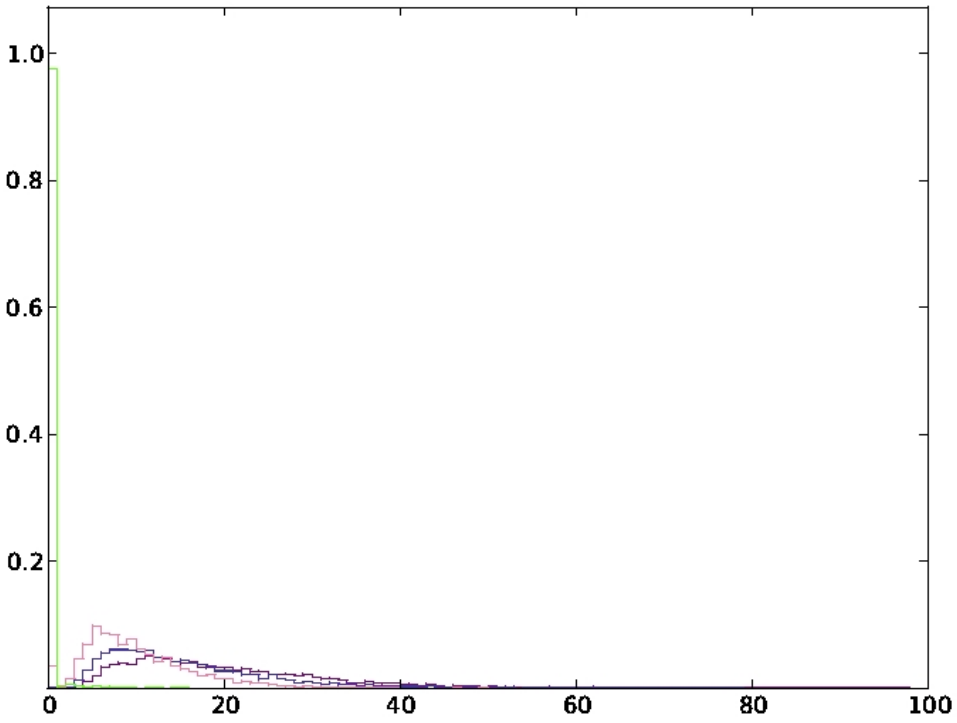

Figure 7.12: Empirical hit time distributions of QR algorithm for uniform doubly stochastic Jacobi matrix initial data with $n = 190$ and deflation tolerance ranging through $\epsilon = 10^{-k}$ where $k = 2, 4, 6, 8$ (type 2). Smaller ϵ corresponds to heavier tail.

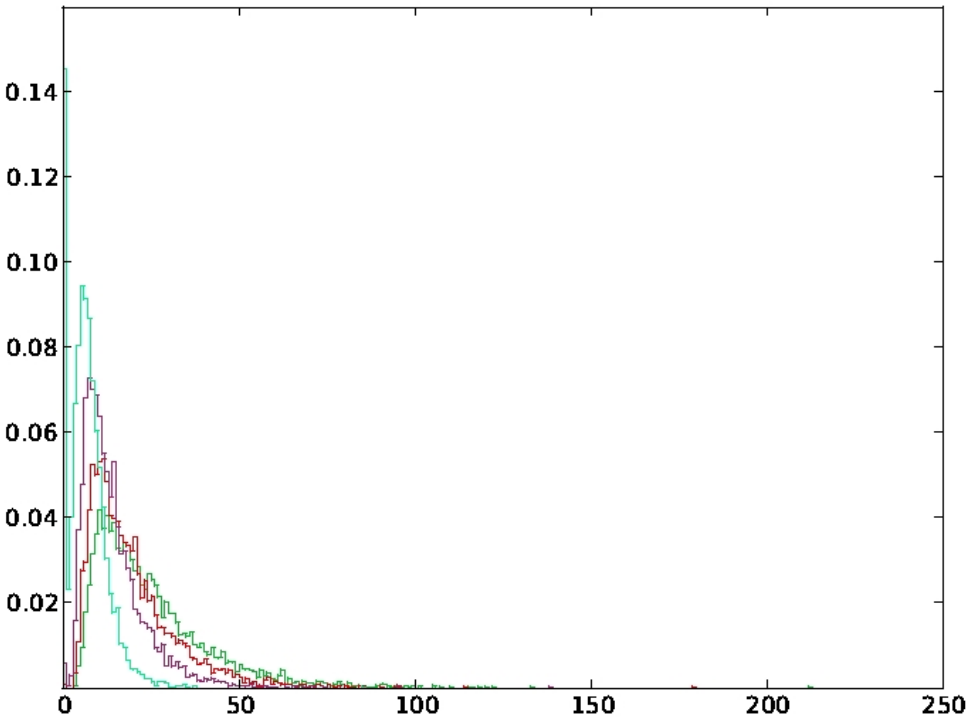

Figure 7.13: Empirical hit time distributions of QR algorithm for Jacobi uniform eigenvalue initial data with $n = 130$ and deflation tolerance ranging through $\epsilon = 10^{-k}$ where $k = 2, 4, 6, 8$ (type 2). Smaller ϵ corresponds to heavier tail.

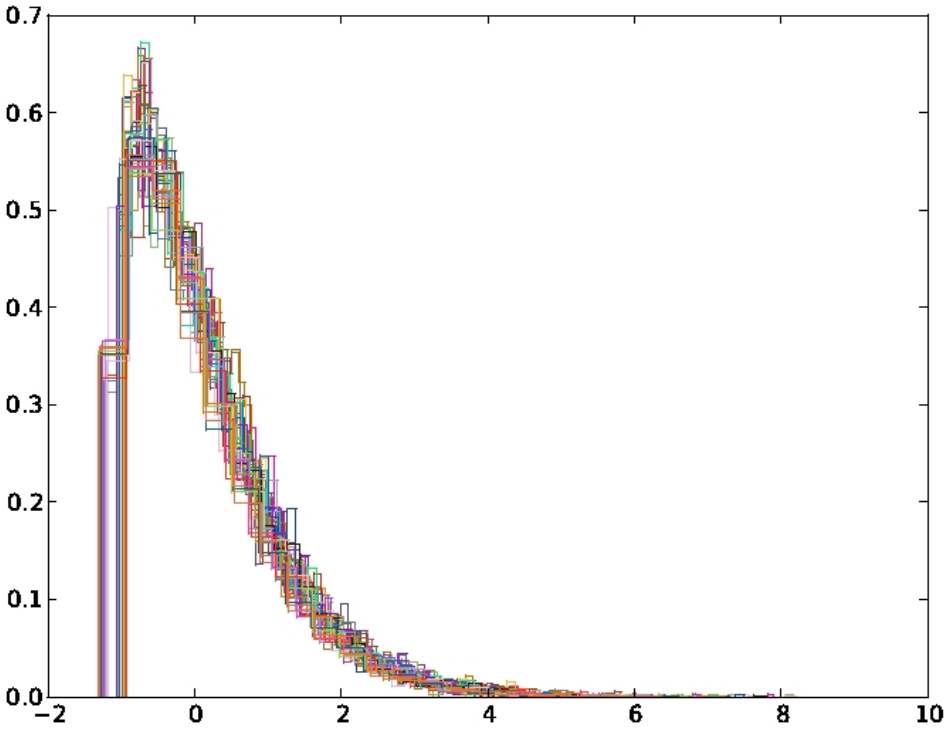

Figure 7.14: Normalized empirical hit time distributions of QR algorithm for Hermite–1 initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 3).

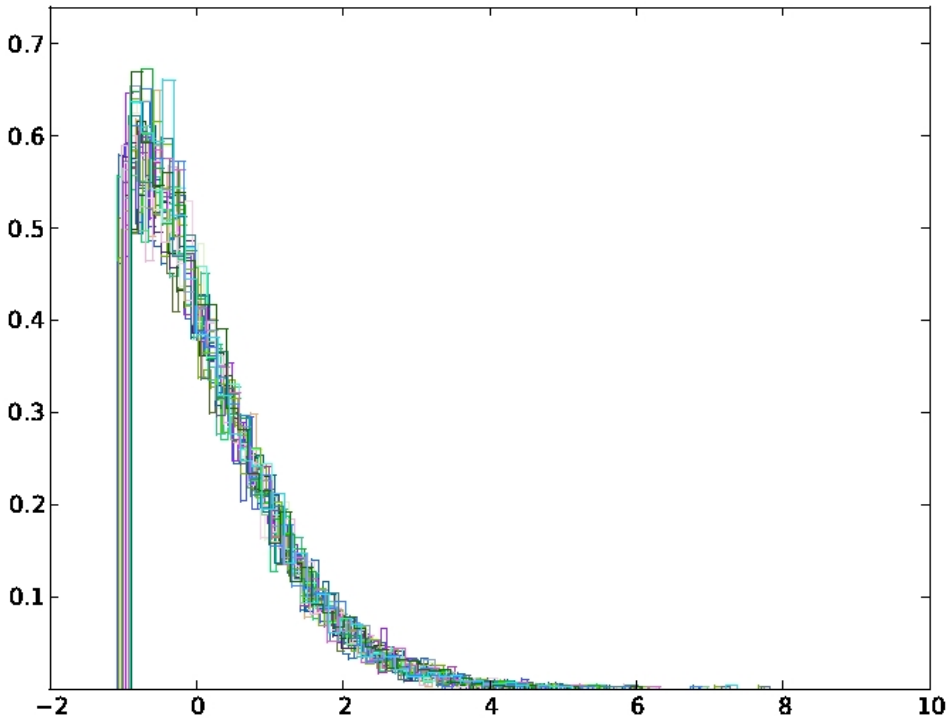

Figure 7.15: Normalized empirical hit time distributions of QR algorithm for GOE initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 3).

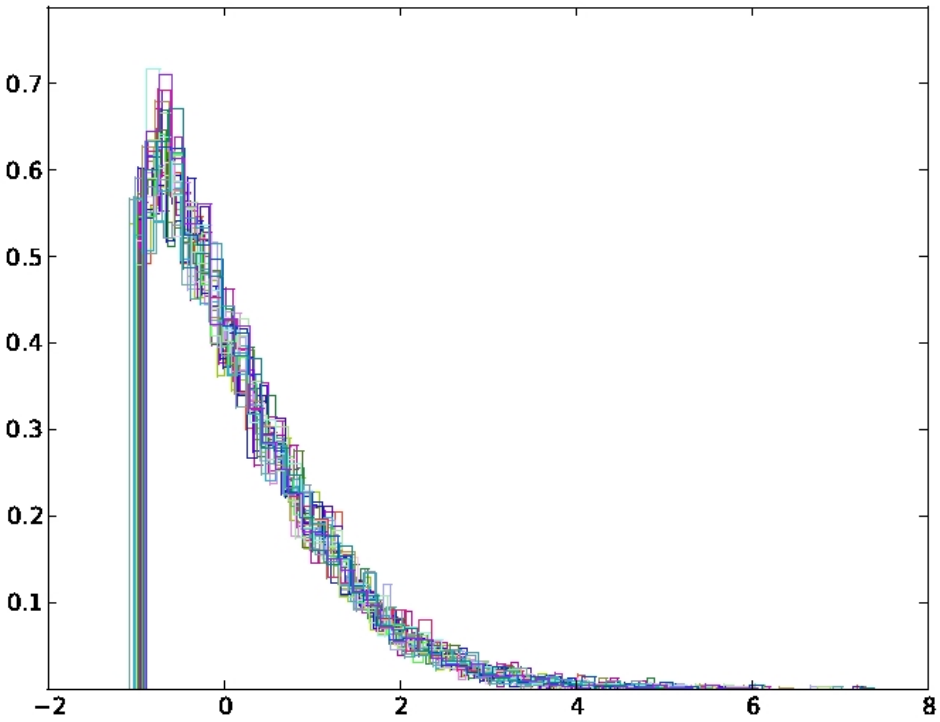

Figure 7.16: Normalized empirical hit time distributions of QR algorithm for Wigner initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 3).

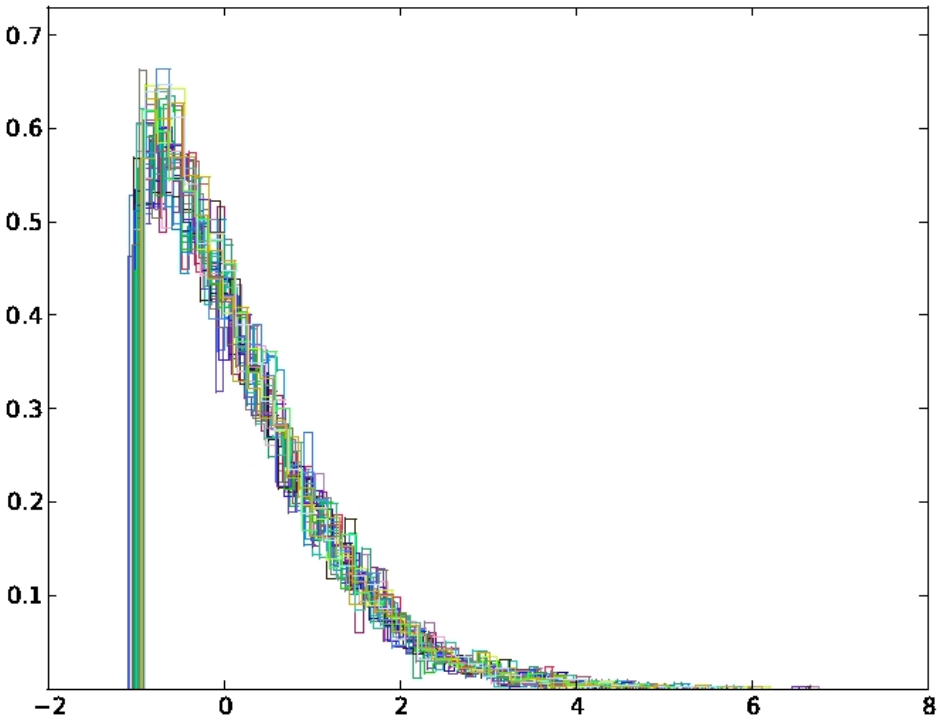

Figure 7.17: Normalized empirical hit time distributions of QR algorithm for Bernoulli initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 3).

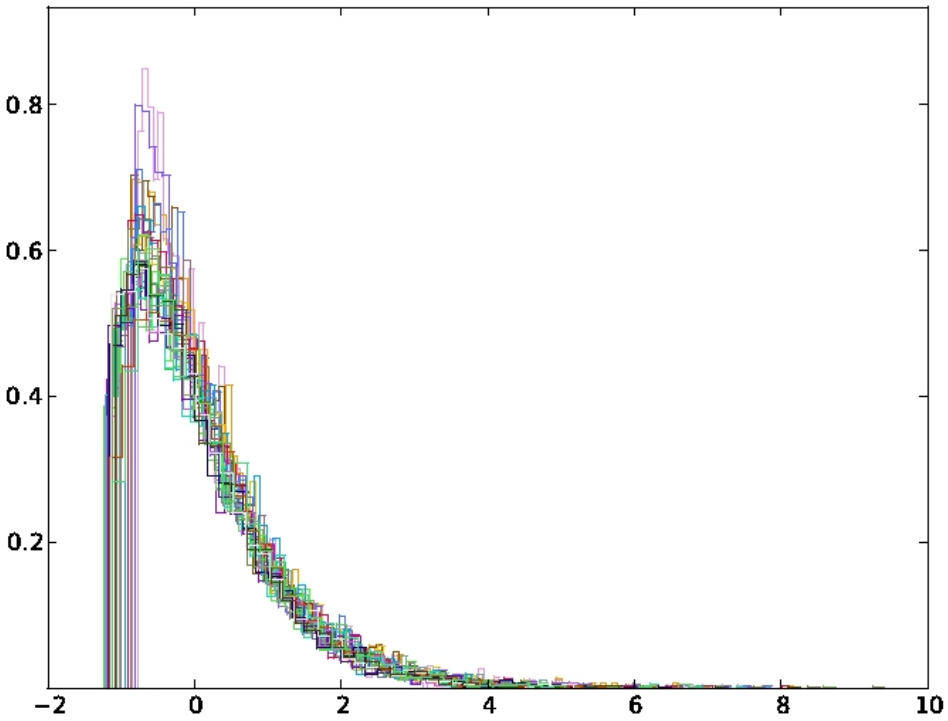

Figure 7.18: Normalized empirical hit time distributions of QR algorithm for uniform doubly stochastic Jacobi initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 3).

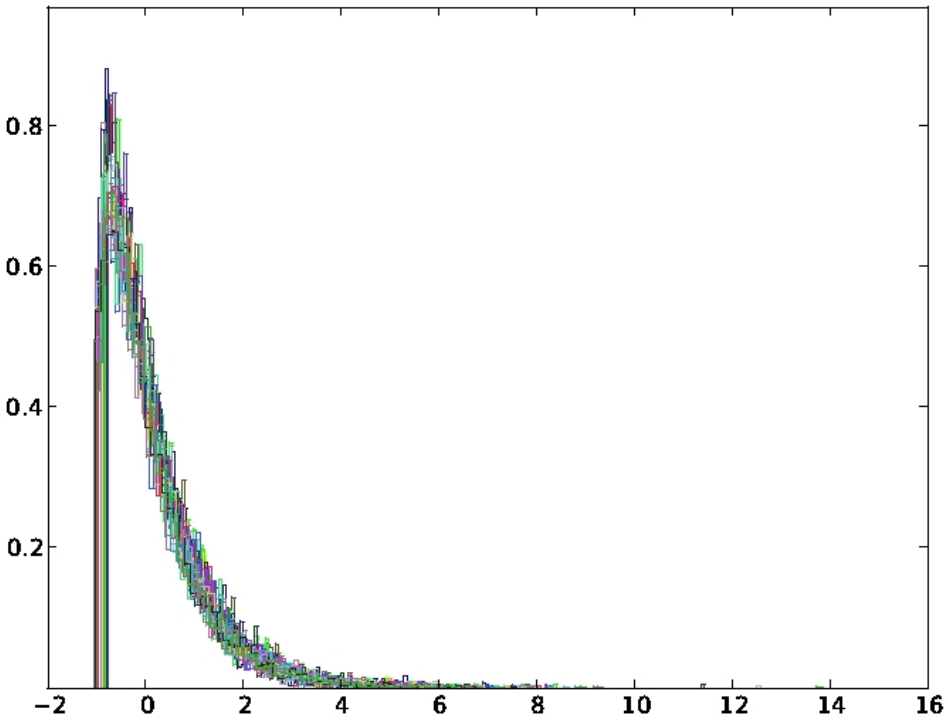

Figure 7.19: Normalized empirical hit time distributions of QR algorithm for Jacobi uniform eigenvalue initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 2).

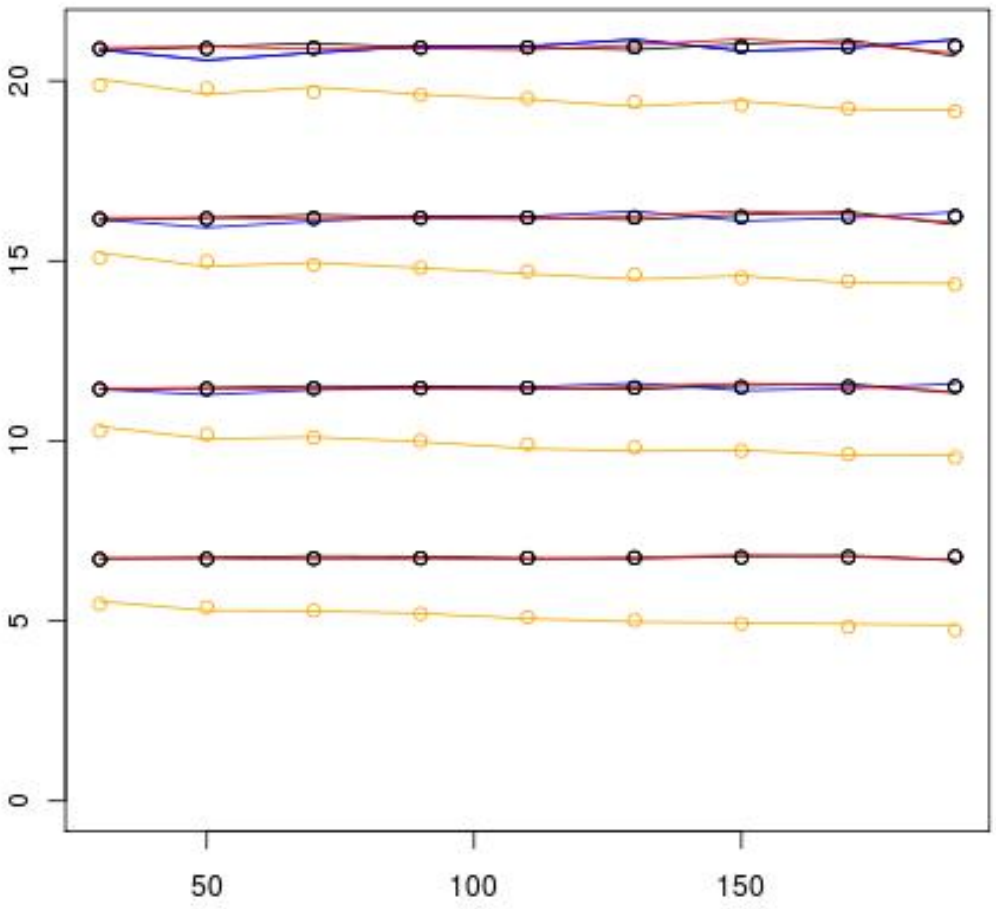

Figure 7.20: Means of the empirical hit time distributions of QR algorithm for all Wigner–type initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 4). Orange corresponds to Hermite–1 and red/ black/ blue to Wigner/ GOE/ Bernoulli. Dots are the regression predictions described in the text.

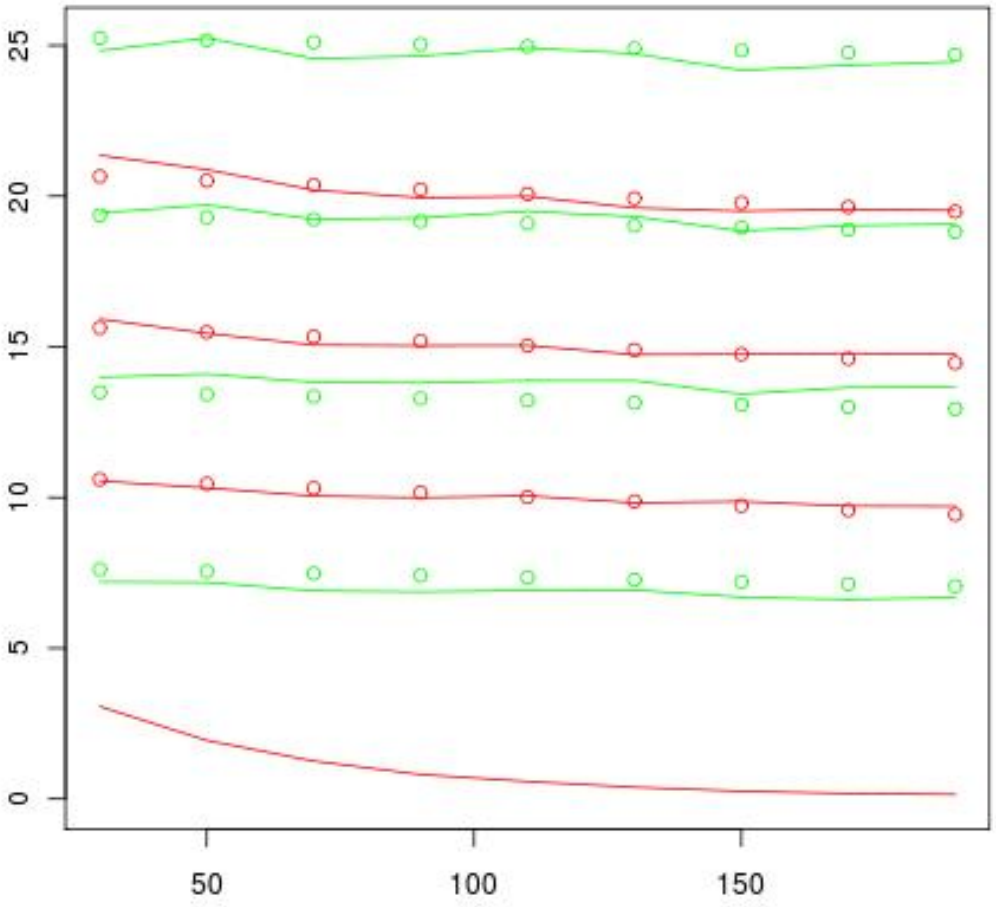

Figure 7.21: Means of the empirical hit time distributions of QR algorithm for non Wigner–type initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 4). Red corresponds to uniform doubly stochastic Jacobi and green to Jacobi with uniform eigenvalues. Dots correspond to regression predictions.

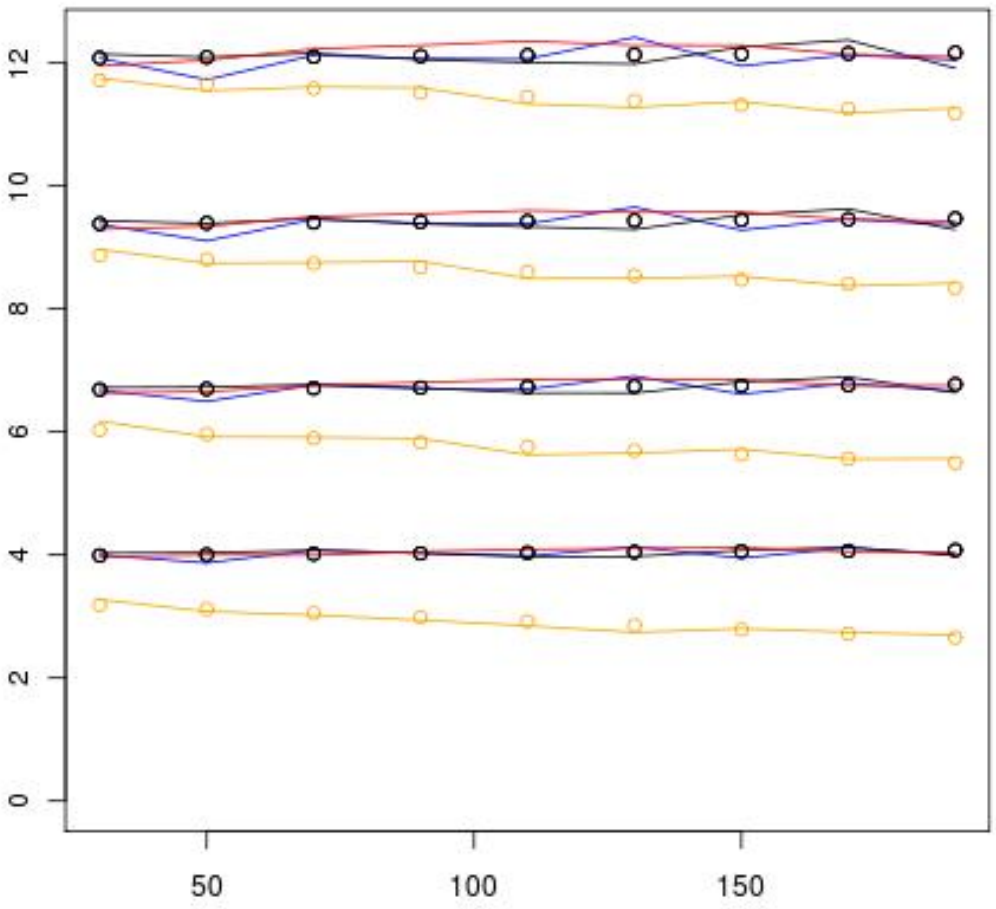

Figure 7.22: Standard deviations of the empirical hit time distributions of QR algorithm for all Wigner–type initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 5). Orange corresponds to Hermite–1 and red/ black/ blue to Wigner/ GOE/ Bernoulli. Dots are the regression predictions described in the text.

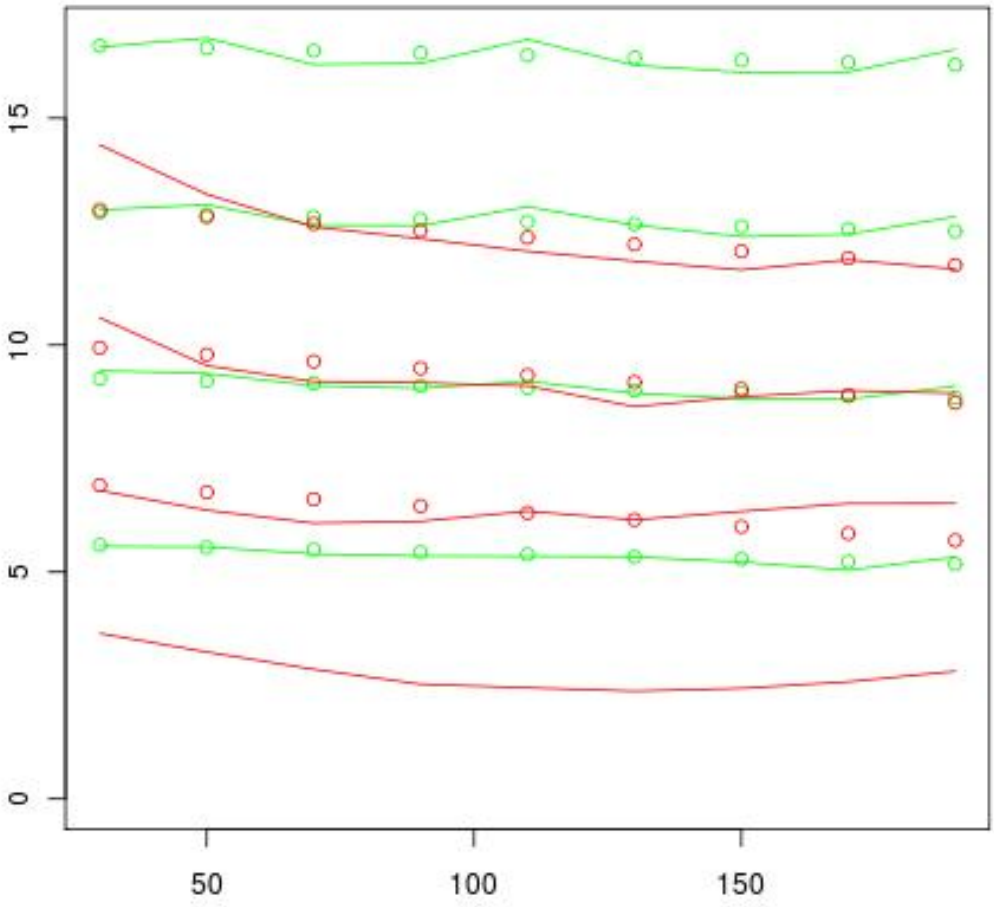

Figure 7.23: Standard deviations of the empirical hit time distributions of QR algorithm for non Wigner–type initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimension ranging through $10, 30, \ldots, 190$ (type 5). Red corresponds to uniform doubly stochastic Jacobi and green to Jacobi with uniform eigenvalues. Dots correspond to regression predictions.

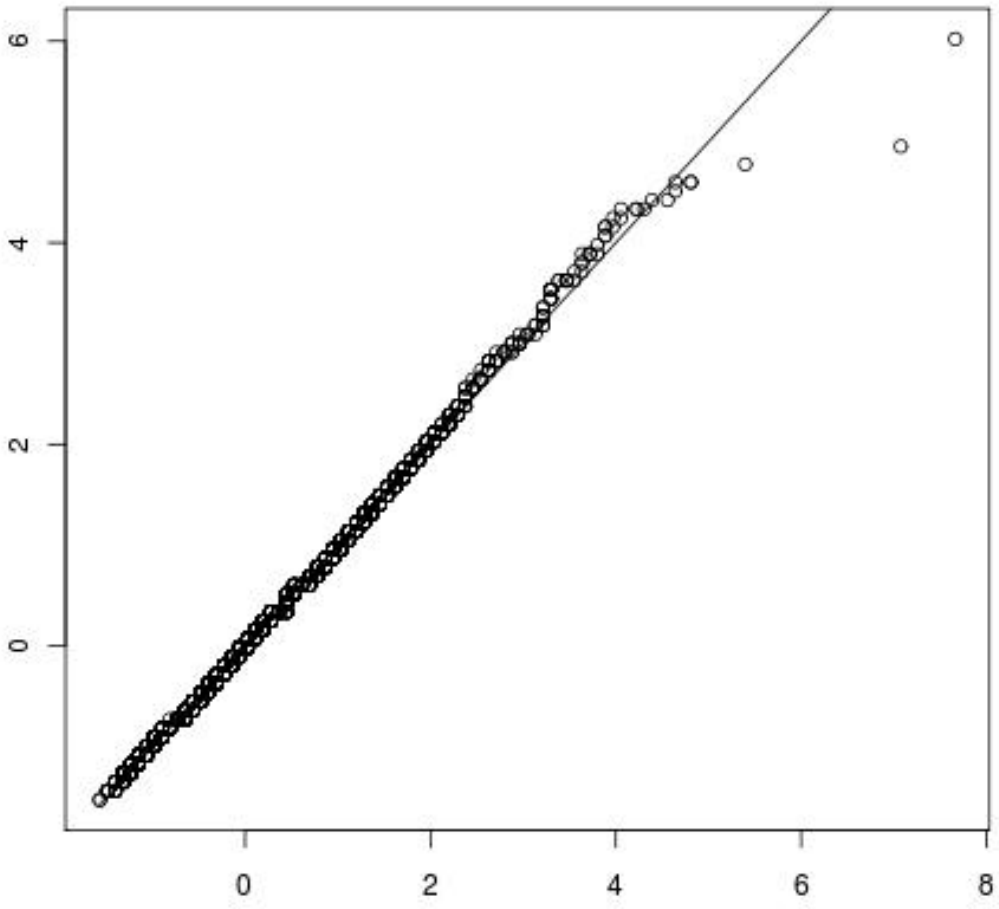

Figure 7.24: QQ plot of the normalized (mean zero, variance one) QR runtime samples of GOE initial data ($n = 190$, $\epsilon = 10^{-8}$) against the normalized runtime samples of Hermite–1 initial data (same parameters).

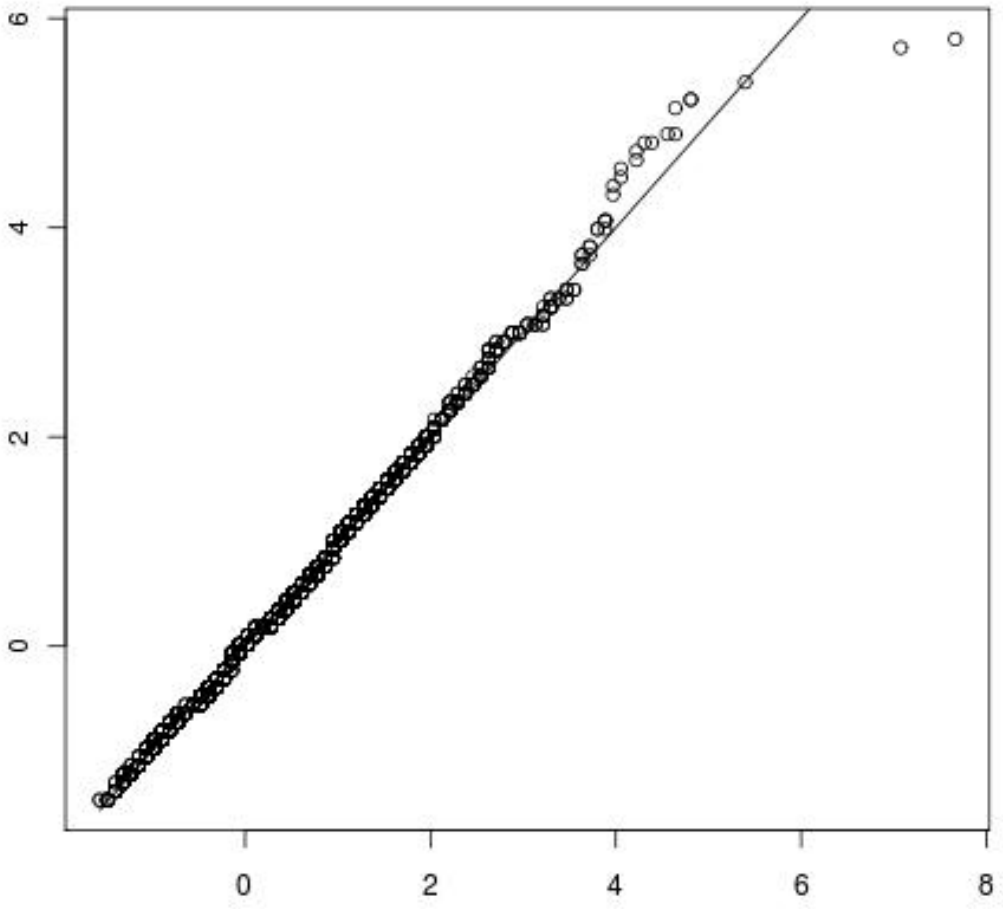

Figure 7.25: QQ plot of the normalized (mean zero, variance one) QR runtime samples of GOE initial data (where matrix size is $n = 190$, $\epsilon = 10^{-8}$) against the normalized runtime samples of Wigner initial data (same parameters).

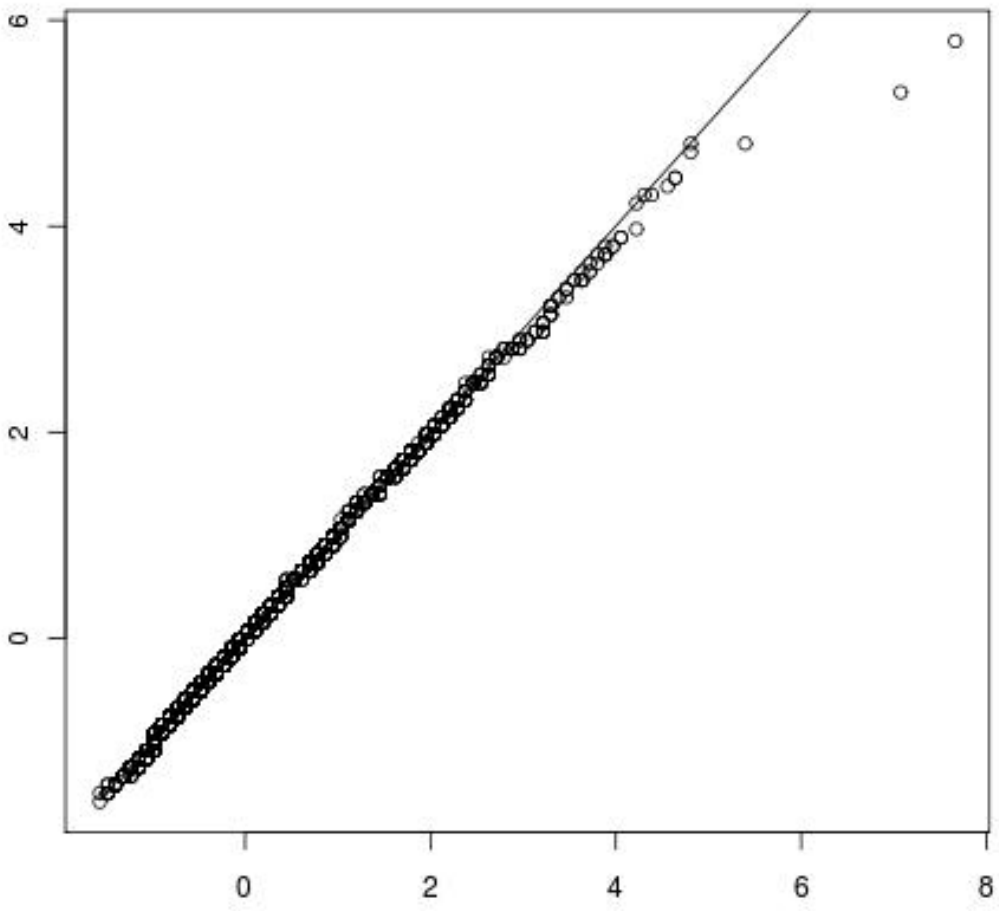

Figure 7.26: QQ plot of the normalized (mean zero, variance one) QR runtime sampes of GOE initial data ($n = 190$, $\epsilon = 10^{-8}$) against the normalized runtime samples of Bernoulli initial data (same parameters).

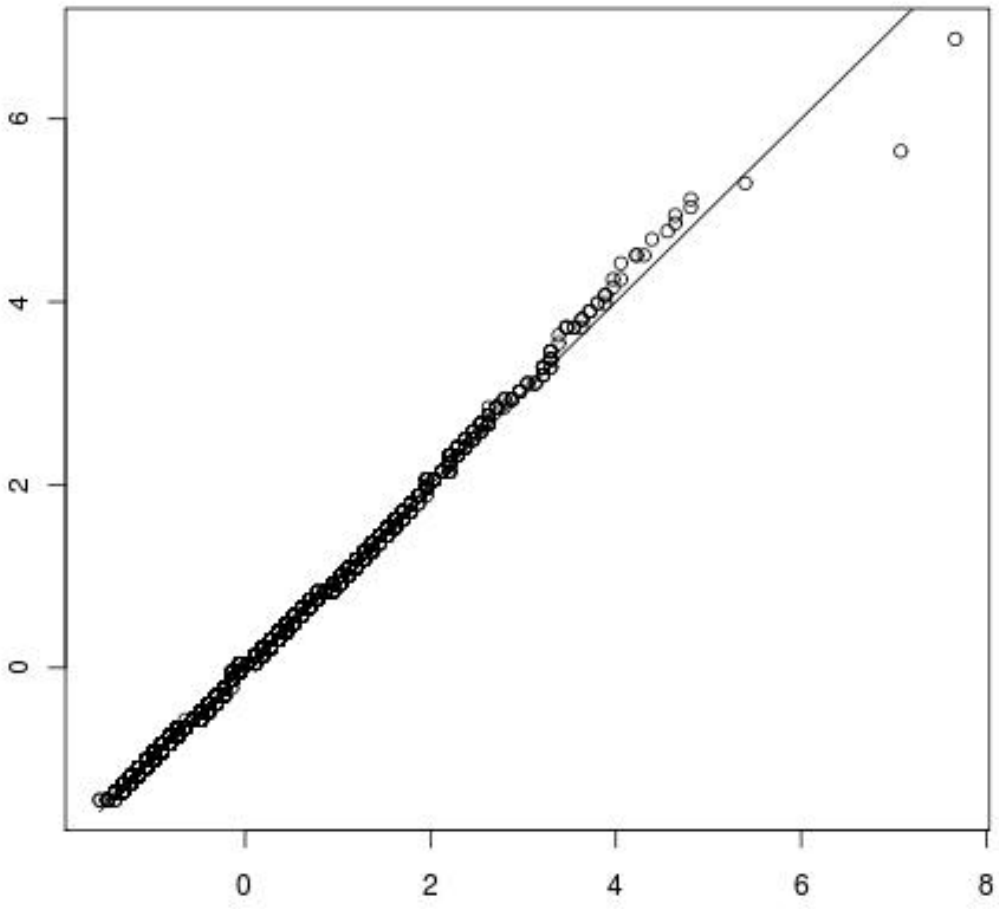

Figure 7.27: QQ plot of the normalized (mean zero, variance one) QR runtime sampes of GOE initial data ($n = 190$, $\epsilon = 10^{-8}$) against the normalized runtime samples of uniform doubly stochastic Jacobi initial data (same parameters).

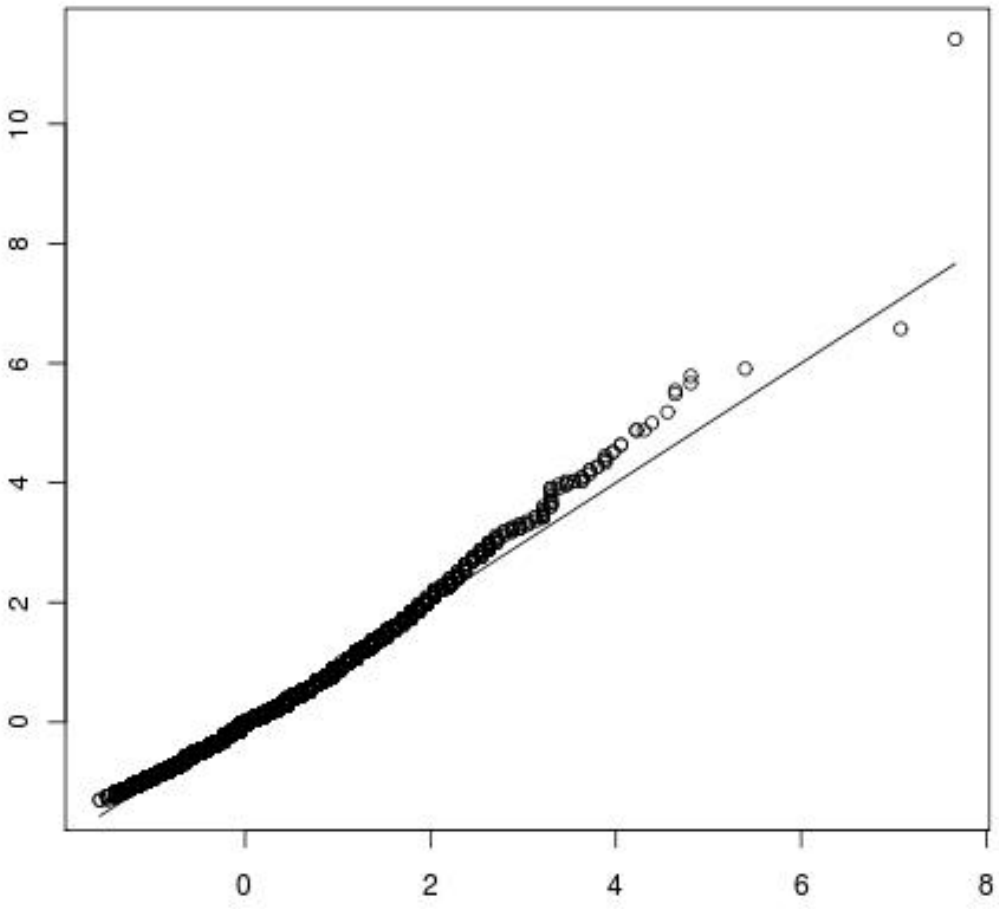

Figure 7.28: QQ plot of the normalized (mean zero, variance one) QR runtime sampes of GOE initial data ($n = 190$, $\epsilon = 10^{-8}$) against the normalized runtime samples of Jacobi with uniform eigenvalues initial data ($n = 130$ same ϵ).

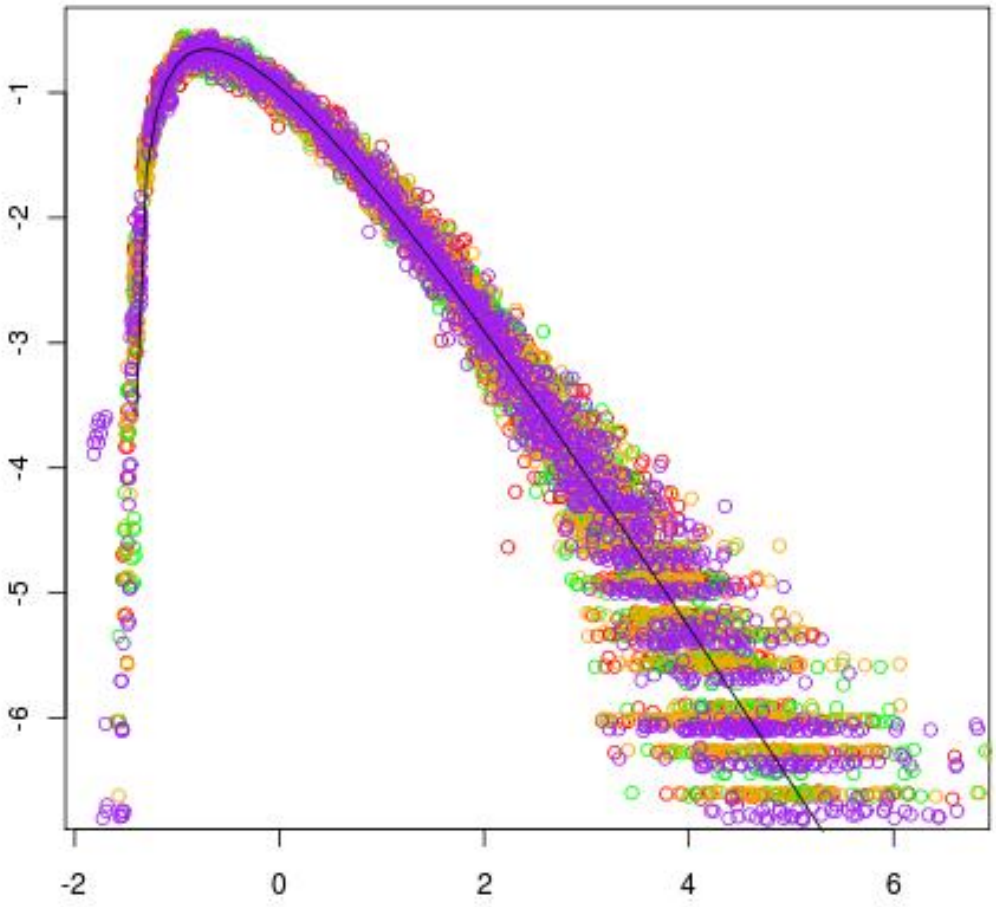

Figure 7.29: Combination of all the QR runtime histograms (type 7), compared with a 'normalized' gamma distribution with parameters $k = 2$ and $\theta = 1$ (red: Bernoulli, green: GOE, orange: Wigner, purple: Hermite–1). Here normalized refers to shifting and scaling to obtain a distribution of mean zero and variance one.

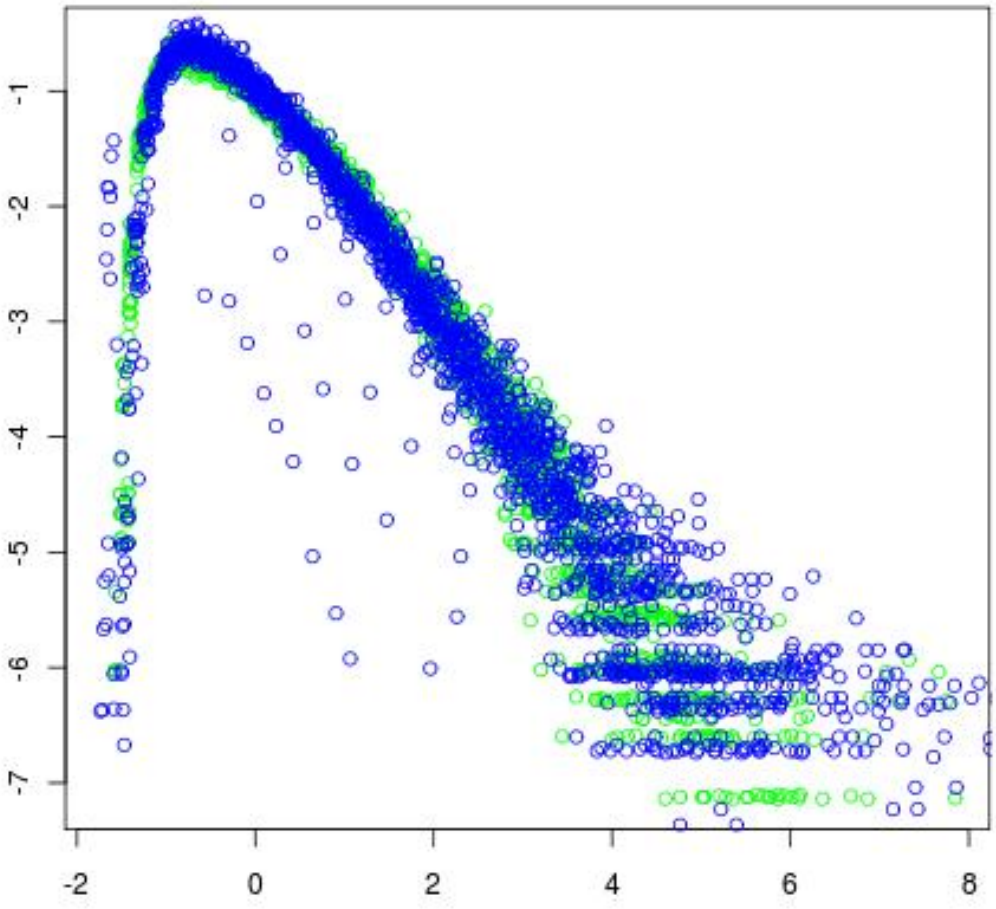

Figure 7.30: Combination of normalized QR runtime histograms (green: GOE initial data, blue: uniform doubly stochastic Jacobi initial data).

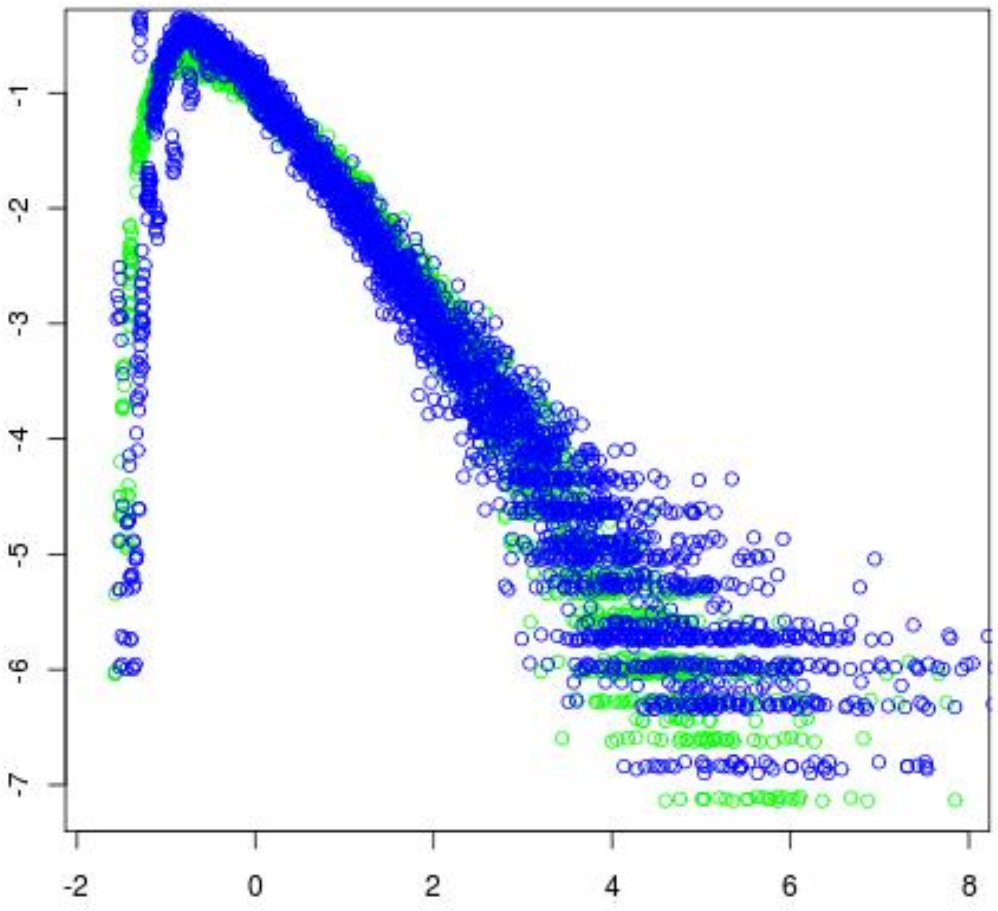

Figure 7.31: Combination of normalized QR runtime histograms (green: GOE initial data, blue: Jacobi with uniform eigenvalues initial data).

7.2.2 Runtime Behavior for the Toda algorithm

In this section we present the results of our simulations of the Toda algorithm runtime distributions in a similar way to our presentation of the QR runtimes in 7.2.1. We present results for initial data of Wigner–type (Hermite–1 distribution, the GOE, the Wigner and the Matrix Bernoulli–distribution) as well as some results for the runtimes under non Wigner–type initial conditions (uniform doubly stochastic Jacobi matrices and Jacobi matrices with uniform eigenvalues). Tolerances ϵ and matrix dimension were chosen as in 7.2.1 except where stated otherwise. Refer to figures 7.32 to 7.37 for the type 1 plots in the Toda case. Again for the Wigner–type initial distributions we note similarities across tolerances ϵ, matrix dimensions n and initial distributions, especially in the shape of the runtime distributions. When we compare with the QR case, however, the matrix size does have some influence on the shape, scale and location of the empirical runtimes. In particular as the matrix size becomes larger, the bulk of the runtime distributions seems to shift towards the right while simultaneously the standard deviation ('scale') increases. Considering the non Wigner–type initial distributions in figures 7.36 and 7.37 we observe that the former shows remarkable similarities to the runtime distributions of the QR algorithm including that apparently for larger dimensions the mean and variances of the runtime distributions stabilize. Also, the scale of the empirical runtime distributions is much larger for this set of initial values than the others we considered. We believe this is due to the fact that the eigenvalues stay bounded (between -1 and 1) for all matrix sizes (recall that if we were to scale the Wigner–type initial distributions by $\sqrt{n}$ so that the eigenvalues stay roughly concentrated in $[-2, 2]$ we would also slow down the algorithm).

When we fix an initial matrix dimension and consider the runtime distributions for several tolerances, we obtain the figures 7.38 to 7.43 (type 2). As in the QR case for

smaller values of ϵ the empirical runtime distributions 'diffuse' to the right. Also, from these pictures we can plausibly suspect that the runtime distributions for the Toda algorithms exhibit Gaussian right tails in the case of Wigner–type initial data. The non Wigner–type initial distributions we considered for the Toda algorithm can be seen in figure 7.43 lead to runtime distributions in strong contrast with the ones derived from Wigner–type initial data. Also, note again the high proportion of 'near immediate' deflations for large values of the deflation tolerance.

The figures of type three for the Toda case are 7.44–7.49. Direct comparison of these figures lets us again observe a universal shape of the renormalized runtime histograms in case of the Wigner–type initial conditions. For the normalized runtime distributions the claim of Gaussian right tails becomes even more apparent. Note that the normalized histograms for the runtimes derived from the non Wigner–type ensembles (figures 7.48 and 7.49) combine to a different shape than the normalized histograms of the Wigner–type initial distributions. In particular the normalized distributions for the uniform doubly stochastic Jacobi initial ensemble show strong resemblance to the QR runtime distributions from section 7.2.1. In case of 7.49, there seem to be heavier left tails than in the case of the normalized runtime distributions for Wigner–type initial matrices.

Type four and type five figures for the current case are to be found in 7.50 to 7.53. All information about the means and standard deviations of the Toda runtime distributions for Wigner–type initial matrices is combined in figures 7.50 and 7.52. The non Wigner–type runtime means and standard deviations are presented in figures 7.51 and 7.53. In the above paragraphs we already alluded to the fact that in the current case both the deflation tolerance and the matrix size seems to have a strong impact on the means and variances of the particular runtime distribution. The dependence on n is however nonlinear. This fact makes it hard to come up with an ad hoc regression model to be fit. We have tried several nonlinearities, but the following

Table 7.7: Regression Parameters for the Means of the Toda runtime distributions

Initial Distribution	a_0	a_1	a_2
Hermite–1	-7.0273	1.6795	-0.7708
Full Sym. Wigner type	-6.0669	1.2888	-0.7302
Uniform doubly stoch. Jacobi	-34.01514	0.02984	-6.60133
Jacobi Uniform eig.	-2.78614	0.05318	-0.74315

model seems as plausible as any other one we tried.

$$\mu_{n,\epsilon} \approx a_0 + a_1 \log n + a_2 \log \epsilon \tag{7.18}$$

$$\sigma_{n,\epsilon} \approx b_0 + b_1 \log n + b_2 \log \epsilon \tag{7.19}$$

The dots in figures 7.50 to 7.53 correspond to the predicted values of means and variances based on this regression, but as expected for a rather ad hoc regression model we do not obtain a particularly good fit. Refer to table 7.7 for the regression parameters of equation (7.18) and to table 7.8 for the parameters of equation (7.19). The behavior of the runtime means and standard deviations for initial data sampled from non Wigner–type ensembles yields remarkably different behavior from the runtimes corresponding to Wigner–type initial data: For both initial ensembles linear dependence on both n and $\log n$ seems plausible at least for larger matrix dimensions. This compels us to fit the following model:

$$\mu'_{n,\epsilon} \approx a_0 + a_1 n + a_2 \log \epsilon \tag{7.20}$$

$$\sigma'_{n,\epsilon} \approx b_0 + b_1 n + b_2 \log \epsilon \tag{7.21}$$

and summarize the coefficients in the corresponding tables 7.7 and 7.8. In order to compare the data across different initial distributions directly we present QQ plots (type 6) of (normalized) runtime samples from the GOE initial ensemble against

Table 7.8: Regression Parameters for the Standard deviations of the Toda runtime distributions

Initial Distribution	b_0	b_1	b_2
Hermite–1	-2.1233	0.6324	-0.1727
Full Sym. Wigner type	-1.6532	0.3347	-0.1569
Uniform doubly stoch. Jacobi	-16.46367	0.04845	-3.10561
Jacobi Uniform eig.	0.97525	0.04068	0.01451

(normalized) runtime samples of other initial distributions in figures 7.54 to 7.58. Similar to the QR case we get very good overlap for all Wigner–type initial distributions, whereas the non Wigner–type initial distributions seems to lead to a somewhat different tail behavior than the GOE runtime distribution (heavier right tail, lighter left tail for uniform doubly stochastic Jacobi and the other way around for Jacobi with uniform eigenvalues). Note that for the plot 7.58 resampling was applied to generate the same number of samples from the non–Wigner initial distribution as there are samples for the GOE initial ensemble.

The last type in this sequence presents the combined histograms in figures 7.59 and 7.61. The combined histograms of the runtime samples for all the Wigner–type initial data (figure 7.59): again show a good overlap across all matrix sizes, tolerances and initial conditions. The normal distribution with mean zero and variance one which was superimposed on this picture shows a good fit especially in the right tail. The figures 7.60 and 7.61 combine the normalized histograms of the GOE runtimes with the normalized runtimes generated using non Wigner–type initial matrices. As expected the overlap is much worse than when we were comparing exclusively Wigner–type initial data. We also reconfirm our suggestion from the above paragraph that the non Wigner–type initial data leads to the described left and right tails. Due to the fact that the runtime behavior of the uniform doubly stochastic Jacobi initial data under the Toda flow resembles the runtime behavior of the QR algorithm we have provided a QQ plot of the normalized runtimes of the QR with GOE initial

data against the normalized runtimes of the Toda algorithm with uniform doubly stochastic Jacobi initial data (figure B.1) as well as combined histograms for the two regimes in appendix two (figure B.2).

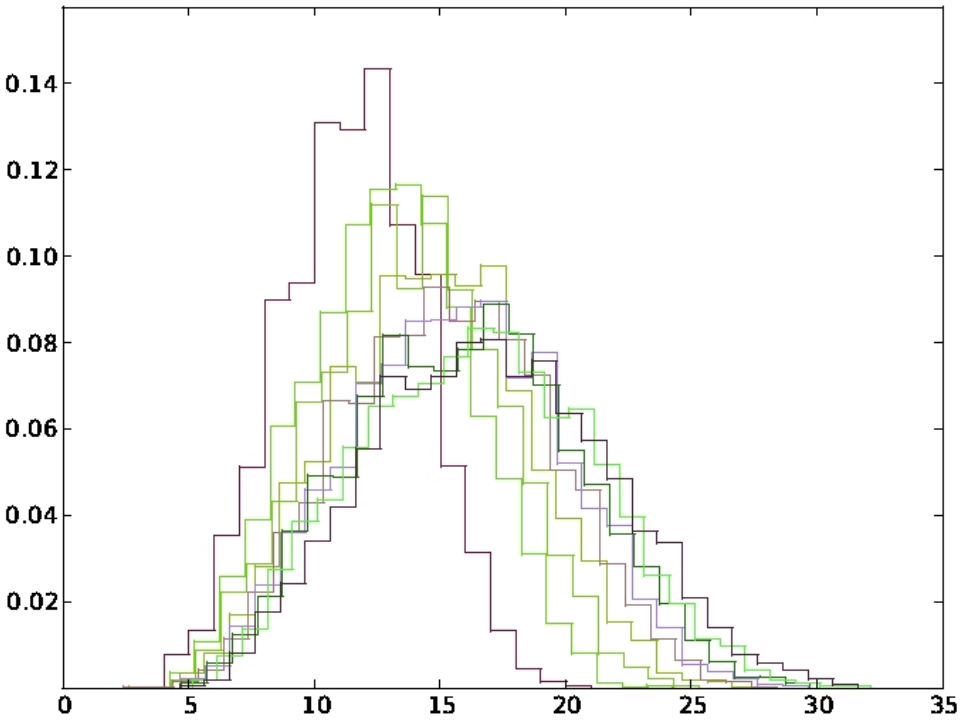

Figure 7.32: Empirical hit time distributions of Toda algorithm for Hermite–1 initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 1).

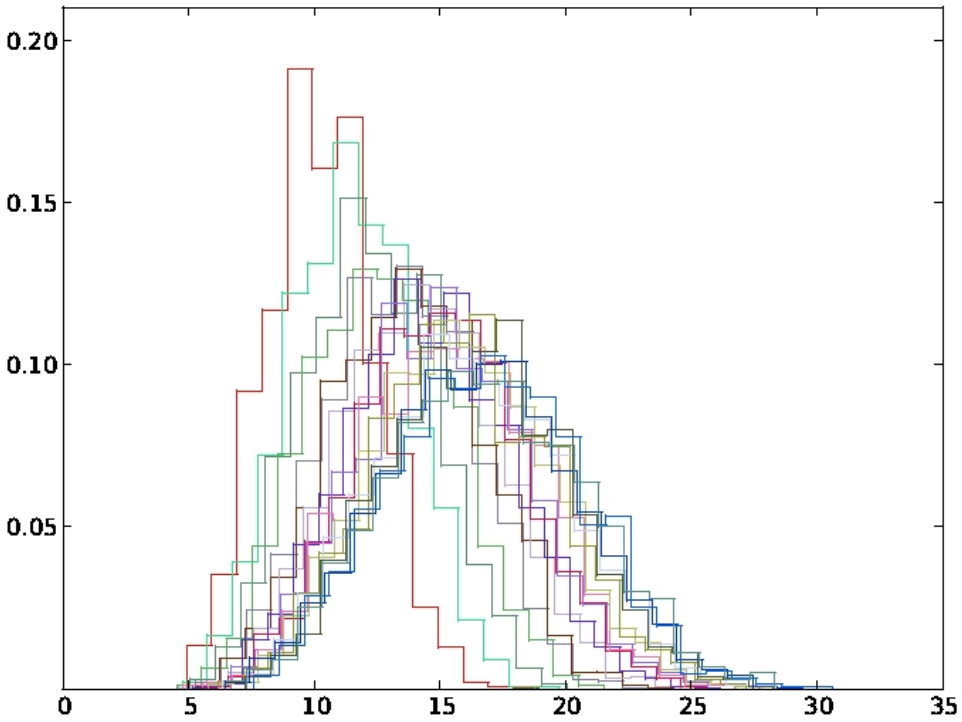

Figure 7.33: Empirical hit time distributions of Toda algorithm for GOE initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 1).

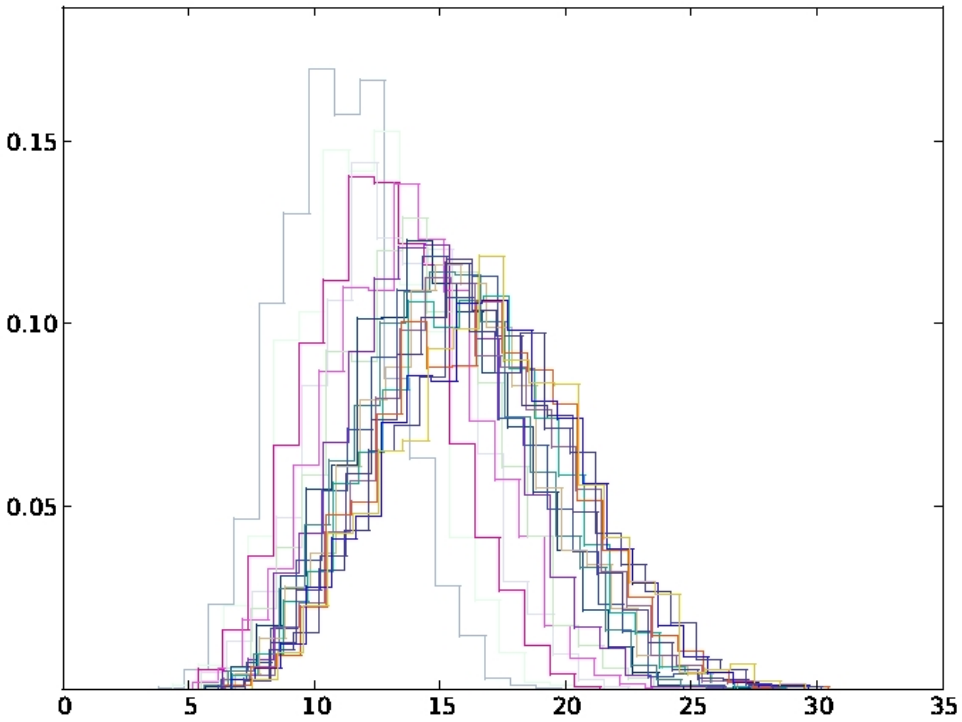

Figure 7.34: Empirical hit time distributions of Toda algorithm for Wigner initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 1).

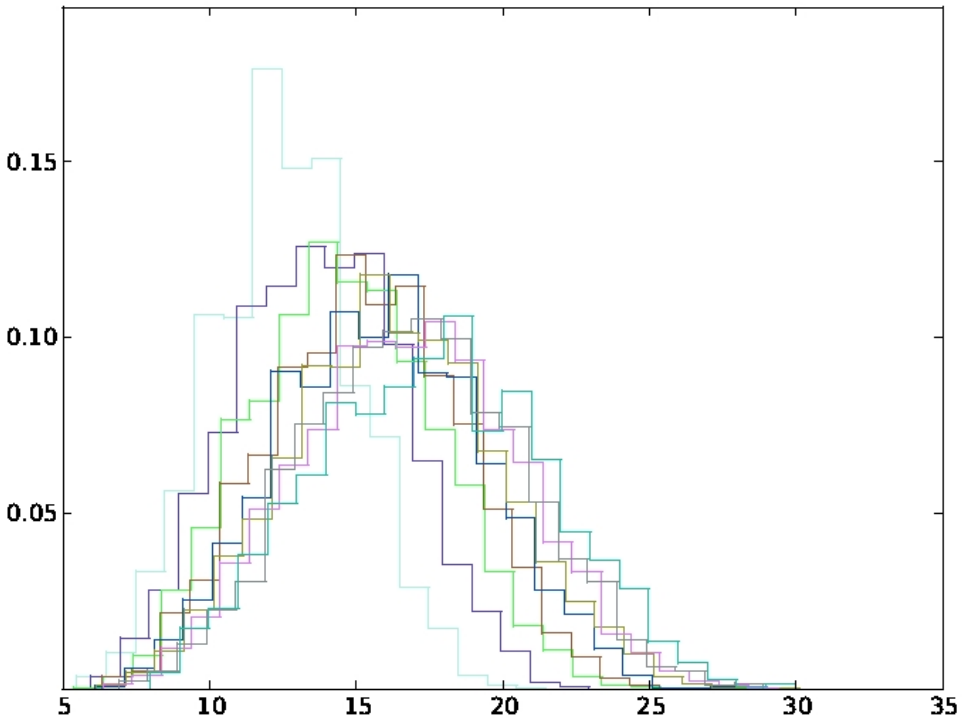

Figure 7.35: Empirical hit time distributions of Toda algorithm for Bernoulli matrix initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 1).

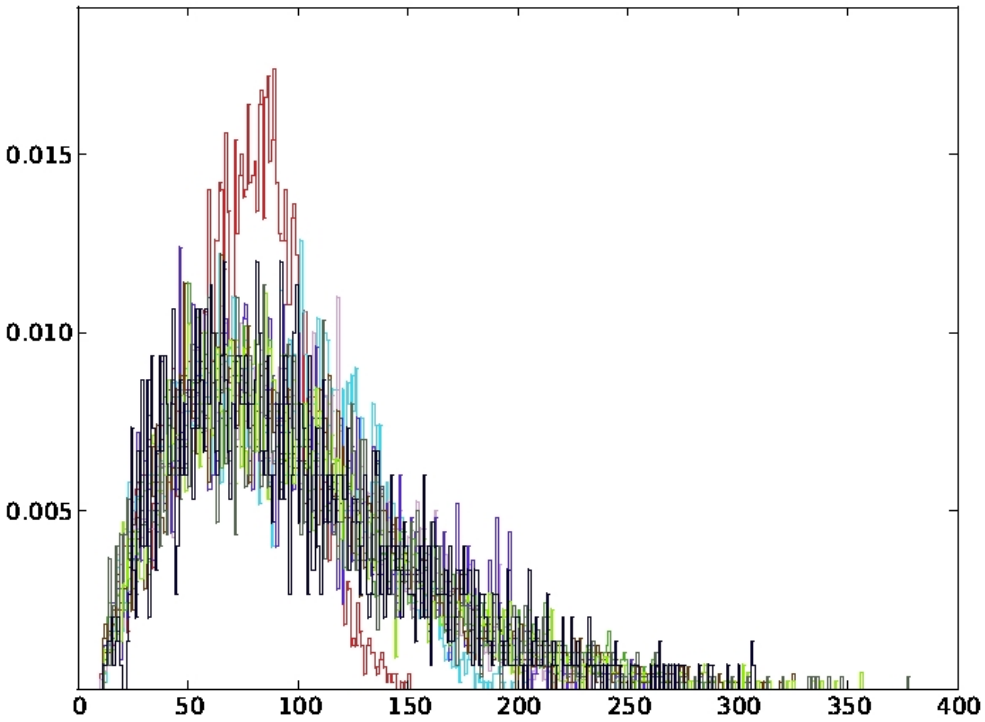

Figure 7.36: Empirical hit time distributions of Toda algorithm for uniform doubly stochastic Jacobi initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 1).

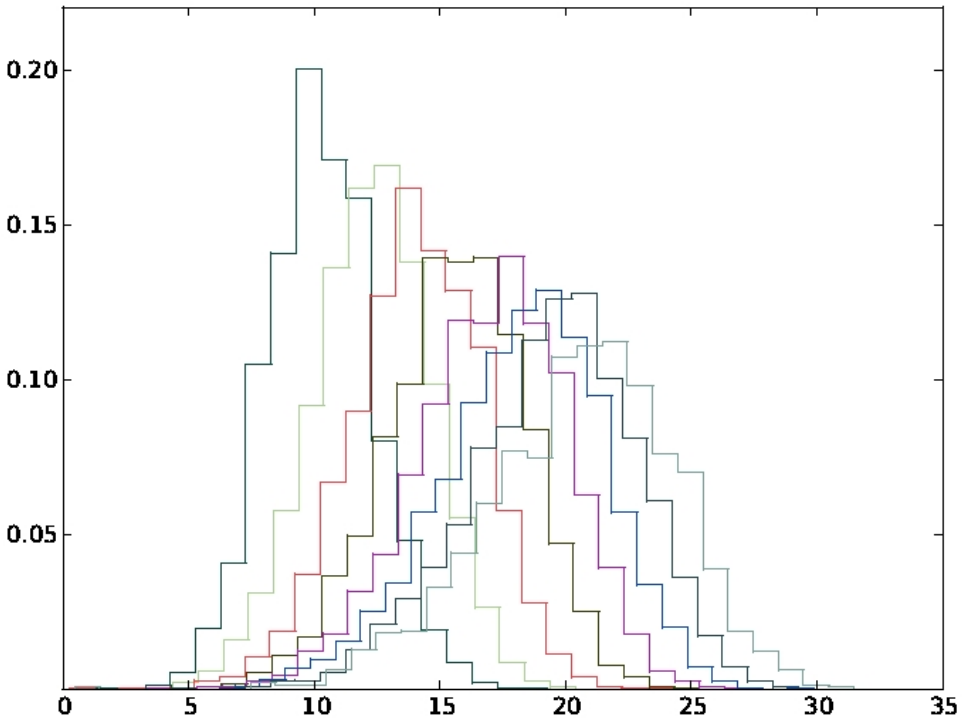

Figure 7.37: Empirical hit time distributions of Toda algorithm for Jacobi with uniform eigenvalue initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 1).

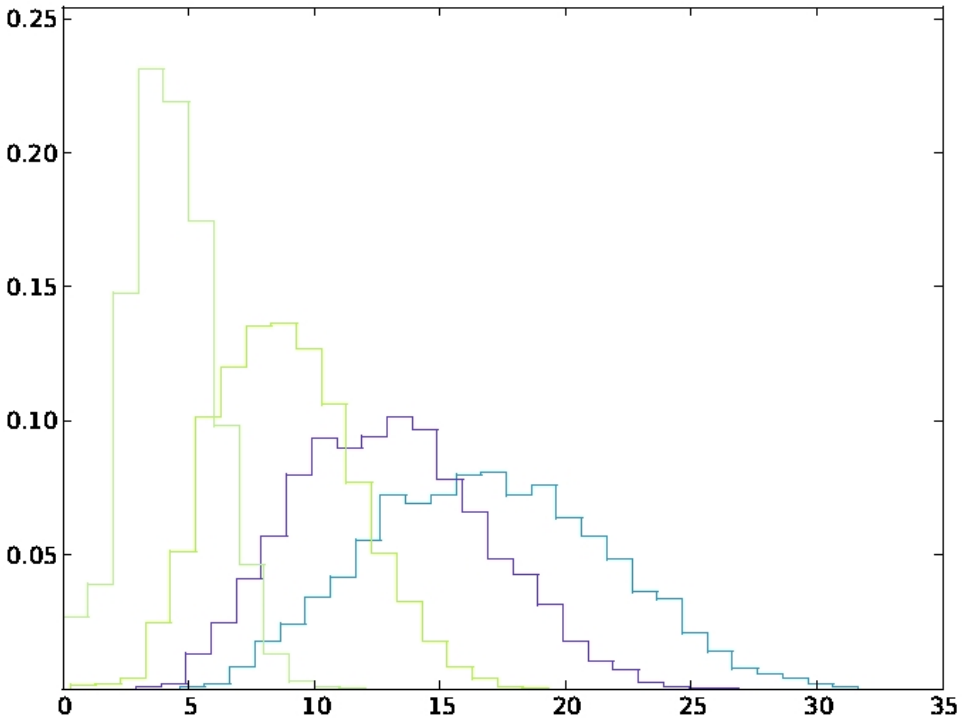

Figure 7.38: Empirical hit time distributions of Toda algorithm for Hermite–1 initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 2).

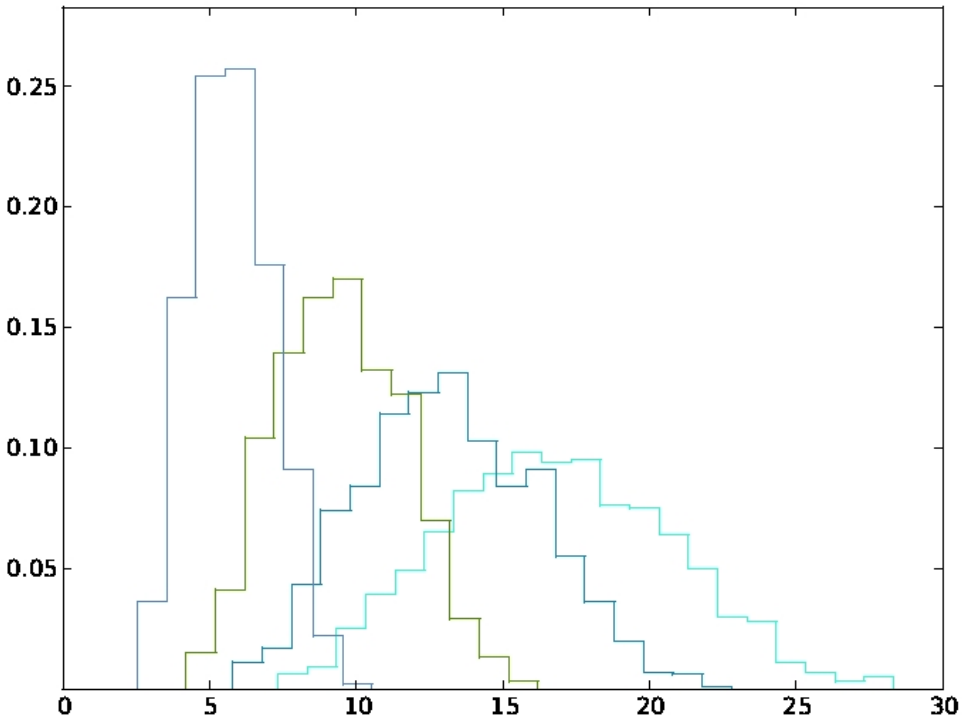

Figure 7.39: Empirical hit time distributions of Toda algorithm for GOE initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 2).

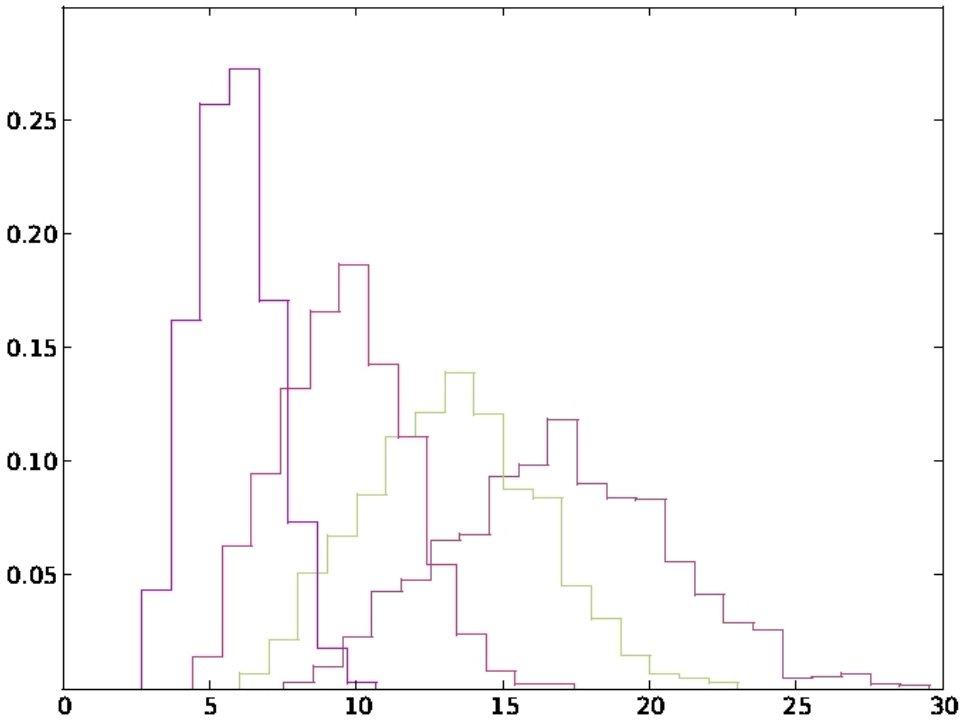

Figure 7.40: Empirical hit time distributions of Toda algorithm for Wigner initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 2).

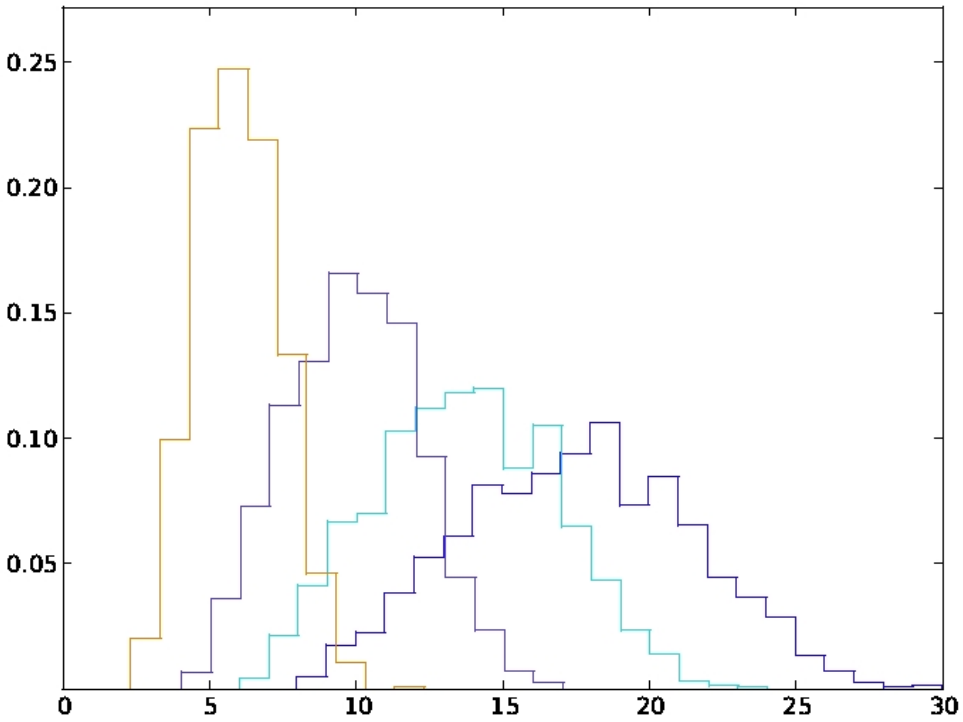

Figure 7.41: Empirical hit time distributions of Toda algorithm for Bernoulli matrix initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 2).

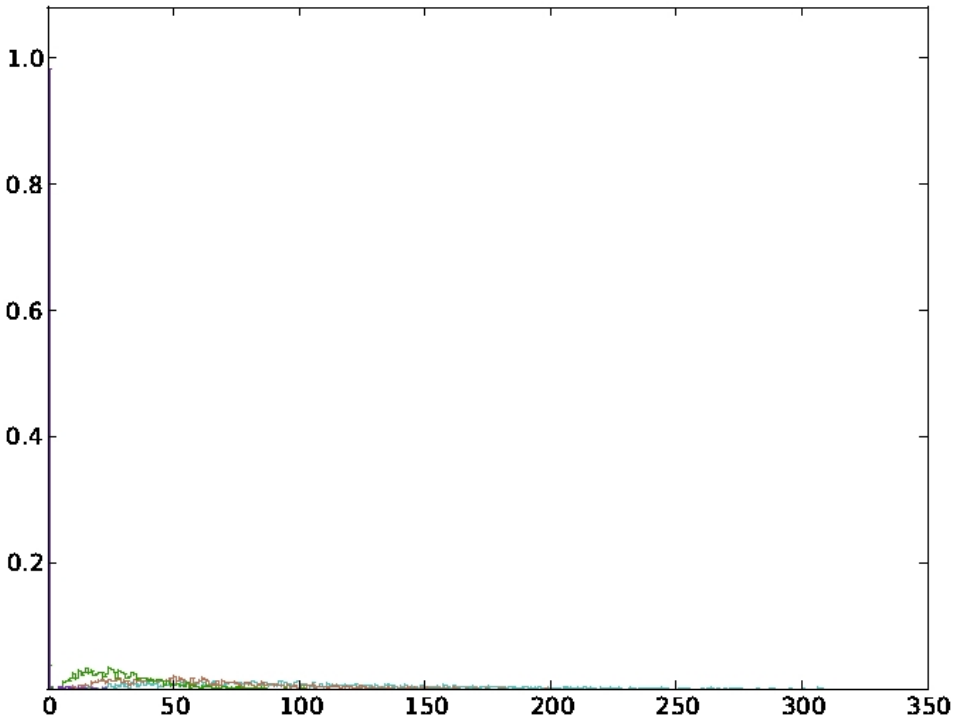

Figure 7.42: Empirical hit time distributions of Toda algorithm for uniform doubly stochastic Jacobi matrix initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 2).

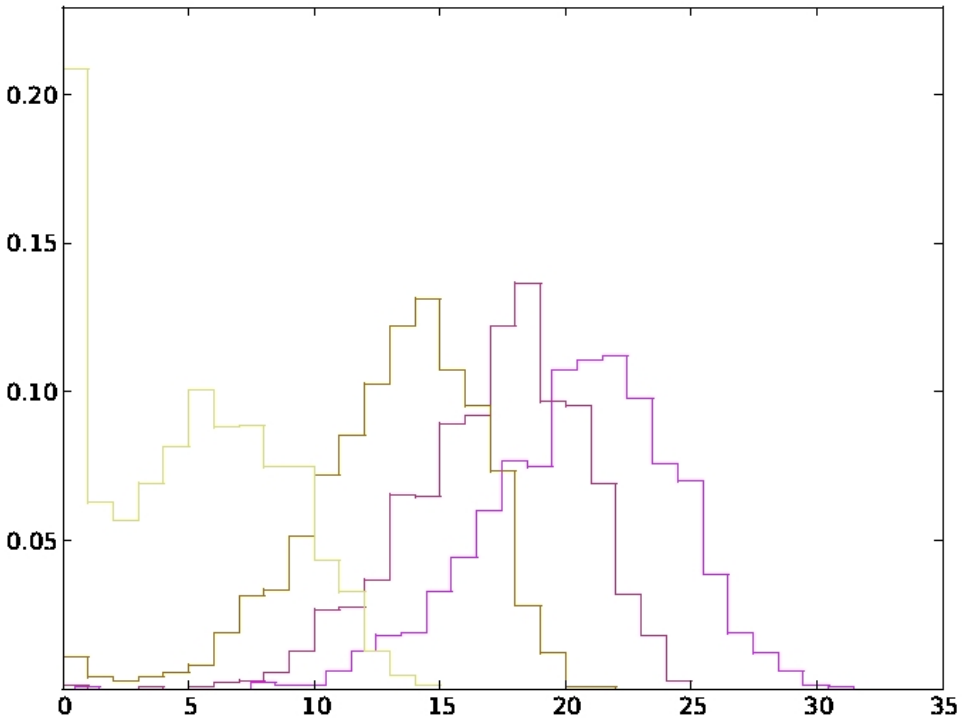

Figure 7.43: Empirical hit time distributions of Toda algorithm for Jacobi with uniform eigenvalue initial data with $\epsilon = 10^{-8}$ and matrix dimensions $10, 30, \ldots, 190$ (type 2).

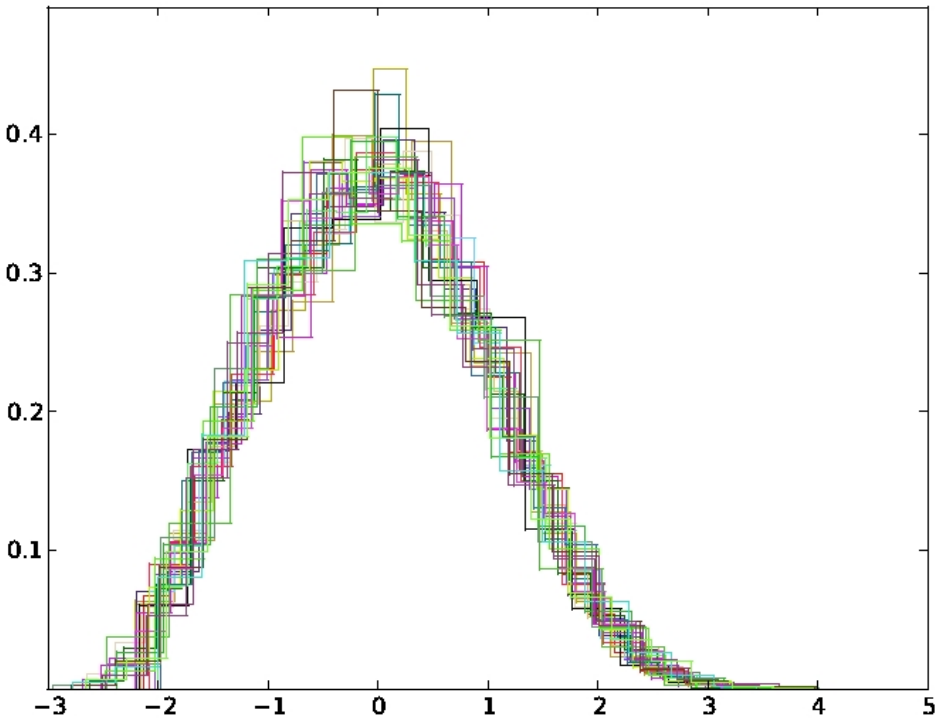

Figure 7.44: Normalized empirical hit time distributions of Toda algorithm for Hermite–1 initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimensions $10, 30, \ldots, 190$ (type 3).

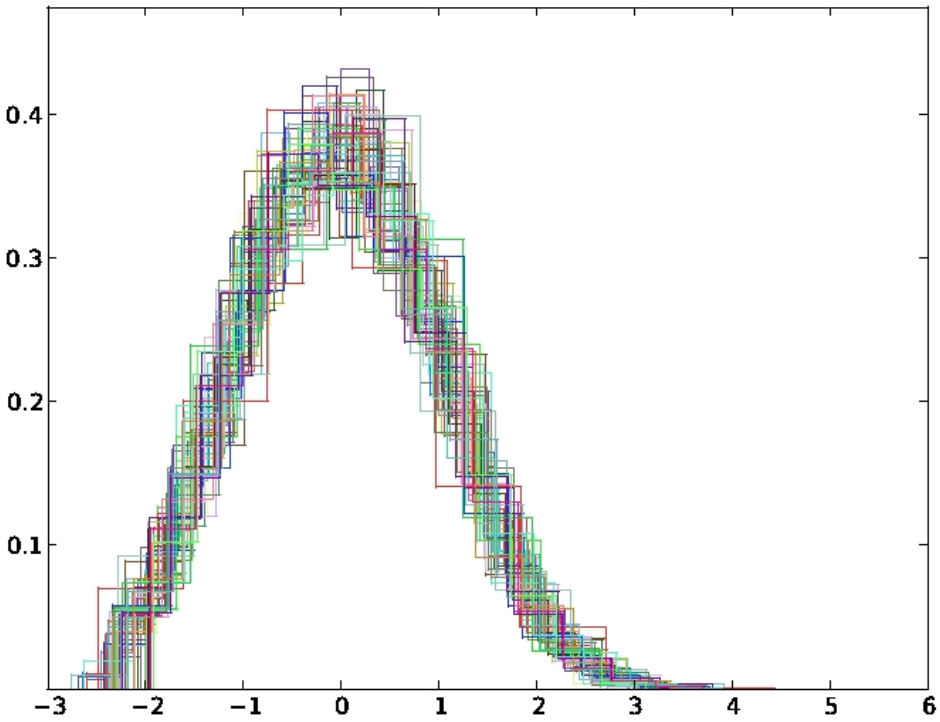

Figure 7.45: Normalized empirical hit time distributions of Toda algorithm for GOE initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimensions $10, 30, \ldots, 190$ (type 3).

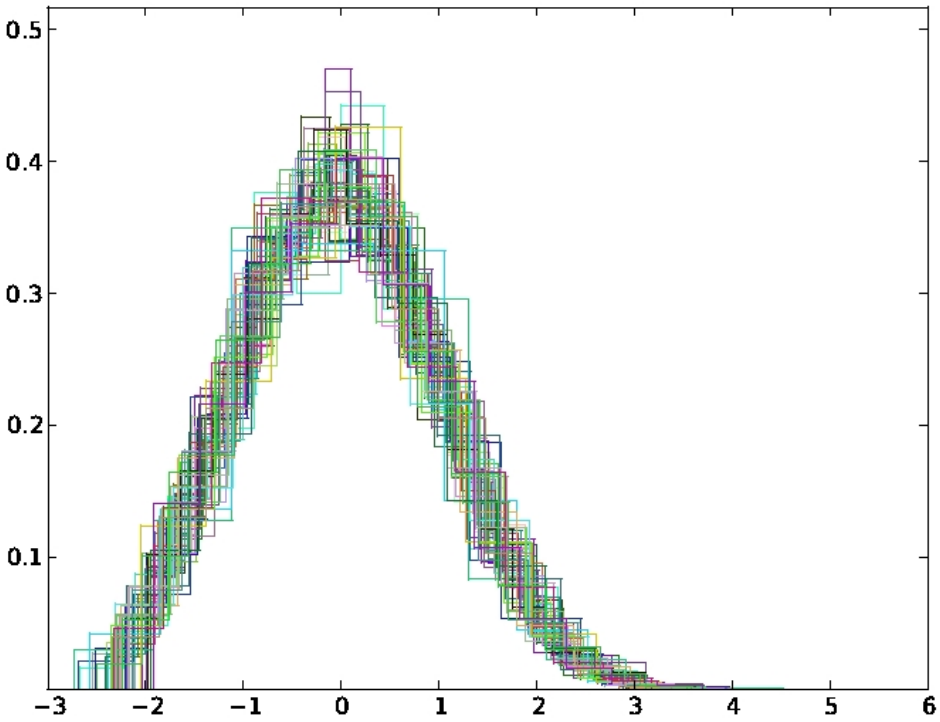

Figure 7.46: Normalized empirical hit time distributions of Toda algorithm for Wigner initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimensions $10, 30, \ldots, 190$ (type 3).

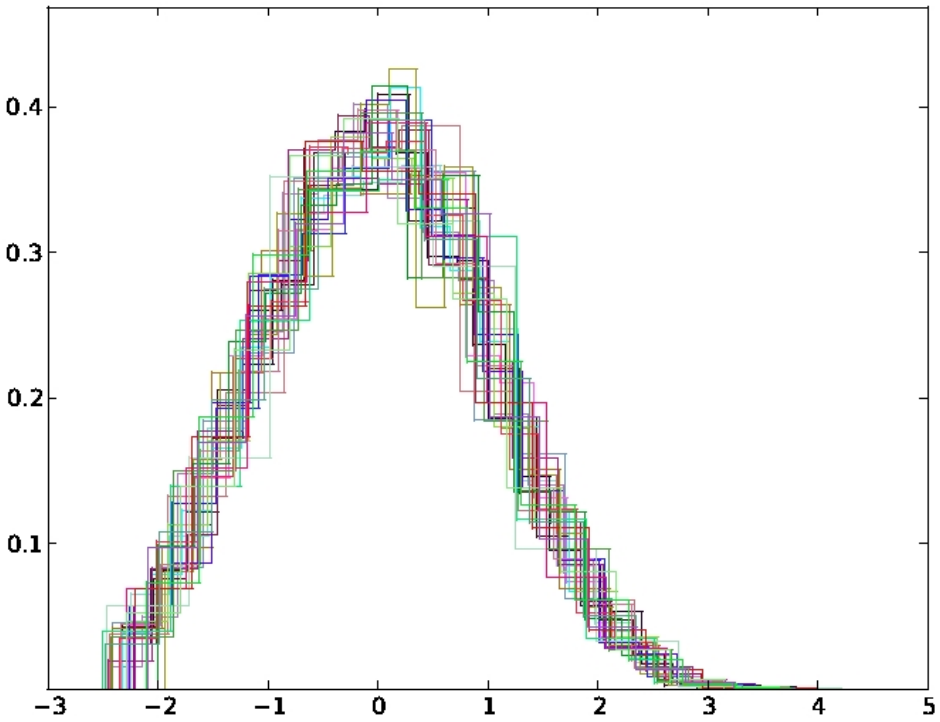

Figure 7.47: Normalized empirical hit time distributions of Toda algorithm for Bernoulli initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimensions $10, 30, \ldots, 190$ (type 3).

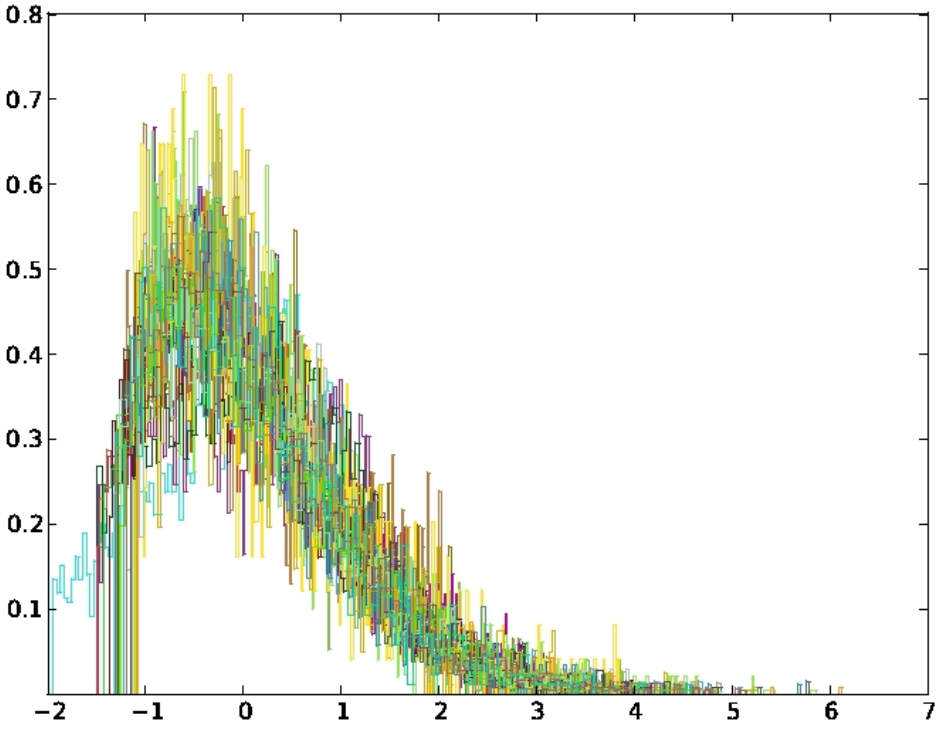

Figure 7.48: Normalized empirical hit time distributions of Toda algorithm for uniform doubly stochastic Jacobi initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimensions $10, 30, \ldots, 190$ (type 3).

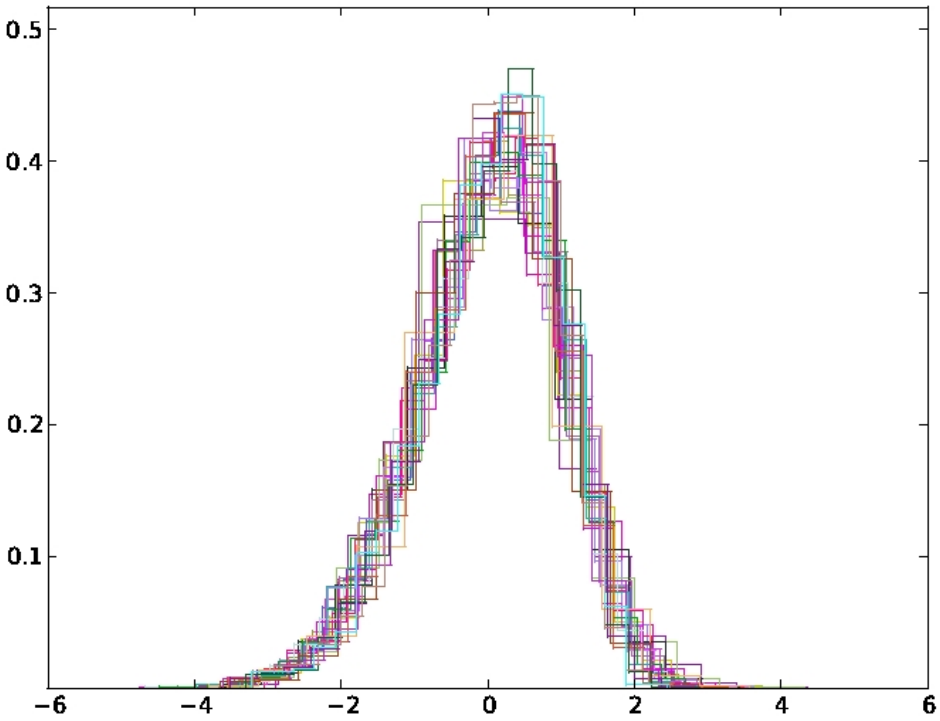

Figure 7.49: Normalized empirical hit time distributions of Toda algorithm for Jacobi with uniform eigenvalue initial data with $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$ and matrix dimensions $10, 30, \ldots, 190$ (type 3).

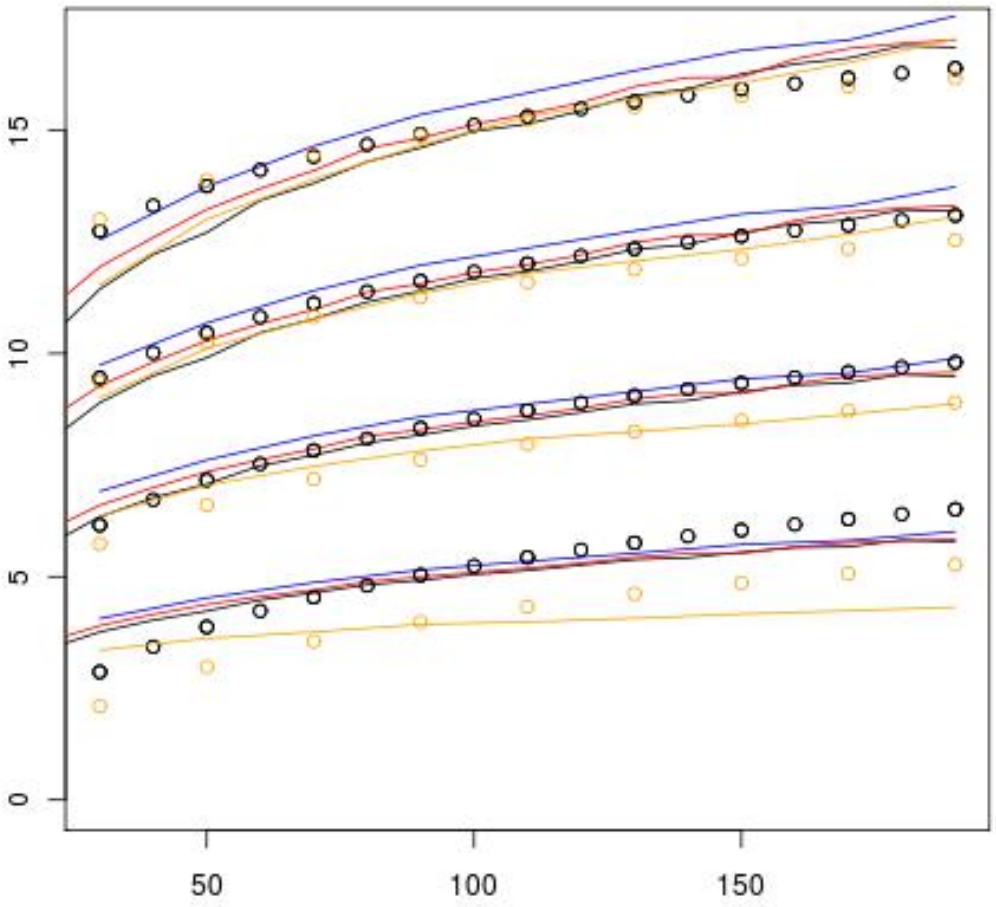

Figure 7.50: Means of the empirical hit time distributions of Toda algorithm for Wigner–type initial data with Orange/ black/ blue/ red correspond to Hermite–1/ GOE/ Bernoulli/ Wigner initial data (type 4). Dots correspond to regression predictions.

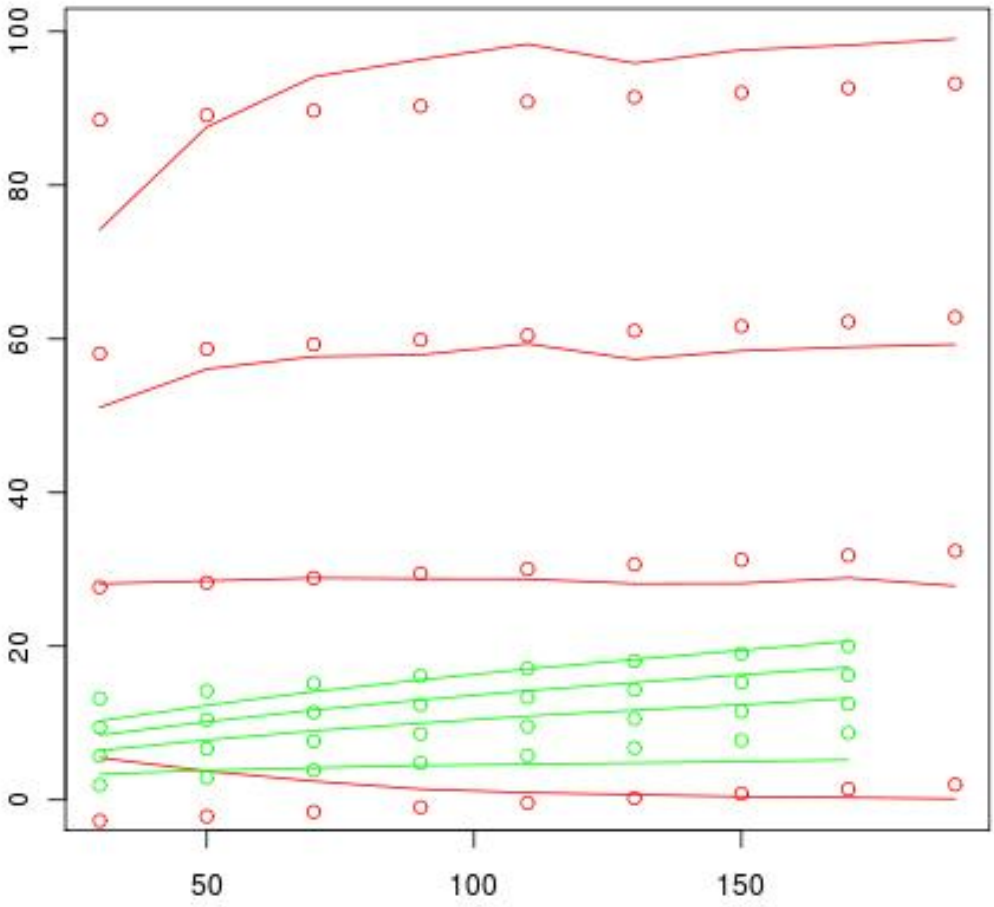

Figure 7.51: Means of the empirical hit time distributions of Toda algorithm for Wigner–type initial data with Red/ green correspond to uniform doubly stochastic Jacobi/ Jacobi with uniform eigenvalue initial data (type 4). Dots correspond to regression predictions.

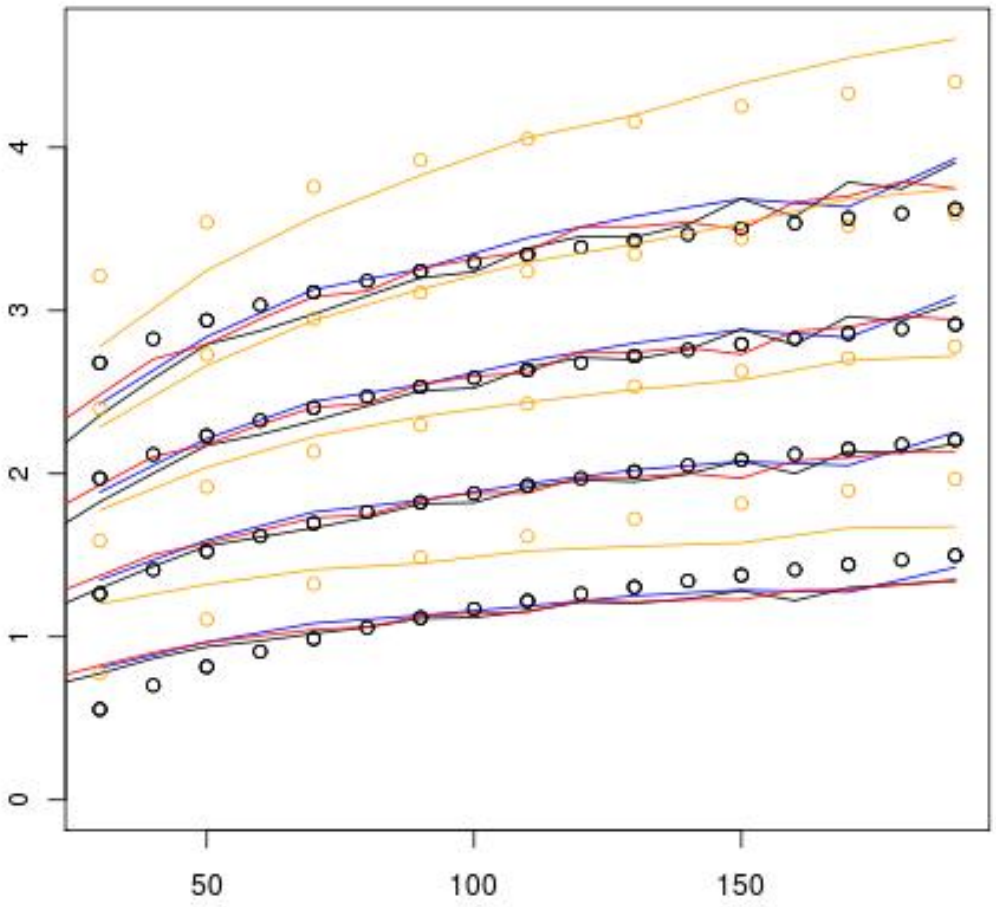

Figure 7.52: Standard deviations of the empirical hit time distributions of Toda algorithm for Wigner–type initial data with Orange/ black/ blue/ red correspond to Hermite–1/ GOE/ Bernoulli/ Wigner initial data (type 5). Dots correspond to regression predictions.

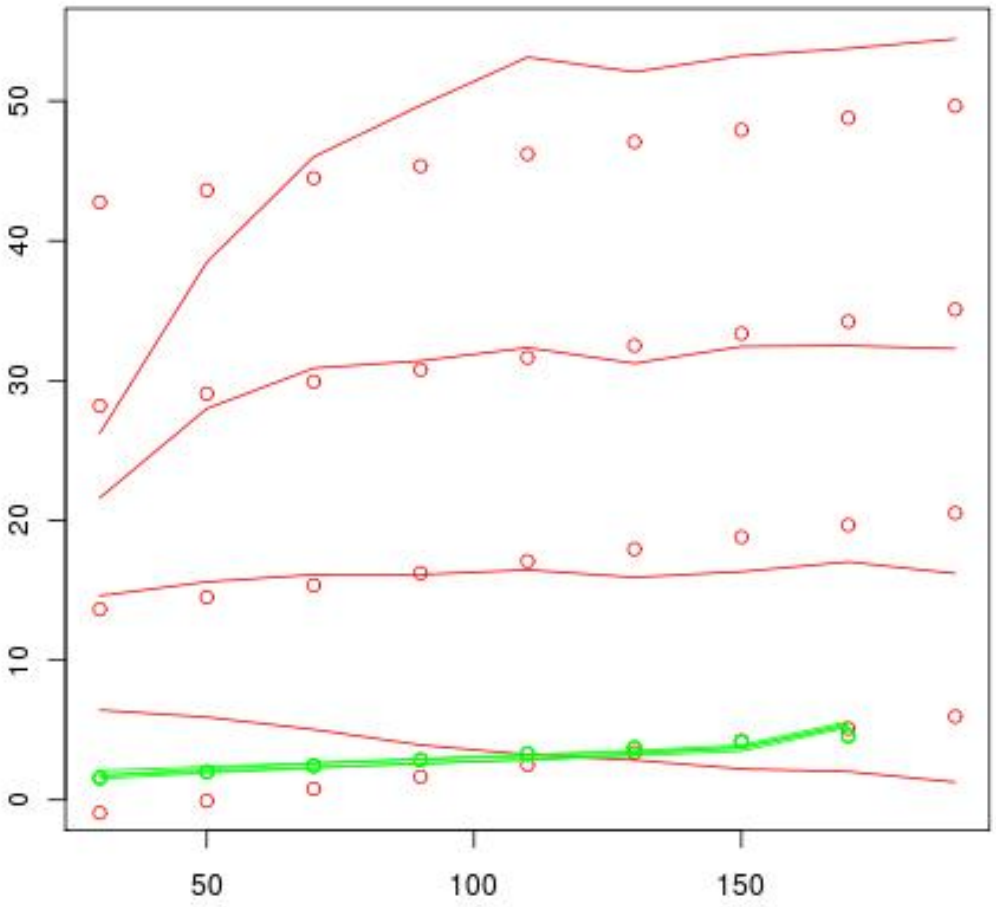

Figure 7.53: Means of the empirical hit time distributions of Toda algorithm for Wigner–type initial data with Red/ green correspond to uniform doubly stochastic Jacobi/ Jacobi with uniform eigenvalue initial data (type 4). Dots correspond to regression predictions.

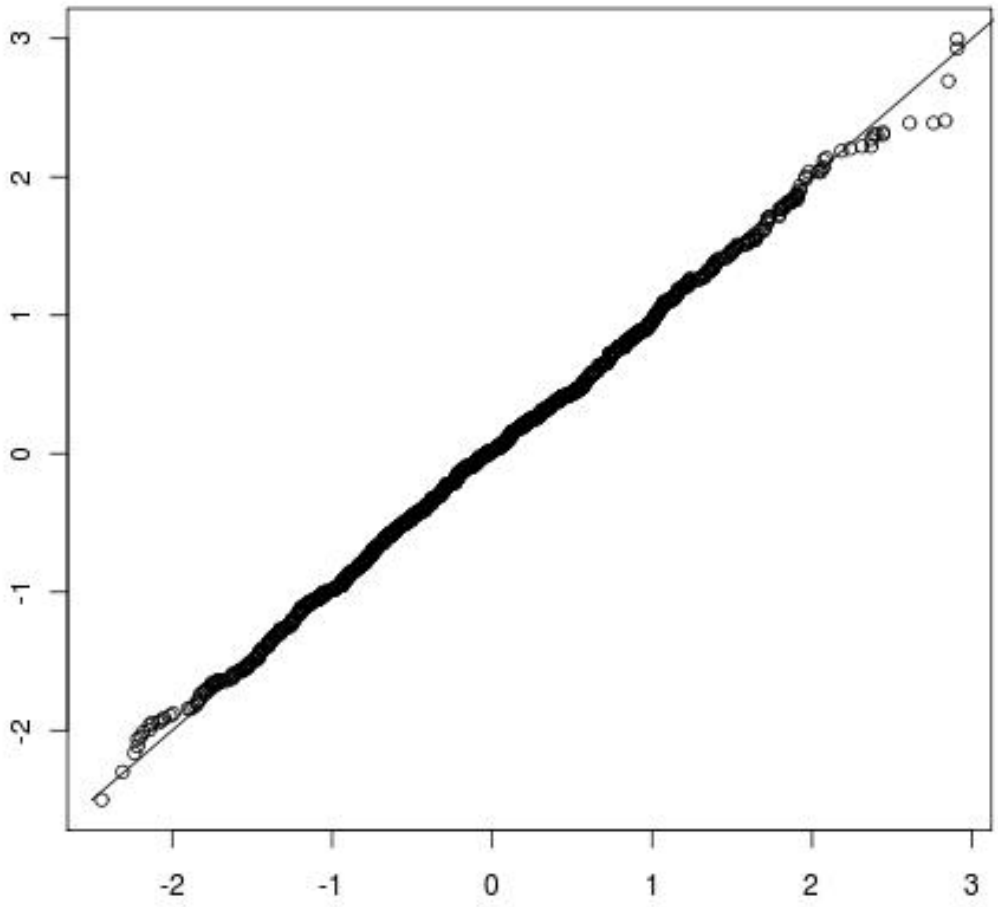

Figure 7.54: QQ plot of the normalized (mean zero, variance one) Toda runtime sampes of GOE initial data ($n = 190$ and $\epsilon = 10^{-8}$) against the normalized runtime samples of Hermite–1 initial data ($n = 170$, same ϵ).

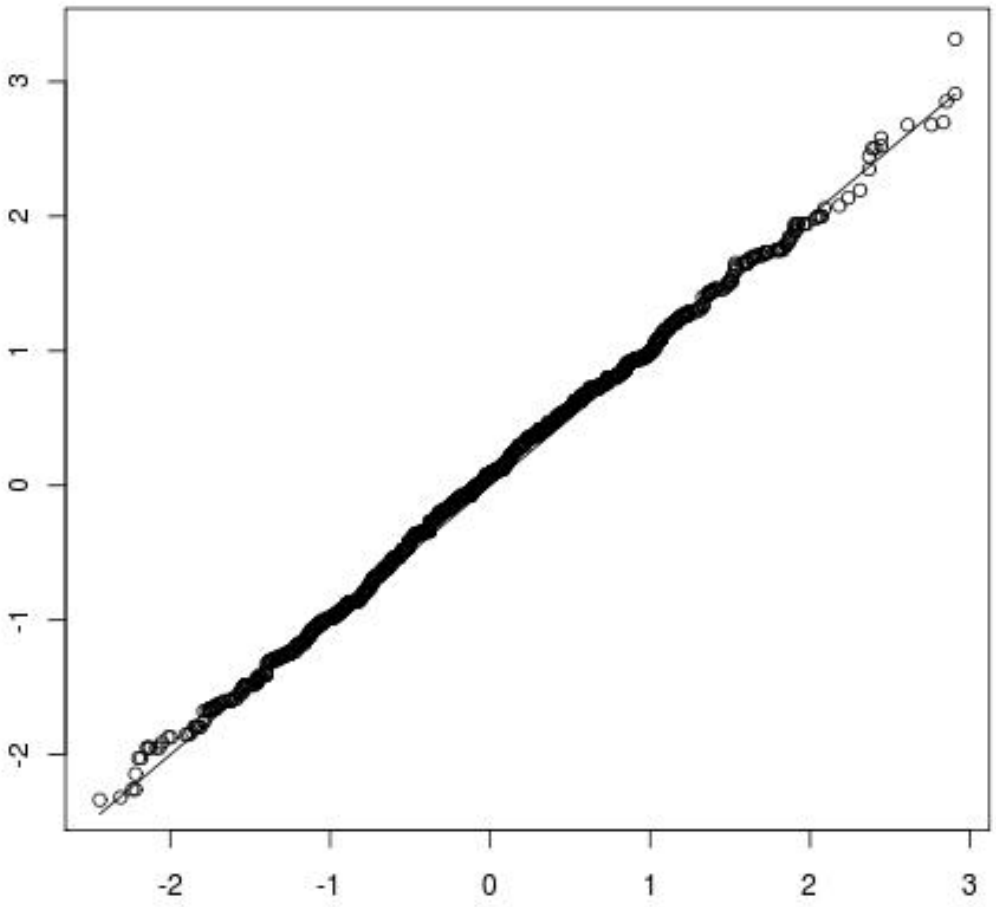

Figure 7.55: QQ plot of the normalized (mean zero, variance one) Toda runtime samples of GOE initial data ($n = 190$ and $\epsilon = 10^{-8}$) against the normalized runtime samples of Wigner initial data (same parameters).

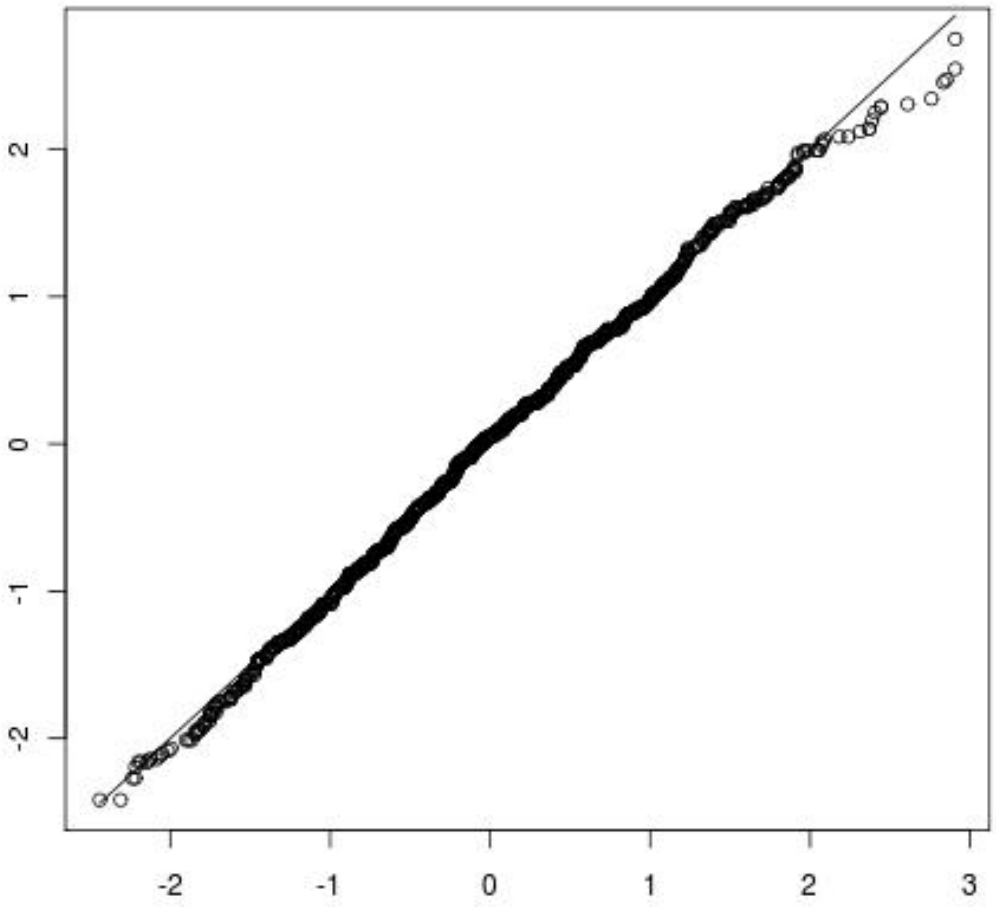

Figure 7.56: QQ plot of the normalized (mean zero, variance one) Toda runtime samples of GOE initial data ($n = 190$ and $\epsilon = 10^{-8}$) against the normalized runtime samples of Bernoulli initial data (same parameters).

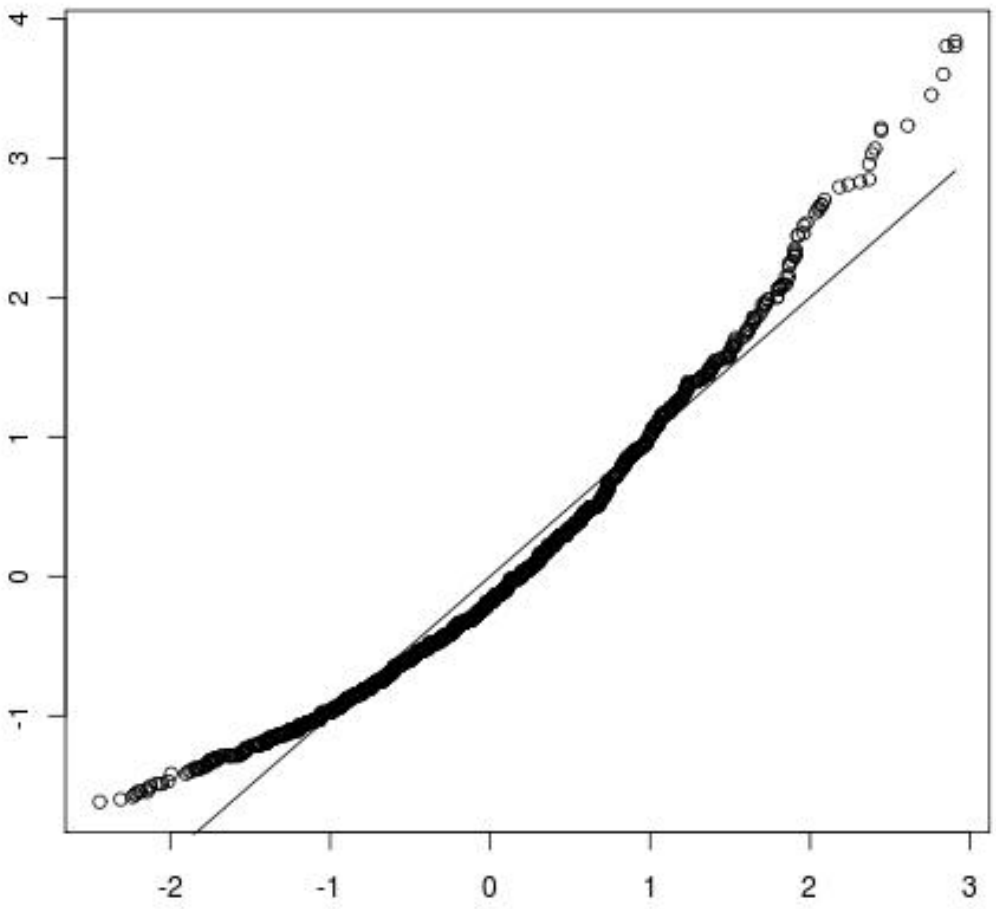

Figure 7.57: QQ plot of the normalized (mean zero, variance one) Toda runtime samples of GOE initial data ($n = 190$ and $\epsilon = 10^{-8}$) against the normalized runtime samples of uniform Jacobi doubly stochastic initial data (same parameters).

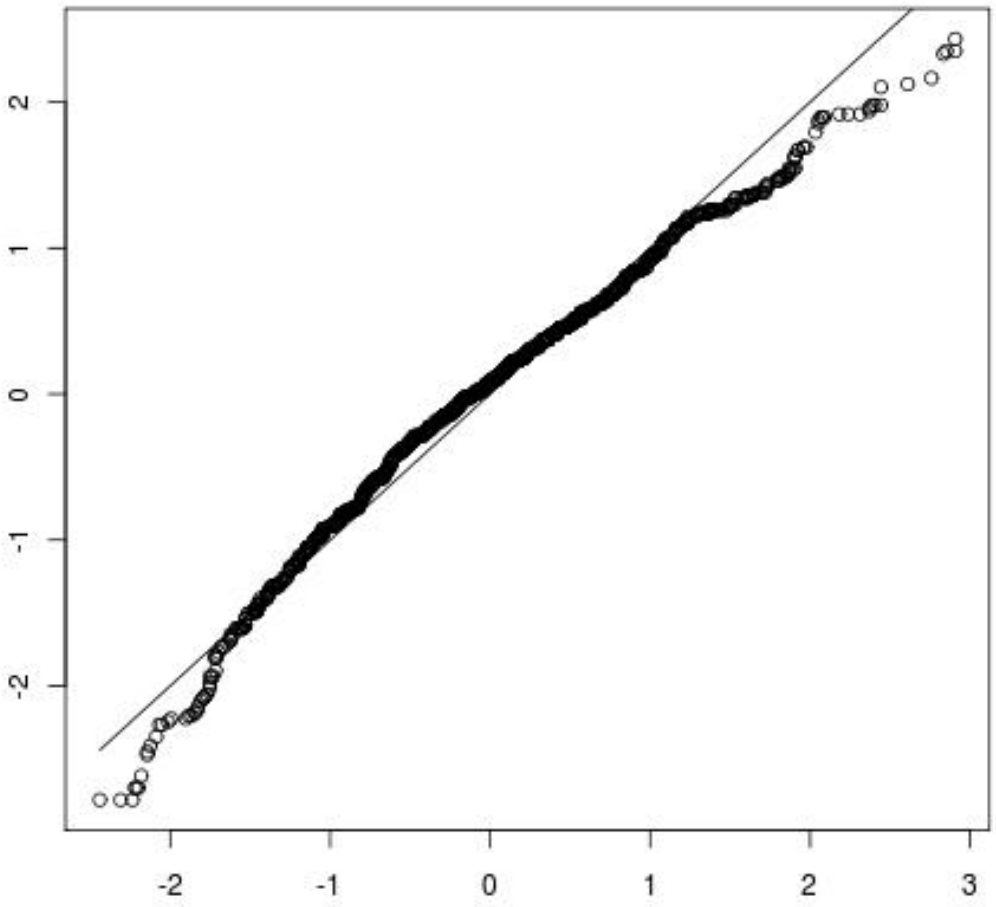

Figure 7.58: QQ plot of the normalized (mean zero, variance one) Toda runtime samples of GOE initial data ($n = 190$, $\epsilon = 10^{-8}$) against the normalized runtime samples of Jacobi with uniform eigenvalue initial data (same parameters).

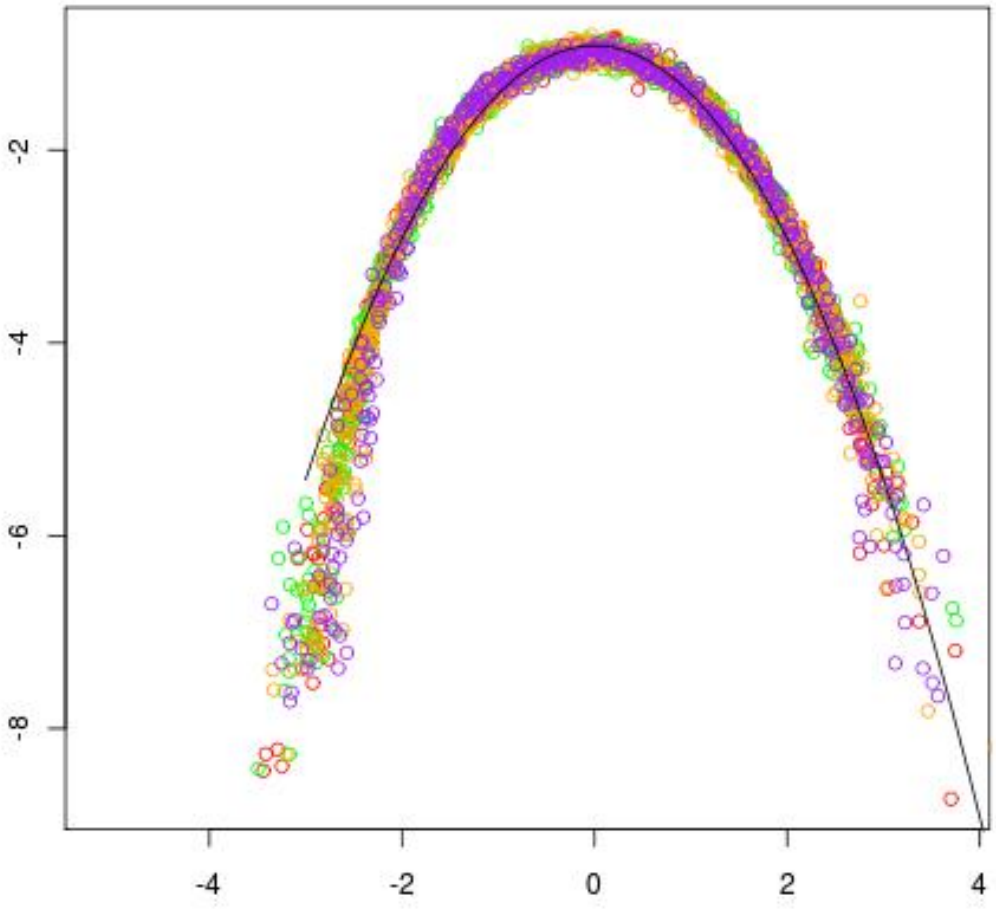

Figure 7.59: Combination of all the normalized Toda runtime histograms compared with a standard normal distribution with parameters $\mu = 0$ and $\sigma = 1$.

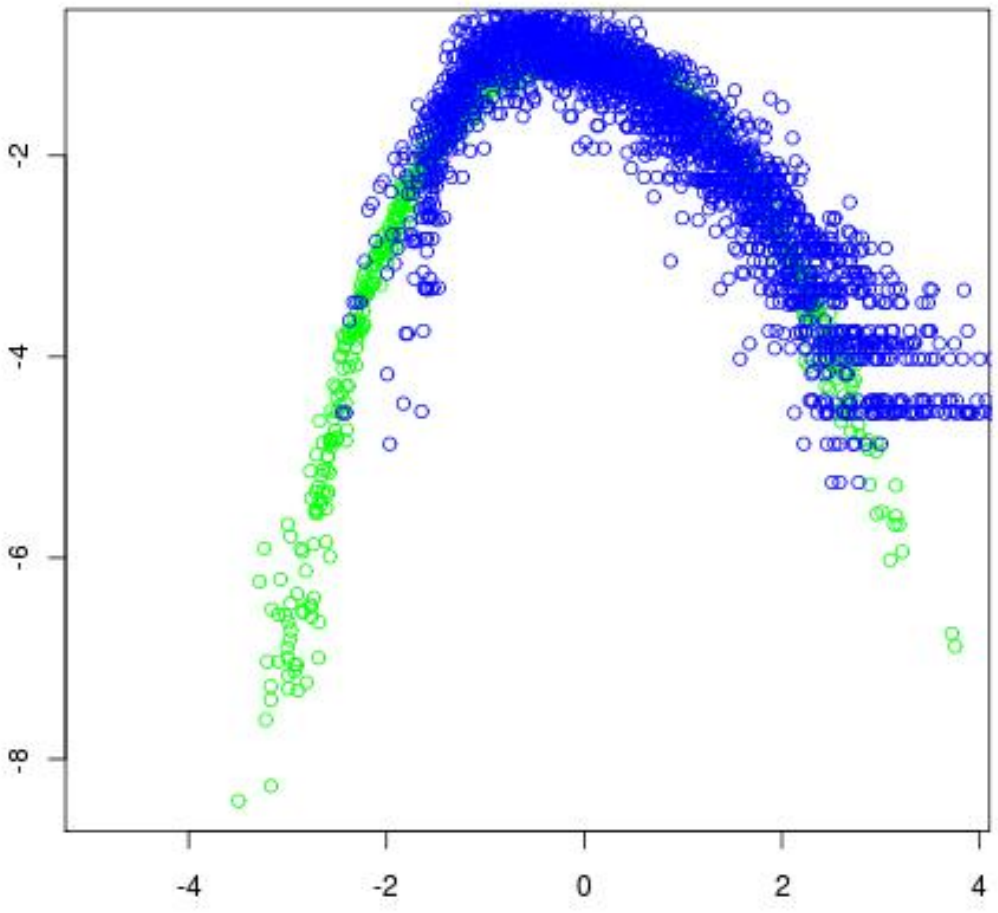

Figure 7.60: Combination of all the normalized Toda runtime histograms with GOE initial data compared with all the runtime histograms with uniform doubly stochastic Jacobi initial data.

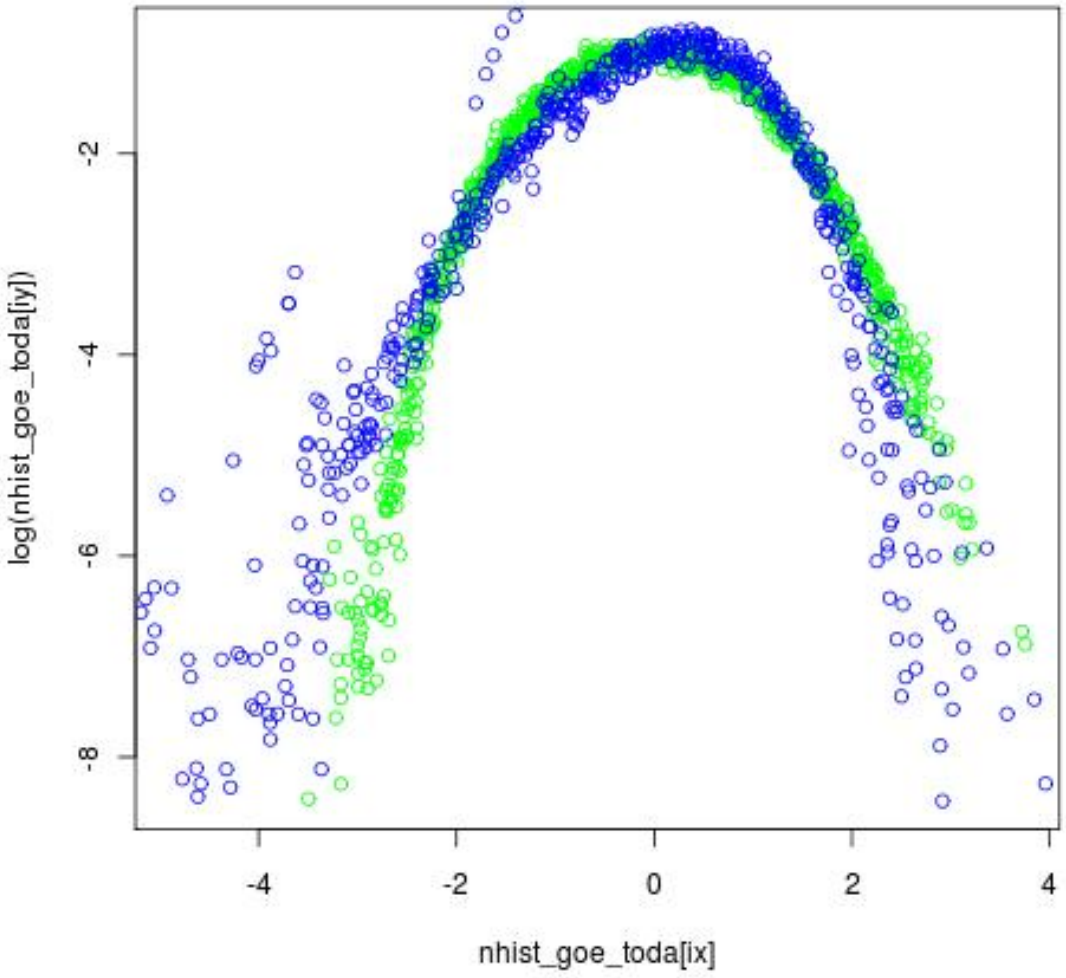

Figure 7.61: Combination of all the Toda runtime histograms with GOE initial data compared with all the runtime histograms with Jacobi with uniform eigenvalues initial data.

7.3 Empirical Results on the Deflation Position

In this section we present empirical results on the distribution of $I_\epsilon(M_0)$, where M_0 is distributed according to various initial distributions. Recall that I_ϵ is defined as the subdiagonal index at which the first deflation occurs. If the matrix M_0 is of dimension $n \times n$ then the function I_ϵ can take $n-1$ different values. Based on the array indexing scheme in Python these values range from zero to $n-2$. Most of the figures presented in this section show empirical histograms of the deflation index position on a log scale in the vertical (y–)direction in order to get a better idea of the speed of decay.

7.3.1 Deflation Position for the QR algorithm

We present typical results of our empirical study of the distribution of the deflation position in the QR algorithm for the initial conditions that we have been considering in the figures 7.62–7.65. It is remarkable that for all cases of initial conditions and matrix sizes, the distribution of deflation indices seems to have essentially the same properties: The deflations are concentrated at the lower end of the matrices and the probability that the first deflation occurs at a given index seems to decay exponentially with decreasing subdiagonal index. These properties are also independent of the deflation tolerance. For our pictures, however, we have fixed $\epsilon = 10^{-8}$.

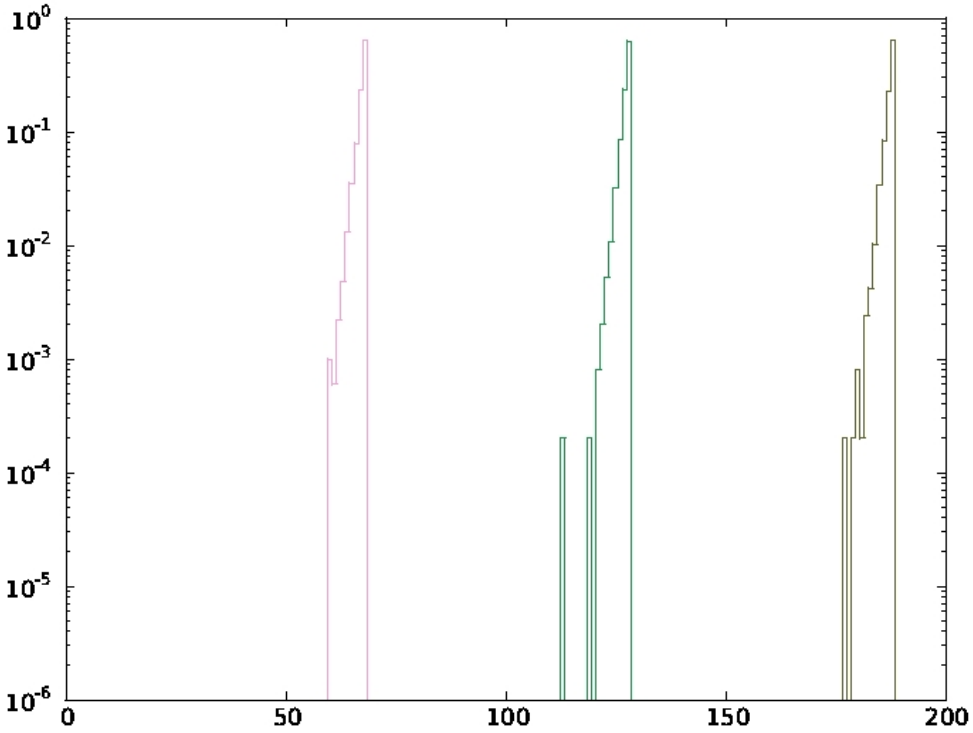

Figure 7.62: Empirical deflation index distributions of QR algorithm for Hermite–1 initial data with $\epsilon = 10^{-8}$ and matrix dimensions $70, 130, 190$.

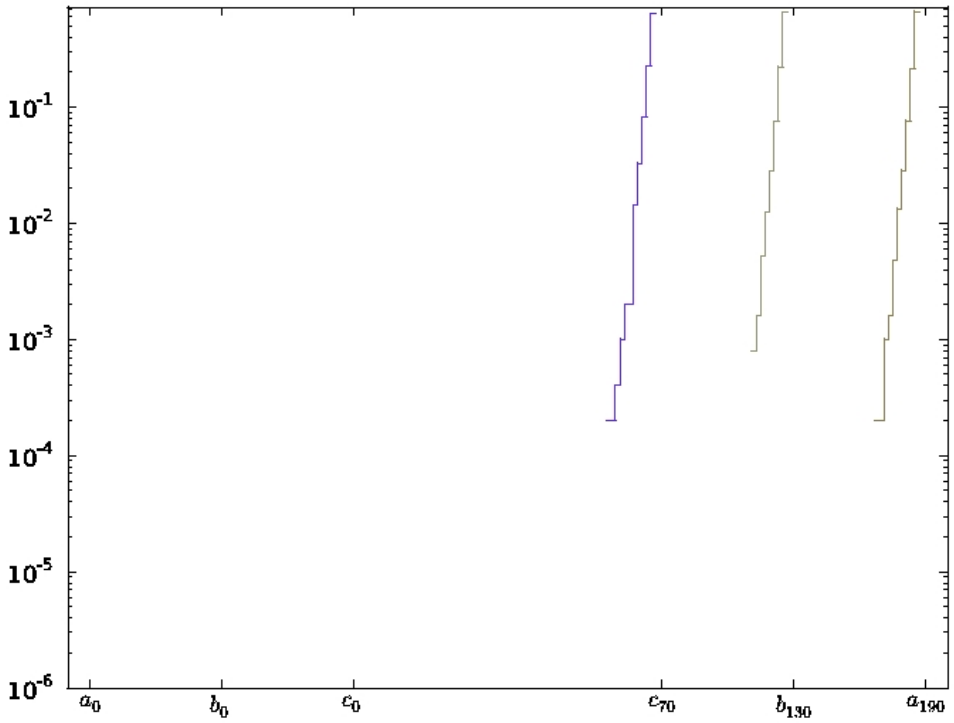

Figure 7.63: Empirical deflation index distributions of QR algorithm for GOE initial data with $\epsilon = 10^{-8}$ and matrix dimensions $70, 130, 190$. We have centered the hit index distributions for better visibility so that the peak at a_0 (b_0, c_0) corresponds to the peak at a_{190} (b_{130}), c_{70} respectively). The subindices correspond to indices on the matrix subdiagonal. Note that the deflation index by definition takes values from zero to $n - 2$.

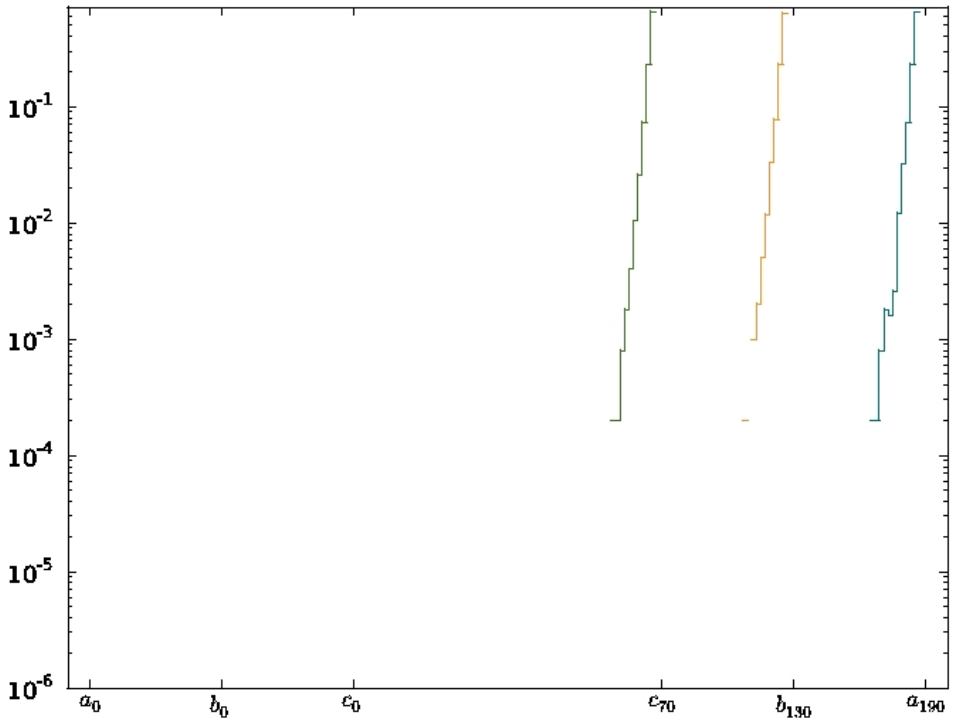

Figure 7.64: Empirical deflation index distributions of QR algorithm for iid Gaussian initial data with $\epsilon = 10^{-8}$ and matrix dimensions $70, 130, 190$. Hit index distributions are centered as in 7.63 with same meaning of a_i, b_i, c_i.

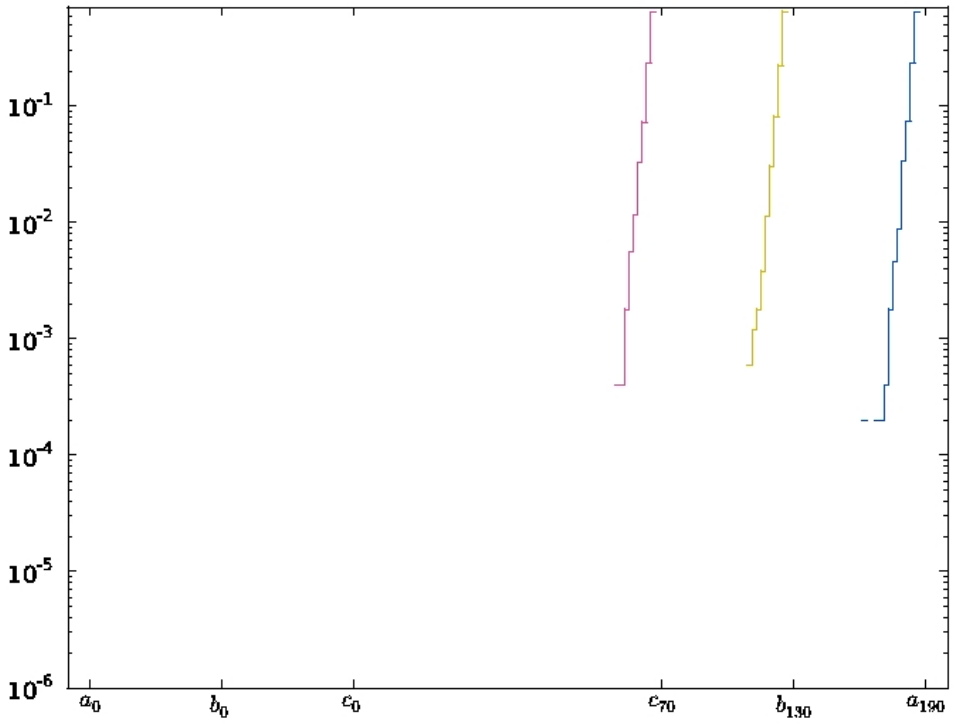

Figure 7.65: Empirical deflation index distributions of QR algorithm for Bernoulli matrix initial data with $\epsilon = 10^{-8}$ and matrix dimensions $70, 130, 190$. Hit index distributions are centered as in 7.63 with same meaning of a_i, b_i, c_i.

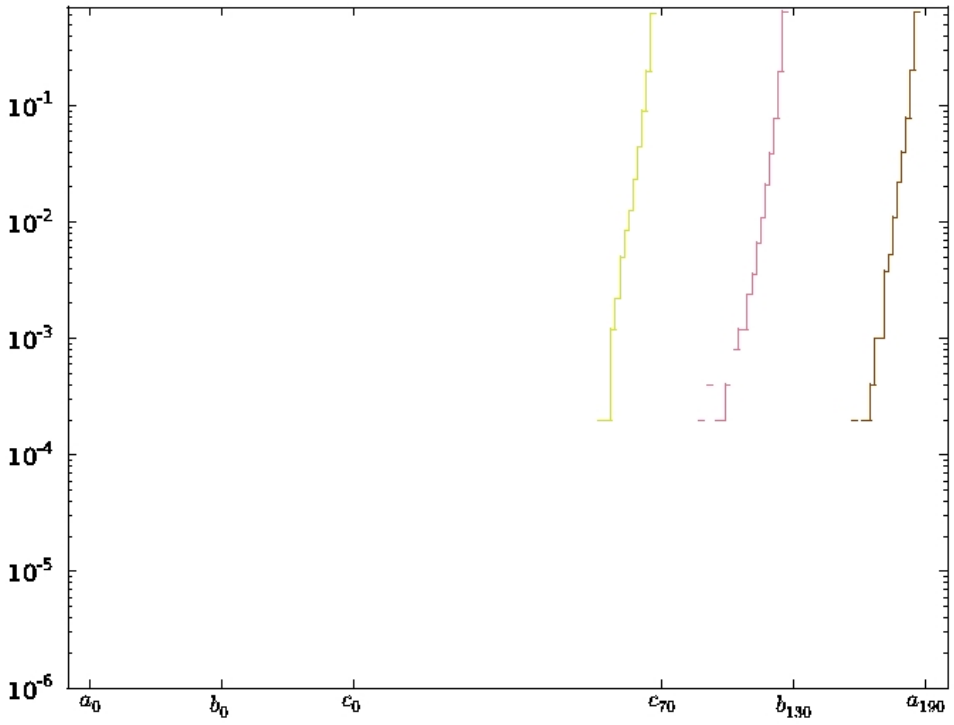

Figure 7.66: Empirical deflation index distributions of QR algorithm for uniform doubly stochastic Jacobi matrix initial data with $\epsilon = 10^{-8}$ and matrix dimensions $70, 130, 190$. Hit index distributions are centered as in 7.63 with same meaning of a_i, b_i, c_i.

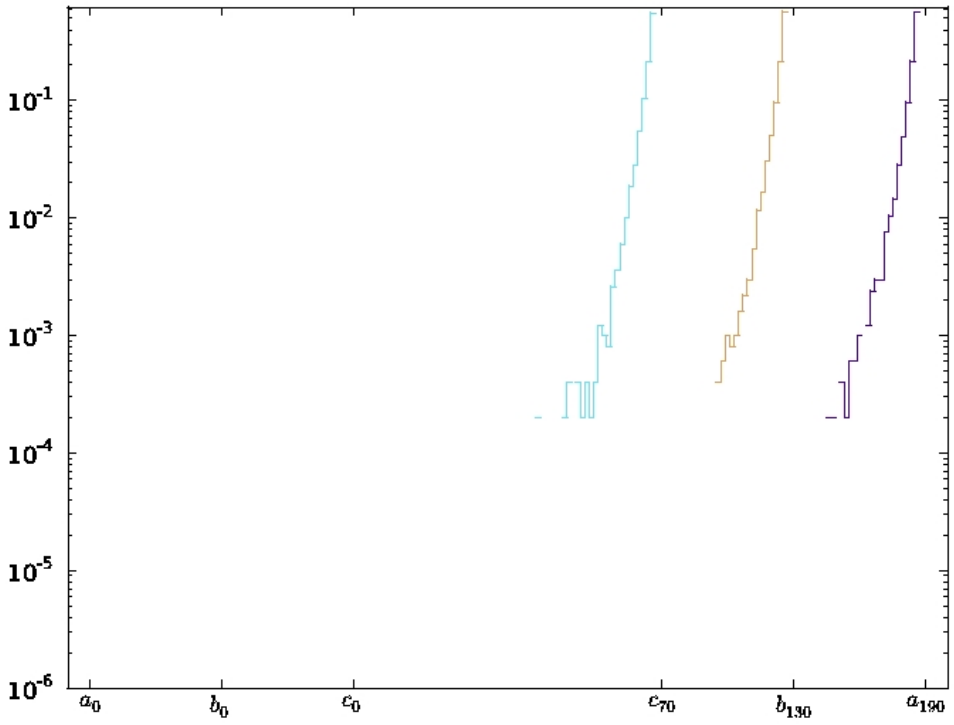

Figure 7.67: Empirical deflation index distributions of QR algorithm for Jacobi matrix with uniform eigenvalues initial data with $\epsilon = 10^{-8}$ and matrix dimensions $70, 130, 190$. Hit index distributions are centered as in 7.63 with same meaning of a_i, b_i, c_i.

7.3.2 Deflation Position for the Toda algorithm

The figures 7.68 – 7.71 present the empirical results for the distribution of the deflation position when the Toda algorithm is applied to the four Wigner–type initial distributions under discussion. Again, the shapes of these distributions are similar accross various matrix dimensions, initial distributions (and even deflation tolerances, although here we only present plots for $\epsilon = 10^{-8}$): Deflations occur typically close to the upper and lower end of the matrices – this is a first difference to the deflation behavior of the QR algorithm. Note that for Hermite–1 initial data (figure 7.68) deflations seem to be more likely to occur at the top end of the matrix (small subdiagonal indices) than at the bottom end. In the case of full symmetric matrix ensembles the distribution seems to be more symmetric. Another difference lies in the fact that deflations occur not only at the extreme positions of the subdiagonal but also at positions towards the center. In contrast, we conclude from our study of the QR algorithms that deflation at such 'central' positions is much more unlikely in that case than in the Toda case.

The deflation positions for non Wigner–type initial matrices are represented in figures 7.72 and 7.73. We can clearly see the contrast in both cases to the Wigner–type initial data. While in the case of uniform doubly stochastic Jacobi matrices deflations seem to take place exclusively at the bottom of the matrix, the deflation events are spread much more 'uniformly' across the subdiagonal indices for Jacobi matrices with uniform eigenvalue initial data.

Lastly we conduct a pictorial analysis of the behavior of the runtime of the Toda algorithm in the Wigner–type initial data case when conditioned on deflation occuring towards the top or the bottom of the matrix. To do this we present scatter plots of the deflation index against the runtime of the Toda algorithm for a large dimension n and the deflation tolerance $\epsilon = 10^{-8}$, along with the corresponding (conditional

run time histograms). First we consider Hermite–1 initial data ($n = 170$). In figure 7.74 we can see that in our experiments deflation occurs at the upper end (small subdiagonal indices) of the matrix before it starts occuring at the lower end of the matrix. This relationship can reverse for Hermite–1 initial data and larger tolerance values (cf. figure 7.77). To underline this point we have included figures 7.75 and 7.76 (for $\epsilon = 10^{-8}$) as well as 7.78 and 7.79 (for $\epsilon = 10^{-2}$). For these figures we first fixed a tolerance and then separated our We have centered the hit index distributions for better visibility so that the peak at a_0 (b_0, c_0) corresponds to the peak at a_{190} (b_{130}), c_{70} respectively). The subindices correspond to indices on the matrix subdiagonal. Note that the deflation index by definition takes values from zero to $n - 2$.samples according to whether the first deflation occured above or below the middle of the matrix. For these groups we then computed the runtime histograms separately.

This behavior is markedly different from the behavior of the Toda algorithm on full symmetric matrices. For these matrices we present figure 7.80 as a representative of their deflation/ runtime behavior: The effect that deflations at the lower end of the matrix start occuring at earlier times than at the upper end is less pronounced for full symmetric matrices. Overall the scatter plots look more 'symmetric' across all our experiments with full symmetric matrices than for the ones with Hermite–1 initial data. This symmetry is also supported by conditional histograms 7.81 and 7.82, which were created as in the Jacobi matrix case.

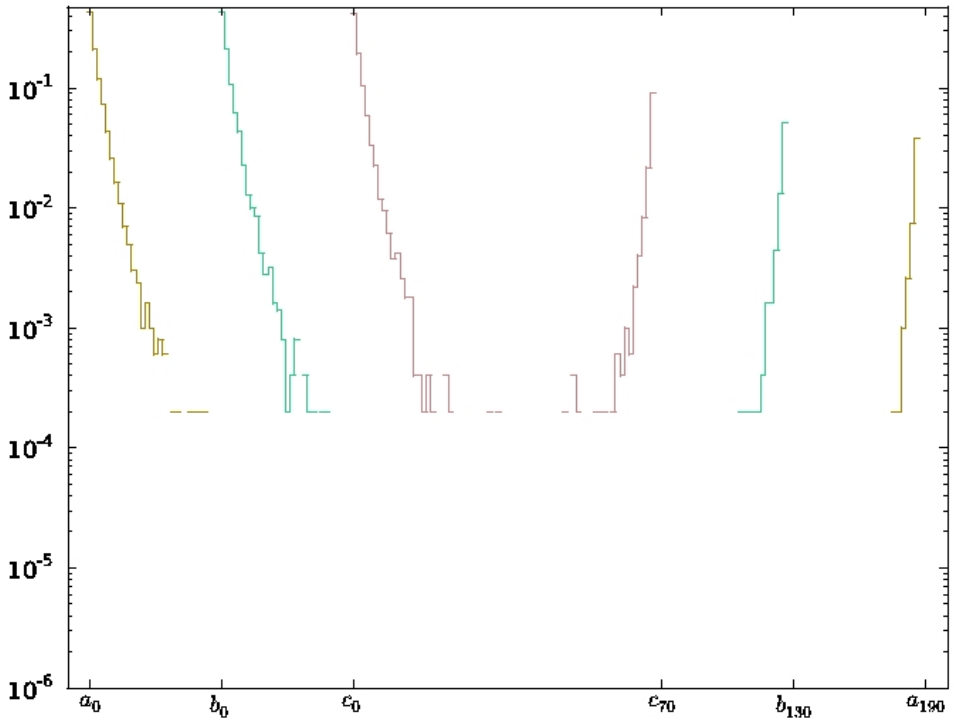

Figure 7.68: Empirical deflation index distributions of Toda algorithm for Hermite–1 initial data with $\epsilon = 10^{-8}$ and matrix dimensions $n = 70, 130, 190$. Hit index distributions are centered as in 7.63 with same meaning of a_i, b_i, c_i.

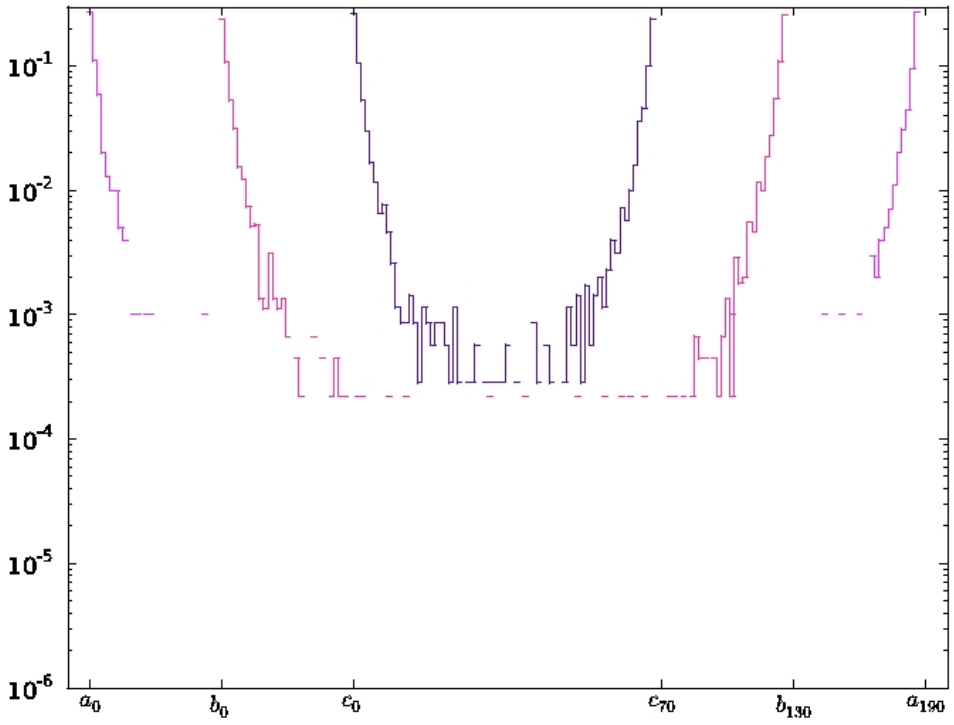

Figure 7.69: Empirical deflation index distributions of Toda algorithm for GOE initial data with $\epsilon = 10^{-8}$ and matrix dimensions $n = 70, 130, 190$. Hit index distributions are centered as in 7.63 with same meaning of a_i, b_i, c_i.

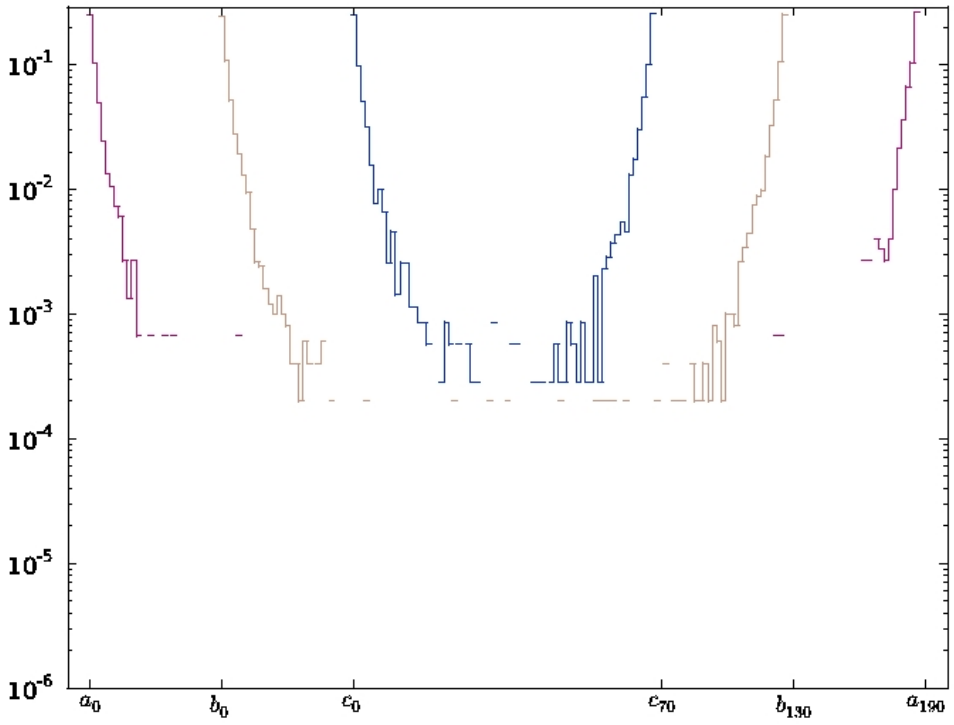

Figure 7.70: Empirical deflation index distributions of Toda algorithm for Wigner initial data with $\epsilon = 10^{-8}$ and matrix dimensions $n = 70, 130, 190$. Hit index distributions are centered as in 7.63 with same meaning of a_i, b_i, c_i.

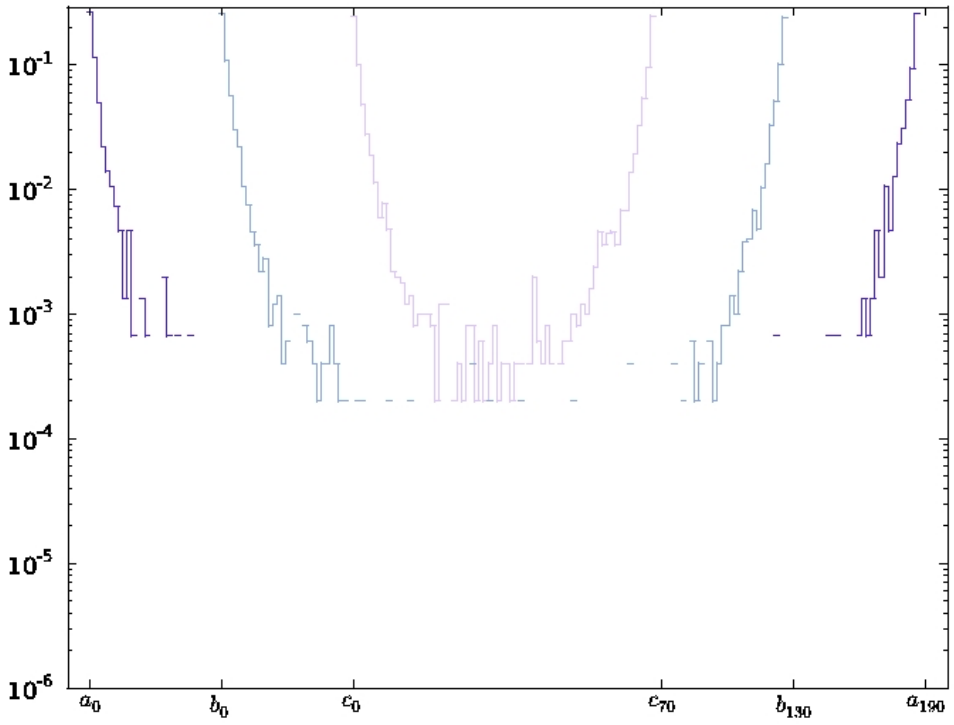

Figure 7.71: Empirical deflation index distributions of Toda algorithm for Bernoulli initial data with $\epsilon = 10^{-8}$ and matrix dimensions $n = 70, 130, 190$. Hit index distributions are centered as in 7.63 with same meaning of a_i, b_i, c_i.

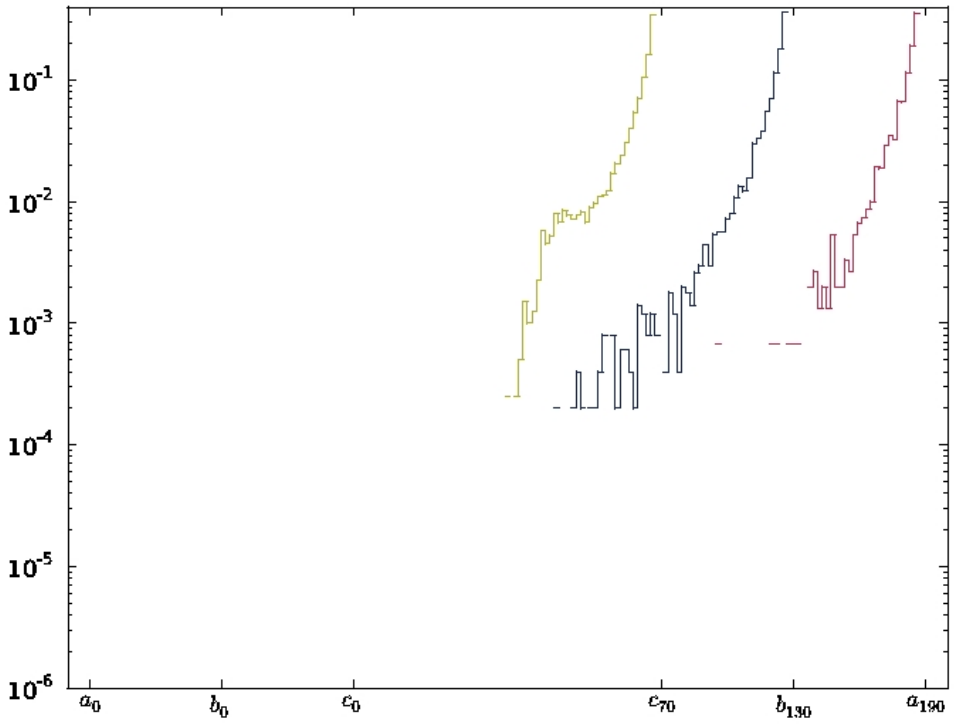

Figure 7.72: Empirical deflation index distributions of Toda algorithm for uniform doubly stochastic Jacobi initial data with $\epsilon = 10^{-8}$ and matrix dimensions $n = 70, 130, 190$. Hit index distributions are centered as in 7.63 with same meaning of a_i, b_i, c_i.

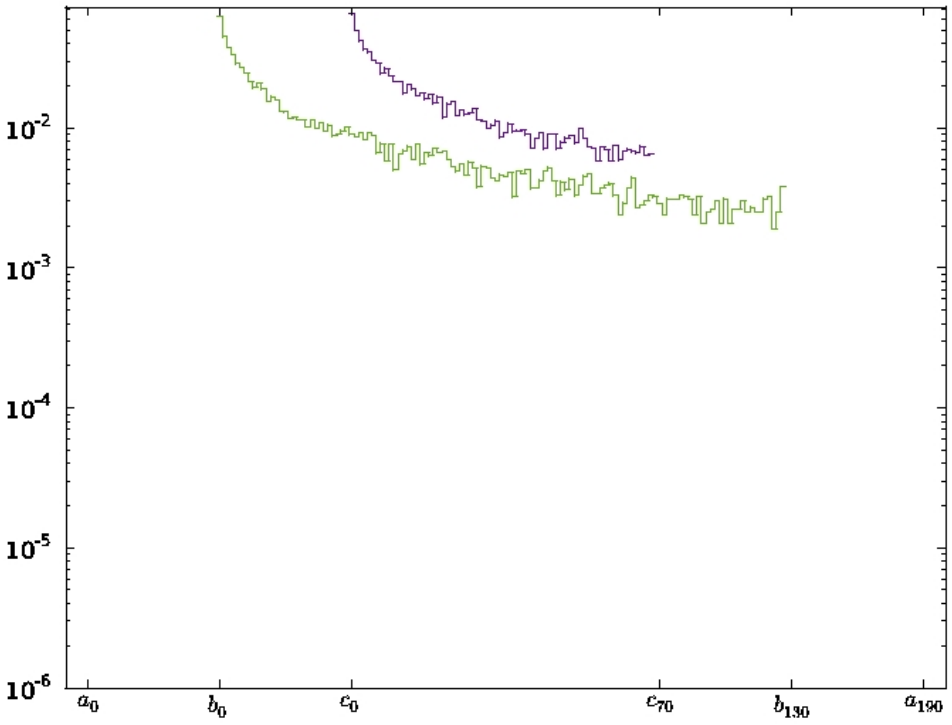

Figure 7.73: Empirical deflation index distributions of Toda algorithm for Bernoulli initial data with $\epsilon = 10^{-8}$ and matrix dimensions $n = 50, 90, 130$. Hit index distributions are centered as in 7.63 with same meaning of a_i, b_i, c_i.

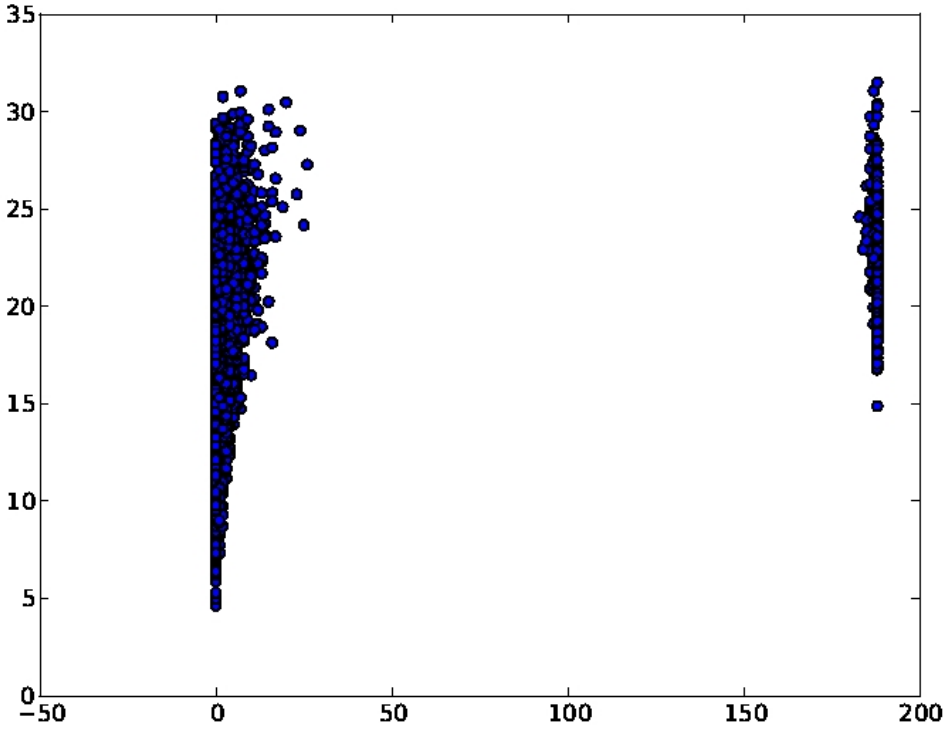

Figure 7.74: Scatter plot of deflation index I_ϵ (on x–axis) agains T_ϵ (on y–axis) for $\epsilon = 10^{-8}$ and $n = 190$ with Hermite–1 initial data.

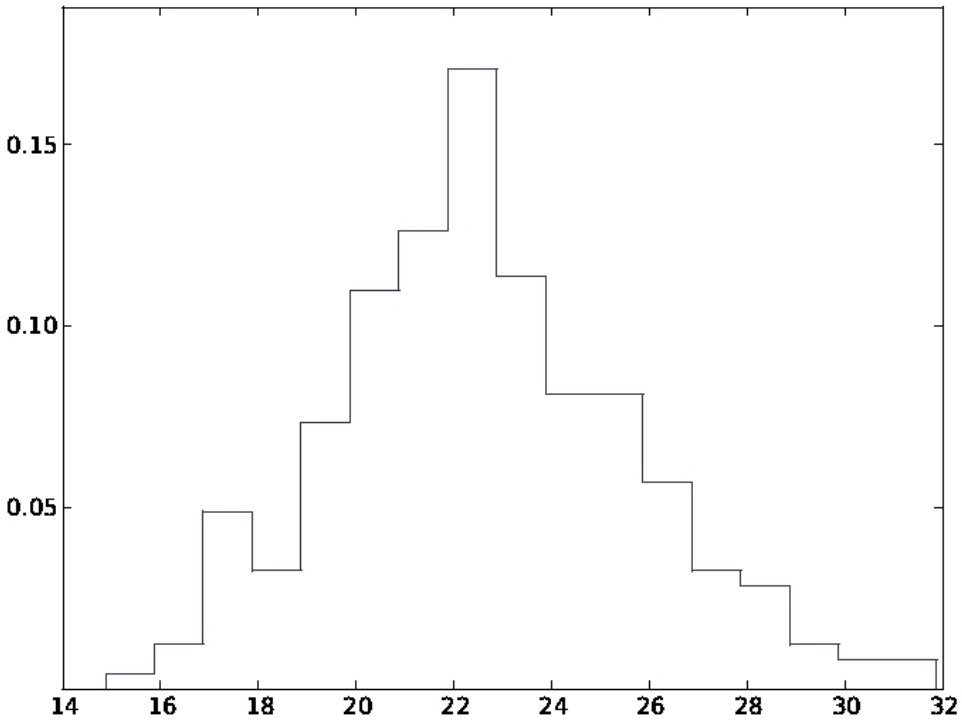

Figure 7.75: 'Conditional' histogram of all the deflation time T_ϵ samples where deflation occured in the bottom half of the matrix $\epsilon = 10^{-8}$ and $n = 190$ with Hermite–1 initial data.

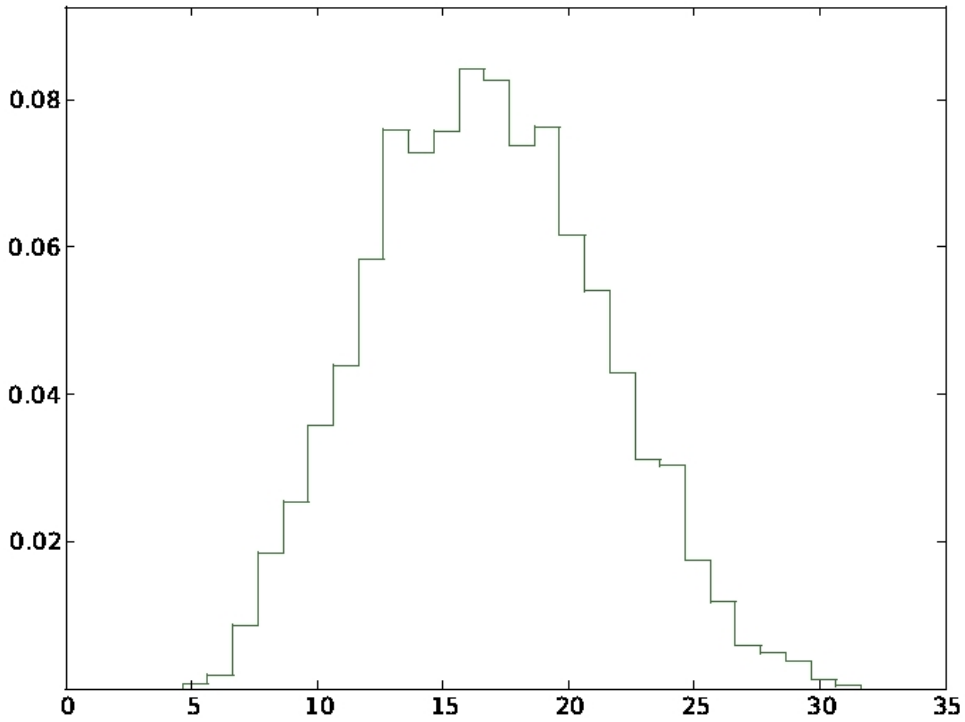

Figure 7.76: 'Conditional' histogram of all the deflation time T_ϵ samples where deflation occured in the top half of the matrix $\epsilon = 10^{-8}$ and $n = 190$ with Hermite–1 initial data.

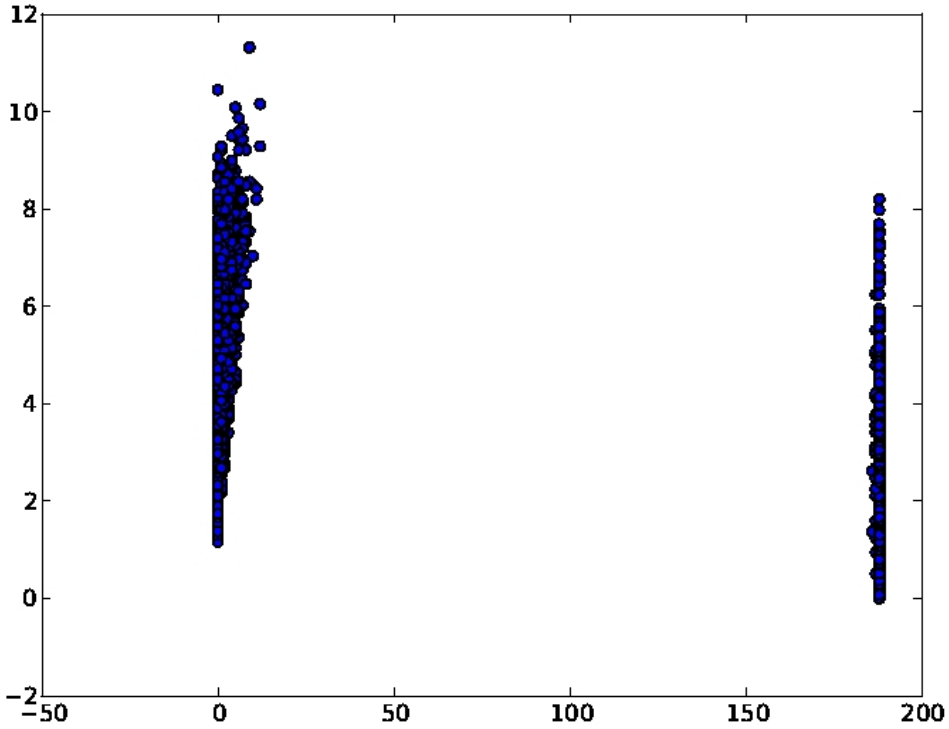

Figure 7.77: Scatter plot of deflation index I_ϵ (on x–axis) against T_ϵ (on y–axis) for $\epsilon = 10^{-2}$ and $n = 190$ with Hermite–1 initial data.

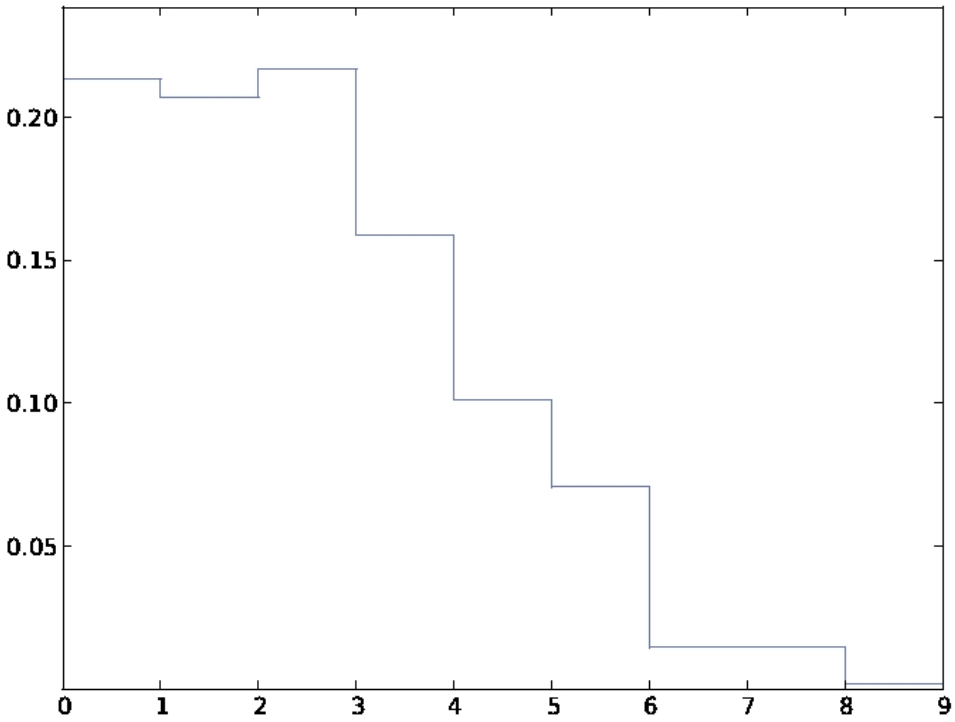

Figure 7.78: 'Conditional' histogram of all the deflation time T_ϵ samples where deflation occured in the bottom half of the matrix $\epsilon = 10^{-2}$ and $n = 190$ with Hermite–1 initial data.

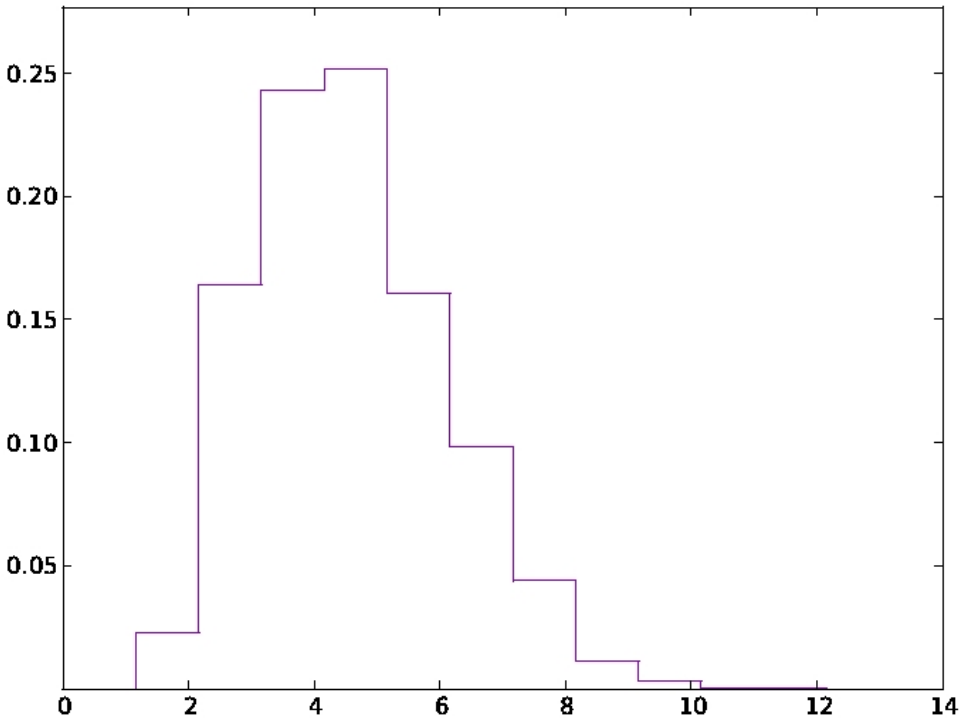

Figure 7.79: 'Conditional' histogram of all the deflation time T_ϵ samples where deflation occured in the top half of the matrix $\epsilon = 10^{-2}$ and $n = 170$ with Hermite–1 initial data.

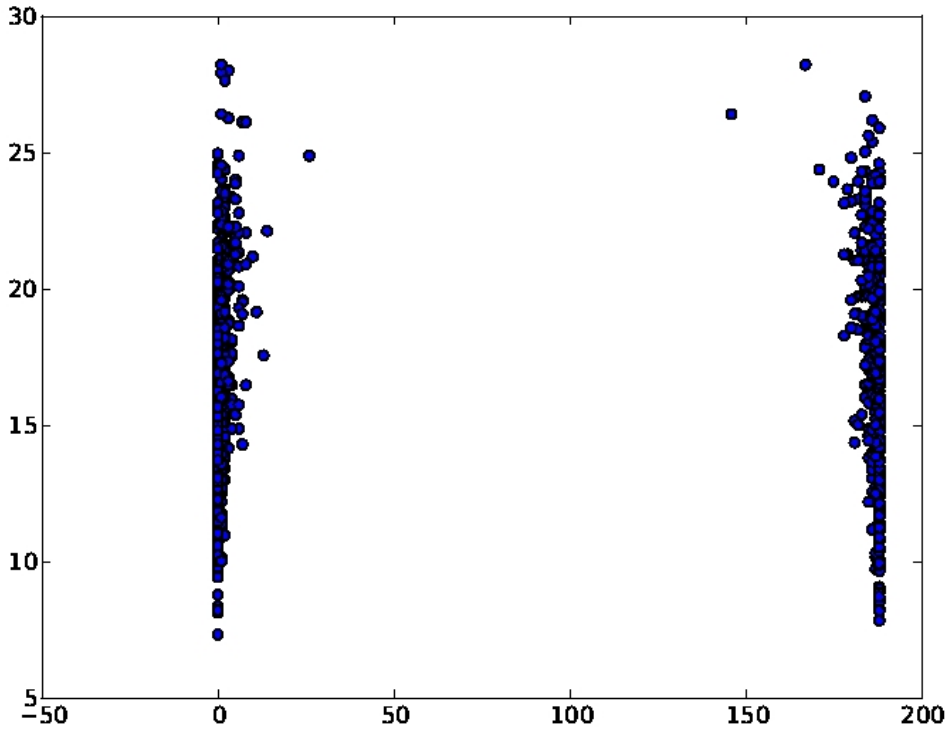

Figure 7.80: Scatter plot of deflation index I_ϵ (on x–axis) againts T_ϵ (on y–axis) for $\epsilon = 10^{-8}$ and $n = 190$ with GOE initial data.

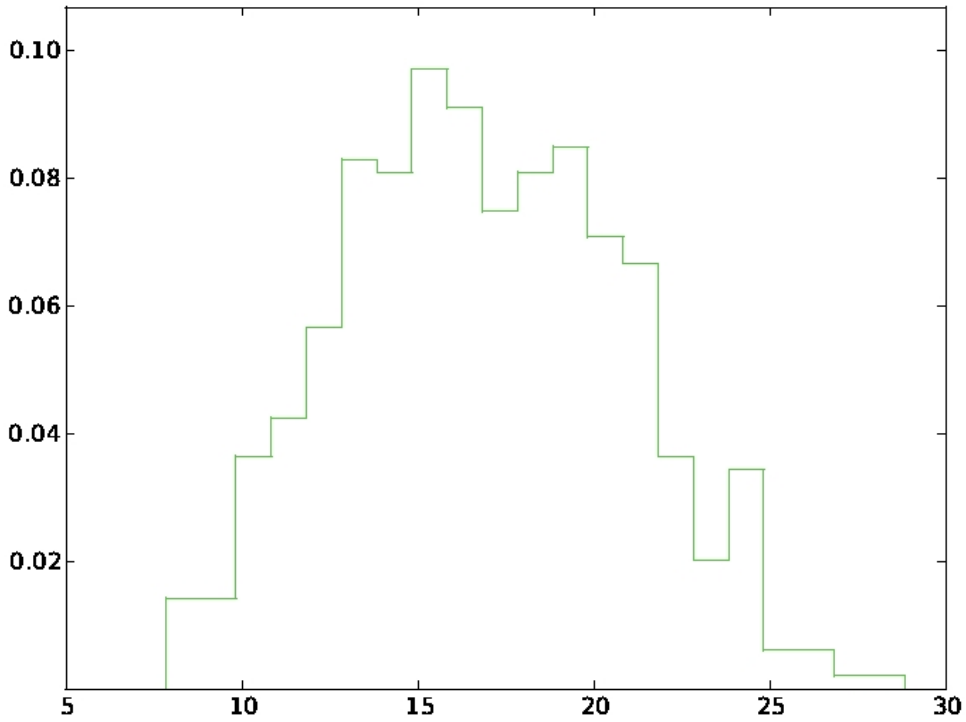

Figure 7.81: 'Conditional' histogram of all the deflation time T_ϵ samples where deflation occured in the bottom half of the matrix $\epsilon = 10^{-8}$ and $n = 190$ with GOE initial data.

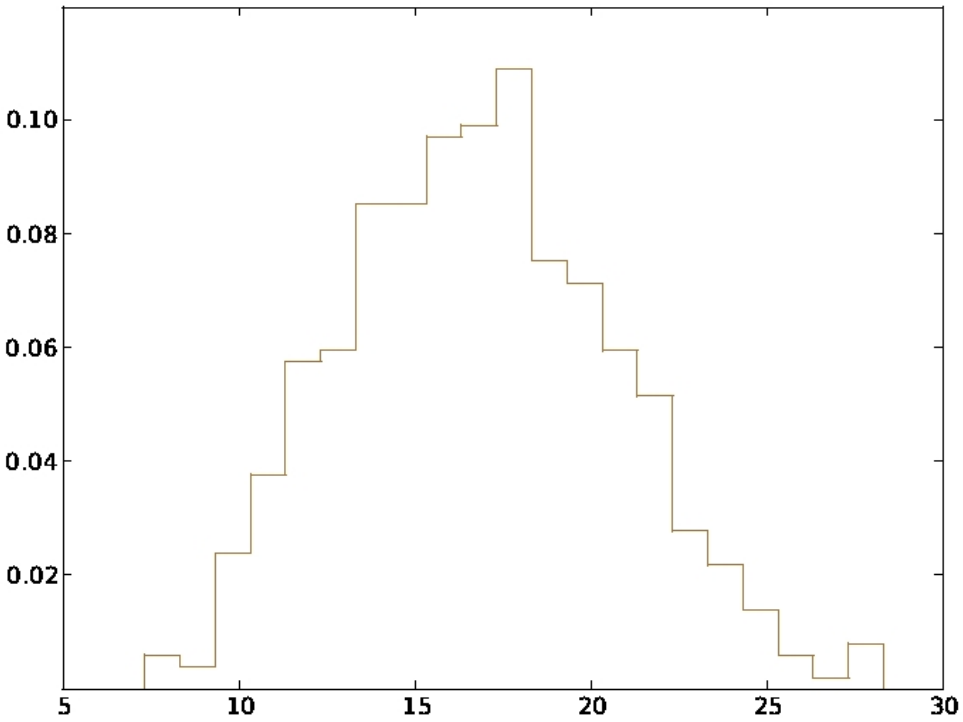

Figure 7.82: 'Conditional' histogram of all the deflation time T_ϵ samples where deflation occured in the top half of the matrix $\epsilon = 10^{-8}$ and $n = 190$ with GOE initial data.

7.3.3 Splitting the Atom

Divide and conquer is a time tested strategy in computer science for creating efficient algorithms in many areas. This general technique has been shown to be useful in solving dense eigenvalue problems in [4]: The idea is to find a procedure that reduces a matrix in an isospectral fashion to a near block matrix and then to apply the same procedure recursively to the blocks that were found in this way. It is interesting in this context that we have found a Hamiltonian that can be used in the DLNT framework which leads to a distribution of the deflation position that is concentrated around the center of the matrix. This is very desirable, because the divide and conquer strategy heuristically leads to the largest speed ups when the problem is divided into subproblems of the same size. This Hamiltonian is given by:

$$H(L) := \mathrm{tr}|L| \tag{7.22}$$

ie. the sum of the absolute values of the eigenvalues of the Jacobi matrix L. This implies by theorem 5.1 that the first row q of the eigenvector matrix evolves as

$$q(t) = \frac{e^{t\cdot\mathrm{sign}(\Lambda)}q_0}{\|e^{t\cdot\mathrm{sign}(\Lambda)}q_0\|} \tag{7.23}$$

We can thus use the `TodaExplicit` implementation strategy from section 6.1 to test our claim. We provide several figures to illuminate the properties of this algorithm: Firstly, to illustrate our claim that the Hamiltonian 7.22 leads to deflations in the middle of the matrix we ask to refer to figure 7.83. Here we provide plots of the empirical deflation position distribution for deflation tolerance $\epsilon = 10^{-8}$ and three different matrix sizes ($n = 90, 130, 170$) in one figure. It is apparent, that for each of the dimensions n the deflation position concentrates around $n/2$ as claimed. Next, we present runtime histograms for two different tolerance values: Figure 7.84 for $\epsilon =$

Table 7.9: Regression Parameters for the means and Standard deviations of the runtime distribution corresponding to the 'splitting the atom' algorithm with Hermite–1 initial data

Statistic	a_0	a_1	a_2
Mean	0.6032	0.5411	-0.5045
	b_0	b_1	b_2
Standard deviation	1.854263	0.007012	0.033165

10^{-2} and figure 7.85 for $\epsilon = 10^{-8}$. Note that for the relatively large tolerance value $\epsilon = 10^{-2}$ all of the matrix dimensions show a nontrivial 'immediate' deflation at the very start of the algorithm, which does not occur for the more realistic tolerance $\epsilon = 10^{-8}$. For an explanation of this effect we refer to the discussion of the distribution of the minimum of the subdiagonal entries of the Hermite–1 distribution in section 3.2. The apparent regularity of this runtime behavior is also represented in the figures 7.86 and 7.87. The former (latter) figure shows the dependence of the empirical runtime averages (standard deviations) on the deflation tolerance and the matrix sizes. It is quite interesting that the dependence on the means seems perfectly linear for fixed tolerance ϵ. This finding motivates us to estimate a linear regression of the form

$$\mu_{n,\epsilon} \approx a_0 + a_1 n + a_2 \log \epsilon \tag{7.24}$$

$$\sigma_{n,\epsilon} \approx b_0 + b_1 n + b_2 \log \epsilon \tag{7.25}$$

for values of $\epsilon < .01$. The results of this regression can be found in table 7.9. In the figure 7.87 the individual lines correspond the tolerance values $\epsilon = 10{-}k$, $k = 2, 4, 6, 8$ from top to bottom. The irregular shape of the topmost graph is explained by the 'residual' early deflation for large tolerances that can be seen in the lower left corner of figure 7.84. The difference in the graphs for the two smallest tolerances is too small to be noticeable in the figure.

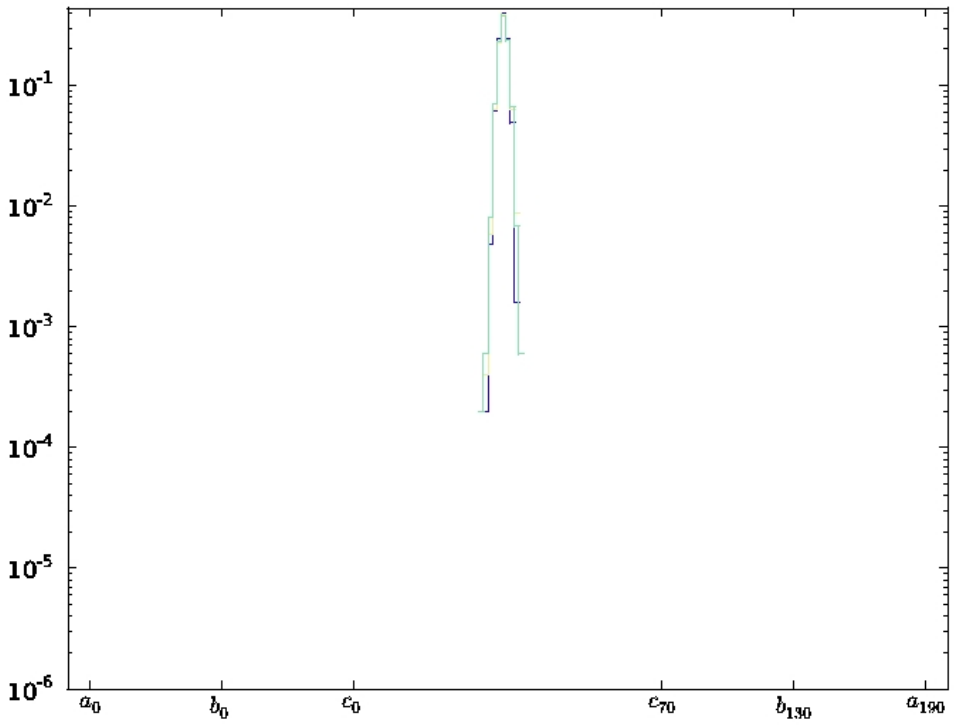

Figure 7.83: Empirical distribution of deflation positions for Hermite–1 initial data under the DLNT algorithm resulting from the Hamiltonian (7.22) on a log scale. Deflation tolerance was chosen as $\epsilon = 10^{-8}$, matrix dimension $n = 90, 130, 170$ and we used 2000 samples for each dimension to create this figure.

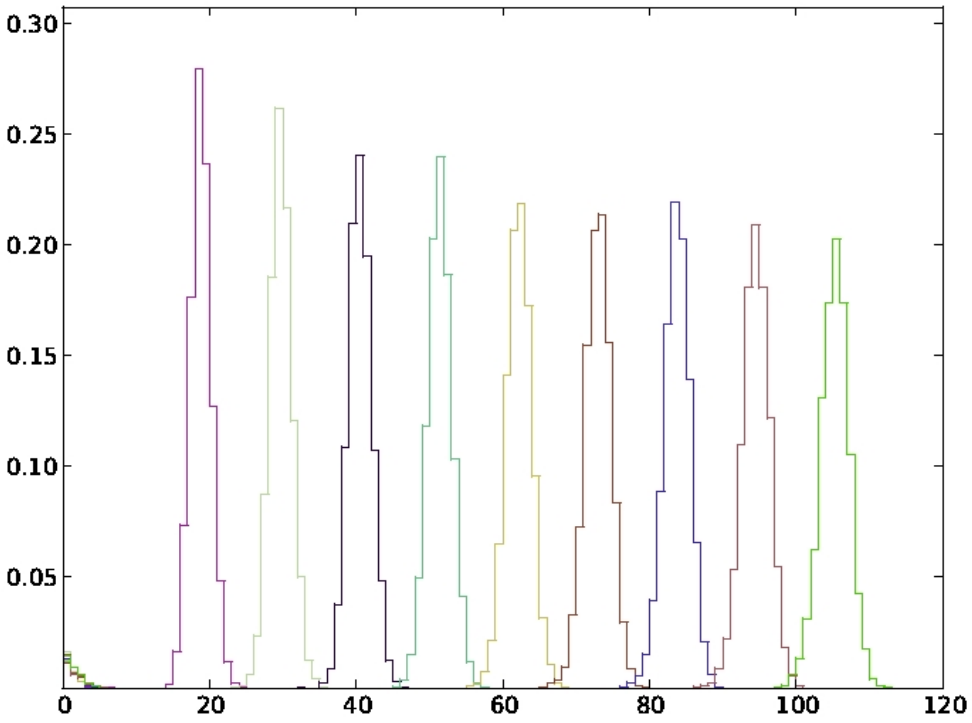

Figure 7.84: Histogram of deflation times for Hermite–1 initial data under the DLNT algorithm resulting from the Hamiltonian (7.22). Deflation tolerance was chosen as $\epsilon = 10^{-2}$, matrix dimension $n = 90, 110, 130, 150, 170$ and we used 2000 samples for each dimension to create this figure. The dimensions correspond to the histograms in order from left to right. Note the 'residual' early deflation at the lower left.

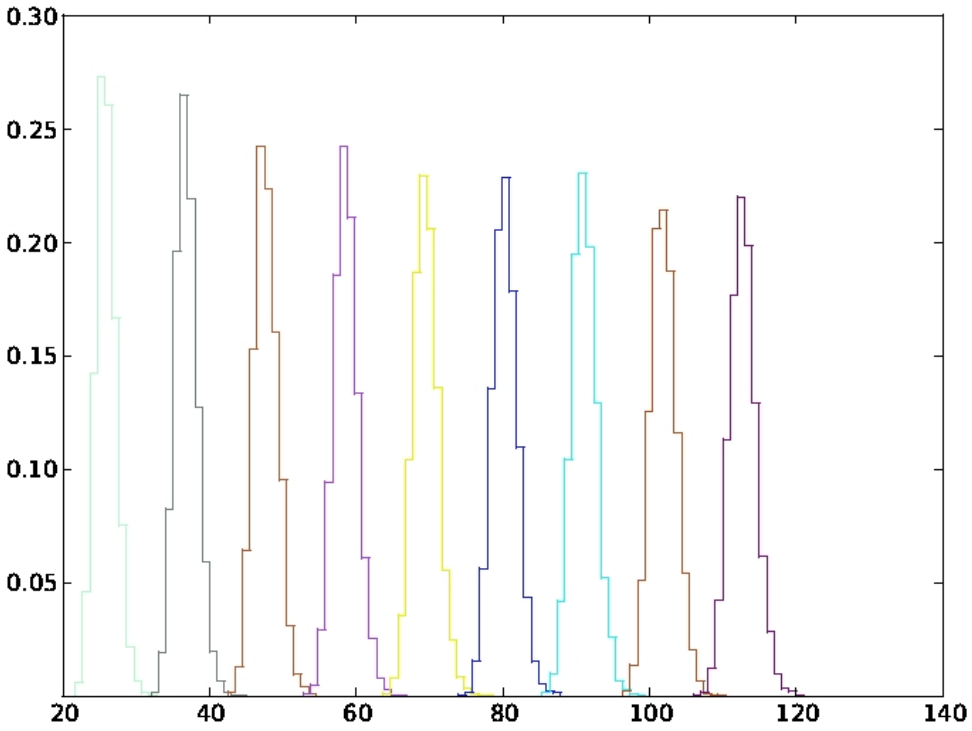

Figure 7.85: Histogram of deflation times for Hermite–1 initial data under the DLNT algorithm resulting from the Hamiltonian (7.22). Deflation tolerance was chosen as $\epsilon = 10^{-8}$, matrix dimension $n = 90, 110, 130, 150, 170$ and we used 2000 samples for each dimension to create this figure. The dimensions correspond to the histograms in order from left to right. Note the absence of 'residual' early deflation.

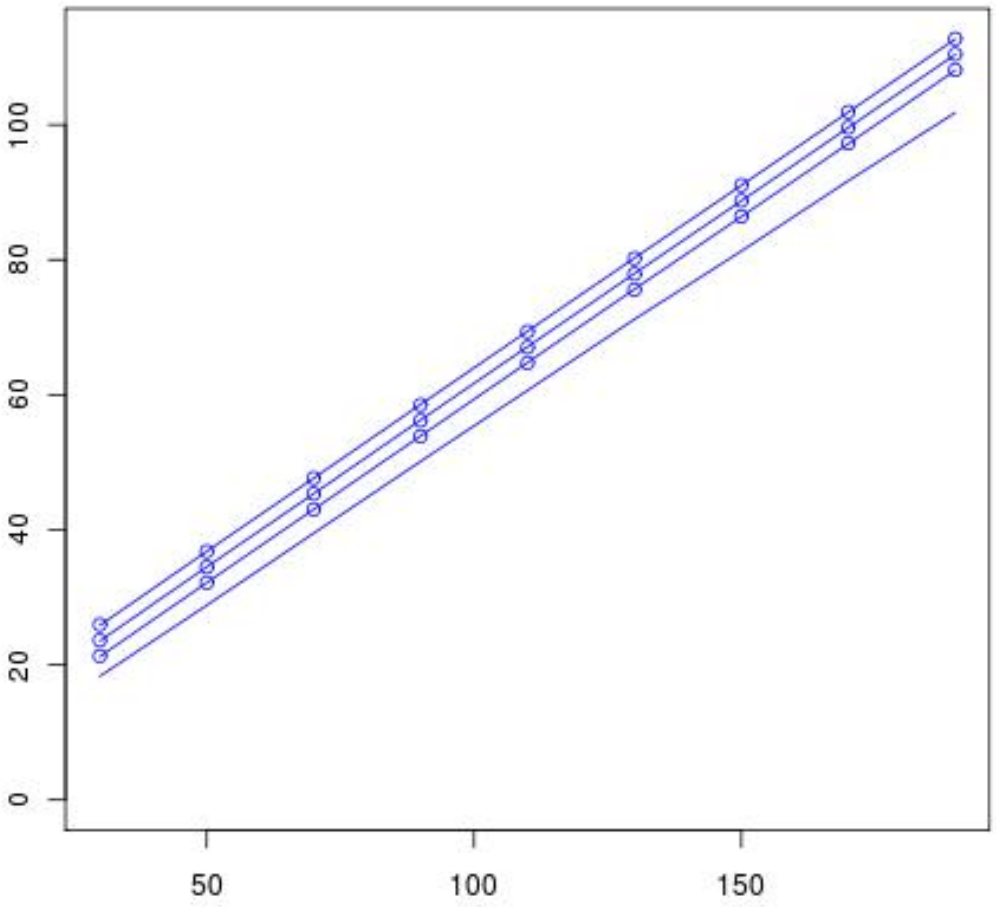

Figure 7.86: Means of the runtime histograms of the DLNT algorithm corresponding to equation (7.22). Each line corresponds to one of the deflation tolerances (from bottom to top) $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$. Dots correspond to regression predictions for $\epsilon < .01$.

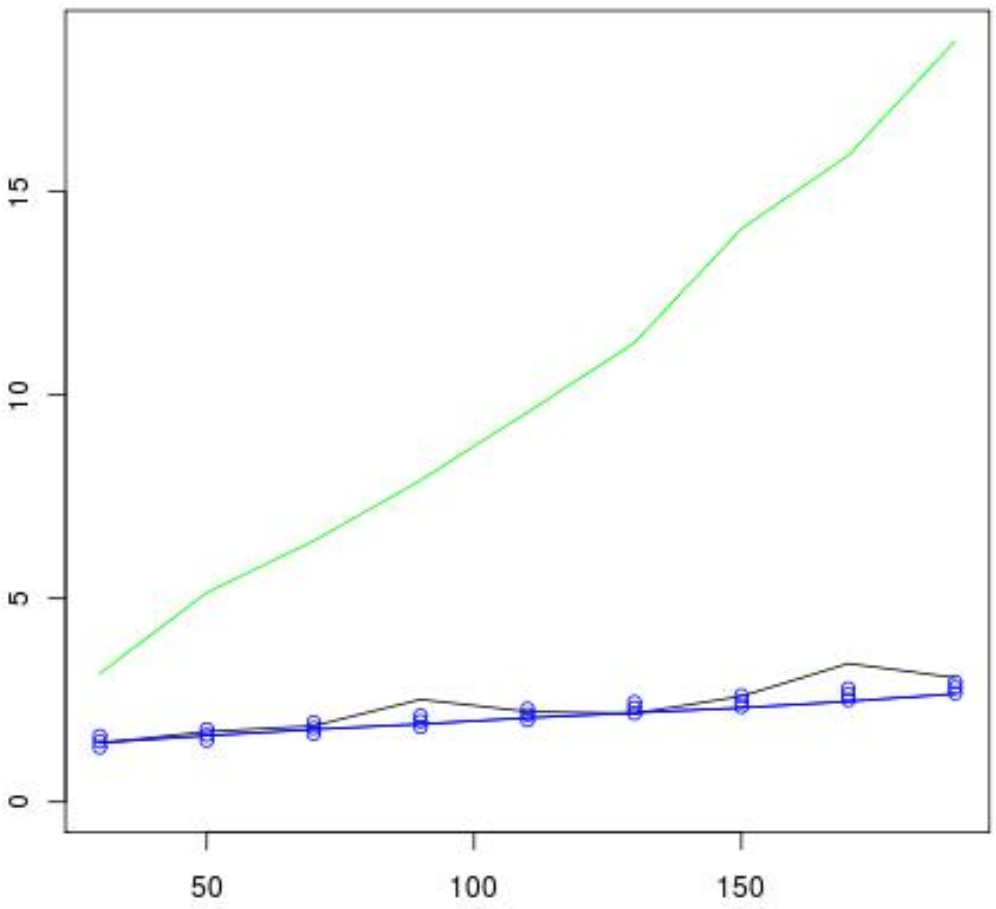

Figure 7.87: Standard deviations of the runtime histograms of the DLNT algorithm corresponding to equation (7.22). Each line corresponds to one of the deflation tolerances (from top to bottom) $\epsilon = 10^{-k}$, $k = 2, 4, 6, 8$. The lines for the two smallest standard deviations overlap and the comparatively large standard deviation corresponding to the tolerance $\epsilon = 10^{-2}$ is due to the residual early deflation visible in 7.84. Dots correspond to regression predictions for $\epsilon < .01$.

7.4 Statistical Analysis of the Runtime Distribution Tails

In this section we go back to the figures of type 2 for the QR (figures 7.14, 7.15, 7.16, 7.17) and the Toda (figures 7.44, 7.45, 7.46, 7.47). Together with the combined plots (figures 7.29 and 7.59) these figures have led us to the informal conclusion that there is a univeral structure in the distribution of the runtime which depends on the algorithm (cf. sections 7.2.1 and 7.2.2). In this section we will try to provide statistical evidence underlining its validity. In particular we will provide evidence for the claim that the right tails for the runtime distributions tend to Gaussian tails for all distributions sampled from the Toda algorithm and to exponential tails for the QR algorithm. Estimating the tails of empirical distributions and validating those estimates is an often encountered problem in many areas of science. However, it is not straightforward to approach the problem in a principled way. The reference [8] provides a framework for approaching this estimation/ validation problem which we adapt to our circumstances. In particular it is suggested to use a hypothesis testing approach to analyze whether the observations are consistent with our tail–hypothesis and to rule out competing hypotheses.

We now introduce two families of distributions each for the QR and the Toda runtime distributions: In the QR case, we suspect that the data shows exponential right tails. We will thus estimate the parameters of what we call the the cut off exponential as well as the cut off gamma distribution

$$f_{\lambda,x_{\min}}(x) = Z_f^{-1}(\lambda, x_{\min}) \cdot e^{-\lambda x} \cdot 1_{x>x_{\min}}(x) \tag{7.26}$$

$$\gamma_{k,\theta,x_{\min}}(x) = Z_\gamma^{-1}(k, \theta, x_{\min}) x^{k-1} e^{-x/\theta} \cdot 1_{x>x_{\min}}(x) \tag{7.27}$$

using the data in the right tail of our data to see which of these families explains the data more plausibly. We take a similar approach in the case of the Toda runtime data. Our proposal distributions in this case will be the cut off Gaussian as well as the cut off Weibull distribution with flexible tails:

$$\eta_{\sigma,x_{\min}}(x) = Z_{\eta}^{-1}(\sigma, x_{\min}) e^{-\frac{(x-x_{\min})^2}{2\sigma^2}} \cdot 1_{x>x_{\min}}(x) \tag{7.28}$$

$$w_{k,\lambda,x_{\min}}(x) = Z_{w}^{-1}(k, \lambda, x_{\min}) x^{k-1} e^{-(x/\lambda)^k} \cdot 1_{x>x_{\min}}(x) \tag{7.29}$$

Here the Z terms denote the necessary normalization constants. The following is a summary of our adaptation of the procedure from [8] which we used to analyze the goodness of fit of the respective runtime distributions with the proposed right tails:

1. By inspection, in particular of figures 7.29 and 7.59, we determine that for each histogram the value $x_{\min}$ is set equal to the empirical mean of the corresponding data set. Both figures indicate that the data to the right of this value seem to align well with the respective decay hypotheses.

2. Next, compute numerically maximum likelihood estimates of the distribution parameters using the runtime data, along with an estimate of the variance of the estimates.

3. Compute a slightly modified Kolmogorov–Smirnov statistic KS^* for the runtime data D and its proposed distribution with the maximum likelihood parameters. We denote the distribution function of the latter distribution by F and the empirical distribution function derived from the right tail data by F_D. In particular:

$$\mathrm{KS}^* = \sup_{t>x_{\min}} |F_D(t) - F(t)| \tag{7.30}$$

4. Use a semiparametric approach to generate new data sets D_i: Here data on the left tail is (re–)sampled from the runtime data and data on the right tail is sampled from the distribution with the proposed tail (ie. a Gaussian in the case of Toda and a Gamma in the case of QR). Use the newly sampled data to infer maximum likelihood estimates for the parameters of the relevant distribution family, compute KS^* for the given data set.

5. This procedure yields samples $\mathrm{KS}^*(D_i)$ which we can compare to $\mathrm{KS}^*(D)$. Intuitively, if enough of the Kolmogorov-Smirnov values generated from the artificial data sets are larger than $\mathrm{KS}^*(D)$ from our actual runtime data, then the right tail of the empirical distribution based on D is not atypically far away from the proposed (Gaussian or exponential) tail. In [8] the authors suggest to reject the hypothesis of a fit at the right tail, if the fraction of $\mathrm{KS}^*(D_i)$ that is larger than $\mathrm{KS}^*(D)$ is less than ten percent. We thus reject if 'there is a probability of one in ten or less that we would merely by chance get data that agree as poorly with the model as the data we have' [8, p. 677].

6. We then compare our models with respect to their goodness of fit with the data. This will be already enough to choose between the two models for each algorithm so that an additional likelihood ratio test is not necessary.

This strategy was implemented in the statistical computing language R [46]. Refer to figure 7.88 for a pictorial summary of the validation procedure for an artificial example created from sampling uniformly in the interval from -5 to 0 and sampling from χ_1. The mean and the standard deviation of the cutoff Gaussian distribution model are estimated as the empirical mean of the data and the empirical standard deviation of the data to the right of the empirical mean, amended by its reflection to the left of the mean. Based on these parameters we can compute the modified KS–statistic to the right of the data–mean. In the second row an artificial dataset

is created (ticks on the $x - -axis$) by resampling from the part below the empirical mean of the data set and sampling from the estimated 'one–sided' Gaussian distribution to the right of the empirical mean. Now we reestimate the parameters (mean and standard deviation) for this dataset and compute the respective modified KS–statistic. Generating many artificial data sets allows for a comparison of the artificial modified KS–statistics with the one derived from the real dataset and thus the computation of an empirical p–value.

We summarize all the information arising from the statistical analysis in tables 7.10 to 7.17 (for QR runtimes on Wigner class initial data) as well as 7.18 to 7.25 (for Toda runtimes on Wigner class initial data). In these tables we list the matrix dimension, the deflation tolerance value and the number N of right tail samples on which the estimates are based. Next we give the estimate itself along with a value I_κ for the parameters κ. This is the coefficient of the inverse of the Hessian corresponding to the second derivative of the likelihood function with respect to the parameter κ. The inverse of the Hessian is an approximation to the Fisher information matrix, so that I_κ/N is an approximation to the variance of the maximum likelihood estimator if asymptotic normality were to hold for the current size of the data. We then present the values of KS^* as well as the 'Monte–Carlo p–value' that we determined by the above description. Note that these tables omit the results for the runtime experiments with deflation tolerance equal to .01. The runtime distributions in these cases can show features very different from the ones at smaller deflation tolerances which makes them less valuable for being fitted to the families we discussed.

It emerges from the summary tables that – as expected – the fit with our distribution families increases with increasing dimension and decreasing deflation tolerance. In all of the tables we are more likely to find higher KS p–values in the top section. It turns out that the goodness of fit of individual runtime distributions is better for the cutoff exponential model than for the cutoff gamma model in the case of QR

runtimes and the goodness of fit for the cut off Gaussian distribution is much better than the goodness of fit for the cutoff Weibull distribution. Note however, that except for the cases of smallest deflation tolerance, we would typically reject even the cutoff exponential model for the studied initial data. This situation is very different from the cut off Gaussian model which can't be rejected (with very high p–values) under almost all regimes and which we thus consider a very robust model for the right tails of our runtime data.

Table 7.10: Summary of statistical results for QR runtime data of Hermite–1 initial matrices under cutoff exponential model.

Dim.	Tol.	N	Rate	I_{Rate}	KS*	KS p-value
30	1.0E-08	1998	0.09	4.00E-06	0.02	0.46
50	1.0E-08	2098	0.09	4.00E-06	0.05	0.00
70	1.0E-08	2149	0.10	4.00E-06	0.06	0.00
90	1.0E-08	2054	0.09	4.00E-06	0.04	0.00
110	1.0E-08	2077	0.09	4.00E-06	0.03	0.04
130	1.0E-08	2049	0.09	4.00E-06	0.02	0.19
150	1.0E-08	2042	0.09	4.00E-06	0.03	0.10
170	1.0E-08	2008	0.09	4.00E-06	0.02	0.18
190	1.0E-08	2015	0.09	4.00E-06	0.02	0.25
30	1.0E-06	1996	0.11	7.00E-06	0.02	0.25
50	1.0E-06	2131	0.13	8.00E-06	0.09	0.00
70	1.0E-06	2199	0.13	8.00E-06	0.10	0.00
90	1.0E-06	2069	0.12	7.00E-06	0.07	0.00
110	1.0E-06	2101	0.13	8.00E-06	0.06	0.00
130	1.0E-06	2068	0.13	8.00E-06	0.04	0.02
150	1.0E-06	2060	0.12	8.00E-06	0.05	0.00
170	1.0E-06	2024	0.12	8.00E-06	0.03	0.09
190	1.0E-06	2046	0.13	8.00E-06	0.03	0.02
30	1.0E-04	1969	0.17	1.40E-05	0.03	0.03
50	1.0E-04	1881	0.17	1.50E-05	0.03	0.34
70	1.0E-04	1942	0.17	1.50E-05	0.04	0.10
90	1.0E-04	2150	0.19	1.70E-05	0.14	0.00
110	1.0E-04	2163	0.20	1.80E-05	0.11	0.00
130	1.0E-04	2107	0.19	1.80E-05	0.09	0.00
150	1.0E-04	2094	0.19	1.70E-05	0.10	0.00
170	1.0E-04	2084	0.19	1.80E-05	0.08	0.00
190	1.0E-04	2085	0.19	1.80E-05	0.08	0.00

Table 7.11: Summary of statistical results for QR runtime data of Hermite–1 initial matrices under cutoff gamma model.

Dim.	Tol.	N	Shape	Scale	I_{Shape}	I_{Shape}	KS*	KS p-value
30	1.0E-08	1998	3.90	6.01	1.89E-01	1.99E-01	0.04	0.37
50	1.0E-08	2098	2.19	7.89	1.51E-01	4.82E-01	0.06	0.00
70	1.0E-08	2149	1.82	8.43	1.47E-01	6.07E-01	0.06	0.00
90	1.0E-08	2054	2.32	7.79	1.51E-01	4.55E-01	0.04	0.00
110	1.0E-08	2077	3.52	6.06	1.84E-01	2.17E-01	0.05	0.00
130	1.0E-08	2049	3.32	6.27	1.78E-01	2.41E-01	0.04	0.03
150	1.0E-08	2042	2.83	6.93	1.66E-01	3.25E-01	0.04	0.01
170	1.0E-08	2008	3.95	5.69	1.91E-01	1.77E-01	0.05	0.03
190	1.0E-08	2015	3.54	6.07	1.81E-01	2.17E-01	0.04	0.13
30	1.0E-06	1996	3.67	4.76	1.81E-01	1.30E-01	0.04	0.03
50	1.0E-06	2131	1.42	7.05	1.34E-01	4.69E-01	0.09	0.00
70	1.0E-06	2199	0.96	7.76	1.29E-01	6.57E-01	0.10	0.00
90	1.0E-06	2069	1.82	6.56	1.39E-01	3.64E-01	0.07	0.00
110	1.0E-06	2101	2.99	4.97	1.73E-01	1.62E-01	0.07	0.00
130	1.0E-06	2068	2.81	5.16	1.66E-01	1.81E-01	0.05	0.00
150	1.0E-06	2060	2.40	5.64	1.56E-01	2.36E-01	0.06	0.00
170	1.0E-06	2024	3.66	4.44	1.85E-01	1.14E-01	0.05	0.00
190	1.0E-06	2046	2.88	5.07	1.67E-01	1.72E-01	0.05	0.00
30	1.0E-04	1969	3.25	3.54	1.68E-01	7.72E-02	0.05	0.00
50	1.0E-04	1881	4.37	2.89	1.95E-01	4.29E-02	0.07	0.16
70	1.0E-04	1942	4.04	2.99	1.92E-01	4.88E-02	0.07	0.09
90	1.0E-04	2150	0.42	6.25	1.12E-01	5.14E-01	0.14	0.00
110	1.0E-04	2163	1.74	4.18	1.50E-01	1.54E-01	0.11	0.00
130	1.0E-04	2107	1.90	4.10	1.48E-01	1.42E-01	0.09	0.00
150	1.0E-04	2094	1.50	4.58	1.35E-01	1.95E-01	0.11	0.00
170	1.0E-04	2084	2.24	3.79	1.55E-01	1.12E-01	0.09	0.00
190	1.0E-04	2085	1.96	4.02	1.49E-01	1.35E-01	0.09	0.00

Table 7.12: Summary of statistical results for QR runtime data of GOE initial matrices under cutoff exponential model.

Dim.	Tol.	N	Rate	I_{Rate}	KS*	KS p-value
30	1.0E-08	2131	0.09	4.00E-06	0.05	0.00
50	1.0E-08	2153	0.09	4.00E-06	0.06	0.00
70	1.0E-08	2003	0.08	4.00E-06	0.03	0.35
90	1.0E-08	2159	0.09	4.00E-06	0.06	0.00
110	1.0E-08	2154	0.09	4.00E-06	0.06	0.00
130	1.0E-08	2119	0.09	4.00E-06	0.06	0.00
150	1.0E-08	2003	0.08	4.00E-06	0.02	0.36
170	1.0E-08	2019	0.08	3.00E-06	0.02	0.36
190	1.0E-08	2054	0.09	4.00E-06	0.04	0.00
30	1.0E-06	2007	0.11	6.00E-06	0.03	0.16
50	1.0E-06	2036	0.11	6.00E-06	0.02	0.17
70	1.0E-06	2015	0.11	6.00E-06	0.02	0.19
90	1.0E-06	2030	0.11	6.00E-06	0.02	0.37
110	1.0E-06	2023	0.11	6.00E-06	0.02	0.17
130	1.0E-06	1988	0.11	6.00E-06	0.02	0.18
150	1.0E-06	2007	0.11	6.00E-06	0.02	0.17
170	1.0E-06	2042	0.11	6.00E-06	0.02	0.41
190	1.0E-06	1933	0.11	6.00E-06	0.03	0.44
30	1.0E-04	2023	0.15	1.20E-05	0.04	0.02
50	1.0E-04	2068	0.16	1.20E-05	0.05	0.00
70	1.0E-04	2050	0.15	1.20E-05	0.05	0.01
90	1.0E-04	2073	0.16	1.20E-05	0.05	0.00
110	1.0E-04	2062	0.16	1.20E-05	0.05	0.00
130	1.0E-04	2023	0.16	1.20E-05	0.04	0.00
150	1.0E-04	2026	0.15	1.20E-05	0.05	0.00
170	1.0E-04	2079	0.15	1.20E-05	0.06	0.00
190	1.0E-04	1968	0.15	1.20E-05	0.03	0.03

Table 7.13: Summary of statistical results for QR runtime data of GOE initial matrices under cutoff gamma model.

Dim.	Tol.	N	Shape	Scale	I_{Shape}	I_{Shape}	KS*	KS p-value
30	1.0E-08	2131	2.81	7.28	1.68E-01	3.58E-01	0.06	0.00
50	1.0E-08	2153	2.62	7.48	1.69E-01	3.97E-01	0.06	0.00
70	1.0E-08	2003	4.97	5.32	2.19E-01	1.33E-01	0.04	0.36
90	1.0E-08	2159	2.66	7.41	1.69E-01	3.86E-01	0.06	0.00
110	1.0E-08	2154	2.10	8.23	1.59E-01	5.49E-01	0.06	0.00
130	1.0E-08	2119	2.41	7.80	1.64E-01	4.58E-01	0.06	0.00
150	1.0E-08	2003	4.29	5.90	1.99E-01	1.81E-01	0.03	0.46
170	1.0E-08	2019	4.34	5.89	1.99E-01	1.77E-01	0.04	0.24
190	1.0E-08	2054	3.42	6.54	1.85E-01	2.61E-01	0.04	0.00
30	1.0E-06	2007	4.56	4.36	2.04E-01	9.37E-02	0.04	0.11
50	1.0E-06	2036	4.18	4.57	2.00E-01	1.11E-01	0.04	0.08
70	1.0E-06	2015	4.44	4.45	2.03E-01	1.00E-01	0.04	0.03
90	1.0E-06	2030	4.56	4.32	2.09E-01	9.30E-02	0.05	0.04
110	1.0E-06	2023	3.92	4.74	1.97E-01	1.26E-01	0.04	0.05
130	1.0E-06	1988	4.26	4.52	2.04E-01	1.08E-01	0.04	0.14
150	1.0E-06	2007	3.98	4.81	1.90E-01	1.25E-01	0.04	0.00
170	1.0E-06	2042	3.70	5.04	1.82E-01	1.43E-01	0.04	0.00
190	1.0E-06	1933	5.09	4.05	2.19E-01	7.64E-02	0.04	0.48
30	1.0E-04	2023	3.91	3.41	1.87E-01	6.34E-02	0.06	0.00
50	1.0E-04	2068	3.17	3.83	1.73E-01	9.21E-02	0.07	0.00
70	1.0E-04	2050	3.44	3.70	1.77E-01	8.11E-02	0.06	0.00
90	1.0E-04	2073	3.53	3.60	1.81E-01	7.57E-02	0.06	0.00
110	1.0E-04	2062	2.98	3.92	1.74E-01	1.02E-01	0.06	0.00
130	1.0E-04	2023	3.28	3.75	1.78E-01	8.79E-02	0.06	0.00
150	1.0E-04	2026	3.28	3.84	1.71E-01	8.99E-02	0.06	0.00
170	1.0E-04	2079	2.66	4.30	1.56E-01	1.26E-01	0.06	0.00
190	1.0E-04	1968	4.05	3.35	1.90E-01	6.02E-02	0.06	0.00

Table 7.14: Summary of statistical results for QR runtime data of Wigner initial matrices under cutoff exponential model.

Dim.	Tol.	N	Rate	I_{Rate}	KS*	KS p-value
30	1.0E-08	2114	0.09	4.00E-06	0.06	0.00
50	1.0E-08	2131	0.09	4.00E-06	0.06	0.00
70	1.0E-08	2120	0.09	4.00E-06	0.06	0.00
90	1.0E-08	2131	0.09	4.00E-06	0.06	0.00
110	1.0E-08	2098	0.09	4.00E-06	0.05	0.00
130	1.0E-08	2168	0.09	4.00E-06	0.07	0.00
150	1.0E-08	1996	0.08	3.00E-06	0.03	0.12
170	1.0E-08	2007	0.08	4.00E-06	0.03	0.34
190	1.0E-08	2095	0.09	4.00E-06	0.05	0.00
30	1.0E-06	1992	0.11	6.00E-06	0.03	0.08
50	1.0E-06	2011	0.11	6.00E-06	0.03	0.11
70	1.0E-06	1993	0.11	6.00E-06	0.02	0.19
90	1.0E-06	2005	0.11	6.00E-06	0.02	0.32
110	1.0E-06	1974	0.11	6.00E-06	0.03	0.23
130	1.0E-06	2027	0.11	6.00E-06	0.03	0.07
150	1.0E-06	2030	0.11	6.00E-06	0.03	0.06
170	1.0E-06	2024	0.11	6.00E-06	0.02	0.16
190	1.0E-06	1968	0.11	6.00E-06	0.02	0.37
30	1.0E-04	2025	0.16	1.20E-05	0.05	0.00
50	1.0E-04	2042	0.16	1.20E-05	0.04	0.01
70	1.0E-04	2016	0.15	1.20E-05	0.04	0.01
90	1.0E-04	2036	0.15	1.20E-05	0.04	0.01
110	1.0E-04	2001	0.15	1.10E-05	0.04	0.02
130	1.0E-04	2052	0.16	1.20E-05	0.05	0.01
150	1.0E-04	2050	0.15	1.20E-05	0.07	0.00
170	1.0E-04	2063	0.16	1.20E-05	0.05	0.00
190	1.0E-04	1996	0.15	1.20E-05	0.04	0.00

Table 7.15: Summary of statistical results for QR runtime data of Wigner initial matrices under cutoff gamma model.

Dim.	Tol.	N	Shape	Scale	I_{Shape}	I_{Shape}	KS*	KS p-value
30	1.0E-08	2114	2.88	7.14	1.76E-01	3.47E-01	0.06	0.00
50	1.0E-08	2131	3.10	6.87	1.80E-01	3.05E-01	0.06	0.00
70	1.0E-08	2120	2.30	8.13	1.54E-01	4.98E-01	0.06	0.00
90	1.0E-08	2131	2.31	8.11	1.54E-01	4.92E-01	0.06	0.00
110	1.0E-08	2098	2.83	7.43	1.61E-01	3.64E-01	0.05	0.00
130	1.0E-08	2168	1.90	8.74	1.49E-01	6.37E-01	0.07	0.00
150	1.0E-08	1996	4.55	5.71	2.07E-01	1.63E-01	0.03	0.25
170	1.0E-08	2007	4.58	5.61	2.11E-01	1.57E-01	0.04	0.33
190	1.0E-08	2095	2.09	8.45	1.52E-01	5.72E-01	0.05	0.00
30	1.0E-06	1992	4.75	4.22	2.15E-01	8.71E-02	0.04	0.04
50	1.0E-06	2011	4.87	4.15	2.17E-01	8.22E-02	0.05	0.07
70	1.0E-06	1993	3.99	4.80	1.88E-01	1.24E-01	0.05	0.13
90	1.0E-06	2005	3.93	4.84	1.86E-01	1.27E-01	0.05	0.12
110	1.0E-06	1974	4.42	4.55	1.93E-01	1.02E-01	0.05	0.16
130	1.0E-06	2027	3.62	5.07	1.82E-01	1.48E-01	0.04	0.04
150	1.0E-06	2030	3.66	5.06	1.84E-01	1.47E-01	0.04	0.01
170	1.0E-06	2024	3.98	4.76	1.94E-01	1.24E-01	0.04	0.00
190	1.0E-06	1968	3.62	5.09	1.81E-01	1.51E-01	0.05	0.30
30	1.0E-04	2025	3.83	3.43	1.90E-01	6.60E-02	0.06	0.00
50	1.0E-04	2042	3.90	3.38	1.91E-01	6.32E-02	0.06	0.00
70	1.0E-04	2016	3.38	3.75	1.72E-01	8.40E-02	0.06	0.00
90	1.0E-04	2036	3.15	3.91	1.66E-01	9.47E-02	0.05	0.00
110	1.0E-04	2001	3.44	3.77	1.68E-01	8.25E-02	0.06	0.00
130	1.0E-04	2052	2.74	4.22	1.59E-01	1.20E-01	0.06	0.00
150	1.0E-04	2050	2.67	4.31	1.58E-01	1.28E-01	0.08	0.00
170	1.0E-04	2063	2.90	4.05	1.67E-01	1.09E-01	0.06	0.00
190	1.0E-04	1996	2.85	4.15	1.59E-01	1.15E-01	0.06	0.00

Table 7.16: Summary of statistical results for QR runtime data of Bernoulli initial matrices under cutoff exponential model.

Dim.	Tol.	N	Rate	I_{Rate}	KS*	KS p-value
30	1.0E-08	2153	0.09	4.00E-06	0.06	0.00
50	1.0E-08	2116	0.09	4.00E-06	0.04	0.01
70	1.0E-08	2136	0.09	4.00E-06	0.05	0.00
90	1.0E-08	2171	0.09	4.00E-06	0.06	0.00
110	1.0E-08	2176	0.09	4.00E-06	0.08	0.00
130	1.0E-08	2008	0.08	3.00E-06	0.02	0.48
150	1.0E-08	2148	0.09	4.00E-06	0.05	0.00
170	1.0E-08	2165	0.09	4.00E-06	0.06	0.00
190	1.0E-08	2051	0.09	4.00E-06	0.03	0.22
30	1.0E-06	2014	0.11	6.00E-06	0.03	0.10
50	1.0E-06	2176	0.12	7.00E-06	0.09	0.00
70	1.0E-06	2003	0.11	6.00E-06	0.03	0.34
90	1.0E-06	2068	0.11	6.00E-06	0.03	0.03
110	1.0E-06	2027	0.11	6.00E-06	0.03	0.12
130	1.0E-06	2025	0.11	6.00E-06	0.03	0.01
150	1.0E-06	2027	0.11	6.00E-06	0.03	0.24
170	1.0E-06	2029	0.11	6.00E-06	0.03	0.11
190	1.0E-06	2057	0.11	6.00E-06	0.03	0.02
30	1.0E-04	2048	0.16	1.20E-05	0.04	0.01
50	1.0E-04	2006	0.16	1.20E-05	0.03	0.06
70	1.0E-04	2030	0.15	1.20E-05	0.03	0.02
90	1.0E-04	2111	0.16	1.20E-05	0.05	0.00
110	1.0E-04	2063	0.16	1.20E-05	0.05	0.00
130	1.0E-04	2061	0.15	1.20E-05	0.07	0.00
150	1.0E-04	2062	0.16	1.20E-05	0.04	0.00
170	1.0E-04	2048	0.16	1.20E-05	0.05	0.00
190	1.0E-04	2089	0.16	1.20E-05	0.06	0.00

Table 7.17: Summary of statistical results for QR runtime data of Bernoulli initial matrices under cutoff gamma model.

Dim.	Tol.	N	Shape	Scale	$I_{extShape}$	$I_{extShape}$	KS*	KS p-value
30	1.0E-08	2153	2.88	7.12	1.73E-01	3.39E-01	0.06	0.00
50	1.0E-08	2116	3.03	6.79	1.82E-01	3.07E-01	0.05	0.00
70	1.0E-08	2136	2.68	7.44	1.65E-01	3.83E-01	0.05	0.00
90	1.0E-08	2171	2.88	7.08	1.76E-01	3.38E-01	0.06	0.00
110	1.0E-08	2176	2.37	7.80	1.64E-01	4.57E-01	0.08	0.00
130	1.0E-08	2008	3.45	6.84	1.77E-01	2.79E-01	0.04	0.15
150	1.0E-08	2148	2.60	7.42	1.71E-01	3.97E-01	0.05	0.00
170	1.0E-08	2165	2.12	8.26	1.56E-01	5.41E-01	0.06	0.00
190	1.0E-08	2051	5.15	5.11	2.31E-01	1.21E-01	0.04	0.13
30	1.0E-06	2014	5.09	4.02	2.20E-01	7.42E-02	0.04	0.16
50	1.0E-06	2176	1.96	6.41	1.57E-01	3.44E-01	0.09	0.00
70	1.0E-06	2003	4.39	4.48	1.99E-01	1.01E-01	0.05	0.25
90	1.0E-06	2068	4.48	4.33	2.09E-01	9.48E-02	0.05	0.04
110	1.0E-06	2027	4.54	4.34	2.09E-01	9.46E-02	0.05	0.05
130	1.0E-06	2025	2.96	5.77	1.65E-01	2.17E-01	0.05	0.00
150	1.0E-06	2027	4.29	4.44	2.06E-01	1.04E-01	0.05	0.27
170	1.0E-06	2029	3.80	4.88	1.88E-01	1.34E-01	0.04	0.07
190	1.0E-06	2057	4.78	4.17	2.20E-01	8.45E-02	0.04	0.01
30	1.0E-04	2048	4.09	3.28	1.94E-01	5.72E-02	0.06	0.00
50	1.0E-04	2006	4.25	3.16	2.03E-01	5.27E-02	0.07	0.00
70	1.0E-04	2030	3.60	3.60	1.77E-01	7.40E-02	0.06	0.00
90	1.0E-04	2111	3.54	3.55	1.85E-01	7.37E-02	0.06	0.00
110	1.0E-04	2063	3.68	3.51	1.86E-01	7.03E-02	0.06	0.00
130	1.0E-04	2061	2.13	4.84	1.44E-01	1.82E-01	0.08	0.00
150	1.0E-04	2062	3.31	3.68	1.81E-01	8.37E-02	0.06	0.00
170	1.0E-04	2048	2.86	4.11	1.62E-01	1.12E-01	0.06	0.00
190	1.0E-04	2089	3.80	3.41	1.95E-01	6.59E-02	0.07	0.00

Table 7.18: Summary of statistical results for Toda runtime data of Hermite–1 initial matrices under cutoff Gaussian model.

Dim.	Tol.	N	σ	I_σ	KS*	KS p-value
30	1.0E-08	2502	2.75	1.52E-03	0.02	0.63
50	1.0E-08	2570	3.20	1.99E-03	0.03	0.00
70	1.0E-08	2484	3.60	2.61E-03	0.01	0.89
90	1.0E-08	2489	3.85	2.98E-03	0.02	0.42
110	1.0E-08	2504	4.08	3.32E-03	0.02	0.09
130	1.0E-08	2487	4.23	3.60E-03	0.02	0.22
150	1.0E-08	2512	4.44	3.92E-03	0.01	0.91
170	1.0E-08	2503	4.60	4.22E-03	0.02	0.10
190	1.0E-08	2474	4.79	4.64E-03	0.02	0.30
30	1.0E-06	2504	2.27	1.03E-03	0.02	0.25
50	1.0E-06	2519	2.67	1.42E-03	0.03	0.07
70	1.0E-06	2457	3.01	1.84E-03	0.01	0.96
90	1.0E-06	2495	3.17	2.02E-03	0.02	0.44
110	1.0E-06	2448	3.40	2.36E-03	0.02	0.57
130	1.0E-06	2428	3.53	2.56E-03	0.02	0.56
150	1.0E-06	2428	3.68	2.79E-03	0.02	0.63
170	1.0E-06	2448	3.81	2.97E-03	0.02	0.72
190	1.0E-06	2451	3.90	3.11E-03	0.02	0.06
30	1.0E-04	2554	1.77	6.14E-04	0.06	0.00
50	1.0E-04	2505	2.08	8.67E-04	0.05	0.00
70	1.0E-04	2415	2.34	1.13E-03	0.02	0.31
90	1.0E-04	2406	2.47	1.27E-03	0.02	0.44
110	1.0E-04	2420	2.58	1.38E-03	0.03	0.01
130	1.0E-04	2347	2.69	1.54E-03	0.02	0.67
150	1.0E-04	2413	2.71	1.53E-03	0.03	0.14
170	1.0E-04	2446	2.82	1.62E-03	0.02	0.13
190	1.0E-04	2441	2.85	1.67E-03	0.03	0.04

Table 7.19: Summary of statistical results for Toda runtime data of Hermite–1 initial matrices under cutoff Weibull model.

Dim.	Tol.	N	Shape	Scale	I_{Shape}	I_{Shape}	KS*	KS p-value
30	1.0E-08	2502	4.62	12.70	3.13E-02	2.34E-02	0.03	0.11
50	1.0E-08	2570	3.92	13.65	2.92E-02	5.57E-02	0.03	0.00
70	1.0E-08	2484	4.31	15.39	2.82E-02	4.08E-02	0.02	0.09
90	1.0E-08	2489	4.13	16.11	2.64E-02	5.10E-02	0.02	0.02
110	1.0E-08	2504	4.26	17.07	2.69E-02	4.91E-02	0.02	0.02
130	1.0E-08	2487	4.09	17.35	2.57E-02	5.94E-02	0.02	0.05
150	1.0E-08	2512	3.91	17.61	2.46E-02	7.19E-02	0.02	0.08
170	1.0E-08	2503	3.88	18.13	2.44E-02	7.72E-02	0.02	0.08
190	1.0E-08	2474	3.69	18.40	2.45E-02	1.02E-01	0.02	0.01
30	1.0E-06	2504	4.52	10.04	2.89E-02	1.43E-02	0.03	0.05
50	1.0E-06	2519	3.91	10.90	2.66E-02	3.09E-02	0.03	0.02
70	1.0E-06	2457	4.12	12.09	2.57E-02	2.66E-02	0.02	0.14
90	1.0E-06	2495	3.77	12.34	2.38E-02	4.03E-02	0.02	0.11
110	1.0E-06	2448	3.88	13.11	2.40E-02	3.89E-02	0.02	0.13
130	1.0E-06	2428	3.89	13.53	2.33E-02	3.88E-02	0.02	0.08
150	1.0E-06	2428	3.78	13.77	2.29E-02	4.53E-02	0.02	0.13
170	1.0E-06	2448	3.64	13.99	2.21E-02	5.41E-02	0.02	0.12
190	1.0E-06	2451	3.35	13.87	2.22E-02	8.15E-02	0.02	0.03
30	1.0E-04	2554	4.03	7.07	2.53E-02	1.01E-02	0.06	0.00
50	1.0E-04	2505	3.33	7.44	2.22E-02	2.41E-02	0.05	0.00
70	1.0E-04	2415	3.70	8.44	2.18E-02	1.72E-02	0.03	0.02
90	1.0E-04	2406	3.40	8.57	2.03E-02	2.51E-02	0.03	0.04
110	1.0E-04	2420	3.05	8.46	1.98E-02	4.09E-02	0.03	0.00
130	1.0E-04	2347	3.40	9.17	1.97E-02	2.73E-02	0.02	0.27
150	1.0E-04	2413	3.24	9.11	1.99E-02	3.50E-02	0.03	0.02
170	1.0E-04	2446	3.18	9.32	1.90E-02	3.77E-02	0.02	0.02
190	1.0E-04	2441	3.03	9.27	1.89E-02	4.86E-02	0.02	0.02

Table 7.20: Summary of statistical results for Toda runtime data of GOE initial matrices under cutoff Gaussian model.

Dim.	Tol.	N	σ	I_σ	KS*	KS p-value
30	1.0E-08	1774	2.35	1.56E-03	0.04	0.02
50	1.0E-08	1009	2.81	3.91E-03	0.04	0.15
70	1.0E-08	1732	3.02	2.63E-03	0.02	0.71
90	1.0E-08	1726	3.28	3.12E-03	0.02	0.59
110	1.0E-08	501	3.46	1.20E-02	0.05	0.14
130	1.0E-08	2238	3.51	2.75E-03	0.02	0.42
150	1.0E-08	1207	3.81	6.03E-03	0.03	0.26
170	1.0E-08	1247	3.83	5.90E-03	0.02	0.89
190	1.0E-08	489	4.04	1.67E-02	0.03	0.80
30	1.0E-06	1794	1.81	9.13E-04	0.05	0.00
50	1.0E-06	977	2.22	2.52E-03	0.02	0.85
70	1.0E-06	1697	2.38	1.67E-03	0.02	0.70
90	1.0E-06	1707	2.59	1.97E-03	0.02	0.53
110	1.0E-06	486	2.76	7.82E-03	0.02	0.95
130	1.0E-06	2227	2.75	1.70E-03	0.02	0.53
150	1.0E-06	1226	2.96	3.58E-03	0.01	0.98
170	1.0E-06	1242	3.01	3.65E-03	0.01	0.97
190	1.0E-06	493	3.14	9.99E-03	0.03	0.57
30	1.0E-04	1812	1.28	4.51E-04	0.08	0.00
50	1.0E-04	962	1.60	1.34E-03	0.02	0.87
70	1.0E-04	1734	1.69	8.19E-04	0.03	0.10
90	1.0E-04	1717	1.88	1.03E-03	0.04	0.02
110	1.0E-04	482	2.00	4.15E-03	0.03	0.64
130	1.0E-04	2187	2.01	9.21E-04	0.02	0.71
150	1.0E-04	1245	2.12	1.80E-03	0.03	0.18
170	1.0E-04	1251	2.17	1.88E-03	0.02	0.46
190	1.0E-04	485	2.26	5.28E-03	0.03	0.62

Table 7.21: Summary of statistical results for Toda runtime data of GOE initial matrices under cutoff Weibull model.

Dim.	Tol.	N	Shape	Scale	I_{Shape}	I_{Shape}	KS*	KS p-value
30	1.0E-08	1774	4.56	11.91	5.68E-02	4.54E-02	0.04	0.00
50	1.0E-08	1009	4.49	13.49	8.96E-02	9.25E-02	0.04	0.00
70	1.0E-08	1732	4.75	14.87	5.23E-02	5.03E-02	0.03	0.07
90	1.0E-08	1726	4.01	14.98	4.81E-02	1.06E-01	0.02	0.37
110	1.0E-08	501	3.28	14.23	1.53E-01	8.31E-01	0.04	0.09
130	1.0E-08	2238	4.12	16.29	3.77E-02	8.75E-02	0.02	0.14
150	1.0E-08	1207	4.63	17.80	6.66E-02	9.75E-02	0.01	0.72
170	1.0E-08	1247	4.05	17.26	6.51E-02	1.78E-01	0.02	0.24
190	1.0E-08	489	4.19	17.99	1.58E-01	3.81E-01	0.02	0.64
30	1.0E-06	1794	4.44	9.14	5.70E-02	3.10E-02	0.05	0.00
50	1.0E-06	977	4.93	10.88	8.95E-02	3.73E-02	0.03	0.27
70	1.0E-06	1697	4.98	11.81	5.13E-02	2.44E-02	0.03	0.22
90	1.0E-06	1707	4.06	11.79	4.76E-02	6.10E-02	0.02	0.18
110	1.0E-06	486	3.76	11.98	1.53E-01	2.94E-01	0.02	0.64
130	1.0E-06	2227	4.22	12.86	3.81E-02	4.86E-02	0.02	0.09
150	1.0E-06	1226	4.31	13.57	6.48E-02	7.93E-02	0.02	0.36
170	1.0E-06	1242	4.07	13.54	6.39E-02	1.05E-01	0.02	0.26
190	1.0E-06	493	4.18	14.01	1.63E-01	2.44E-01	0.04	0.07
30	1.0E-04	1812	4.05	6.27	5.55E-02	2.28E-02	0.08	0.00
50	1.0E-04	962	4.92	7.80	8.60E-02	1.85E-02	0.04	0.42
70	1.0E-04	1734	4.71	8.29	5.00E-02	1.57E-02	0.03	0.00
90	1.0E-04	1717	3.83	8.29	4.66E-02	3.91E-02	0.04	0.00
110	1.0E-04	482	3.95	8.83	1.51E-01	1.21E-01	0.03	0.09
130	1.0E-04	2187	4.41	9.44	3.81E-02	2.07E-02	0.02	0.33
150	1.0E-04	1245	4.06	9.52	6.47E-02	5.28E-02	0.03	0.00
170	1.0E-04	1251	3.81	9.49	6.16E-02	6.91E-02	0.02	0.10
190	1.0E-04	485	4.53	10.37	1.68E-01	9.11E-02	0.03	0.50
30	1.0E-02	1788	3.93	3.70	5.20E-02	8.48E-03	0.09	0.00
50	1.0E-02	1055	3.61	4.09	8.55E-02	2.48E-02	0.11	0.00
70	1.0E-02	1760	4.30	4.86	4.89E-02	8.12E-03	0.06	0.00
90	1.0E-02	1700	3.86	5.05	4.43E-02	1.31E-02	0.04	0.00
110	1.0E-02	499	3.90	5.26	1.63E-01	5.01E-02	0.06	0.00
130	1.0E-02	2202	4.16	5.64	3.58E-02	9.26E-03	0.03	0.03
150	1.0E-02	1278	3.65	5.53	6.16E-02	2.91E-02	0.05	0.01
170	1.0E-02	1242	3.91	5.84	6.25E-02	2.32E-02	0.05	0.00
190	1.0E-02	499	4.07	6.10	1.54E-01	4.99E-02	0.04	0.04

Table 7.22: Summary of statistical results for Toda runtime data of Wigner initial matrices under cutoff Gaussian model.

Dim.	Tol.	N	σ	I_σ	KS*	KS p-value
30	1.0E-08	2002	2.49	1.55E-03	0.03	0.13
50	1.0E-08	989	2.84	4.07E-03	0.03	0.46
70	1.0E-08	1749	3.12	2.79E-03	0.03	0.18
90	1.0E-08	1720	3.34	3.25E-03	0.02	0.56
110	1.0E-08	1216	3.49	5.01E-03	0.03	0.20
130	1.0E-08	2484	3.56	2.56E-03	0.01	0.95
150	1.0E-08	1216	3.62	5.39E-03	0.01	1.00
170	1.0E-08	1253	3.73	5.56E-03	0.02	0.47
190	1.0E-08	737	3.86	1.01E-02	0.03	0.51
30	1.0E-06	2051	1.91	8.93E-04	0.05	0.00
50	1.0E-06	996	2.21	2.45E-03	0.04	0.06
70	1.0E-06	1727	2.45	1.74E-03	0.02	0.38
90	1.0E-06	1729	2.60	1.96E-03	0.02	0.25
110	1.0E-06	1228	2.70	2.96E-03	0.03	0.22
130	1.0E-06	2474	2.79	1.57E-03	0.01	0.87
150	1.0E-06	1199	2.85	3.38E-03	0.02	0.93
170	1.0E-06	1265	2.91	3.34E-03	0.04	0.04
190	1.0E-06	742	3.02	6.16E-03	0.03	0.68
30	1.0E-04	1921	1.41	5.18E-04	0.03	0.29
50	1.0E-04	1004	1.59	1.27E-03	0.05	0.00
70	1.0E-04	1741	1.76	8.90E-04	0.05	0.01
90	1.0E-04	1737	1.88	1.02E-03	0.03	0.04
110	1.0E-04	1222	1.94	1.53E-03	0.03	0.40
130	1.0E-04	2447	2.02	8.38E-04	0.02	0.56
150	1.0E-04	1232	2.02	1.67E-03	0.03	0.22
170	1.0E-04	1253	2.11	1.78E-03	0.02	0.42
190	1.0E-04	738	2.19	3.27E-03	0.02	0.90

Table 7.23: Summary of statistical results for Toda runtime data of Wigner initial matrices under cutoff Weibull model.

Dim.	Tol.	N	Shape	Scale	I_{Shape}	I_{Shape}	KS*	KS p-value
30	1.0E-08	2002	4.62	12.56	4.68E-02	3.83E-02	0.03	0.01
50	1.0E-08	989	4.54	13.96	9.25E-02	9.96E-02	0.03	0.04
70	1.0E-08	1749	4.26	14.72	4.88E-02	7.80E-02	0.03	0.01
90	1.0E-08	1720	4.42	15.76	5.09E-02	7.60E-02	0.03	0.01
110	1.0E-08	1216	4.09	15.93	6.98E-02	1.57E-01	0.03	0.01
130	1.0E-08	2484	4.26	16.72	3.34E-02	6.88E-02	0.02	0.05
150	1.0E-08	1216	4.09	16.70	6.75E-02	1.70E-01	0.01	0.93
170	1.0E-08	1253	4.08	17.28	6.90E-02	1.90E-01	0.02	0.07
190	1.0E-08	737	3.73	16.99	1.07E-01	4.39E-01	0.02	0.29
30	1.0E-06	2051	4.21	9.40	4.64E-02	3.44E-02	0.05	0.00
50	1.0E-06	996	4.38	10.73	9.23E-02	7.02E-02	0.04	0.00
70	1.0E-06	1727	4.48	11.70	4.94E-02	3.85E-02	0.02	0.04
90	1.0E-06	1729	4.31	12.19	5.05E-02	5.11E-02	0.03	0.03
110	1.0E-06	1228	4.04	12.32	6.94E-02	1.01E-01	0.03	0.03
130	1.0E-06	2474	4.28	13.10	3.36E-02	4.14E-02	0.02	0.03
150	1.0E-06	1199	4.40	13.44	6.75E-02	7.59E-02	0.02	0.43
170	1.0E-06	1265	3.97	13.36	6.91E-02	1.29E-01	0.03	0.00
190	1.0E-06	742	3.70	13.25	1.07E-01	2.74E-01	0.02	0.52
30	1.0E-04	1921	5.01	7.18	4.57E-02	8.05E-03	0.05	0.11
50	1.0E-04	1004	4.17	7.55	9.23E-02	4.38E-02	0.05	0.00
70	1.0E-04	1741	4.20	8.19	4.91E-02	2.60E-02	0.05	0.00
90	1.0E-04	1737	4.10	8.60	4.95E-02	3.20E-02	0.03	0.01
110	1.0E-04	1222	4.16	8.95	6.91E-02	4.54E-02	0.03	0.16
130	1.0E-04	2447	4.31	9.47	3.40E-02	2.09E-02	0.02	0.10
150	1.0E-04	1232	4.13	9.41	6.73E-02	5.14E-02	0.03	0.02
170	1.0E-04	1253	4.09	9.77	6.76E-02	5.82E-02	0.02	0.12
190	1.0E-04	738	3.90	9.78	1.05E-01	1.13E-01	0.02	0.66

Table 7.24: Summary of statistical results for Toda runtime data of Bernoulli initial matrices under cutoff Gaussian model.

Dim.	Tol.	N	σ	I_σ	KS*	KS p-value
30	1.0E-08	2543	2.41	1.14E-03	0.04	0.00
50	1.0E-08	2479	2.85	1.64E-03	0.03	0.60
70	1.0E-08	2471	3.20	2.07E-03	0.02	0.29
90	1.0E-08	2519	3.24	2.09E-03	0.01	0.75
110	1.0E-08	2490	3.48	2.44E-03	0.01	0.97
130	1.0E-08	2445	3.67	2.76E-03	0.01	0.74
150	1.0E-08	2511	3.69	2.72E-03	0.01	0.93
170	1.0E-08	1239	3.71	5.55E-03	0.02	0.82
190	1.0E-08	760	3.92	1.01E-02	0.04	0.25
30	1.0E-06	2486	1.90	7.23E-04	0.03	0.01
50	1.0E-06	2519	2.20	9.65E-04	0.02	0.22
70	1.0E-06	2455	2.51	1.28E-03	0.02	0.31
90	1.0E-06	2525	2.54	1.27E-03	0.02	0.09
110	1.0E-06	2508	2.71	1.46E-03	0.02	0.45
130	1.0E-06	2452	2.88	1.69E-03	0.02	0.23
150	1.0E-06	2512	2.89	1.67E-03	0.01	0.87
170	1.0E-06	1235	2.90	3.40E-03	0.02	0.90
190	1.0E-06	760	3.08	6.26E-03	0.05	0.05
30	1.0E-04	2609	1.32	3.36E-04	0.09	0.00
50	1.0E-04	2565	1.57	4.82E-04	0.05	0.00
70	1.0E-04	2398	1.84	7.03E-04	0.01	0.77
90	1.0E-04	2497	1.84	6.79E-04	0.02	0.24
110	1.0E-04	2531	1.95	7.52E-04	0.04	0.00
130	1.0E-04	2450	2.08	8.85E-04	0.03	0.06
150	1.0E-04	2490	2.09	8.81E-04	0.02	0.50
170	1.0E-04	1248	2.09	1.75E-03	0.03	0.13
190	1.0E-04	748	2.27	3.45E-03	0.03	0.54

Table 7.25: Summary of statistical results for Toda runtime data of Bernoulli initial matrices under cutoff Weibull model.

Dim.	Tol.	N	Shape	Scale	I_{Shape}	I_{Shape}	KS*	KS p-value
30	1.0E-08	2543	4.20	12.40	4.26E-02	5.91E-02	0.04	0.00
50	1.0E-08	2479	5.28	15.00	3.98E-02	2.41E-02	0.03	0.17
70	1.0E-08	2471	4.40	15.38	3.61E-02	5.45E-02	0.02	0.01
90	1.0E-08	2519	4.60	16.19	3.73E-02	5.14E-02	0.02	0.14
110	1.0E-08	2490	4.50	16.80	3.59E-02	5.72E-02	0.02	0.06
130	1.0E-08	2445	4.12	16.93	3.41E-02	8.45E-02	0.02	0.03
150	1.0E-08	2511	4.32	17.57	3.45E-02	7.41E-02	0.02	0.19
170	1.0E-08	1239	4.17	17.52	7.09E-02	1.82E-01	0.02	0.55
190	1.0E-08	760	4.49	18.71	1.17E-01	2.29E-01	0.03	0.06
30	1.0E-06	2486	4.53	9.94	4.24E-02	2.57E-02	0.04	0.00
50	1.0E-06	2519	5.04	11.49	3.96E-02	1.78E-02	0.02	0.03
70	1.0E-06	2455	4.46	12.06	3.63E-02	3.11E-02	0.03	0.03
90	1.0E-06	2525	4.50	12.55	3.69E-02	3.40E-02	0.02	0.04
110	1.0E-06	2508	4.34	12.94	3.55E-02	4.03E-02	0.02	0.06
130	1.0E-06	2452	3.93	12.98	3.37E-02	6.22E-02	0.02	0.21
150	1.0E-06	2512	4.24	13.65	3.44E-02	4.86E-02	0.02	0.09
170	1.0E-06	1235	4.24	13.77	7.09E-02	1.04E-01	0.02	0.27
190	1.0E-06	760	4.55	14.72	1.17E-01	1.33E-01	0.05	0.00
30	1.0E-04	2609	3.84	6.56	4.19E-02	2.57E-02	0.09	0.00
50	1.0E-04	2565	4.63	7.98	3.91E-02	1.29E-02	0.05	0.00
70	1.0E-04	2398	4.65	8.82	3.59E-02	1.30E-02	0.03	0.18
90	1.0E-04	2497	4.59	9.09	3.64E-02	1.57E-02	0.03	0.05
110	1.0E-04	2531	4.04	9.04	3.48E-02	2.76E-02	0.04	0.00
130	1.0E-04	2450	3.88	9.29	3.35E-02	3.34E-02	0.03	0.12
150	1.0E-04	2490	4.31	9.89	3.45E-02	2.36E-02	0.02	0.13
170	1.0E-04	1248	4.07	9.75	7.14E-02	6.43E-02	0.03	0.00
190	1.0E-04	748	4.56	10.69	1.10E-01	6.40E-02	0.02	0.69

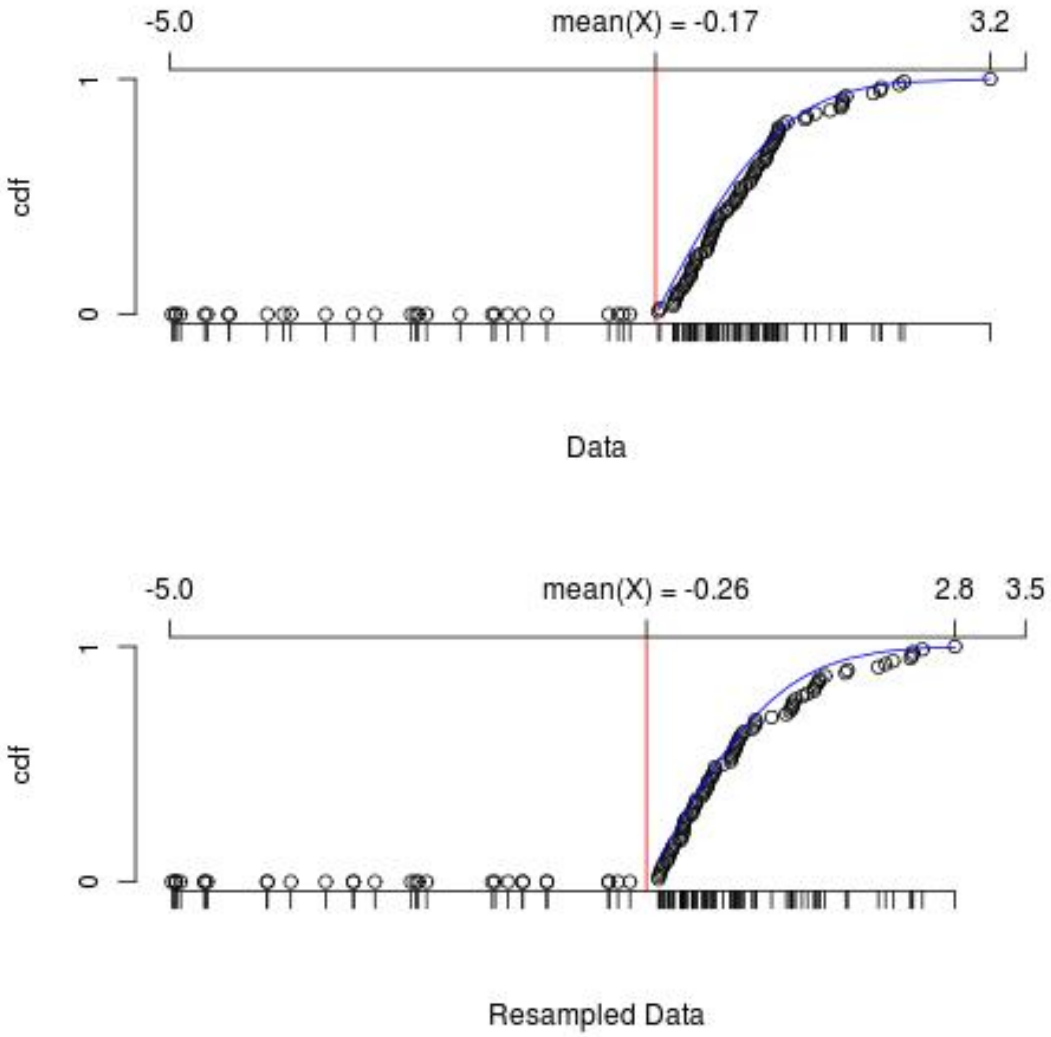

Figure 7.88: Pictorial representation of the statistical validation methodology. The upper row shows the estimation of the 'cutoff Gaussian' parameters from the data set. The lower row shows a resampled data set used to generate a KS–statistic sample for the p–value computation.

Chapter Eight

Conclusion and Future Work

Here we first summarize the information we gained from the numerical simulations in the foregoing chapter.

After that we prove a theorem about the behavior of the inverse spectral map for Jacobi matrices using bidiagonal coordinates and outline some ideas for further research.

8.1 Summary of the Numerical Results and Open Questions

The main results of this thesis are found in chapter 7. There we present interesting numerical observations regarding the behavior of the QR and the Toda algorithms on random initial data. Our data support the following statements:

- For initial data with spectral distribution converging to a Wigner semicircle limit the 'normalized' runtime distributions of the QR as well as the 'normalized' runtime distributions of the Toda algorithm are universal. Here normalized means that the histograms were shifted and scaled to exhibit mean zero and variance one.

- The runtime distributions of the QR algorithm in the case of Wigner–type initial data show exponential right tails, the runtime distribution of the Toda algorithm for these initial distribution shows Gaussian right tails.

- In the case of the QR runtime distributions for Wigner–type initial data, the means and standard deviations are very insensitive to the matrix dimension and they increase linearly in the negative log of the deflation tolerance.

- The same is not true for the Toda algorithm as in this case both means and standard deviations increase monotonously in the matrix dimension.

- Choosing initial data not of Wigner–type leads to different behavior both of the means and standard deviations. In addition non Wigner–type runtime histograms show a worse overlap with Wigner–type runtime histograms than when comparing Wigner–type runtime histograms with each other. In the case of the Toda algorithm, the shapes of the runtime histograms are visibly different in the Wigner and non Wigner–type cases.

- The QR algorithm, for a wide range of initial distributions has the strong tendency to deflate at the bottom of the matrix, whereas the Toda algorithm tends to deflate at both ends. With increasing matrix size deflation at the top end seems to become more likely in the case of Hermite–1 distributed Jacobi matrices.

- We were able to design an algorithm, that tends to split random matrices of Wigner type in the middle. Its runtime distributions have low variance and the means seem to depend linearly on $\log \epsilon$ and n.

It would be very interesting to conduct further numerical study of the above algorithms on other types of initial data. One ensemble that suggests itself for further study is the Wishart ensemble or a tridiagonalized variant of it.

8.2 Future Work

8.2.1 The Geometry of the Inverse Spectral Map near the Boundary

Recently a new parametrization, the so called bidiagonal coordinates, of tridiagonal matrices was found with favorable properties as compared to the standard spectral map [38]. We will use this parametrization to derive estimates for the standard spectral map. In short, there is a smooth and bijective map ψ between Jacobi matrices and bidiagonal matrices (with inverse ϕ). To define these maps, consider a bidiagonal matrix

$$B = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ \beta_1 & \lambda_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \beta_{n-1} & \lambda_n \end{pmatrix} \tag{8.1}$$

and set

$$L = (m_{ij})_{1 \leq ij \leq n} \begin{cases} 0 & i < j \\ 1 & i = j \\ \prod_{k=1}^{i-1} \frac{\beta_k}{\lambda_{\pi(i)} - \lambda_{\pi(k)}} & i > j \end{cases} \tag{8.2}$$

along with $\Lambda = \operatorname{diag}(\lambda)$, then $T = \phi(B) = \mathcal{Q}[L]^T \Lambda \mathcal{Q}[L]$ is the inverse chart and $B = \psi(T) = L^{-1}\Lambda L$, where $T = Q\Lambda Q^T$, is the parametrization. Here $\mathcal{Q}(A)$ is the unique orthogonal matrix in the QR decomposition of A (see section 2.1).

We now fix the following notation: Let $w = Q^T e_1$ and choose a permutation π such

that $w_{\pi(1)} \geq \cdots \geq w_{\pi(n)}$. Let

$$M(\lambda) = (m_{ij})_{1 \leq ij \leq n} = \begin{cases} 0 & i < j \\ 1 & i = j \\ \prod_{k=1}^{j-1} \frac{\lambda_{\pi(i)} - \lambda_{\pi(k)}}{\lambda_{\pi(j)} - \lambda_{\pi(k)}} & i > j \end{cases} \tag{8.3}$$

and define $D(w_\pi) = \operatorname{diag}(w_\pi)$, $W = D(w_\pi)M(\lambda_\pi)$, where $\lambda_\pi = (\lambda_{\pi(1)}, \ldots, \lambda_{\pi(n)})$. We can use the equation

$$\beta_i^\pi = \frac{\left|(\lambda_{\pi(i+1)} - \lambda_{\pi(1)}) \cdots (\lambda_{\pi(i+1)} - \lambda_{\pi(i)}\right|}{\left|(\lambda_{\pi(i)} - \lambda_{\pi(1)}) \cdots (\lambda_{\pi(i)} - \lambda_{\pi(i-1)}\right|} \frac{w_{\pi(i+1)}}{w_{\pi(i)}} \tag{8.4}$$

from [38] to show that

$$L = D(w_\pi)M(\lambda_\pi)D(w_\pi)^{-1} \tag{8.5}$$

and as a consequence we obtain that $\mathcal{Q}[L] = \mathcal{Q}[W]$. It is this equality along with the iterative Gram–Schmidt procedure that we'll use to show the following theorem:

Theorem 1. *Let $Q = \mathcal{Q}[W]$, then the entries follow have the following asymptotics:*

$$q_{ij} = \begin{cases} \mathcal{O}(w_{\pi(j)}/w_{\pi(i)}) & i < j \\ \mathcal{O}(1) & i = j \\ \mathcal{O}(w_{\pi(i)}/w_{\pi(j)}) & i > j \end{cases} \tag{8.6}$$

In particular, it follows that the k–th subdiagonal entry in T satisfies the estimate: $b_k = \mathcal{O}(w_{\pi(k+1)}/w_{\pi(k)})$

Proof. We prove the first statement by induction on j. We denote the columns of a matrix A by $a^{(j)}$. Then the claim holds for $j = 1$, because $w^{(1)} = w_\pi$. In the Gram–Schmidt procedure we have $q^{(1)} = \tilde{q}^{(1)} = w_\pi$ and, because of the fact that

$1 \geq w_\pi(1)^2 \geq 1/n$ we have that $q_j^{(1)} = \mathcal{O}(q_j^{(1)}/q_1^{(1)})$.

Now assume that the statement is true for all the columns from $j = 1$ to $j = k-1$. Then by Gram–Schmidt:

$$\tilde{q}_i^{(k)} = w_i^{(k)} - \langle q^{(1)}, w^{(k)} \rangle q_i^{(1)} - \cdots - \langle q^{(k-1)}, w^{(k)} \rangle q_i^{(k-1)} \tag{8.7}$$

Due to the lower triangular structure of W we have

$$\langle q^{(j)}, w^{(k)} \rangle = \sum_{l=k}^{n} m_{lk}(\lambda) q_l^{(j)} w_{\pi(l)} = \sum_{l=k}^{n} m_{lk}(\lambda) \mathcal{O}(w_{\pi(l)}/w_{\pi(j)}) w_{\pi(l)} = \\ = \mathcal{O}(w_{\pi(k)}^2/w_{\pi(j)}) \tag{8.8}$$

So we obtain:

$$\tilde{q}_i^{(k)} = w_i^{(k)} - \sum_{l<i\wedge k} \langle q^{(l)}, w^{(k)} \rangle q_i^{(l)} - 1_{i<k} \left(\langle q^{(i)}, w^{(k)} \rangle q_i^{(i)} + \sum_{i<l<k} \langle q^{(l)}, w^{(k)} \rangle q_i^{(l)} \right) \tag{8.9}$$

Note that for each individual term in the first sum we have the equation

$$\langle q^{(l)}, w^{(k)} \rangle q_i^{(l)} = \mathcal{O}\left(\frac{w_{\pi(k)}^2}{w_{\pi(l)}} \cdot \frac{w_{\pi(i)}}{w_{\pi(l)}} \right) = \mathcal{O}\left(\frac{w_{\pi(k)}^2}{w_{\pi(l)}} \cdot \frac{w_{\pi(i)}}{w_{\pi(i-1)}} \right) = \\ = \mathcal{O}\left(\frac{w_{\pi(k)}^2}{w_{\pi(i-1)}} \right) = \mathcal{O}\left(\frac{w_{\pi(k)}^2}{w_{\pi(i)}} \right) \tag{8.10}$$

where the last equality holds because of $w_{\pi(l)} > w_{\pi(i)}$. For the term between the two summations we have $\langle q^{(i)}, w^{(k)} \rangle q_i^{(i)} = \mathcal{O}(w_{\pi(k)}^2/w_{\pi(i)}^2)$ and for the individual terms in the last sum we have that

$$\langle q^{(l)}, w^{(k)} \rangle q_i^{(l)} = \mathcal{O}\left(\frac{w_{\pi(k)}^2}{w_{\pi(l)}} \cdot \frac{w_{\pi(l)}}{w_{\pi(i)}} \right) = \mathcal{O}\left(\frac{w_{\pi(k)}^2}{w_{\pi(i)}} \right) \tag{8.11}$$

Summarizing:

$$\tilde{q}_i^{(k)} = \begin{cases} \mathcal{O}\left(w_{\pi(k)}^2 / w_{\pi(i)}\right) & i < k \\ w_{\pi(k)} - \mathcal{O}\left(w_{\pi(k)}^2 / w_{\pi(k-1)}\right) & i = k \\ m_{ik}(\lambda) w_{\pi(i)} - \mathcal{O}\left(w_{\pi(k)}^2 / w_{\pi(i)}\right) & i < k \end{cases} \tag{8.12}$$

This last equation implies that $\left\|\tilde{q}^{(k)}\right\| / w_{\pi(k)}$ is bounded away from zero and, in particular:

$$q_i^{(k)} = \frac{\tilde{q}^{(k)}}{\|\tilde{q}^{(k)}\|} = \begin{cases} \mathcal{O}(w_{\pi(k)} / w_{\pi(i)}) & k > i \\ \mathcal{O}(1) & k = i \\ \mathcal{O}(w_{\pi(i)} / w_{\pi(k)}) & k < i \end{cases} \tag{8.13}$$

as claimed in the statement of the theorem. The claim about the k–th subdiagonal entry follows because of the fact that $b_k = \sum_{j=1}^{n} \lambda_j q_j^{(k)} q_j^{(k+1)}$ □

This theorem predicts the block structure of the matrix Q based on the structure of the first row of Q. It gives us, in turn, an idea of the behavior of the inverse spectral map close to the boundary of the positive orthant of the sphere, because it allows us to predict the structure of the tridiagonal matrix resulting from a first row w close to this boundary: For example, in order for the matrix T to deflate at the k–th subdiagonal index and at no other position, we require two 'blocks' of coefficients in w, where the entries in each block are roughly of the same magnitude, but the difference in magnitudes between the two blocks is of order ϵ. Similarly we can give qualitative conditions for the matrix T to deflate at two or more different subdiagonal positions.

We have thought about going the opposite way: Perturb a reduced (=block) Jacobi matrix so that it becomes unreduced and estimate the effect on the offdiagonal blocks of the eigenvector matrix. We believe that a theorem that proves this effect to be

of order ϵ is well within reach and we think that we can prove even finer estimates concerning perturbation of a multiply reduced Jacobi matrix at all its deflation indices.

8.2.2 Explaining the Deflation Behavior in the Jacobi case

One of the interesting numerical observations is the distinct behavior of the deflation index distribution in the case of Toda, QR and the algorithm which splits the matrix in the middle.
The understanding we are beginning to develop of the structure of the inverse spectral map for Jacobi matrices implies that

- for the QR algorithm with Hermite–1 initial data the evolution of $q(t)$, the first row of the eigenvector matrix, is such that with high probability a small number of coefficients becomes very small as compared to the rest.

- for the Toda algorithm with Hermite–1 initial data $q(t)$ two scenarios are highly likely: Either a small number of coefficients becomes very small compared to the rest as in the QR case, or a small number of coefficients becomes very large very quickly (compared to the rest). Recall the perceived asymmetry of the hitting index distributions: We believe that this asymmetry can be explained by concentration of measure on the positive orthant of the sphere. This orthant has most mass close to the vertices, but according to our theorem in section 8.2.1 points close to the vertices correspond to Jacobi matrices 'nearly deflated' at the top.

- for the 'splitting the atom algorithm' the result in section 8.2.1 implies that $q(t)$ is highly likely to separate into two groups of coefficients, each group roughly of

the same size so that within the groups the coefficients are roughly of the same size whereas one group of coefficients is much smaller relative to the other.

Each of these three behaviors can intuitively be explained based on the Hermite–1 initial distribution and the explicit solution

$$q(t) = \frac{\exp\{tG(\Lambda)\}q_0}{\|\exp\{tG(\Lambda)\}q_0\|} \tag{8.14}$$

where G is associated with the respective flow:

- in the QR case $q(t)$ evolves according to the above formula where $G = \log$. Clearly, the coefficients of q which correspond to eigenvectors of small eigenvalues closest in absolute value to zero will become small much quicker than the other eigenvalues.
- in the Toda case we have $G = \mathrm{id}$. In this case both the coefficients of q corresponding to largest and the smallest eigenvalues are the ones that are most likely to separate from the others in scale.
- in the case of $G = \mathrm{sign}$, the splitting the atom case, clearly all coefficients corresponding to positive eigenvalues decay with the same speed, as do the ones corresponding to negative eigenvalues. These two groups are roughly of the same size and they separate with exponential speed as t becomes large.

Let π_t denote a the permutation that orders $q(t)$ by decreasing size and let

$$\text{maxgap } q(t) = \max_{1\le i\le n-1} \frac{q_{\pi_t(i)}}{q_{\pi_t(i+1)}} \tag{8.15}$$

$$\text{argmaxgap } q(t) = \arg\max_{1\le i\le n-1} \frac{q_{\pi_t(i)}}{q_{\pi_t(i+1)}} \tag{8.16}$$

denote For the further study of this problem we believe it would be interesting to investigate properties of the quantity

$$p_\epsilon(t,k) := \mathbb{P}\left(\text{argmaxgap } q(t) = k, \text{maxgap } q(t) \geq \epsilon^{-1}, \forall s < t : \text{maxgap } q(s) < \epsilon^{-1}\right) \tag{8.17}$$

which describes the probability that t was the first time at which the group of largest coefficients was separated from the group of smallest coefficients by a factor of ϵ and that the size of the group of larger coefficients was exactly k. Of course $p_\epsilon(t,k)$ could be studied for different initial conditions and different Hamiltonians $\text{tr}\hat{G}$, but a setting that suggests itself would be $G = \text{sign}$ with q_0 distributed uniformly on the positive sphere orthant as well as λ iid uniform in $[-2\sqrt{n}, 2\sqrt{n}]$. An instances of this setting is given in figure 8.1 (splitting the atom), figure 8.2 (Toda) and figure 8.3 (QR).

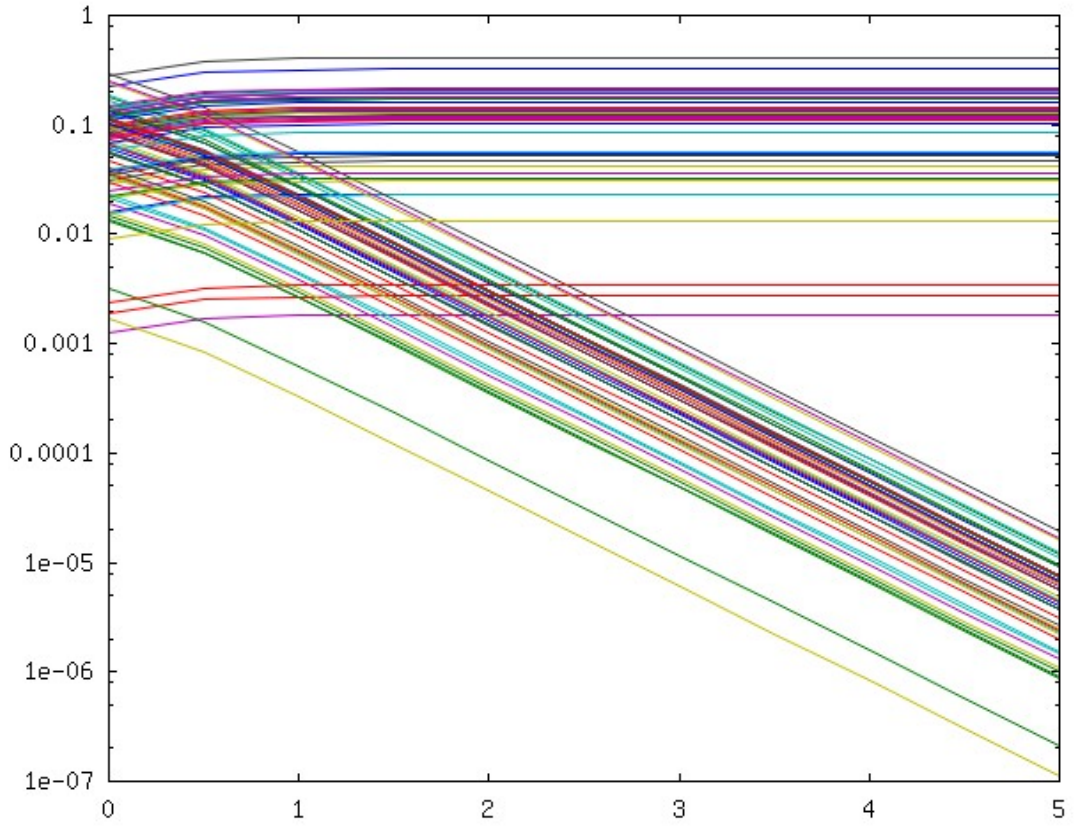

Figure 8.1: Evolution of the spectral coordinate q under the 'splitting the atom' regime. q_0 is distributed uniformly on the positive orthant of the sphere S^{n-1} and λ is distributed iid uniform in $[-2\sqrt{n}, 2\sqrt{n}]$, where $n = 90$.

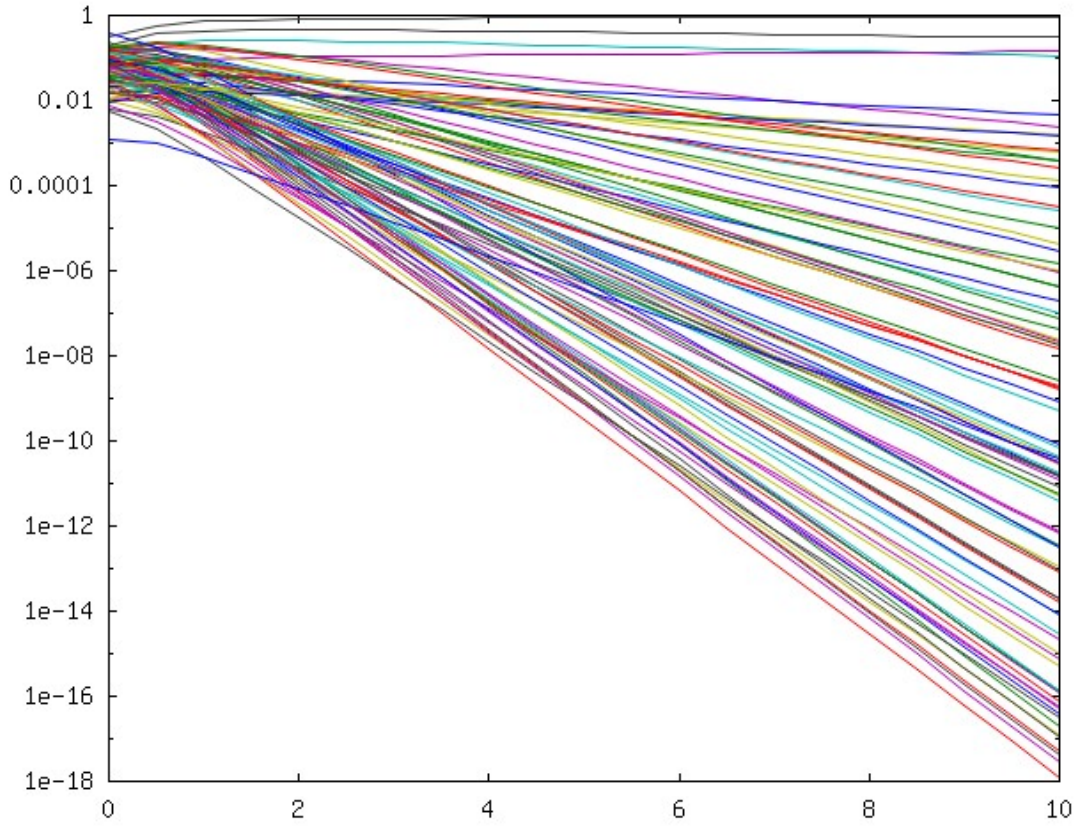

Figure 8.2: Evolution of the spectral coordinate q under the Toda flow. q_0 is distributed uniformly on the positive orthant of the sphere S^{n-1} and λ is distributed iid uniform in $[-2\sqrt{n}, 2\sqrt{n}]$, where $n = 90$.

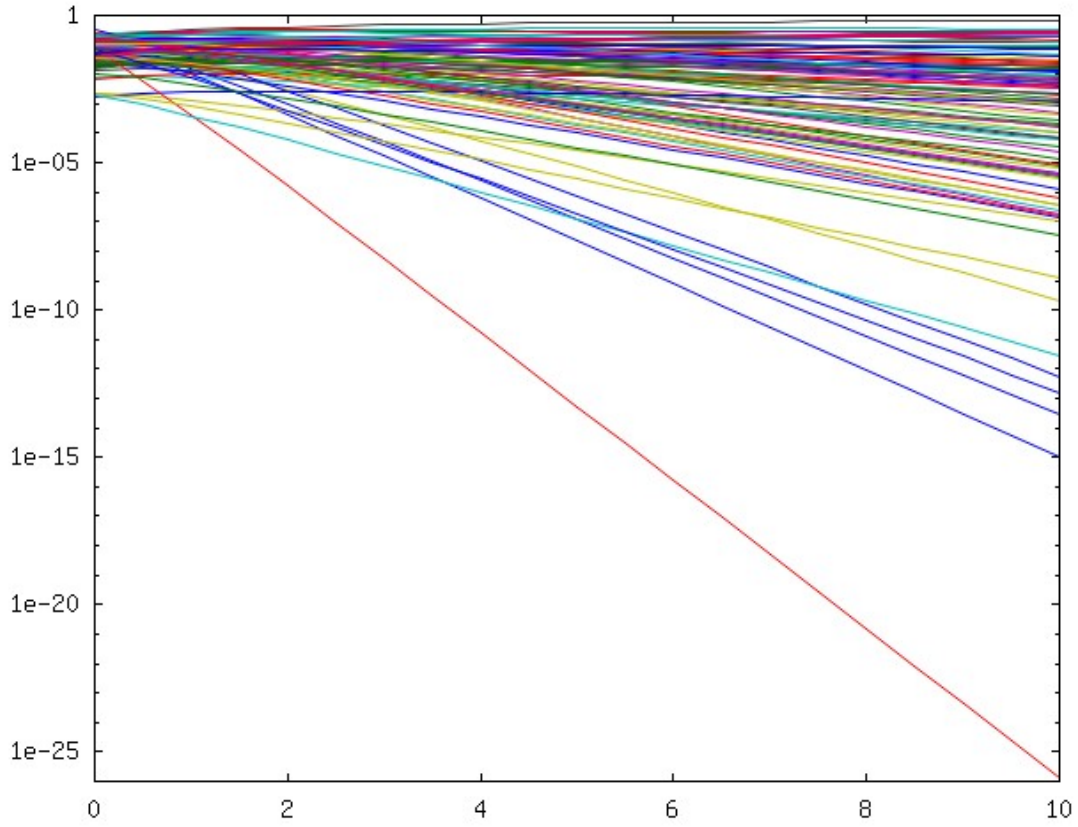

Figure 8.3: Evolution of the absolute value of the spectral coordinate q under the QR flow. q_0 is distributed uniformly on the positive orthant of the sphere S^{n-1} and λ is distributed iid uniform in $[-2\sqrt{n}, 2\sqrt{n}]$, where $n = 90$.

Appendix A

Deferred Proofs

Proof of Proposition 3.3

Proof. First we'll evaluate the uniform measure on the sphere. Let $u \in \mathbb{R}^n$ be a point on $\mathbb{S}^{n-1}$. Then we are looking for a constant $f \equiv f(u)$.

$$f \cdot \int_{\mathbb{S}_{n-1}} dA(u) = f \int_0^{\pi} \cdots \int_0^{2\pi} \left[\mathrm{Jac}(\phi)\right](\theta_1, \ldots, \theta_{n-1})\, d\theta_1 \ldots d\theta_{n-1}$$

where ϕ are the standard spherical coordinates (cf. [32], pp. 326):

$$\phi : \mathbb{R}_+ \times [0,\pi]^{n-2} \times [0, 2\pi] \to \mathbb{S}^{n-1}$$

$$\phi : (r, \theta_1, \ldots, \theta_{n-1}) \mapsto \begin{pmatrix} r\cos\theta_1 \\ r\sin\theta_1\cos\theta_2 \\ \cdots \\ r\sin\theta_1 \cdots \sin\theta_{n-2}\cos\theta_{n-1} \\ r\sin\theta_1 \cdots \sin\theta_{n-2}\sin\theta_{n-1} \end{pmatrix}$$

Its Jacobian is given as:

$$\mathrm{Jac}(\phi) = |\det D\phi| = r^{n-1} \prod_{j=1}^{n-2} \sin^{n-1-j}(\theta_j)$$

Clearly $f = \kappa_{n-1}$ (the surface area of the n–dimensional sphere) and the density that the uniform distribution on $\mathbb{S}_{n-1}$ induces on $(\theta_1, \ldots, \theta_{n-1}) \in [0,\pi]^{n-2} \times [0, 2\pi]$ (via ϕ^{inv}) is given by

$$f(\theta_1, \ldots, \theta_{n-1}) = \kappa_{n-1}^{-1} \cdot \prod_{j=1}^{n-2} \sin^{n-1-j}(\theta_j)$$

Consider now the pushforward of this density under

$$\psi : [0,\pi]^{n-2} \times [0,2\pi] \to \Sigma_{n-1}$$

$$\psi : \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{n-1} \end{pmatrix} \mapsto \begin{pmatrix} \cos^2\theta_1 \\ \sin^2\theta_1 \cos^2\theta_2 \\ \vdots \\ \prod_{j=1}^{N-2} \sin^2(\theta_j)\cos^2(\theta_{n-1}) \end{pmatrix}$$

where Σ_k is the probability simplex for $k+1$ different possible outcomes. Let us restrict ψ to $\psi|_{[0,\pi/2]^{n-1}}$ (corresponding to the intersection of the unit sphere with the positive quadrant of $\mathbb{R}^n$. Clearly there are overall 2^n quadrants). Due to symmetry considerations the density on the restricted set is equal to 2^n times the original density.

Let us evaluate the Jacobian of ψ. Clearly the Jacobi matrix is lower triangular, so it suffices to compute the entries on the diagonal:

$$\frac{\partial \mu_i}{\partial \theta_i} = -2\cos(\theta_i)\sin(\theta_i)\prod_{j=1}^{i-1}\sin^2(\theta_j) \tag{A.1}$$

Thus

$$\mathrm{Jac}(\psi) = 2^{n-1}\prod_{j=1}^{n-2}\left[\sin^2(\theta_j)\right]^{n-1-j}\prod_{j=1}^{n-1}\cos(\theta_j)\prod_{j=1}^{n-1}\sin(\theta_j)$$

The density of the pushforward onto the simplex will be given by:

$$2^n\kappa_{n_1}^{-1}[\mathrm{Jac}(\phi)](\theta_1,\ldots,\theta_{n-1})d\theta_1\ldots d\theta_{n-1} =$$

$$2^n\kappa_{n_1}^{-1}[\mathrm{Jac}(\phi)](\theta_1(\mu_1,\ldots,\mu_{n-1}),\ldots,\theta_{n-1}(\mu_1,\ldots,\mu_{n-1}))$$

$$[\mathrm{Jac}(\psi^{\mathrm{inv}})](\mu_1,\ldots,\mu_{n-1})d\mu_1\ldots d\mu_{n-1} =$$

Since

$$\mathrm{Jac}(\psi^{\mathrm{inv}}) = [\mathrm{Jac}^{-1}(\psi)](\theta_1(\mu_1,\ldots,\mu_{n-1}),\ldots,\theta_{n-1}(\mu_1,\ldots,\mu_{n-1})) \tag{A.2}$$

we have:

$$[\mathrm{Jac}(\phi)](\theta_1(\mu_1,\ldots,\mu_{n-1}),\ldots,\theta_{n-1}(\mu_1,\ldots,\mu_{n-1}))[\mathrm{Jac}(\psi^{\mathrm{inv}})](\mu_1,\ldots,\mu_{n-1}) =$$
$$= \left[2^{n-1}\prod_{j=1}^{n-1}\sin^{n-j}(\theta_j(\mu))\prod_{j=1}^{n-1}\cos(\theta_j(\mu))\right]^{-1}$$

The following is quite easy to prove:

$$\theta_i = \arccos\sqrt{\frac{\mu_i}{1-\sum_{j=1}^{i-1}\mu_i}}$$
$$\cos(\theta_i) = \sqrt{\frac{\mu_i}{1-\sum_{j=1}^{i-1}\mu_i}}$$
$$\sin(\theta_i) = \sqrt{1-\frac{\mu_i}{1-\sum_{j=1}^{i-1}\mu_i}}$$

We now compute:

$$\prod_{j=1}^{n-1}\sin^{n-j}(\theta_j(\mu))\prod_{j=1}^{n-1}\cos(\theta_j(\mu)) =$$
$$= \prod_{i=1}^{n-1}\left[\sqrt{1-\frac{\mu_i}{1-\sum_{j=1}^{i-1}\mu_i}}\right]^{n-i}\prod_{j=1}^{n-1}\sqrt{\frac{\mu_j}{1-\sum_{m=1}^{j-1}\mu_m}} =$$
$$= \sqrt{1-\mu_1}^{n-1}\prod_{i=2}^{n-1}\left[\sqrt{1-\frac{\mu_i}{1-\sum_{j=1}^{i-1}\mu_i}}\right]^{n-i}\frac{\sqrt{\mu_1,\ldots,\mu_{n-1}}}{\prod_{j=2}^{n-1}\sqrt{1-\sum_{m=1}^{j-1}\mu_m}} =$$
$$= \sqrt{1-\mu_1}^{n-2}\prod_{i=2}^{n-1}\left[\sqrt{1-\frac{\mu_i}{1-\sum_{j=1}^{i-1}\mu_i}}\right]^{n-i}\frac{\sqrt{\mu_1,\ldots,\mu_{n-1}}}{\prod_{j=3}^{n-1}\sqrt{1-\sum_{m=1}^{j-1}\mu_m}} =$$

$$= \sqrt{1-\mu_1-\mu_2}^{n-2} \prod_{i=3}^{n-1} \left[\sqrt{1 - \frac{\mu_i}{1-\sum_{j=1}^{i-1}\mu_i}} \right]^{n-i} \frac{\sqrt{\mu_1, \ldots, \mu_{n-1}}}{\prod_{j=3}^{n-1}\sqrt{1-\sum_{m=1}^{j-1}\mu_m}} =$$

$$= \sqrt{1-\mu_1-\mu_2}^{n-3} \prod_{i=3}^{n-1} \left[\sqrt{1 - \frac{\mu_i}{1-\sum_{j=1}^{i-1}\mu_i}} \right]^{n-i} \frac{\sqrt{\mu_1, \ldots, \mu_{n-1}}}{\prod_{j=4}^{n-1}\sqrt{1-\sum_{m=1}^{j-1}\mu_m}} =$$

$$= \sqrt{1-\sum_{k=1}^{l}\mu_k}^{n-(l+1)} \prod_{i=l+1}^{n-1} \left[\sqrt{1 - \frac{\mu_i}{1-\sum_{j=1}^{i-1}\mu_i}} \right]^{n-i} \frac{\sqrt{\mu_1, \ldots, \mu_{n-1}}}{\prod_{j=l+2}^{n-1}\sqrt{1-\sum_{m=1}^{j-1}\mu_m}} =$$

$$\overset{l=n-1}{=} \sqrt{\mu_1, \ldots, \mu_{n-1}} \cdot \sqrt{1-\sum_{k=1}^{n-1}\mu_k}$$

So that overall we obtain the distribution

$$\mathbb{P}(d\mu_1, \ldots, d\mu_{-1}) = 2\kappa_{n-1}^{-1}\mu_1^{-\frac{1}{2}} \cdots \mu_{n-1}^{-\frac{1}{2}} \cdot \left(1-\sum_{j=1}^{n-1}\mu_j\right)^{-\frac{1}{2}} d\mu_1 \cdots d\mu_{n-1}$$

Note that $2\kappa_{n-1}^{-1} = \frac{\Gamma\left(\frac{n}{2}\right)}{\pi^{n/2}} = \frac{\Gamma\left(\frac{n}{2}\right)}{\Gamma^n\left(\frac{1}{2}\right)}$. Thus, what we obtain is exactly the Dirichlet distribution with parameters $\alpha = (\frac{1}{2}, \ldots, \frac{1}{2})$. □

Proof. A simpler computation of this using the Gaussian distribution. We know that

$$\int_{\mathbb{R}^n} e^{-|x|^2/2} dx = (2\pi)^{n/2} \tag{A.3}$$

By converting this integral to spherical coordinates we obtain:

$$(2\pi)^{n/2} = \kappa_n \int_0^\infty e^{-r^2/2} r^{n-1} dr \overset{s=r^2/2}{=} \kappa_n \int_0^\infty e^{-s} \sqrt{2s}^{n-1} \frac{ds}{\sqrt{2s}} =$$

$$= \sqrt{2}^{n-2} \kappa_n \Gamma\left(\frac{n}{2}\right)$$

So that $\kappa_n = \frac{2\pi^{n/2}}{\Gamma\left(\frac{n}{2}\right)}$. Now let's compute the pushforward: Clearly

$$(2\pi)^{n/2} = \int_{\mathbb{R}^n} e^{-|y|^2/2} dy = 2^n \int_0^\infty \cdots \int_0^\infty e^{-|y|^2/2} dy$$

Now substitute $x_i = y_i^2$ to obtain

$$(2\pi)^{n/2} = \int_0^\infty \cdots \int_0^\infty e^{-\frac{1}{2}\sum_{j=1}^n x_j} \prod_{j=1}^n x_j^{-1/2} dx_j \tag{A.4}$$

Apply one last transform of the form $x_i = p_i s$ for $1 \leq i \leq n-1$ and $s = \sum_{j=1}^n x_j$. The Jacobi matrix of the transformation $\phi : (p, s) \mapsto x$ is given by:

$$\begin{pmatrix} s & \cdots & 0 & p_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & s & p_{n-1} \\ -s & \cdots & -s & 1 - \sum_{j=1}^{n-1} p_j \end{pmatrix}$$

the last row holds because $x_n = s\left(1 - \sum_{j=1}^{n-1} p_j\right)$. Overall we obtain $\mathrm{Jac}(\phi) = s^{n-1}$ (by adding the initial $n-1$ rows to the last row), which after being put together with (A.4) yields:

$$(2\pi)^{n/2} = \int_0^\infty \int_{\sum_{j=1}^n p_j = 1} e^{-s/2} s^{n/2-1} \left(1 - \sum_{j=1}^{n-1} p_j\right)^{-1/2} \prod_{j=1}^{n-1} p_j^{-1/2} dp ds \tag{A.5}$$

where $p_n = 1 - \sum_{j=1}^n p_j$. This means that overall (by integrating out the s direction) I obtain the Dirichlet $(1/2, \ldots, 1/2)$–distribution for $s = 1$ as claimed. $\square$

Appendix B

Additional Figures

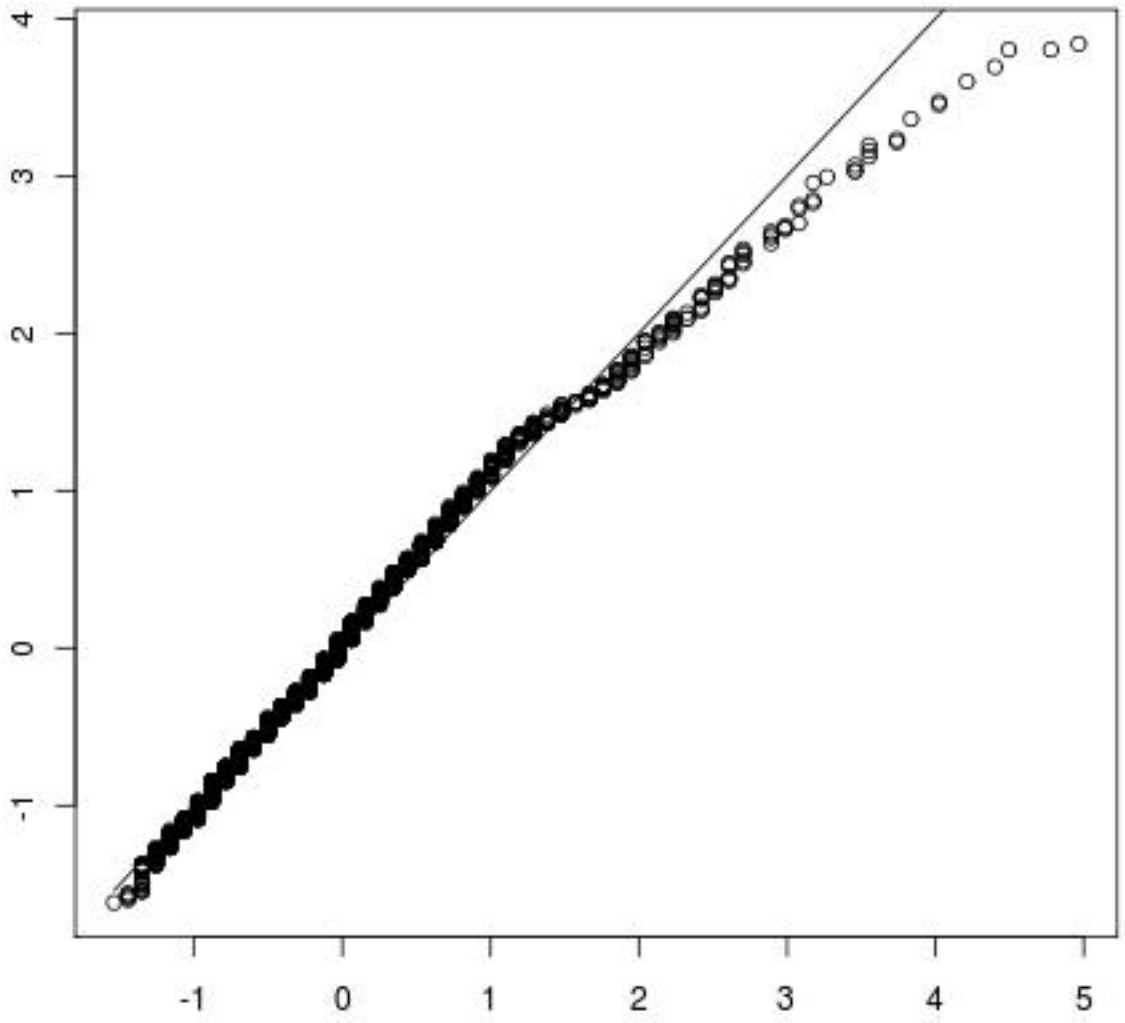

Figure B.1: QQ plot of the normalized runtime of the QR algorithm on GOE initial data of dimension $n = 190$ and deflation tolerance $\epsilon = 10^{-k}$ against the corresponding data for the Toda algorithm on uniform double stochastic Jacobi initial data.

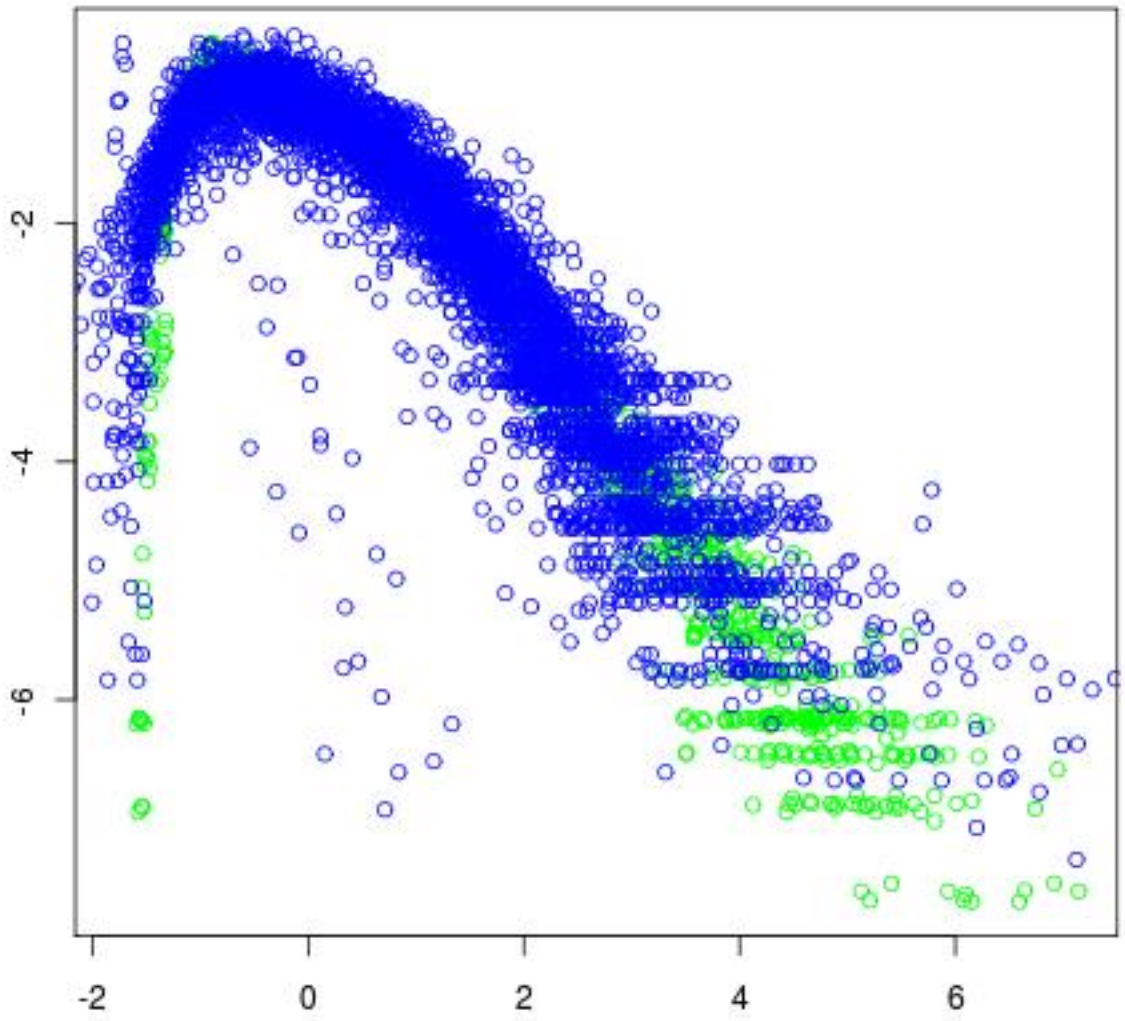

Figure B.2: Combination of all the QR runtime histograms for GOE initial data and all the Toda runtime histograms for uniform doubly stochastic Jacobi initial data.

Bibliography

[1] M. Adler. On a trace functional for formal pseudo-differential operators and the symplectic structure of the korteweg-devries type equations. *Inventiones Mathematicae*, 50:219–248, 1978. 10.1007/BF01410079.

[2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[3] Jinho Baik, Thomas Kriecherbauer, Luen-Chau Li, Kenneth McLaughlin, and Carlos Tomei, editors. *Integrable Systems and Random Matrices*, volume 458 of *Contemporary Mathematics*, chapter Some Open Problems in Random Matrix Theory and the Theory of Integrable Systems, pages 419–430. American Mathematical Society, Providence, First edition, 2008.

[4] G. Ballard, J. Demmel, and I. Dumitriu. Minimizing Communication for Eigenproblems and the Singular Value Decomposition. *ArXiv e-prints*, November 2010.

[5] Steve Batterson. Convergence of the shifted qr algorithm on 33 normal matrices. *Numerische Mathematik*, 58:341–352, 1990. 10.1007/BF01385629.

[6] G. Ben Arous and A. Guionnet. Large Deviations for Wigner's Law and Voiculescu's non–commutative Entropy. *Probability Theory and Related Fields*, 108(4):517–542, August 1997.

[7] R.W. Brockett. Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems. In *Decision and Control, 1988., Proceedings of the 27th IEEE Conference on*, pages 799 –803 vol.1, December 1988.

[8] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[9] Percy Deift. *Orthogonal Polynomials and Random Matrices: A Riemann Hilbert Approach*. Courant Lecture Notes in Mathematics. Courant Institute of Mathematical Sciences, New York, First edition, 1999.

[10] Percy Deift and Dimitri Gioev. Universality at the Edge of the Spectrum for Unitary, Orthogonal and Symplectic Ensembles of Random. *Communications in Pure and Applied Mathematics*, 60(6):867–910, 2007.

[11] Percy Deift and Dimitri Gioev. *Random Matrix Theory: Invariant Ensembles and Universality*. American Mathematical Society, Providence, First edition, 2009.

[12] Percy Deift, C. David Levermore, and C. Eugene Wayne, editors. *Dynamical Systems and Probabilistic Methods in Partial Differential Equations*, volume 31 of *Lecture Notes in Applied Mathematics*, chapter Integrable Hamiltonian Systems, pages 103–138. American Mathematical Society, Providence, First edition, 1996.

[13] Percy Deift, L. C. Li, T. Nanda, and C. Tomei. The Toda Flow on a Generic Orbit is Integrable. *Communications on Pure and Applied Mathematics*, 39(2):183–232, March 1986.

[14] Percy Deift, Luen Chau Li, and Carlos Tomei. Matrix Factorizations and Integrable Systems. *Communications on Pure and Applied Mathematics*, 42:443–521, 1989.

[15] Percy Deift, T. Nanda, and C. Tomei. Ordinary Differential Equations and the Symmetric Eigenvalue Problem. *SIAM Journal on Numerical Analysis*, 20(1):1–22, February 1983.

[16] James W. Demmel. The Probability that a Numerical Analysis Problem is Difficult. *Mathematics of Computation*, 50(182):449–480, April 1988.

[17] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, First edition, 1997.

[18] Persi Diaconis and Philip Matchett Wood. Random Doubly Stochastic Jacobi Matrices. 2010.

[19] Ioana Dumitriu and Alan Edelman. Matrix Models for beta Ensembles. *Journal of Mathematical Physics*, 43(11):5830–5847, November 2002.

[20] Alan Edelman. Eigenvalues and Condition Numbers of Random Matrices. *Siam J. Matrix Anal. Appl.*, 9(4):543–560, October 1988.

[21] William Feller. *An Introduction to Probability Theory and its Applications*, volume 2. John Wiley & Sons, New York, London, Sydney, Toronto, Second edition, 1966.

[22] H. Flaschka. The toda lattice. ii. existence of integrals. *Phys. Rev. B*, 9(4):1924–1925, Feb 1974.

[23] Peter J. Forrester. *Log–Gases and Random Matrices*. Princeton University Press, Princeton, Woodstock, First edition, 2010.

[24] J. G. F. Francis. The QR Transformation A Unitary Analogue to the LR TransformationPart 1. *The Computer Journal*, 4(3):265–271, 1961.

[25] Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1 edition, November 1994.

[26] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore and London, third edition, 1996.

[27] William B. Gragg and William J. Harrod. The Numerically Stable Reconstruction of Jacobi Matrices from Spectral Data. *Numerische Mathematik*, 44(3):317–335, October 1984.

[28] Misha Gromov. *Metric Structures for Riemannian and Non–Riemannian Spaces*, volume 152 of *Progress in Mathematics*. Birkhäuser, Boston, Basel, Berlin, Second edition, 2001.

[29] Alice Guionnet. *Large Random Matrices: Lectures on Macroscopic Asymptotics*, volume 1957 of *Lecture Notes in Mathematics*. Springer, Berlin, Heidelberg, New York, First edition, 2009.

[30] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *ArXiv e-prints*, September 2009.

[31] J. Harnad, C. A. Tracy, and H. Widom. Hamiltonian Structure of Equations Appearing in Random Matrices. *ArXiv High Energy Physics - Theory e-prints*, January 1993.

[32] Anthony Knapp. *Basic Real Analysis*. Cornerstones. Birkhäuser, Boston, Basel, Berlin, First edition, 2005.

[33] Bertram Kostant. The solution to a generalized toda lattice and representation theory. *Advances in Mathematics*, 34(3):195 – 338, 1979.

[34] V. N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *USSR Computational Mathematics and Mathematical Physics*, 1(3):637 – 657, 1962.

[35] Peter D. Lax. Integrals of nonlinear equations of evolution and solitary waves. In Peter Sarnak and Andrew Majda, editors, *Selected Papers Volume I*, pages 366–389. Springer New York, 2005.

[36] Peter D. Lax. *Linear Algebra and Its Applicatoins*. Wiley Interscience, Hoboken, New Jersey, second edition, 2007.

[37] Ricardo Leite, Nicolau Saldanha, and Carlos Tomei. The asymptotics of wilkinsons shift: Loss of cubic convergence. *Foundations of Computational Mathematics*, 10:15–36, 2010. 10.1007/s10208-009-9047-3.

[38] Ricardo S. Leite, Nicolau C. Saldanha, and Carlos Tomei. An atlas for tridiagonal isospectral manifolds. *Linear Algebra and its Applications*, 429(1):387 – 402, 2008.

[39] Madan Lal Mehta. *Random Matrices*. Elsevier, San Diego and London, third edition, 2004.

[40] Cleve Moler and Charles Van Loan. Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty–Five Years Later. *Siam Review*, 45(1):3–49, 2003.

[41] Jürgen Moser, editor. *Dynamical Systems Theory and Applications*, volume 38 of *Lecture Notes in Physics*, chapter Finitely many Mass Points on the Line under the Influence of an Exponential Potential – an Integrable System, pages 467–497. Springer, New York, Berlin, Heidelberg, First edition, 1975.

[42] Jürgen Moser and Eduard J. Zehnder. *Notes on Dynamical Systems*. Courant Lecture Notes in Mathematics. American Mathematical Society, Providence, First edition, 2005.

[43] T. Nanda. Differential Equations and the QR Algorithm. *SIAM Journal on Numerical Analysis*, 22(2):310–321, April 1985.

[44] B. N. Parlett and Jr. W. G. Poole. A geometric theory for the qr, lu and power iterations. *SIAM Journal on Numerical Analysis*, 10(2):389–412, 1973.

[45] Beresford N. Parlett. *The symmetric eigenvalue problem*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.

[46] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[47] N. Raj Rao and Alan Edelman. The polynomial method for random matrices. *Found. Comput. Math.*, 8:649–702, November 2008.

[48] Arvind Sankar, Daniel A. Spielman, and Shang-Hua Teng. Smoothed Complexity of the Condition Numbers and Growth Factors of Matrices. *Siam J. Matrix Anal. Appl.*, 28(2):446–476, 2006.

[49] William W. Symes. Hamiltonian Group Actions and Integrable Systems. *Physica 1D*, pages 339–370, 1980.

[50] William W. Symes. The QR Algorithm and Scattering for the Finite Nonperiodic Toda Lattice. *Physica 4D*, pages 275–280, 1982.

[51] Terence Tao and Van Vu. From the Littlewood–Offord Problem to the Circular Law: Universality of the Spectral Distribution of Random Matrices. *Bulletin (New Series) of the American Mathematical Society*, 46(3):377–396, July 2009.

[52] David S. Watkins. Understanding the QR Algorithm. *SIAM Review*, 24(4):427–440, October 1982.

[53] David S. Watkins. Perspectives on the Eigenvalue Problem. *SIAM Review*, 35(3):430–471, September 1993.

[54] Eugene P. Wigner. On the distribution of the Roots of Certain Symmetric Matrices. *The Annals of Mathematics*, 67(2):325–327, March 1958.

[55] J. H. Wilkinson, editor. *The algebraic eigenvalue problem*. Oxford University Press, Inc., New York, NY, USA, 1988.