

Abstracts of Dissertations

Institute of Information and
Communication Technologies

BULGARIAN ACADEMY OF
SCIENCES



1 / 2026



STOACHASTIC
NUMERICAL METHODS
FOR EIGENVALUE
ESTIMATION

Silvi-Maria T Gurova

СТОХАСТИЧНИ
ЧИСЛЕНИ МЕТОДИ ЗА
ОЦЕНКА НА
СОБСТВЕНИ
СТОЙНОСТИ

Силви-Мария Гюрова

Автореферати на дисертации

Институт по информационни и
комуникационни технологии

БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ

ISSN: 1314-6351

Поредицата „Автореферати на дисертации на Института по информационни и комуникационни технологии при Българската академия на науките“ представя в електронен формат автореферати на дисертации за получаване на научната степен „Доктор на науките“ или на образователната и научната степен „Доктор“, защитени в Института по информационни и комуникационни технологии при Българската академия на науките. Представените трудове отразяват нови научни и научно-приложни приноси в редица области на информационните и комуникационните технологии като Компютърни мрежи и архитектури, Паралелни алгоритми, Научни пресмятания, Лингвистично моделиране, Математически методи за обработка на сензорна информация, Информационни технологии в сигурността, Технологии за управление и обработка на знания, Грид-технологии и приложения, Оптимизация и вземане на решения, Обработка на сигнали и разпознаване на образи, Интелигентни системи, Информационни процеси и системи, Вградени интелигентни технологии, Йерархични системи, Комуникационни системи и услуги и др.

Редактори

Геннадий Агре

Институт по информационни и комуникационни технологии, Българска академия на науките
E-mail: agre@iinf.bas.bg

Райна Георгиева

Институт по информационни и комуникационни технологии, Българска академия на науките
E-mail: rayna@parallel.bas.bg

Даниела Борисова

Институт по информационни и комуникационни технологии, Българска академия на науките
E-mail: dborissova@iit.bas.bg

Настоящото издание е обект на авторско право. Всички права са запазени при превод, разпечатване, използване на илюстрации, цитирания, разпространение, възпроизвеждане на микрофилми или по други начини, както и съхранение в бази от данни на всички или част от материалите в настоящето издание. Копирането на изданието или на част от съдържанието му е разрешено само със съгласието на авторите и/или редакторите

The series Abstracts of Dissertations of the Institute of Information and Communication Technologies at the Bulgarian Academy of Sciences presents in an electronic format the abstracts of Doctor of Sciences and PhD dissertations defended in the Institute of Information and Communication Technologies at the Bulgarian Academy of Sciences. The studies provide new original results in such areas of Information and Communication Technologies as Computer Networks and Architectures, Parallel Algorithms, Scientific Computations, Linguistic Modelling, Mathematical Methods for Sensor Data Processing, Information Technologies for Security, Technologies for Knowledge management and processing, Grid Technologies and Applications, Optimization and Decision Making, Signal Processing and Pattern Recognition, Information Processing and Systems, Intelligent Systems, Embedded Intelligent Technologies, Hierarchical Systems, Communication Systems and Services, etc.

Editors

Gennady Agre

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
E-mail: agre@iinf.bas.bg

Rayna Georgieva

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
E-mail: rayna@parallel.bas.bg

Daniela Borissova

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
E-mail: dborissova@iit.bas.bg

This work is subjected to copyright. All rights are reserved, whether the whole or part of the materials is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in other ways, and storage in data banks. Duplication of this work or part thereof is only permitted under the provisions of the authors and/or editor.



BULGARIAN ACADEMY OF SCIENCES

Abstract of PhD Thesis

STOACHASTIC NUMERICAL METHODS FOR EIGENVALUE ESTIMATION

Silvi-Maria Todorova Gurova

Supervisor: Prof. Aneta Karaivanovava

Approved by Supervising Committee:

Prof. Michail Todorov,
Prof. Nadia Zlateva
Assoc. Prof. Ivan Hristov
Assoc. Prof. Sofiya Ivanovska
Prof. Ivan Georgiev



**INSTITUTE OF INFORMATION AND
COMMUNICATION TECHNOLOGIES**

**Department of Scalable Algorithms and Cybersecurity
with the High-Performance Computing Center**

General Characteristic of the Dissertation

Relevance of the Topic

The problem of finding and estimating extreme eigenvalues of square, in particular symmetric, matrices occupies a central place in linear algebra due to its key role in a wide range of scientific and applied fields. Such problems arise in quantum mechanics, where, in the Schrödinger equation [65], the eigenvalues of the Hamiltonian operator determine the energy spectrum of the system, and the extreme values provide information about the ground and excited states [64, 73]. In financial mathematics, the analysis of extreme eigenvalues of covariance matrices is essential for risk modeling and portfolio optimization, including the assessment of systematic market risk and diversification opportunities [51, 72]. In spectral graph theory, these values are associated with structural properties such as connectivity, expansion, and robustness, which are decisive for the efficiency of algorithms in machine learning and network analysis [8, 12, 26]. In image processing and computer vision, eigenvalues play a crucial role in principal component analysis, dimensionality reduction, noise suppression, and spectral segmentation [47, 34]. In tensor analysis and multidimensional data processing, the maximum eigenvalue and spectral norms determine the behavior of iterative methods and regularization techniques, with applications in signal processing and artificial intelligence [49, 56, 80].

There exist various numerical approaches for computing eigenvalues and eigenvectors: exact (direct), iterative, and stochastic. Direct methods include the QR decomposition and solving the characteristic polynomial for small dimensions. They provide a solution in a finite number of steps and have computational complexity of order $\mathcal{O}(n^3)$. In high dimensions, they suffer from the accumulation of numerical errors due to rounding in arithmetic operations [87, 90].

Iterative methods, such as the power method, inverse power method, Jacobi method, and the Arnoldi and Lanczos methods [33, 43], have lower computational complexity $\mathcal{O}(n^2k)$, where n is the matrix dimension and k is the number of iterations. However, these methods are often characterized by slow convergence, strong dependence on the initial approximation, significant memory requirements, and limited parallel efficiency.

Stochastic numerical methods offer an effective alternative by providing linear dependence of the required memory on the problem dimension, as well as natural parallelism and relative ease of implementation. Although they exhibit slower convergence and lower precision than deterministic methods, they enable the solution of very high-dimensional problems. They are particularly well-suited for estimating extreme eigenvalues of symmetric matrices.

In the context of the rapid development of high-performance computing (HPC) systems and the emergence of new hardware architectures, there is a growing need for the development and deployment of software solutions based on efficient stochastic algorithms for high-dimensional problems, including the estimation of extreme eigenvalues [18, 24, 59]. This has led to increased research interest in improving accuracy, accelerating convergence, and expanding the applicability of stochastic methods in scientific computing, machine learning, and data analysis, aligned with modern computational capabilities.

Overview of the Main Results in the Field

The earliest publications on Monte Carlo (MC) methods in linear algebra date back to the 1950s and 1960s, and they did not initially address the problem of approximate eigenvalue computation. The task of finding the eigenvalues and eigenvectors of matrices usually arises when solving systems of linear algebraic equations (LAEs) and stochastic differential equations (SDEs). In practice, for high-dimensional matrices, it is usually necessary to compute only the extreme eigenvalues, which determine the condition number and key properties of the system [63, 87, 90]. Such problems arise in image and signal processing, control systems, machine learning, and quantum mechanics, where, for example, the energy levels of quantum systems are determined by the eigenvalues of the Hamiltonian operator [64, 73]. In addition to these, deterministic methods such as the power method, the Rayleigh quotient iteration, the resolvent power method, and the Lanczos algorithm are widely used [33, 43].

The Monte Carlo method for estimating the dominant eigenvalue was proposed by Sobol in 1973 [92], while Mikhailov introduced the method for the smallest eigenvalue in 1987 [55]. Significant contributions to the development of the Power Monte Carlo (PMC) method were made by Dimov and Karaivanova during the period 1996-2000 [15, 18, 21, 23, 24, 25]. In the following decades, iterative MC and Quasi-Monte Carlo (QMC) algorithms were developed to find extreme eigenvalues, based on the power and resolvent power methods [3, 19, 21, 25, 60]. In particular, the use of the resolvent matrix to estimate the smallest eigenvalue was introduced by Dimov and Karaivanova in 1998 [24].

The first QMC method for finding extreme eigenvalues were proposed by Karaivanova and Mascagni in the period 2001–2003 [58, 59, 60, 61]. A number of studies have used QMC methods based on low-discrepancy sequences, including the Sobol sequences [74, 75, 76], the Halton sequences [39, 40], and the Faure sequences [28, 29, 30], providing a more uniform and deterministic coverage of the sample space [2, 3, 10, 11, 44, 48]. Compared to MC approaches, QMC methods are distinguished by greater flexibility, applicability to large-scale matrices, relative simplicity, and efficient parallelization [16, 22, 61].

Over the past two decades, various strategies for parallelizing MC algorithms have been developed and analyzed [4, 5, 6, 78, 79, 83]. Parallel MC methods for solving systems of linear algebraic equations based on the Jacobi iteration have been presented by Dimov [16], as well as approaches using minimal Markov chains and minimal communication [27, 82]. A key idea in these developments is the balancing of stochastic and systematic errors [16]. Hybrid MC algorithms for matrix computations have also been developed [1, 84], including parallel implementations to estimate the largest and smallest eigenvalues of matrices proposed by Dimov, Mascagni and Karaivanova [20, 24, 59].

Goals and Tasks of the Dissertation

The main goal of the dissertation is to propose and investigate stochastic numerical methods for estimating eigenvalues and to develop and analyze Monte Carlo and randomized Quasi-Monte Carlo algorithms to approximate the extreme eigenvalues of symmetric square matrices. To achieve the main goal, the following tasks have been formulated.

1. Consider and study the Power (Resolvent) Monte Carlo method and the randomized Power (Resolvent) QMC method for estimating the extreme eigenvalues of symmetric matrices, and analyze the balance between the resulting stochastic and systematic errors.
2. Develop and investigate efficient Monte Carlo and randomized Quasi-Monte Carlo algorithms based on these stochastic methods for numerically calculating the extreme eigenvalues of symmetric square matrices, using generators of pseudorandom numbers and low-discrepancy sequences.
3. Conduct numerical experiments with the developed algorithms, justifying the choice of specialized software for generating low-discrepancy sequences, in particular, Sobol and Halton sequences, and to numerically investigate the balance between stochastic and systematic error depending on their defining parameters.
4. Formulate and investigate a problem in the field of financial mathematics related to the management of the market risk of an asset portfolio, in which the risk assessment is based on the determination of the maximum eigenvalue of a correlation matrix. Apply the Power Monte Carlo (PMC) and randomized Power Quasi-Monte Carlo (PQMC) algorithms developed in Chapter 1 to numerically estimate the largest eigenvalue and to analyze their effectiveness in assessing market risk.

Research Methodology

The research methodology in this dissertation is based on fundamental scientific results in the following areas:

- Probability theory and mathematical statistics - probability spaces; random variables and their characteristics such as distribution functions, probability density function, mathematical expectation, variance; law of large numbers; central limit theorem; probability error, confidence interval, Markov chain; samples of random variables; statistical tests for studying generators of pseudorandom variables.
- Mathematical analysis - series and sequences, convergence of series, convergence criteria, systematic error, norms, and metric spaces.
- Numerical methods, linear and computational algebra - iterative methods for SLAE and eigenvalues; matrix norms; optimization methods; algorithms for generating pseudorandom numbers, stochastic numerical methods (Monte Carlo and Quasi-Monte Carlo methods).
- Number theory, more precisely, discrete and analytic number theory - uniformly distributed sequences, discrepancy, radical-inverse functions, decomposition into prime bases.
- Computer science - algorithms and data structures; programs/codes written in MATLAB, C++ using pseudo-random number generators such as MT and MS; libraries for randomized Sobol and Halton sequences.

Part of the numerical experiments were performed on the high-performance cluster, which consists of 12 Fujitsu Primergy RX 2540 M4 servers. Each server is configured with an NVIDIA Tesla V100 32GB GPU and two x Intel Xeon Gold 5118 2.30 GHz, 24 CPU cores, with 128 GB RAM. The Sobol sequences were obtained using BRODA's Sobol Randomized Sequence Generator (RSG) [7].

In the following sections, we have formulated the basic problem of estimating eigenvalues of a symmetric matrix and described the MC and QMC approaches for numerically solving multidimensional problems, along with the errors that characterize these methods.

Problem Formulation

Consider a matrix $A = \{a_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$, with $a_{ij} = a_{ji}$, $i, j = 1, \dots, n$. The problem of finding the eigenvalues $\lambda(A)$ is formulated as follows:

$$A\mathbf{x} = \lambda\mathbf{x}, \text{ where } \mathbf{x} \in \mathbb{R}^n \text{ is an arbitrary nonzero vector.} \quad (1)$$

All eigenvalues of symmetric matrices are real numbers. Without loss of generality, the following ordering is assumed:

$$\lambda_{min} = |\lambda_n| \leq |\lambda_{n-1}| \leq \dots \leq |\lambda_2| < |\lambda_1| = \lambda_{max}. \quad (2)$$

The matrix A can also be represented in vector form as:

$$A = \{a_{ij}\}_{i,j=1}^n = (\mathbf{a}_1, \dots, \mathbf{a}_i, \dots, \mathbf{a}_n)^T, \text{ where } \mathbf{a}_i = (a_{i1}, \dots, a_{in}), i = 1, \dots, n.$$

The symbol T denotes the transpose operation. For this study, the following vector and matrix norms are used:

$$\|\mathbf{h}\| = \|\mathbf{h}\|_1 = \sum_{i=1}^n |h_i|, \quad \|\mathbf{a}_i\| = \|\mathbf{a}_i\|_1 = \sum_{j=1}^n |a_{ij}|, \quad \|A\| = \|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|, \quad j = 1, \dots, n.$$

The application of stochastic numerical methods to estimate the extreme eigenvalues λ_{min} and λ_{max} of the problem defined above involves developing and analysing Power (Resolvent) Monte Carlo and randomized Power (Resolvent) Quasi-Monte Carlo algorithms for their approximate computation. By stochastic numerical methods, we will further understand Monte Carlo methods and their variants, the randomized Quasi-Monte Carlo methods.

Monte Carlo Methods

Monte Carlo (MC) methods are based on the construction of a random variable (r.v.) θ , which is in fact a measurable function $\theta = \theta(\omega)$ defined on the probability space $(\Omega, \mathfrak{A}, \mathbb{P})$ [88], with domain Ω and range $\mathbb{R}$ (the set of real numbers). The condition in constructing the random variable θ is related to its mathematical expectation. The requirement is that the mathematical expectation should equal the desired solution to a given problem or a functional of the solution to the original problem.

$$E[\theta] = I, \quad (3)$$

where I is the sought solution. By definition, $E[\theta]$ exists if and only if $E|\theta|$ exists.

Depending on the specific problem, the random variable θ may be discrete or continuous. In the discrete case, the expectation is given by:

$$E[\theta(\omega)] = \sum_i \mathbb{P}\{\theta(\omega) = x_i\} x_i = \sum_i p_i x_i = I, \text{ where } \sum_i p_i = 1,$$

and in the continuous case:

$$E[\theta] = \int_{-\infty}^{\infty} x p(x) dx, \text{ where } \int_{-\infty}^{\infty} p(x) dx = 1,$$

where $p(x) \geq 0$ is the probability density function. The distribution function $F_\theta(x) = \mathbb{P}\{\theta < x\}$ describes the cumulative probability and is related to $p(x)$ through standard differential relationships [88, 91].

When implementing an MC algorithm, a sample $\theta_1, \theta_2, \dots, \theta_N, \dots$ is generated from identically distributed independent realizations of the random variable θ , and the approximate value of I is determined by the mean:

$$\bar{\theta}_N = \frac{1}{N} \sum_{i=1}^N \theta_i. \quad (4)$$

Convergence and Error Estimation in Monte Carlo Methods

The theoretical justification of MC methods is based on the Law of Large Numbers (LLN). When the mathematical expectation is finite, the sequence of mean values $\bar{\theta}_N$ converges in probability to I as $N \rightarrow \infty$ (The Weak Law of Large Numbers, Khinchin's theorem) [14, 88]. The necessary and sufficient condition for the convergence to be almost sure (Strong Law of Large Numbers, Kolmogorov's theorem) is that there is a mathematical expectation.

Suppose that the random variable θ has finite variance, i.e.

$$\text{Var}[\theta] = E[\theta - E\theta]^2 = E[\theta^2] - (E[\theta])^2 < \infty.$$

Then the Central Limit Theorem (CLT) ensures an asymptotically normal distribution of the error [31, 88]. As a result, the following probabilistic estimate is obtained:

$$\lim_{N \rightarrow \infty} \mathbb{P} \left\{ \left| \frac{1}{N} \sum_{i=1}^N (\theta_i - I) \right| < x \sigma N^{-1/2} \right\} = \Phi(x),$$

where $\sigma = \sqrt{\text{Var}[\theta]}$, and $\Phi(x) = \frac{2}{\sqrt{2\pi}} \int_0^x e^{-\frac{t^2}{2}} dt$ is called the probability integral and is determined by the distribution function of a standard normal random variable in the interval $(0, 1)$ [92].

The quantity

$$r_N = x_\beta \sigma N^{-\frac{1}{2}} \quad (5)$$

is called the probability error, with the parameter β determining the level of confidence. At $\beta = 0.5$, the so-called probability error is obtained, which characterizes the typical convergence rate of MC methods of the order of $\mathcal{O}(N^{-1/2})$.

In practical applications, the variance of the random variable θ is empirically estimated by the formula:

$$\text{Var}[\theta] \approx \frac{1}{N} \sum_{i=1}^N (\theta_i)^2 - (\bar{\theta}_N)^2. \quad (6)$$

When the sample size N of random variable values is small, the more accurate formula is usually used [92]:

$$\text{Var}[\theta] \approx \frac{1}{(N-1)} \sum_{i=1}^N (\theta_i)^2 - \frac{N}{N-1} (\bar{\theta}_N)^2. \quad (7)$$

Two types of MC methods are distinguished to reduce error [22, 89, 92]:

- Efficient MC methods, which aim to minimize the variance through variance reduction techniques (separation of a principal part, symmetrization of the subintegral function, essential sample method, importance partitioning method, etc.), without changing the asymptotic order of the error.
- Superconvergent MC methods, in which by sampling the values of a random variable in uniform non-overlapping subdomains and under additional conditions for smoothness of the subintegral function, a higher order of convergence is achieved $r_N = \mathcal{O}(N^{-1/2-\varepsilon(k)})$, where $\varepsilon(k) > 0$ depends on the dimensionality of the problem.

Pseudorandom number generators are used to simulate the values of random variables.

Pseudorandom Number Generators

The practical implementation of MC methods requires the use of pseudorandom number generators (PRNGs), which generate uniformly distributed values in $(0, 1)$ [53, 81]. The main requirements for good PRNGs include statistical randomness, a long period, reproducibility, efficiency, and the ability to generate in parallel [50, 53]. Among the most commonly used generators are linear and combined congruential generators [32, 54, 71], shift register generators (e.g., Mersenne Twister [62]), and generators based on other mathematical constructs

(e.g., Middle Square PRNG [70]). In the dissertation research for the implementation of stochastic algorithms for estimating extreme eigenvalues, Mersenne Twister (MT) and Middle Square (MS) pseudorandom number generators were used.

Quasi-Monte Carlo Methods

Quasi-Monte Carlo (QMC) methods are a deterministic alternative to Monte Carlo (MC) methods and use low-discrepancy sequences instead of pseudorandom points [39, 66, 74]. They provide a more uniform coverage of the unit hypercube $[0, 1]^k$ and lead to improved convergence rates of order $\mathcal{O}(N^{-1}(\log N)^k)$, provided the integrand is sufficiently smooth [9, 52]. When estimating multidimensional integrals, the approximate solution has the form:

$$\int_{[0,1]^k} f(x)dx \approx \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i), \quad (8)$$

where $\mathbf{x}_i$ are elements of a low-discrepancy sequence.

Low-Discrepancy Sequences and Error Estimation

The key characteristic of these sequences is the star-discrepancy $D_N^*(S)$, which measures the deviation from a uniform distribution:

$$D_N^*(S) = \sup_{\mathbf{z} \in [0,1]^k} \left| \frac{1}{N} \sum_{i=1}^N \mathbf{1}_Z(\mathbf{x}_i) - \prod_{j=1}^k z_j \right|. \quad (9)$$

A sequence is said to have low-discrepancy if

$$D_N(S) = \mathcal{O}(N^{-1}(\log N)^k) \text{ or } D_N(S) \leq C_k \frac{(\log N)^k}{N}, \quad (10)$$

where the constant C_k in (10) depends only on the dimension k .

The error in Quasi-Monte Carlo (QMC) methods is estimated using the Koksma-Hlawka inequality [9, 72]:

$$\left| I(f) - \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \right| \leq V(f) D_N^*(S), \quad (11)$$

where $D_N^*(S)$ is the star-discrepancy of the sequence $S = \{\mathbf{x}_i\}_{i=1}^\infty \in [0, 1]^k$. $V(f)$ is the variation of the function in the sense of Hardy-Krause. This inequality shows that the error depends on both the smoothness of the integrand and on the uniformity of the point sequence used. For sufficiently smooth functions, an asymptotic error order of $\mathcal{O}(N^{-1}(\log N)^k)$ is achieved, which is theoretically better than that of classical MC methods.

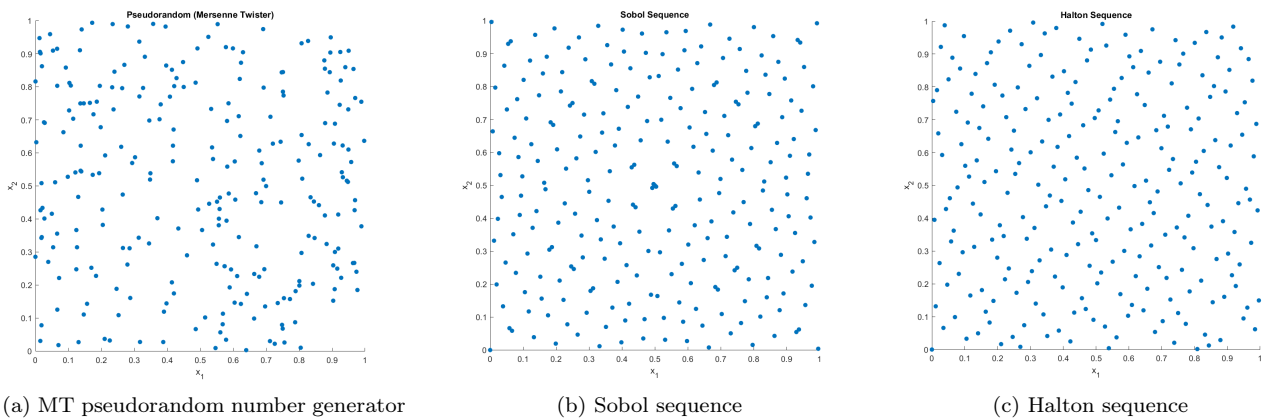


Figure 1: Comparison of a two-dimensional pseudorandom sequence generated with an MT PRNG with two-dimensional projections of the 16-dimensional Sobol and Halton sequences. 256 points are generated, and the projection is made along the first two coordinates.

The first constructions of low-discrepancy sequences originate from the van der Corput sequence [85]. Halton extended this idea to the multidimensional case by using radical inversion in different prime bases for each coordinate [39]. Halton sequences are conceptually simple, but in higher dimensions they may exhibit correlations, which necessitates the use of modifications [41, 86]. Sobol sequences, introduced in 1976 [76], are digital (t, k) -sequences in base 2. They are constructed using direction numbers derived from primitive polynomials over the finite field $\mathbb{F}_2 = \{0, 1\}$ (with addition modulo 2), which ensures high uniformity and good projection properties [74, 89]. Due to these properties, Sobol sequences are among the most widely used in QMC applications.

Figure (1) presents a comparison between a two-dimensional pseudorandom sequence generated by the Mersenne Twister (MT) and two-dimensional projections of 16-dimensional Sobol and Halton sequences. A clearly more uniform coverage of the unit square is observed for the low-discrepancy sequences compared to the pseudorandom sequence.

Scrambled Low-Discrepancy Sequences

The deterministic nature of classical low-discrepancy sequences leads to the absence of a natural probabilistic framework for error estimation and to the risk of correlations in high dimensions. These drawbacks are addressed by introducing scrambled (randomized) low-discrepancy sequences, in which controlled stochasticity preserves the uniform coverage of the unit hypercube while overcoming the limitations of fixed geometry [13].

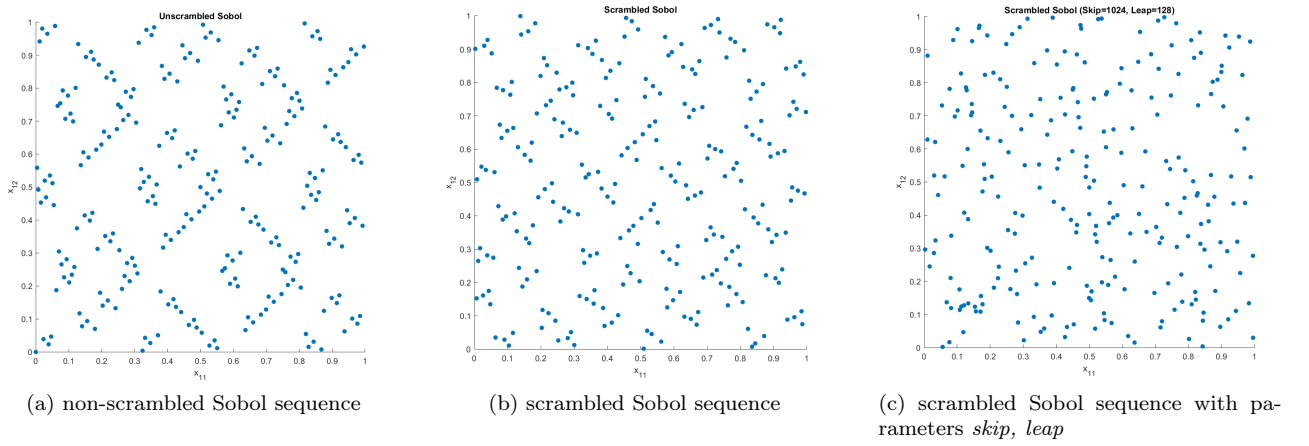


Figure 2: Comparison between a two-dimensional projection with the presence of correlation in the 11th and 12th coordinates of the Sobol sequence and the corresponding two-dimensional projections of the scrambled Sobol sequence without and with the additionally included the parameters $skip = 1024$ and $leap = 128$.

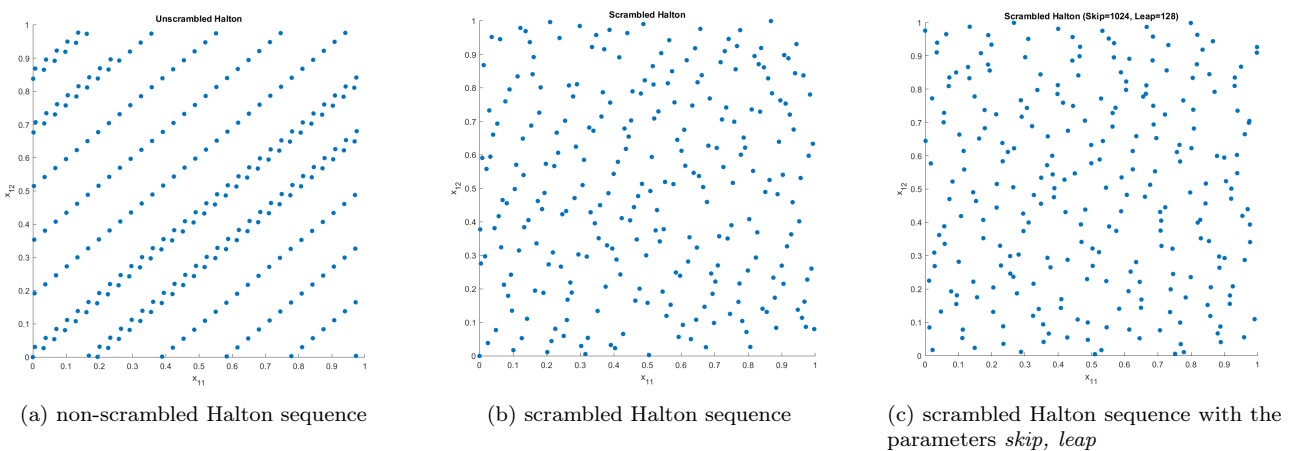


Figure 3: Comparison between a two-dimensional projection with the presence of correlation at the 11th and 12th coordinates of the Halton sequence and the corresponding two-dimensional projections of the scrambled Halton sequence without and with the additional the parameters $skip = 1024$ and $leap = 128$.

In addition to scrambled low-discrepancy sequences, techniques can be used to skip initial elements of the sequence by applying a skipping operator and a leaping operator. In this way, the low-discrepancy property is preserved, while introducing controlled randomness, allowing statistical error estimation.

Figures (2) and (3) demonstrate the presence of correlations in non-scrambled Sobol and Halton sequences in higher coordinates, as well as the effect of applying the scrambling method. Scrambling leads to a significant improvement in the uniformity of the point distribution, as shown in Figure (2b) and Figure (3b). Figure (2c) and Figure (3c) also illustrate the additional effect of combining scrambling with the *skip* and *leap* parameters. These techniques reduce correlations and improve the practical performance of sequences in higher-dimensional problems.

Modern modifications of the Halton and Sobol sequences include randomization aimed at reducing correlations and improving convergence [52, 66] for specific problems. Randomization is implemented by applying specially constructed transformations (scramblings) to already established low-discrepancy sequences. There are two main approaches to scrambling. The first is based on random permutations of digits, where the digits of the coordinates in the base b (in particular, the base 2 in binary representation) are subjected to random permutations [67]. The second approach is based on addition modulo 1 with a random number or a random vector generated by a pseudorandom generator. This method is also known as random shifting (or shift) modulo 1 and is characterized by a simple and efficient implementation [45, 46]. In practice, the two approaches are often combined, leading to numerous variants and hybrid techniques [69, 68].

Stochastic Methods with Markov Chains

Monte Carlo (MC) and Quasi-Monte Carlo (QMC) methods are conventionally classified into two main groups: direct and iterative.

Direct MC/QMC methods are characterized by the fact that the approximate solution is influenced solely by stochastic (statistical) error. In the classical MC approach, this error is of order $\mathcal{O}(N^{-1/2})$ and is estimated via the inequality (5). In contrast, in QMC methods it is described by the Koksma–Hlawka inequality (11) and has asymptotic order $\mathcal{O}(N^{-1}(\log N)^k)$ when low-discrepancy sequences such as Sobol and Halton sequences are used. In this dissertation, randomized low-discrepancy sequences are employed to achieve more uniform coverage of the unit cube while maintaining controlled stochasticity. Therefore, the errors estimated by (5) and (11) are interpreted as stochastic errors depending on the sample size N .

Iterative MC/QMC methods, in addition to stochastic error, also include a systematic error, since they do not estimate the exact solution directly but rather a finite iteration of a convergent iterative process. As with classical numerical methods in linear algebra, such as the simple iteration method and the Gauss–Seidel method [90], the systematic error depends on the properties of the iterative scheme and the norm of the matrix involved. In stochastic iterative methods for linear algebra problems, a random variable $\theta^{(k)}$ is constructed by a Markov chain with k transitions [16, 22, 92]. The number of transitions can be fixed or random, with a mathematical expectation of k . The mathematical expectation of the random variable $\theta^{(k)}$ approximates the desired solution by a functional involving matrix-vector products. In this context, the systematic error is determined by the number of iterations k , while the stochastic error depends on the sample size N . Therefore, balancing systematic and stochastic errors and reducing variance are fundamental research tasks in the development of efficient stochastic iterative methods and algorithms for linear algebra.

Structure of the Dissertation

The dissertation is structured into an introduction, three chapters, a conclusion, an appendix, and a bibliography. It contains 118 pages, including 15 figures and 16 tables, a bibliography of 121 references, and a list of mathematical notation and abbreviations.

- The **Introduction** presents the relevance of the topic, provides an overview of the main scientific results in the field, defines the aims and objectives of the dissertation, and describes in detail the research methodology.
- **Chapter 1** "Stochastic Power Method: Development and Analysis of Efficient Algorithms for Estimating the Maximum Eigenvalue" is devoted to developing efficient stochastic power algorithms for estimating the maximum eigenvalue of dense symmetric matrices. The problem is defined, and the methods used to solve it are considered: the stochastic Power Method in two variants, Power Monte Carlo (PMC) and Power Quasi-Monte Carlo (PQMC). The construction of a random variable using a Markov chain is described in detail, with emphasis on the use of almost optimal probabilities. The pseudocode of the almost optimal PMC and PQMC algorithms developed is presented and conditions are determined to balance stochastic

and systematic errors. The chapter concludes with a discussion of numerical experiments conducted on matrices of different dimensions.

- **Chapter 2** "Stochastic Resolvent Method: Development and Analysis of Efficient Algorithms for Estimating the Minimum Eigenvalue" addresses the problem of estimating the minimum eigenvalue of symmetric square matrices using their resolvent matrix. The Resolvent Power Method and its stochastic implementations via Resolvent Monte Carlo (RMC) and Resolvent Quasi-Monte Carlo (RQMC) approaches are presented. The chapter investigates the role of parameters affecting the convergence of the infinite series representing the resolvent matrix. Almost optimal RMC and RQMC algorithms are constructed and the balance between stochastic and systematic errors is analyzed. This balance depends on the number of realizations of the random variables, the length of the Markov chain, the degree of the resolvent matrix, and the acceleration parameter included in the power series representation of the resolvent. The final section presents numerical experiments related to estimating the minimum eigenvalue of test symmetric matrices.
- **Chapter 3** "Stochastic Methods for Estimating the Market Risk of Investment Portfolios" is devoted to applying the algorithms developed in Chapter 1 to a real-world problem in financial mathematics: estimating the market risk of an investment portfolio. This is achieved through the maximum eigenvalue of the portfolio's correlation matrix. The largest eigenvalue is computed using the almost optimal PMC and PQMC algorithms. When applying the almost optimal PQMC algorithm, two variants are considered: (1) using scrambled Sobol and Halton sequences with default settings; (2) using scrambled Sobol and Halton sequences with the *skip* and *leap* parameters. Numerical experiments on real financial data demonstrate the applicability and efficiency of the proposed algorithms.

The concluding section includes a list of publications, approbation, scientific and applied contributions, and acknowledgements.

1. Stochastic Power Method: Development and Investigation of Efficient Algorithms for Estimating the Maximum Eigenvalue

1.1 Problem Formulation

Let $A = \{a_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$ be a given nonsingular symmetric matrix. The eigenvalues (spectrum) of the matrix A are defined as the real values of the parameter $\lambda(A)$ for which the equation:

$$A\mathbf{x} = \lambda(A)\mathbf{x}, \text{ where } \mathbf{x} \in \mathbb{R}^n, \quad (1.1)$$

has a nonzero solution, and the corresponding nonzero vectors $\mathbf{x}$ are called eigenvectors of A . A necessary and sufficient condition for the existence of a nonzero solution is that the parameter $\lambda(A)$ be a root of the characteristic equation $|A - \lambda(A)\mathbf{I}| = 0$, where $\mathbf{I}$ is the unit n -dimensional matrix [90].

It is assumed that the eigenvalues of A are ordered by absolute value and satisfy the inequalities:

$$\lambda_{min} = |\lambda_n| \leq |\lambda_{n-1}| \leq |\lambda_{n-2}| \leq \dots \leq |\lambda_2| < |\lambda_1| = \lambda_{max}. \quad (1.2)$$

We consider the task of developing and investigating efficient algorithms based on the Stochastic Power method to estimate λ_{max} , which, according to (1.2), is unique.

1.2 Methods for Estimating the Largest Eigenvalue

1.2.1 Power Method

The Power method is a *classical* iterative numerical method for approximating the maximum eigenvalue λ_{max} of square matrices [33, 43], based on the Rayleigh quotient [43]:

$$r(\mathbf{x}) = \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}}.$$

The algorithm is implemented by initializing with a normalized initial vector, iteratively updating the vector and evaluating the eigenvalue after k iterations, i.e.

$$\lambda_{max} \approx \lambda_{max}^{(k)} = r(\mathbf{x}_k).$$

According to conditions (1.2), the obtained approximation corresponds to the dominant eigenvalue.

Systematic error: In the case of a symmetric matrix, the error of the truncation of the Power method when representing the vectors $\mathbf{x}^T$ and $\mathbf{x}$ as linear combinations of an orthonormal basis is of the order:

$$|\lambda_{max}^{(k)} - \lambda_{max}| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right), \quad (1.3)$$

which shows a strong dependence on the spectral distance between the first two eigenvalues [43].

Convergence: The convergence rate of the iteration process for symmetric matrices is determined by the coefficient $\left|\frac{\lambda_2}{\lambda_1}\right|^2$, and for dominant eigenvalues in the modulus, the convergence slows down and may require the application of preprocessing techniques.

Computational complexity: The computational complexity of the Power method is $\mathcal{O}(kn^2)$, where k is the number of iterations and n is the dimension of the matrix A , which makes it suitable for large but structured symmetric problems.

1.2.2 Stochastic Power Method

The stochastic Power method, studied in the dissertation, was developed by Karaivanova and Dimov [21, 22] and is a stochastic implementation of the *classical* Power method for the eigenvalue problem (1.1) under condition (1.2). The basis of the method is the representation of the largest eigenvalue as a limit of the Rayleigh quotient:

$$\lambda_{max} = \lim_{k \rightarrow \infty} \frac{(\mathbf{h}, A^k \mathbf{f})}{(\mathbf{h}, A^{k-1} \mathbf{f})}, \quad (1.4)$$

where $\mathbf{f}$ and $\mathbf{h}$ are arbitrarily chosen and normalized vectors. If k is an arbitrarily large natural number, then

$$\lambda_{max}^{(k)} = \frac{(\mathbf{h}, A^k \mathbf{f})}{(\mathbf{h}, A^{k-1} \mathbf{f})} \quad (1.5)$$

is an approximation of λ_{max} with a certain systematic error.

The stochastic Power method is implemented by constructing random variables whose mathematical expectations coincide with the scalar products in the numerator and denominator of the above relation. To estimate $(\mathbf{h}, A^k \mathbf{f})$, a random variable $\theta^{(k)}$ is defined, based on a discrete finite Markov chain with states:

$$l_0 \rightarrow \dots \rightarrow l_t \rightarrow \dots \rightarrow l_k \quad (1 \leq l_t \leq n, \quad 0 \leq t \leq k), \quad (1.6)$$

with an initial density vector $\mathbf{p} = \{p_i\}_{i=1}^n \in \mathbb{R}^n$ and a transition density matrix $P = \{p_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$. The construction involves iterative calculation of the weight W_k :

$$W_0 = 1, \quad W_t = W_{t-1} \frac{a_{l_{t-1} l_t}}{p_{l_{t-1} l_t}}, \quad t = 1, \dots, k, \quad (1.7)$$

and a random variable $\theta^{(k)}$ is defined by

$$\theta^{(k)} = \frac{h_{l_0}}{p_{l_0}} W_k f_{l_k}. \quad (1.8)$$

The random variable $\theta^{(k)}$ defined by (1.8) has an expectation equal to the scalar product $(\mathbf{h}, A^k \mathbf{f})$, i.e. $E[\theta^{(k)}] = (\mathbf{h}, A^k \mathbf{f})$ [17].

Definition 1.2.1. The initial density vector $\mathbf{p} = \{p_i\}_{i=1}^n \in \mathbb{R}^n$ is called permissible for a vector $\mathbf{h} = \{h_i\}_{i=1}^n \in \mathbb{R}^n$, i.e. $\mathbf{p} \in \mathfrak{P}_{\mathbf{h}}$, if $p_i > 0$, when $h_i \neq 0$ and $p_i = 0$, when $h_i = 0$ for $i = 1, \dots, n$.

Definition 1.2.2. A transition density matrix $P = \{p_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$ is called permissible for a symmetric matrix $A = \{a_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$, i.e. $P \in \mathfrak{P}_A$, if $p_{ij} > 0$, when $a_{ij} \neq 0$ and $p_{ij} = 0$, when $a_{ij} = 0$ for $i, j = 1, \dots, n$.

The introduced concepts of an admissible initial density vector $\mathbf{p} \in \mathfrak{P}_{\mathbf{h}}$ and an admissible transition density matrix $P \in \mathfrak{P}_A$ guarantee the correct construction of the Markov chain. For a sample of N realizations of the random variable $\theta^{(k)}$, the mean is represented by:

$$\bar{\theta}_N^{(k)} = \frac{1}{N} \sum_{s=1}^N (\theta^{(k)})_s, \quad (1.9)$$

where $(\theta^{(k)})_s$ denotes the s -th realization of $\theta^{(k)}$ for $1 \leq s \leq N$. This means provides a stochastic approximation of $(\mathbf{h}, A^k \mathbf{f})$. When pseudorandom number generators are used, this is a Monte Carlo estimate with probabilistic error:

$$R_N^{(k)} = |(\mathbf{h}, A^k \mathbf{f}) - \bar{\theta}_N^{(k)}| < c_1 \sigma(\theta^{(k)}) N^{-1/2}, \quad (1.10)$$

where c_1 is a constant and $\sigma(\theta^{(k)})$ is the standard deviation [17]. When low-discrepancy sequences (Sobol, Halton) are used, $\bar{\theta}_N^{(k)}$ is a Quasi-Monte Carlo approximation with error:

$$QR_N^{(k)} = |(\mathbf{h}, A^k \mathbf{f}) - \bar{\theta}_N^{(k)}| < c_2 \frac{(\log N)^{k+1}}{N}, \quad (1.11)$$

obtained by the Koksma-Hlawka inequality, where $(k+1)$ is the dimension of the sequences used.

Theorem 1.2.1.¹ Let $\theta^{(k)}$ and $\theta^{(k-1)}$ be random variables for fixed steps k and $k-1$ of the Markov chain (1.6). Then

$$\frac{E[\theta^{(k)}]}{E[\theta^{(k-1)}]} = \lambda_{max}^{(k)}. \quad (1.12)$$

Remark 1. The stochastic Power method is called the Power Monte Carlo (PMC) method when the samples for $\theta^{(k)}$ and $\theta^{(k-1)}$ are generated using random number generators to construct the Markov chain (1.6).

In this case, the stochastic error of the PMC method is determined by the probabilistic error defined in (1.10).

¹The proof of Theorem 1.2.1 is provided in the dissertation.

Remark 2. When scrambled low-discrepancy sequences, such as scrambled Sobol sequences [75], are used to construct the Markov chain for generating samples of $\theta^{(k)}$ and $\theta^{(k-1)}$, the stochastic Power method is called the Power Quasi-Monte Carlo (PQMC) method.

The stochastic error of the PQMC method is determined by the error defined in (1.11).

Reformulating the problem in terms of integral equations with appropriately chosen kernels and domains allows the use of the Koksma-Hlawka inequality when analyzing the convergence of the considered algorithms. In general, the use of low-discrepancy sequences (so-called quasi-sequences) improves the convergence rate of the corresponding QMC algorithms. Still, it does not allow for a posteriori error estimation. Scrambling (randomization) of the quasi-sequence makes it possible to overcome this drawback.

The theoretical comparison of the error in the PMC and PQMC methods shows that in both cases the error is a product of two factors: one depending on the sequence and the other on the function or the random variable. In this context, the Koksma-Hlawka inequality represents a worst-case bound, whereas the error in the PMC approach has a probabilistic nature. Moreover, computational experience shows that the variation $V(f)$ in the Koksma-Hlawka inequality is usually overestimated, while the second factor, the discrepancy of the sequence, reflects the actual order of the error.

1.3 Almost Optimal PMC and PQMC Methods and Algorithms

1.3.1 Refining the Construction of $\theta^{(k)}$ and $\theta^{(k-1)}$ with *Almost Optimal* Probabilities for the Markov Chain: Error Balancing

Let the symmetric matrix A be represented by its rows $\mathbf{a}_i$, introducing standard ℓ_1 -norms for vectors and matrices:

$$\|\mathbf{h}\| = \|\mathbf{h}\|_1 = \sum_{i=1}^n |h_i|, \quad \|\mathbf{a}_i\| = \|\mathbf{a}_i\|_1 = \sum_{j=1}^n |a_{ij}|, \quad \|A\| = \|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|, \quad j = 1, \dots, n.$$

The probability error of the MC approximations, according to (1.10), depends on the standard deviation $\sigma(\theta^{(k)})$ and the number of realizations N . For a fixed N , it is determined mainly by the variance of the random variable $\theta^{(k)}$. The variance depends on the choice of the admissible initial density vector $\mathbf{p} \in \mathfrak{P}_{\mathbf{h}}$ and the transition density matrix $P \in \mathfrak{P}_A$. All such choices can be used to construct estimates for $\lambda_{max}^{(k)}$, provided that the variances are finite.

The most commonly used approach to constructing a Markov chain for linear algebra problems is based on the uniform distribution. In this case, the coordinates of the density vector $\mathbf{p}$ and the elements of the transition density matrix P are set equal to $\frac{1}{n}$, i.e.,

$$\mathbf{p} = \{p_i\}_{i=1}^n = \frac{1}{n} \text{ and } P = \{p_{ij}\}_{i,j=1}^n = \frac{1}{n}, \quad i, j = 1, \dots, n. \quad (1.13)$$

When choice (1.13) is used to construct a Markov chain, the Power MC and Power QMC methods are called *classical*.

Appropriate choices of $\mathbf{p}$ and P allow the construction of more efficient algorithms, leading to a reduction in variance, a reduction in computational complexity, and greater precision compared to the *classical* case. There is an *almost optimal* choice, represented by the formulas:

$$\mathbf{p} = \{p_i\}_{i=1}^n, \quad p_i = \frac{|h_i|}{\|\mathbf{h}\|} \text{ and } P = \{p_{ij}\}_{i,j=1}^n, \quad p_{ij} = \frac{|a_{ij}|}{\|\mathbf{a}_i\|}, \quad i, j = 1, \dots, n. \quad (1.14)$$

This choice leads to a significant reduction in variance and the construction of the random variable $\theta^{(k)}$ differs significantly from the *classical* PMC/PQMC method.

Theorem 1.3.1. ² Let $\mathbf{p}$ and P be the almost optimal initial density vector and the almost optimal transition density matrix (see (1.14)). Then the mean value (1.9) can be expressed as follows:

$$\bar{\theta}_N^{(k)} = \frac{1}{N} \sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \|\mathbf{h}\| \left\{ \prod_{t=1}^{t=k} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \|\mathbf{a}_{l_{t-1}^{(s)}}\| \right\} f_{l_k^{(s)}}. \quad (1.15)$$

The following corollaries follow from Theorem 1.2.1 and Theorem 1.3.1:

²The proof of Theorem 1.3.1 is given in the dissertation.

Corollary 1. Let $\mathbf{p}$ and P be the almost optimal initial density vector and the almost optimal density matrix. Then the Stochastic Power Method for estimating $\lambda_{max}^{(k)}$ is defined as the ratio of the sample means of the random variables $\bar{\theta}_N^{(k)}$ and $\bar{\theta}_N^{(k-1)}$ given by (1.15), i.e.

$$\lambda_{max}^{(k)} \approx \frac{\sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \left\{ \prod_{t=1}^{t=k} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \|\mathbf{a}_{l_{t-1}^{(s)}}\| \right\} f_{l_k^{(s)}}}{\sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \left\{ \prod_{t=1}^{t=k-1} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \|\mathbf{a}_{l_{t-1}^{(s)}}\| \right\} f_{l_{k-1}^{(s)}}}. \quad (1.16)$$

Corollary 2. Assume that the L_1 -norm of all row vectors of the symmetric matrix A is equal to a constant a , i.e.

$$\|\mathbf{a}_i\|_1 = \|\mathbf{a}_i\| = \sum_{j=1}^n |a_{ij}| = a, \quad i = 1, \dots, n. \quad (1.17)$$

Then the estimator of $\lambda_{max}^{(k)}$ given by (1.16) can be rewritten in a simpler form:

$$\lambda_{max}^{(k)} \approx \frac{\sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \left\{ \prod_{t=1}^{t=k} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \right\} a f_{l_k^{(s)}}}{\sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \left\{ \prod_{t=1}^{t=k-1} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \right\} f_{l_{k-1}^{(s)}}}. \quad (1.18)$$

Corollary 3. Let the L_1 -norm of all row vectors of the symmetric matrix A satisfy condition (1.17). Assume that the following condition holds for all elements of A :

$$\text{sign}(h_{l_0^{(s)}}) \prod_{t=1}^{t=k-1} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) = 1, \quad s = 1, \dots, N. \quad (1.19)$$

Then $\lambda_{max}^{(k)}$ can be estimated by an even simpler formula:

$$\lambda_{max}^{(k)} \approx \frac{\sum_{s=1}^N \text{sign}(a_{l_{k-1}^{(s)} l_k^{(s)}}) a f_{l_k^{(s)}}}{\sum_{s=1}^N f_{l_{k-1}^{(s)}}}. \quad (1.20)$$

It should be noted that by Corollary (2) and Corollary (3) simplified estimates are constructed for an approximate calculation of $\lambda_{max}^{(k)}$ for certain classes of symmetric matrices. According to Remark 3, these classes are nonempty sets.

Remark 3. The set of symmetric matrices that satisfy the conditions of Corollary 3 is nonempty. For example, consider a symmetric matrix A with positive entries, for which all diagonal elements are equal, i.e., $a_{ii} = b > 0$, $i = 1, \dots, n$, and all off-diagonal elements are also equal, i.e., $a_{ij} = d > 0$, $i \neq j$, $i, j = 1, \dots, n$.

Since the systematic error ε_1 depends on the number k of transitions in the Markov chain, and the stochastic error ε_2 depends on the number of realizations N , an estimate of the corresponding errors is obtained:

$$\left| \lambda_{max} - \frac{\bar{\theta}_N^{(k)}}{\bar{\theta}_N^{(k-1)}} \right| = \left| \lambda_{max} - \lambda_{max}^{(k)} + \lambda_{max}^{(k)} - \frac{\bar{\theta}_N^{(k)}}{\bar{\theta}_N^{(k-1)}} \right| \leq \left| \lambda_{max} - \lambda_{max}^{(k)} \right| + \left| \lambda_{max}^{(k)} - \frac{\bar{\theta}_N^{(k)}}{\bar{\theta}_N^{(k-1)}} \right| < \varepsilon_1 + \varepsilon_2. \quad (1.21)$$

An optimal balance between the two types of errors (systematic and stochastic) is achieved when $\varepsilon_1 = \varepsilon_2 = \varepsilon$.

Definition 1.3.1. PMC and PQMC methods for estimating the maximum eigenvalue of a symmetric matrix, with the choice of the initial density vector $\mathbf{p}$ and the transition density matrix P according to equation (1.14), are referred to as almost optimal PMC and PQMC methods.

1.3.2 Almost Optimal PMC and Almost Optimal PQMC Algorithms: Computational Complexity

The pseudocode in **Algorithm 1** describes a procedure to estimate the maximum eigenvalue of a dense symmetric matrix based on formula (1.16) and Corollary 1. The algorithm is defined as an *almost optimal* PQMC algorithm when low-discrepancy sequences (Sobol or Halton) are used in Step 4 and as an *almost optimal* PMC algorithm, indicated by **Algorithm (1a)**, when pseudorandom number generators are used. If the conditions

of Corollary 2 and Corollary 3 are satisfied, the formulas in Step 6 can be replaced by the simpler expressions (1.18) or (1.20).

Algorithm 1 Pseudocode for computing the *Almost Optimal* PQMC algorithm

- 1: INPUT: matrix $A = \{a_{ij}\} \in \mathbb{R}^{n \times n}$, vectors $\mathbf{h}, \mathbf{f} \in \mathbb{R}^n$ and positive integers n, N, k
- 2: COMPUTE: $\|\mathbf{a}_i\| = \sum_{j=1}^n |a_{ij}|$, $1 \leq i \leq n$ and $\|\mathbf{h}\| = \sum_{i=1}^n |h_i|$
- 3: COMPUTE: The coordinates of a permissible density vector $\mathbf{p}$ and the elements of a permissible density matrix P

$$p_i = \frac{|h_i|}{\|\mathbf{h}\|} \quad \text{and} \quad p_{ij} = \frac{|a_{ij}|}{\|\mathbf{a}_i\|}, \quad i, j = 1, \dots, n.$$

- 4: GENERATE: N elements of the $(k+1)$ - dimensional Sobol (or Halton) sequence
- 5: CONSTRUCT: N realizations of the Markov chain with integer elements

$$l_0^{(s)} \rightarrow \dots \rightarrow l_t^{(s)} \rightarrow \dots \rightarrow l_{k-1}^{(s)} \rightarrow l_k^{(s)}, \quad 1 \leq l_t^{(s)} \leq n, \quad 0 \leq t \leq k, \quad 1 \leq s \leq N$$

- 6: COMPUTE:

$$\begin{aligned} \bar{\theta}_N^{(k)} &= \frac{1}{N} \sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \|\mathbf{h}\| \left\{ \prod_{t=1}^{t=k} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \|\mathbf{a}_{l_{t-1}^{(s)}}\| \right\} f_{l_k^{(s)}}; \\ \bar{\theta}_N^{(k-1)} &= \frac{1}{N} \sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \|\mathbf{h}\| \left\{ \prod_{t=1}^{t=k-1} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \|\mathbf{a}_{l_{t-1}^{(s)}}\| \right\} f_{l_{k-1}^{(s)}}; \end{aligned}$$

- 7: OUTPUT:

$$\lambda_{max}^{(k)} := \frac{\bar{\theta}_N^{(k)}}{\bar{\theta}_N^{(k-1)}}$$

The computational complexity of **Algorithm 1** is determined by the number of realizations N of the Markov chain, the average time required to compute a single realization of the random variables $\theta^{(k)}$ and $\theta^{(k-1)}$, and the preprocessing cost associated with the construction of the transition density matrix P . Steps 2 and 3 require $\mathcal{O}(n^2)$ operations and are treated as preprocessing, while the main computational effort is concentrated in Steps 4 to 6. For fixed k , the time for a single realization is $\tau_k = \mathcal{O}(nk)$; therefore, the total computational complexity of the algorithm is:

$$\text{Cost}(\text{Alg1}) = \mathcal{O}(Nnk) + \mathcal{O}(n^2) = \mathcal{O}(Nnk + n^2),$$

which reduces to $\mathcal{O}(Nnk)$ when $N > n$. Numerical experiments show that for dense symmetric matrices, a balance between systematic and stochastic errors is achieved for $N = \mathcal{O}(n)$, while for sparse matrices, an additional reduction in computational cost can be expected.

1.4 Numerical Results and Discussion

In the numerical experiments, the *almost optimal* PMC algorithm (**Algorithm (1a)**) is implemented using two pseudorandom number generators: Mersenne Twister (MT) and Middle Square (MS). The corresponding implementations are denoted by $PMC_{(MT)}$ and $PMC_{(MS)}$. The results obtained are compared with those of the *classical* PMC algorithm, implemented using the standard choice of the initial density vector and the transition density matrix according to formula (1.14), denoted by $PMC_{(MT)}^{(n)}$ and $PMC_{(MS)}^{(n)}$. The *almost optimal* PQMC algorithm (**Algorithm 1**) is implemented using low-discrepancy sequences (Sobol and Halton) denoted by $PQMC_{(S)}$ and $PQMC_{(H)}$, and is compared with the *classical* PQMC approach and its variants $PQMC_{(S)}^{(n)}$ and $PQMC_{(H)}^{(n)}$. To reduce correlations in high dimensions, the sequences are scrambled using “RR2” (reverse-radix) for Halton and “MatousekAffineOwen” for Sobol. All generators and sequences are implemented in MATLAB 9.10 (R2021a).

The numerical experiments are carried out on two symmetric dense matrices of sizes $n = 100$ and $n = 500$, with randomly generated entries and eigenvalues $(\lambda_1, \lambda_2) = (50.0408, 4.0522)$ and $(\lambda_1, \lambda_2) = (250.2454, 9.0721)$ respectively. The vectors $\mathbf{h}$ and $\mathbf{f}$ are chosen to have unit norm L_1 . To ensure convergence, the matrix A is scaled by the parameter q as:

$$q = \frac{1}{|\lambda_1| + \delta_1}, \quad \delta_1 > 0,$$

or alternatively

$$q = \frac{1}{\max_i \|\mathbf{a}_i\| + \delta}, \quad \delta > 0, \quad i = 1, \dots, n,$$

where $\|\mathbf{a}_i\|$ denotes the L_1 -norm of the i -th row of A . The choice of well-separated eigenvalues guarantees fast convergence and allows for an experimental investigation of the balance between systematic and stochastic errors. The influence of the number of transitions k and the number of realizations N on accuracy is analyzed, as well as the effect of variance reduction achieved by the *almost optimal* PMC/PQMC algorithms under fixed computational complexity.

1.4.1 Case (i): Matrix size $n = 100$

Table 1.1 presents the results for estimating the largest eigenvalue of a symmetric matrix ($n = 100$, $\lambda_1 = 50.0408$) using the *almost optimal* $PMC_{(MT)}$ and the *classical* $PMC_{(MT)}^{(n)}$ algorithm. The optimal balance between systematic and stochastic error is achieved for $k = 8$ and $k = 9$. The minimum absolute error for $PMC_{(MT)}$ is 5×10^{-5} at $k = 8, N = 512$, whereas for the *classical* $PMC_{(MT)}^{(n)}$ an error of 1.5×10^{-3} is observed in $k = 8, N = 2048$. This clearly demonstrates the superiority of the almost optimal variant.

Table 1.1: Results of $PMC_{(MT)}$ and $PMC_{(MT)}^{(n)}$ at different N and k for a matrix A with size $n = 100$, for which $\lambda_2/\lambda_1 \approx 0.081$, where $\lambda_1 = 50.0408$ and $\lambda_2 = 4.0522$.

N	k	$\lambda_1^{(k)}$ using $PMC_{(MT)}$	Absolute Error $PMC_{(MT)}$	$\lambda_1^{(k)}$ using $PMC_{(MT)}^{(n)}$	Absolute Error $PMC_{(MT)}^{(n)}$
128	8	50.039901	0.00090	50.019883	0.0209
	9	50.040709	0.00009	50.037795	0.0030
	10	50.045295	0.00450	50.066104	0.0253
256	7	50.023923	0.01688	50.012158	0.0286
	8	50.040628	0.00017	50.039528	0.0013
	9	50.044896	0.00410	50.050593	0.0098
512	7	50.035045	0.00576	49.996040	0.0448
	8	50.040750	0.00005	50.039264	0.0015
	9	50.052027	0.01123	50.050158	0.0094
1024	7	50.029828	0.01097	50.019435	0.0214
	8	50.040720	0.00008	50.039735	0.0011
	9	50.045403	0.00460	50.054351	0.0136
2048	7	50.036291	0.00451	50.029563	0.0112
	8	50.041086	0.00029	50.041728	0.0009
	9	50.043850	0.00305	50.057179	0.0164

*The highlighted rows represent the optimal k values for each N .

Table 1.2 shows the analogous results for $PMC_{(MS)}$ and $PMC_{(MS)}^{(n)}$. The smallest absolute error for the *almost optimal* MS algorithm is 1.8×10^{-4} in $k = 8$ and $N = 512$, while the *classical* MS algorithm reaches an error of 1.8×10^{-3} . This confirms the general trend that *almost optimal* algorithms reduce variance and yield more accurate estimates than *classical* ones, regardless of the pseudorandom number generator used.

Figure 1.1 shows that the *almost optimal* versions ($PMC_{(MT)}$ and $PMC_{(MS)}$) follow uniform trajectories close to the exact value λ_{max} even for smaller values of N , while the *classical* versions, especially $PMC_{(MS)}^{(n)}$, demonstrate larger fluctuations and weaker stability. The main reason is the reduced variance in the construction of random variables $\theta^{(k)}$ and $\theta^{(k-1)}$ using the optimal transition density matrix, which confirms the higher precision and computational efficiency of the *almost optimal* PMC algorithms.

Table 1.3 presents the results for the *almost optimal* and *classical* PQMC algorithms based on Halton sequences, when evaluating $\lambda_1 = 50.0408$ for a matrix of size $n = 100$. A monotonic decrease in absolute error is observed as the number of transitions k increases, with optimal convergence achieved at $k = 11$ -12. For $N = 2048$, the algorithm $PQMC_{(H)}$ reaches an absolute error of 8.2×10^{-4} , while the *classical* $PQMC_{(H)}^{(n)}$ remains significantly less accurate with an error of 0.0707. This clearly demonstrates the advantage of the *almost optimal* construction. Table 1.4 contains the analogous results for the algorithms based on Sobol sequences. The *almost optimal* algorithm $PQMC_{(S)}$ systematically outperforms both its *classical* counterpart and the variant based on Halton sequences. For example, for $N = 2048$ and an optimal number of transitions, $PQMC_{(S)}$ achieves an absolute error of 5×10^{-5} , while $PQMC_{(S)}^{(n)}$ reaches 0.0336. Even for smaller values of N , such as $N = 256$ and $k = 11$, the difference is significant: 8.1×10^{-4} versus 0.1455 in favor of the *almost optimal* algorithm.

Table 1.2: Results of $PMC_{(MS)}$ and $PMC_{(MS)}^{(n)}$ at different N and k for the matrix A with size $n = 100$, for which $\lambda_2/\lambda_1 \approx 0.081$, where $\lambda_1 = 50.0408$ and $\lambda_2 = 4.0522$.

N	k	$\lambda_1^{(k)}$ using $PMC_{(MS)}$	Absolute Error $PMC_{(MS)}$	$\lambda_1^{(k)}$ using $PMC_{(MS)}^{(n)}$	Absolute Error $PMC_{(MS)}^{(n)}$
128	8	50.039440	0.00136	50.019388	0.0214
	9	50.040595	0.00020	50.029883	0.0109
	10	50.051262	0.01046	50.060923	0.0201
256	7	50.031044	0.00976	49.984883	0.0559
	8	50.042490	0.00169	50.028211	0.0126
	9	50.050490	0.00969	50.051251	0.0105
512	7	50.030518	0.01028	49.997989	0.0428
	8	50.040617	0.00018	50.033570	0.0072
	9	50.058913	0.01811	50.053570	0.0128
	10	50.061048	0.02025	50.091035	0.0502
1024	7	50.031472	0.00933	50.009835	0.0310
	8	50.041209	0.00041	50.038850	0.0020
	9	50.044938	0.00414	50.056396	0.0156
2048	7	50.038998	0.00180	50.022658	0.0181
	8	50.041243	0.00044	50.038951	0.0018
	9	50.043045	0.00225	50.063537	0.0227

*The highlighted rows represent the optimal k values for each N .

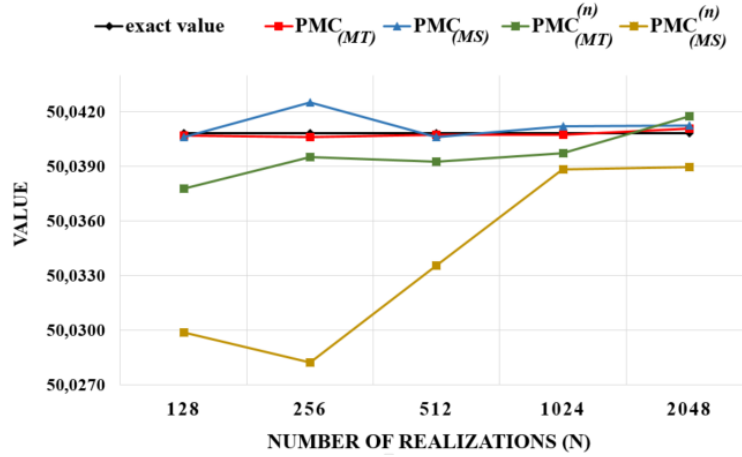


Figure 1.1: Comparison of the approximate results for $\lambda_{max}^{(k)}$ with $\lambda_{max} = 50.0408$ in the balanced error case, i.e. when $(k = 9, N = 128)$ and $(k = 8, N > 128)$. Applied algorithms: $PMC_{(MT)}$, $PMC_{(MS)}$, $PMC_{(MT)}^{(n)}$ and $PMC_{(MS)}^{(n)}$.

The visualization in Figure 1.2 confirms the tabular results, showing a smoother and more stable convergence of the algorithms $PQMC_{(S)}$ and $PQMC_{(H)}$ toward the exact value λ_{max} for all values of N . In contrast, the *classical* variants $PQMC_{(S)}^{(n)}$ and $PQMC_{(H)}^{(n)}$ demonstrate larger fluctuations and initial overestimation of the eigenvalue, and their convergence is significantly slower.

Figure 1.3 shows a comparison between the *almost optimal* PMC and PQMC algorithms for the balanced cases. In PMC methods, $PMC_{(MT)}$ reaches a minimum error of 5×10^{-5} at $N = 512$ and $k = 8$, while $PMC_{(MS)}$ shows a larger but still small error of 1.8×10^{-4} . On the PQMC side, $PQMC_{(S)}$ is the most accurate, with an error of 5×10^{-5} at $N = 2048$ and $k = 11$, while $PQMC_{(H)}$ reaches 8.2×10^{-4} . These results clearly show that the use of scrambled Sobol sequences and the *almost optimal* transition matrix leads to lower variance and an optimal balance between stochastic and systematic error compared to their *classical* variants.

1.4.2 Case (ii): Matrix size $n = 500$

Table 1.5 presents the results for estimating the maximum eigenvalue λ_{max} of a symmetric dense matrix of size $n = 500$, obtained with the *almost optimal* algorithm $PMC_{(MT)}$ and its *classical* analogue $PMC_{(MT)}^{(n)}$. Numerical data show that for a fixed number of realizations N , increasing the number of transitions k leads to a decrease in the absolute error, with the optimal balance between systematic and stochastic error being

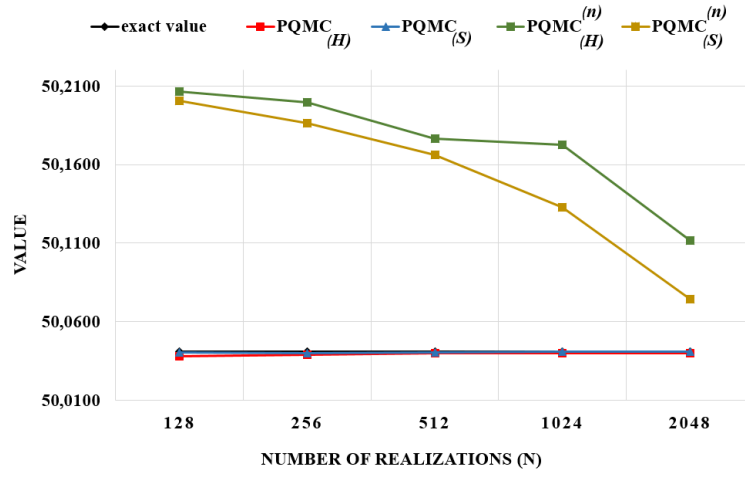


Figure 1.2: Comparison of the approximate results for $\lambda_{max}^{(k)}$ with $\lambda_{max} = 50.0408$ in the balanced error case, i.e. when $(k = 12, N = 128)$ and $(k = 11, N > 128)$. Applied algorithms: $PQMC_{(S)}$, $PQMC_{(H)}$, $PQMC_{(S)}^{(n)}$ and $PQMC_{(H)}^{(n)}$.

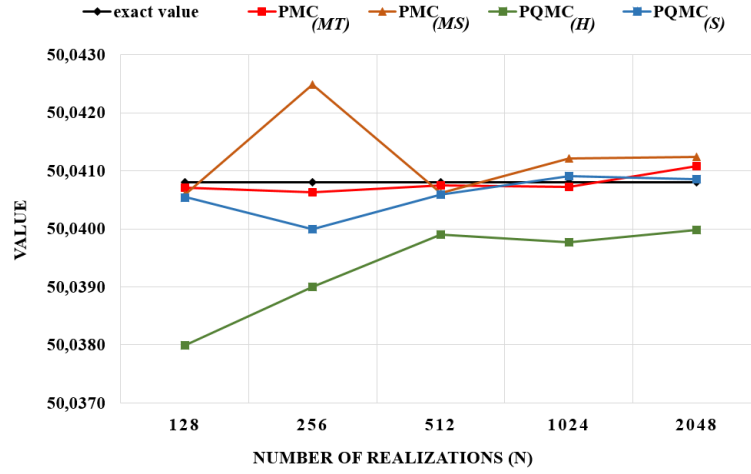


Figure 1.3: Comparison of λ_{max} of the symmetric matrix A ($n = 100$) with $\lambda_{max}^{(k)}$ in the balancing cases. The *almost optimal* PMC and PQMC algorithms are used.

achieved at $k = 9$. In this balanced case, the *almost optimal* algorithm reaches an absolute error of the order of 10^{-4} (e.g., for $N = 512$), while in the *classical* variant the error remains of the order of 10^{-2} under the same parameters, clearly demonstrating significantly higher accuracy.

Similar conclusions can be drawn from Table 1.6, which presents the results for the *almost optimal* algorithm $PMC_{(MS)}^{(n)}$ and the *classical* $PMC_{(MS)}^{(n)}$. In this case, optimal convergence is observed at $k = 9$, with the minimum absolute error for the *almost optimal* algorithm achieved at $N = 128$ on the order of 10^{-4} . In contrast, the *classical* variant demonstrates significantly larger errors, on the order of 10^{-2} , which confirms the advantage of the optimized design even when using a weaker pseudorandom generator.

The visualization in Figure 1.4 confirms the results in Table 1.5 and Table 1.6. The *almost optimal* PMC algorithms show smoother and more stable convergence to λ_{max} with increasing number of realizations N . This is due to the significantly reduced variance in the construction of the random variables $\theta^{(k)}$ and $\theta^{(k-1)}$. The *classical* algorithm $PMC_{(MS)}^{(n)}$ demonstrates fluctuations and even overestimation of the eigenvalue at larger values of N , while $PMC_{(MT)}^{(n)}$ is characterized by more stable convergence. This indicates that the MT pseudorandom number generator is preferable for high-dimensional problems.

Table 1.7 presents the results for estimating λ_{max} using the *almost optimal* and *classical* PQMC algorithms based on scrambled Halton sequences. For all algorithms, the absolute error decreases as the number of transitions k increases, with the optimal balance achieved at $k = 12$. In this case, the *almost optimal* algorithm $PQMC_{(H)}$ achieves an absolute error on the order of 10^{-3} (e.g., 4.59×10^{-3} at $N = 2048$), while the *classical* variant $PQMC_{(H)}^{(n)}$ demonstrates significantly larger errors, reaching 0.0726 for the same number of realizations. This

Table 1.3: Results of $PQMC_{(H)}$ and $PQMC_{(H)}^{(n)}$ for different N and k for a matrix A with size $n = 100$, for which $\lambda_2/\lambda_1 \approx 0.081$, where $\lambda_1 = 50.0408$ and $\lambda_2 = 4.0522$.

N	k	$\lambda_1^{(k)}$ using $PQMC_{(H)}$	Absolute Error $PQMC_{(H)}$	$\lambda_1^{(k)}$ using $PQMC_{(H)}^{(n)}$	Absolute Error $PQMC_{(H)}^{(n)}$
128	11	49.801798	0.23900	49.574856	0.4659
	12	50.037989	0.00281	50.206208	0.1654
	13	50.152986	0.11219	50.314121	0.2733
256	10	50.001834	0.03897	49.739820	0.3010
	11	50.038996	0.00180	50.199525	0.1587
	12	50.067986	0.02719	50.229280	0.1885
512	10	50.017989	0.02281	49.749688	0.2911
	11	50.039899	0.00090	50.176724	0.1359
	12	50.178512	0.13771	50.196566	0.1558
1024	10	49.976987	0.06381	49.806135	0.2347
	11	50.039772	0.00103	50.172345	0.1315
	12	50.107985	0.06719	50.295866	0.2551
2048	10	49.973344	0.06746	49.907160	0.1336
	11	50.039982	0.00082	50.111519	0.0707
	12	50.183245	0.14245	50.275169	0.2344

*The highlighted rows represent the optimal k values for each N .

Table 1.4: Results of $PQMC_{(S)}$ and $PQMC_{(S)}^{(n)}$ for different N and k for a matrix A with size $n = 100$, for which $\lambda_2/\lambda_1 \approx 0.081$, where $\lambda_1 = 50.0408$ and $\lambda_2 = 4.0522$.

N	k	$\lambda_1^{(k)}$ using $PQMC_{(S)}$	Absolute Error $PQMC_{(S)}$	$\lambda_1^{(k)}$ using $PQMC_{(S)}^{(n)}$	Absolute Error $PQMC_{(S)}^{(n)}$
128	11	49.679383	0.36142	49.624160	0.4166
	12	50.040544	0.00026	50.200481	0.1597
	13	50.091116	0.05032	50.299826	0.2590
256	10	49.994318	0.04648	49.742961	0.2978
	11	50.039989	0.00081	50.186264	0.1455
	12	50.049407	0.00861	50.225854	0.1851
512	10	50.050000	0.00920	49.866152	0.1746
	11	50.040585	0.00021	50.166400	0.1256
	12	50.148019	0.10722	50.223877	0.1831
1024	10	49.986946	0.05385	49.933245	0.1076
	11	50.040913	0.00011	50.132603	0.0918
	12	50.087435	0.04664	50.167252	0.1265
2048	10	49.972883	0.06792	49.957605	0.0832
	11	50.040854	0.00005	50.074364	0.0336
	12	50.155992	0.11519	50.186986	0.1462

*The highlighted rows represent the optimal k values for each N .

clearly shows the advantage of the optimized construction of the algorithm.

Table 1.8 presents similar results for algorithms using Sobol sequences. The *almost optimal* algorithm $PQMC_{(S)}$ demonstrates the highest accuracy, with the absolute error in the balanced case at $k = 12$ on the order of 10^{-4} (e.g. 5.4×10^{-4} at $N = 2048$). In contrast, the *classical* variant $PQMC_{(S)}^{(n)}$ achieves significantly larger deviations, with an error of 0.3172 at $N = 256$. This confirms the better uniformity of Sobol sequences and their advantage in higher-dimensional problems.

Figure 1.6 shows the comparisons between the *almost optimal* PMC and PQMC algorithms. The *almost optimal* PMC algorithms $PMC_{(MT)}$ and $PMC_{(MS)}$ achieve minimum errors of 3.8×10^{-4} and 6.5×10^{-4} , respectively, at $N = 1024$ and $k = 9$, while the *almost optimal* PQMC algorithm $PQMC_{(S)}$ provides a comparable or higher accuracy at $k = 12$.

Table 1.9 compares the empirical estimates for the variances of the random variable $\theta^{(k)}$ for the *classical* and *almost optimal* PMC algorithms. Estimates were performed for matrices of dimensions $n = 100$ and $n = 500$. The results show that for the *almost optimal* PMC algorithms, the variance is about two orders of magnitude smaller compared to their *classical* variants. This estimate does not depend significantly on the generator used or on the number of realizations N , which is in agreement with the theoretical results. The fact that the variance

Table 1.5: Results of $PMC_{(MT)}$ and $PMC_{(MT)}^{(n)}$ for different N and k for a matrix A with size $n = 500$, for which $\lambda_2/\lambda_1 \approx 0.0363$, where $\lambda_1 = 250.2454$ and $\lambda_2 = 9.0721$.

N	k	$\lambda_1^{(k)}$ using $PMC_{(MT)}$	Absolute Error $PMC_{(MT)}$	$\lambda_1^{(k)}$ using $PMC_{(MT)}^{(n)}$	Absolute Error $PMC_{(MT)}^{(n)}$
128	8	250.222396	0.02300	249.492834	0.7526
	9	250.232288	0.01311	250.032970	0.2124
	10	250.295027	0.04963	250.149293	0.0961
256	8	250.228998	0.01640	250.124994	0.1204
	9	250.250130	0.00473	250.222275	0.0231
	10	250.283657	0.03826	250.311181	0.0658
512	8	250.235302	0.01011	250.115109	0.1303
	9	250.245110	0.00029	250.192305	0.0531
	10	250.268302	0.02290	250.328365	0.0830
1024	8	250.235306	0.01009	250.164200	0.0812
	9	250.245015	0.00038	250.205542	0.0399
	10	250.269913	0.02451	250.300328	0.0549
2048	8	250.226172	0.01923	250.175939	0.0695
	9	250.246938	0.00154	250.224397	0.0210
	10	250.251779	0.00638	250.295497	0.0501

*The highlighted rows represent the optimal k values for each N .

Table 1.6: Results of $PMC_{(MS)}$ and $PMC_{(MS)}^{(n)}$ at different N and k for a matrix A with size $n = 500$, for which $\lambda_2/\lambda_1 \approx 0.0363$, where $\lambda_1 = 250.2454$ and $\lambda_2 = 9.0721$.

N	k	$\lambda_1^{(k)}$ using $PMC_{(MS)}$	Absolute Error $PMC_{(MS)}$	$\lambda_1^{(k)}$ using $PMC_{(MS)}^{(n)}$	Absolute Error $PMC_{(MS)}^{(n)}$
128	8	250.211445	0.03395	249.462722	0.7827
	9	250.245247	0.00015	250.003068	0.2423
	10	250.279476	0.03408	250.120230	0.1252
256	8	250.203720	0.04168	250.015811	0.2296
	9	250.252044	0.00664	250.221130	0.0243
	10	250.290513	0.04511	250.363882	0.1185
512	8	250.206884	0.03852	250.058060	0.1873
	9	250.244890	0.00051	250.183410	0.0620
	10	250.283279	0.03788	250.356760	0.1114
1024	8	250.238977	0.00642	250.143770	0.1016
	9	250.244751	0.00065	250.268955	0.0236
	10	250.270855	0.02546	250.396782	0.1514
2048	8	250.237612	0.00779	250.103221	0.1422
	9	250.247710	0.00231	250.320873	0.0755
	10	250.275206	0.02981	250.347819	0.1024

*The highlighted rows represent the optimal k values for each N .

for the *almost optimal* versions is two orders of magnitude smaller than the *classical* versions confirms again the conclusions that these algorithms provide a more accurate estimation of the maximum eigenvalue of symmetric matrices with reduced computational complexity.

1.5 Summary

This chapter investigates the efficiency of the *almost optimal* PMC and PQMC algorithms for estimating the largest eigenvalue of dense symmetric matrices for $n = 100$ and $n = 500$. Numerical experiments show that the optimized algorithms significantly outperform the *classical* ones in accuracy and computational complexity by purposefully choosing the transition matrix and balancing systematic and stochastic error. In PMC algorithms, the optimal number of transitions is $k = 8$ for $n = 100$ and $k = 9$ for $n = 500$, with the Mersenne Twister (MT) generator providing a smaller variance than the Middle Square (MS). In the PQMC, the optimal number of transitions is $k = 11$ for $n = 100$ and $k = 12$ for $n = 500$, with the scrambled Sobol sequences leading to smoother convergence and smaller errors than Halton. Absolute errors in *almost optimal* algorithms are on the order of 10^{-4} , while in *classical* ones they reach 10^{-2} , confirming the significant reduction in variance and the

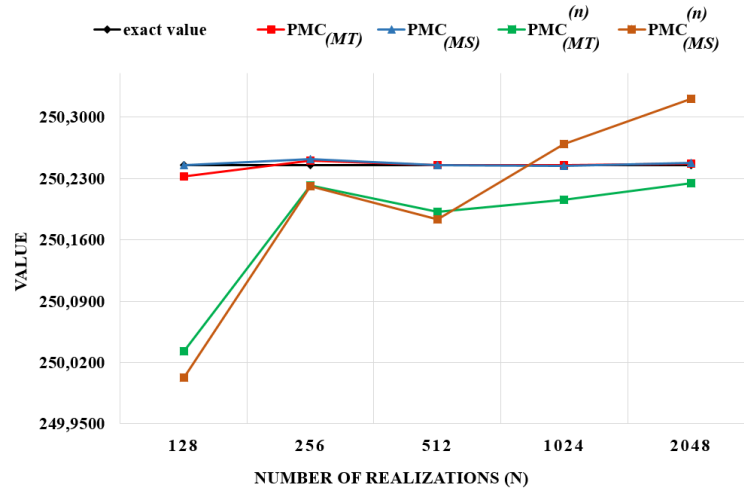


Figure 1.4: Comparison of λ_{max} for the matrix A ($n = 500$) with the approximate results for $\lambda_{max}^{(k)}$ obtained in the balanced error case, i.e. when ($k = 9, N \geq 128$). Applied algorithms: $PMC_{(MT)}$, $PMC_{(MS)}$, $PMC_{(MT)}^{(n)}$ and $PMC_{(MS)}^{(n)}$.

Table 1.7: Results of $PQMC_{(H)}$ and $PQMC_{(H)}^{(n)}$ for different N and k for matrix A with size $n = 500$. The absolute error is for the case $\lambda_2/\lambda_1 \approx 0.0363$.

N	k	$\lambda_1^{(k)}$ using $PQMC_{(H)}$	Absolute Error $PQMC_{(H)}$	$\lambda_1^{(k)}$ using $PQMC_{(H)}^{(n)}$	Absolute Error $PQMC_{(H)}^{(n)}$
128	11	250.167707	0.07769	251.486594	1.2412
	12	250.249391	0.00399	249.309133	0.9363
	13	250.357823	0.11242	251.999063	1.7537
256	11	250.197841	0.04756	249.470953	0.7744
	12	250.239655	0.00574	250.770441	0.5250
	13	250.357573	0.11217	251.711421	1.4660
512	11	250.089218	0.15618	249.591525	0.6539
	12	250.236033	0.00937	250.669970	0.4246
	13	250.369509	0.12411	250.978674	0.7333
1024	11	250.127756	0.11764	249.672942	0.5725
	12	250.250718	0.00532	250.090116	0.1553
	13	250.387049	0.14165	250.829267	0.5839
2048	11	250.166549	0.07885	249.987224	0.2582
	12	250.249986	0.00459	250.172833	0.0726
	13	250.321274	0.07587	250.581459	0.3361

*The highlighted rows represent the optimal k values for each N .

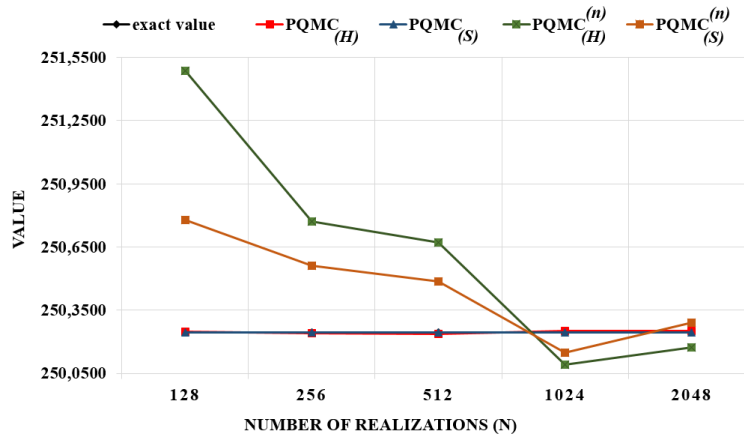


Figure 1.5: Comparison of λ_{max} for the matrix A ($n = 500$) with the approximate results for $\lambda_{max}^{(k)}$ obtained in the balanced error case, i.e. when ($k = 12, N \geq 128$). Applied algorithms: $PQMC_{(S)}$, $PQMC_{(H)}$, $PQMC_{(S)}^{(n)}$ and $PQMC_{(H)}^{(n)}$.

Table 1.8: Results of $PQMC_{(S)}$ and $PQMC_{(S)}^{(n)}$ for different N and k for matrix A with size $n = 500$. The absolute error is for the case $\lambda_2/\lambda_1 \approx 0.0363$.

N	k	$\lambda_1^{(k)}$ using $PQMC_{(S)}$	Absolute Error $PQMC_{(S)}$	$\lambda_1^{(k)}$ using $PQMC_{(S)}^{(n)}$	Absolute Error $PQMC_{(S)}^{(n)}$
128	11	250.236415	0.00898	249.415659	0.8297
	12	250.245037	0.00036	250.777310	0.5319
	13	250.330000	0.08460	250.989461	0.7441
256	11	250.199189	0.04621	249.483392	0.7620
	12	250.245119	0.00028	250.562622	0.3172
	13	250.291367	0.04597	250.704555	0.4592
512	11	250.099484	0.14592	249.632500	0.6129
	12	250.244872	0.00053	250.486175	0.2408
	13	250.378047	0.13265	250.679597	0.4342
1024	11	250.140000	0.10540	249.811533	0.4339
	12	250.244996	0.00040	250.148126	0.0973
	13	250.351139	0.10574	250.651883	0.4065
2048	11	250.193656	0.05174	249.937040	0.3084
	12	250.245937	0.00054	250.290864	0.0455
	13	250.272562	0.02716	250.555662	0.3103

*The highlighted rows represent the optimal k values for each N .

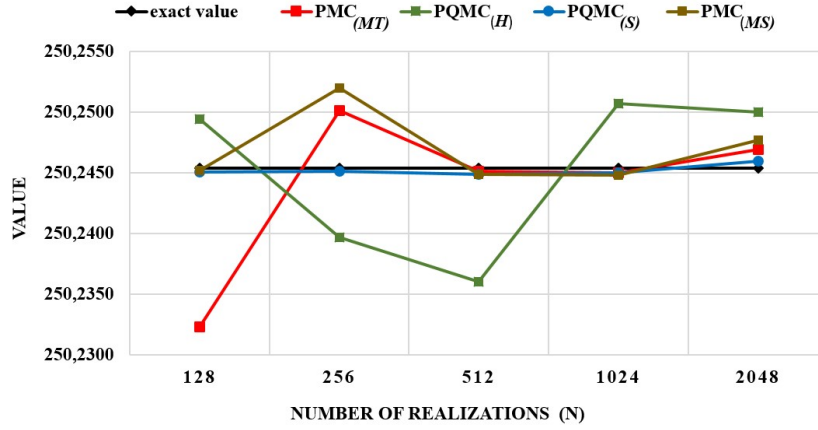


Figure 1.6: Comparison of λ_{max} of the symmetric matrix A ($n = 500$) with $\lambda_{max}^{(k)}$ in the balancing cases. The *almost optimal* PMC and PQMC algorithms are used.

Table 1.9: Variance of the maximum eigenvalue estimates for the *almost optimal* and *classical* PMC algorithms for different matrix dimensions, number of realizations (N), number of transitions (k), and the pseudorandom number generators used (MT and MS).

matrix size	N	k	$Var(PMC_{(MT)})$	$Var(PMC_{(MT)}^{(n)})$	$Var(PMC_{(MS)})$	$Var(PMC_{(MS)}^{(n)})$
100	512	8	1.169203e-06	1.720495e-04	1.236100e-06	1.781005e-04
	2048	8	1.300635e-06	1.772653e-04	1.326996e-06	1.792424e-04
500	512	9	1.186151e-08	1.229778e-05	1.269925e-08	1.285468e-05
	2048	9	1.189763e-08	1.184918e-05	1.268721e-08	1.299595e-05

increase in accuracy.

In conclusion, the *almost optimal* PMC and PQMC algorithms achieve higher accuracy with the same computational complexity, making them a preferred choice for practical applications. When constructing them, we must also take into account the choice of pseudorandom number generators and the choice of low-discrepancy sequences. The described results are published in [36, 37].

2. Stochastic Resolvent Method: Development and Investigation of Efficient Algorithms for Minimum Eigenvalue

2.1 Problem Formulation

Let $A = \{a_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$ be a nonsingular symmetric matrix. The eigenvalues of A are defined as all real values of the parameter $\lambda(A)$ for which the equation

$$A\mathbf{x} = \lambda(A)\mathbf{x}, \quad \text{where } \mathbf{x} \in \mathbb{R}^n, \quad (2.1)$$

has a nonzero solution. It is assumed that the eigenvalues satisfy the ordering

$$0 < \lambda_{\min} = \lambda_n < \lambda_{n-1} \leq \lambda_{n-2} \leq \dots \leq \lambda_2 < \lambda_1 = \lambda_{\max}. \quad (2.2)$$

The minimal eigenvalue $\lambda_{\min}$ is positive and, according to (2.2), is unique. Consider the task of developing and investigating numerically efficient stochastic algorithms for its estimation.

2.2 Method for Computing Eigenvalues

Let $R = \{r_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$ be a nonsingular symmetric matrix whose eigenvalues satisfy the ordering:

$$\mu_{\min} = |\mu_n| < |\mu_{n-1}| \leq |\mu_{n-2}| \leq \dots \leq |\mu_2| < |\mu_1| = \mu_{\max}, \quad (2.3)$$

and for which the equation:

$$R\mathbf{y} = \mu(R)\mathbf{y} \quad (2.4)$$

has a nonzero solution. From linear algebra theory, it is known that under certain relationships between the matrices R and A , there exist connections between their eigenvalues and eigenvectors. Three main cases are considered:

- For the inverse matrix $R = A^{-1}$, the eigenvalues are related by $\mu = \frac{1}{\lambda}$. This allows the use of the *Inverse Power* method to compute the minimal eigenvalue of A , but requires the expensive computation of the inverse matrix.
- For the shifted matrix $R_q = A - q\mathbf{I}$, the relation $\mu = \lambda - q$ holds. This leads to the *Shift Power* method, where the choice of the parameter q influences the approximation of the minimal eigenvalue.
- For the resolvent matrix $R_q = (\mathbf{I} - qA)^{-1}$, the eigenvalues are related by $\mu = \frac{1}{1 - q\lambda}$, and the corresponding method is called the *Resolvent Power* method. The matrix R_q can be represented as the limit of an infinite convergent series under the condition $|q\lambda| < 1$, or equivalently $|q|\|A\| < 1$, namely:

$$R_q = \sum_{i=0}^{\infty} q^i A^i = [\mathbf{I} - qA]^{-1}. \quad (2.5)$$

Depending on the sign of the parameter q , different relations for the maximal eigenvalue are obtained:

$$\mu_{\max} = \frac{1}{1 - q\lambda_{\min}}, \quad \text{for } q < 0, \quad (2.6)$$

$$\mu_{\max} = \frac{1}{1 - q\lambda_{\max}}, \quad \text{for } q > 0. \quad (2.7)$$

Within the scope of the present study, the main interest is in the case $q < 0$, which enables accelerated approximation and efficient estimation of the extreme eigenvalues.

2.2.1 Resolvent Power Method

In the *classical* Power method, the largest eigenvalue $\lambda_{\max}$ of a symmetric matrix A is represented as the limit of the Rayleigh quotient (1.4), which allows a practical approximation using finite powers. On this basis, the Resolvent Power method is introduced, in which the resolvent matrix replaces the matrix A , $R_q = (\mathbf{I} - qA)^{-1}$,

which exists under the condition $|q||A| < 1$. The resolvent matrix admits a Neumann series expansion, and its power R_q^m can be represented by a generalized binomial series:

$$R_q^m = (\mathbf{I} - qA)^{-m} = \sum_{i=0}^{\infty} q^i C_{i+m-1}^i A^i, \quad C_{i+m-1}^i = \frac{(m+i-1)!}{(m-1)!i!}, \quad i = 0, 1, 2, \dots \quad (2.8)$$

The infinite series for R_q^m (where q is called the acceleration parameter) can be written as

$$R_q^m = (I - qA)^{-m} = \mathbf{I} + mqA + \frac{m(m+1)}{2} q^2 A^2 + \frac{m(m+1)(m+2)}{6} q^3 A^3 + \dots \quad (2.9)$$

Applying the Power method to the resolvent matrix (R_q) leads to the Resolvent Power method:

$$\lim_{m \rightarrow \infty} \frac{(\mathbf{h}, R_q^m \mathbf{f})}{(\mathbf{h}, R_q^{m-1} \mathbf{f})} = \lim_{m \rightarrow \infty} \frac{(\mathbf{h}, (\mathbf{I} - qA)^{-m} \mathbf{f})}{(\mathbf{h}, (\mathbf{I} - qA)^{-(m-1)} \mathbf{f})} = \mu_{\max} = \frac{1}{1 - q\lambda}, \quad (2.10)$$

which enables the computation of the maximal eigenvalue $\mu_{\max}$ of the resolvent matrix and, consequently, the estimation of the extreme eigenvalues of A . Depending on the sign of the parameter q , the method can be used to approximate $\lambda_{\max}$ or $\lambda_{\min}$, according to [19, 20, 22]:

$$\lambda = \frac{(\mathbf{h}, AR_q^m \mathbf{f})}{(\mathbf{h}, R_q^m \mathbf{f})} \approx \lambda_{\max}, \quad \text{if } q > 0, \quad (2.11)$$

$$\frac{(\mathbf{h}, AR_q^m \mathbf{f})}{(\mathbf{h}, R_q^m \mathbf{f})} \approx \frac{1}{q} \left(1 - \frac{1}{\mu^{(m)}} \right) \approx \lambda_{\min}, \quad \text{if } q < 0. \quad (2.12)$$

In practice, the resolvent matrix is approximated by truncating the series to a finite number of terms k :

$$R_{q,k}^m = \sum_{i=0}^k C_{i+m-1}^i q^i A^i, \quad (2.13)$$

which leads to a systematic error equal to the “tail” of the series:

$$\Delta_k^{(m)}(q) = (I - qA)^{-m} - R_{q,k}^m = \sum_{i=k+1}^{\infty} C_{i+m-1}^i q^i A^i. \quad (2.14)$$

Estimates for this error and conditions for selecting the parameters m and k such that it is smaller than a prescribed ε have been obtained, i.e.,

$$|\Delta_k^{(m)}(q)| < \varepsilon. \quad (2.15)$$

The following bound holds:

$$|\Delta_k^{(m)}(t)| \leq T_{m,k}(t) = \sum_{i=k+1}^{\infty} C_{i+m-1}^i t^i, \quad t = |q| |\lambda_{\max}| < 1. \quad (2.16)$$

There is a linear relationship between the parameters m and k under the condition $t = |q| |\lambda_{\max}| < 1$. The following estimates are obtained for the relationship between k and m :

- Accuracy inequality:

$$T_{m,k}(t) \leq \frac{b_{k+1}}{1 - \alpha} = \frac{\binom{m+k}{k+1} t^{k+1}}{1 - \frac{m+k+1}{k+2} t} < \varepsilon, \quad b_i = C_{i+m-1}^i t^i, \quad \alpha := \frac{m+k+1}{k+2}, \quad t < 1. \quad (2.17)$$

- Linear relations between the parameters:

$$m < \frac{1-t}{t}(k+1) + \frac{1}{t}, \quad \text{or} \quad m_{\max} = \left\lfloor \frac{(1-t)k+2}{t} - 1 \right\rfloor. \quad (2.18)$$

- Asymptotic relationship (for growing m and fixed t):

$$k \gtrsim \frac{t}{1-t} m, \quad \text{for large values of } m. \quad (2.19)$$

- Minimal value of k for fixed t :

$$k_{\min} = \left\lceil \frac{tm - 1}{1 - t} - 1 \right\rceil. \quad (2.20)$$

The derived relations enable an efficient choice of parameters in numerical implementations and form the basis for the construction of a Stochastic Resolvent method, including a balance between systematic and stochastic errors.

2.2.2 Stochastic Resolvent Method

The Stochastic Resolvent method, introduced by Karaivanova and Dimov [20, 25], represents a generalization of the Resolvent Power method to estimate the extreme eigenvalues of a matrix A . A fundamental component is the construction of random variables $\theta^{(i)}$, defined by a Markov chain, whose mathematical expectations coincide with the scalar products $(\mathbf{h}, A^i \mathbf{f})$, $i = 1, 2, 3, \dots$, i.e.

$$E[\theta^{(i)}] = (\mathbf{h}, A^i \mathbf{f}), \quad i = 1, 2, 3, \dots \quad (2.21)$$

Under the condition on the parameter q satisfying $|q \lambda_{\max}| < 1$, it has been proven in [17] that the scalar product with the resolvent matrix can be represented as an infinite series:

$$(\mathbf{h}, R_q^m \mathbf{f}) = (\mathbf{h}, (\mathbf{I} - qA)^{-m} \mathbf{f}) = \sum_{i=0}^{\infty} q^i C_{i+m-1}^i E[\theta^{(i)}], \quad (2.22)$$

which allows for stochastic estimation using Monte Carlo methods. In practice, mathematical expectations are approximated by mean values:

$$\bar{\theta}_N^{(i)} = \frac{1}{N} \sum_{s=1}^N (\theta^{(i)})_s, \quad i = 1, 2, 3, \dots \quad (2.23)$$

where the stochastic error is of the order of $\mathcal{O}(N^{-1/2})$. For a fixed accuracy ε , the problem of determining the minimum number of realizations N required to guarantee an error below a prescribed threshold has been solved under the assumptions of bounded variance and $\|A\| < 1$:

$$N \geq \left\{ \frac{c_k \sigma}{\varepsilon} \right\}^2. \quad (2.24)$$

Due to the truncation of the infinite series to a finite number of terms k , two errors arise: (1) stochastic (Monte Carlo approximation) and (2) systematic (from truncation of the series). The choice of parameters m and k controls the systematic error, and N controls the stochastic error. The balance between them is crucial for accuracy and computational efficiency. The approximate eigenvalue formula is as follows:

$$\lambda \approx \frac{E[\theta^{(1)}] + qC_m^1 E[\theta^{(2)}] + q^2 C_{m+1}^2 E[\theta^{(3)}] + \dots + q^k C_{k+m-1}^k E[\theta^{(k+1)}]}{1 + qC_m^1 E[\theta^{(1)}] + q^2 C_{m+1}^2 E[\theta^{(2)}] + \dots + q^k C_{k+m-1}^k E[\theta^{(k)}]}, \quad (2.25)$$

where the sign of the parameter q determines whether the maximum ($\lambda_{\max}$ at $q > 0$) or the minimum eigenvalue ($\lambda_{\min}$ at $q < 0$) is evaluated. When pseudorandom numbers are used, the method becomes the Resolvent Monte Carlo (RMC) method, whereas when low-discrepancy sequences are employed, it becomes the Resolvent Quasi-Monte Carlo (RQMC) method. The construction of random variables is based on an appropriate choice of the initial density and the transition probability matrix, and numerical results demonstrate a higher accuracy than *classical* deterministic algorithms.

2.3 Almost Optimal RMC and RQMC Methods and Algorithms

Let the matrix A be represented by the rows $\mathbf{a}_i$, introducing L_1 -norms for vectors and matrices, which serve as a basis for constructing *almost optimal* stochastic resolvent algorithms. The choice of the initial density vector and the transition probability matrix is implemented by normalizing the absolute values of the components of $\mathbf{h}$ and the rows of matrix A , which leads to an *almost optimal* structure of the Markov chain.

The *almost optimal* Resolvent Quasi-Monte Carlo (RQMC) algorithm is presented for estimating the minimal eigenvalue of a symmetric matrix. The algorithm employs low-discrepancy sequences (for example, a scrambled Sobol sequence) to generate realizations in the Markov chain and to compute the random variables $\theta^{(i)}$. When pseudorandom generators are used, the method reduces to an *almost optimal* Resolvent Monte Carlo (RMC)

algorithm. In Step 4 of the algorithm, a scrambled Sobol sequence is used as a low-discrepancy sequence to compute the values of the random variables $\theta^{(i)}$, $i = 1, 2, \dots, k+1$, via a Markov chain.

Algorithm 2 is presented in pseudocode and describes the *almost optimal* RQMC algorithm to estimate the minimal eigenvalue of the symmetric matrix A . If pseudorandom number generators (PRNGs) are used in Step 4, the algorithm is referred to as the *almost optimal* RMC algorithm. In the following exposition, the *almost optimal* RMC algorithm will be denoted by **Algorithm 2a**. Both algorithms belong to the class of iterative stochastic methods and are characterized by the presence of two types of errors: stochastic and systematic. The stochastic error in the MC methods is of the order of $\mathcal{O}(N^{-1/2})$, while in the QMC methods based on low-discrepancy sequences it is $\mathcal{O}(N^{-1}(\log N)^k)$. High dimensionality may lead to correlations between the coordinates of quasirandom sequences, which can be mitigated by using scrambled quasirandom sequences.

Algorithm 2 Pseudocode for computing the *almost optimal* Resolvent QMC algorithm

- 1: INPUT: matrix $A = \{a_{ij}\} \in \mathbb{R}^{n \times n}$; vectors $\mathbf{h}, \mathbf{f} \in \mathbb{R}^n$, positive integers n, N, k, m and the real number q , where $(-1)/\|A\| < q < 0$
- 2: COMPUTE: $p_i = \frac{|h_i|}{\sum_{i=1}^n |h_i|}$; $p_{ij} = \frac{|a_{ij}|}{\sum_{j=1}^n |a_{ij}|}$, $1 \leq i \leq n$, $1 \leq j \leq n$
- 3: COMPUTE: $a_i = \sum_{j=1}^n |a_{ij}|$, $1 \leq i \leq n$, $\|\mathbf{h}\| = \sum_{i=1}^n |h_i|$
- 4: GENERATE: N elements of the $(k+1)$ -dimensional Sobol sequence
- 5: CONSTRUCT: N realizations of the Markov chain with integer elements

$$l_1^{(s)} \rightarrow \dots \rightarrow l_i^{(s)} \rightarrow \dots \rightarrow l_k^{(s)} \rightarrow l_{k+1}^{(s)}, \quad 1 \leq l_i^{(s)} \leq n, \quad 1 \leq i \leq k+1, \quad 1 \leq s \leq N$$

- 6: COMPUTE:

$$\bar{\theta}^{(i)} = \frac{1}{N} \|\mathbf{h}\| \sum_{s=1}^N \text{sign}\{h_{l_1^{(s)}} \prod_{i=1}^{k+1} (a_{l_i^{(s)} l_{i+1}^{(s)}})\} \prod_{i=1}^{k+1} a_{l_i^{(s)}} f_{l_{i+1}^{(s)}}; \quad 1 \leq i \leq k+1$$

- 7: COMPUTE:

$$C_{i+m-1}^i = \frac{(i+m-1)!}{i!(m-1)!}, \text{ where } 1 \leq i \leq k$$

- 8: OUTPUT:

$$\lambda_{min} \approx \frac{\bar{\theta}^{(1)} + qC_m^1 \bar{\theta}^{(2)} + q^2 C_{m+1}^2 \bar{\theta}^{(3)} + \dots + q^k C_{k+m-1}^k \bar{\theta}^{(k+1)}}{1 + qC_m^1 \bar{\theta}^{(1)} + q^2 C_{m+1}^2 \bar{\theta}^{(2)} + \dots + q^k C_{k+m-1}^k \bar{\theta}^{(k)}}$$

The systematic error arises from the use of a finite (truncated) Markov chain and is analogous to the truncated Neumann series in deterministic methods. It depends on the truncation parameter k in the representation of the resolvent matrix $R_{q,k}^m$ as a limit of infinite order under the condition $|q|||A|| < 1$. The parameter m determines the degree of the resolvent matrix and affects the convergence of the method:

$$\mathcal{O}\left(\left|\frac{\mu_2}{\mu_1}\right|^m\right), \quad (2.26)$$

where μ_1, μ_2 are the largest eigenvalues of the resolvent matrix. Expressed in terms of the eigenvalues of A :

$$\mathcal{O}\left(\left|\frac{1-q\lambda_1}{1-q\lambda_2}\right|^m\right), \text{ for } q > 0 \text{ and } \mathcal{O}\left(\left|\frac{1-q\lambda_n}{1-q\lambda_{n-1}}\right|^m\right), \text{ for } q < 0. \quad (2.27)$$

The choice of the acceleration parameter q determines whether the algorithm evaluates the minimum ($q \in (-\frac{1}{\|A\|}, 0)$) or the maximum eigenvalue ($q > 0$). The computational process involves generating N realizations of the Markov chain, computing sample means for the random variables, and using binomial coefficients to estimate the eigenvalue.

The balance between stochastic and systematic error is essential for achieving a prescribed accuracy ε at optimal computational cost. This balance is controlled through the parameters N (number of realizations), k (chain length), m (resolvent power), and q (acceleration parameter). Numerical experiments demonstrate that the choice of these parameters must be consistent with the derived theoretical convergence estimates.

2.4 Numerical Results and Discussion

The numerical results were obtained by applying the *almost optimal* Resolvent Quasi-Monte Carlo (RQMC) algorithm (**Algorithm 2**) and the *almost optimal* Resolvent Monte Carlo (RMC) algorithm (**Algorithm 2a**) to estimate the minimal eigenvalue of randomly generated symmetric matrices of sizes $n = 500$ and $n = 1000$. The *almost optimal* RQMC algorithm is implemented using BRODA's Sobol RSG generator, which enables the generation of scrambled Sobol sequences with the required dimensionality for constructing the Markov chain. The *almost optimal* RMC algorithm is implemented using the Mersenne Twister (MT) pseudorandom number generator. Both algorithms are implemented in C++, with a preliminary subroutine used to generate a random symmetric matrix A whose spectrum lies within the unit circle and satisfies the following condition:

$$0 < c < \lambda_n < \lambda_{n-1} \leq \dots \leq \lambda_2 < \lambda_1 < 1,$$

where the constant c is chosen to be $c = 0.15$. This guarantees positivity and spectral separation, which are necessary for the convergence of the resolvent power method.

Two test cases are considered:

- (i) Matrix A of dimension $n = 500$, with the two largest and two smallest eigenvalues being: $\lambda_1 = 0.620713$, $\lambda_2 = 0.335683$, $\lambda_{499} = 0.15452$ and $\lambda_{500} = 0.152992$.
- (ii) Matrix A with dimension $n = 1000$, with the two largest and two smallest eigenvalues being: $\lambda_1 = 0.688181$, $\lambda_2 = 0.443078$, $\lambda_{999} = 0.154138$ and $\lambda_{1000} = 0.151911$.

The separation between the two largest and the two smallest eigenvalues ensures a reliable convergence of the method when estimating λ_{min} .

The numerical experiments are conducted on a high-performance computing cluster (IICT-BAS), consisting of 12 Fujitsu Primergy RX 2540 M4 servers, each configured with: 128 GB RAM; two CPUs – Intel Xeon Gold 5118 (2.30 GHz) with 24 cores; two 800 GB SSDs; 36 TB HDD; and an NVIDIA Tesla V100 GPU with 32 GB memory. The servers are interconnected via InfiniBand for fast, low-latency data exchange. The installed software includes Red Hat Enterprise Linux 7.6 and NVIDIA GRID.

The implementation of **Algorithm 2** is hybrid: the generation of scrambled Sobol sequences is performed on the GPU (NVIDIA Tesla V100, 5120 CUDA cores), while the remaining computations are executed on the CPU. **Algorithm 2a** is implemented on the CPU using the Mersenne Twister generator.

The arbitrarily vectors $\mathbf{h}$ and $\mathbf{f}$ are chosen with a L_1 -norm as follows:

$$\|\mathbf{h}\| = \|\mathbf{f}\| = \sum_{i=1}^n \frac{1}{n} = 1,$$

to ensure proper normalization in the stochastic estimations. Numerical tests show that for large values of the number of realizations N , the execution time of the RQMC algorithm ($RQMC_{(S)}$) is significantly lower than that of the RMC algorithm ($RMC_{(MT)}$). They demonstrate the advantage of the Quasi-Monte Carlo approach combined with GPU acceleration in large-scale problems for estimating the minimal eigenvalue.

2.4.1 Case (i): Matrix size $n=500$

Table 2.1 presents numerical results to estimating the minimum eigenvalue of a symmetric matrix of dimension $n = 500$, obtained by the *almost optimal* algorithms ($RMC_{(MT)}$ and $RQMC_{(S)}$) for different values of the parameters N, k and m . The table also includes the corresponding absolute errors of the estimates obtained. The acceleration parameter is fixed at $q = -0.157428$, and the systematic error is preset at the level $\varepsilon_{sys} = 0.001$. The analysis of the results is based on the balancing of the two main sources of error - systematic and stochastic. The systematic error arises from the truncation of the Neumann series at $i = k + 1$, which corresponds to a finite number of transitions in the Markov chain, as well as from the choice of the degree m of the resolvent matrix. The parameters m and k are chosen based on theoretical estimates of the tail of the series $T_{m,k}(t)$, for $t = |q||\lambda_1| < 1$, so that the condition $T_{m,k}(t) \leq \varepsilon_{sys} = 0.001$ is satisfied. In this way, the systematic error is controlled in advance and remains on the order of $\mathcal{O}(10^{-3})$.

The stochastic error arises from the approximation of mathematical expectations by a finite number of realizations N . In the Monte Carlo approach ($RMC_{(MT)}$), the stochastic error is on the order of $\mathcal{O}(N^{-1/2})$ for a fixed confidence probability. In the Quasi-Monte Carlo approach ($RQMC_{(S)}$), faster convergence is expected. To estimate the statistical error, scrambled Sobol sequences are used, which combine low-discrepancy properties with controlled randomness. It should be noted that the stochastic error depends not only on N , but also on the parameter k ; increasing k increases the number of estimated random variables $\theta^{(i)}$, $i = 1, \dots, k + 1$.

Table 2.1: Numerical results for the matrix $A \in \mathbb{R}^{500 \times 500}$ for different values of N, k and m with the acceleration parameter $q = -0.157428$. The exact minimum eigenvalue $\lambda_{\min} = \lambda_{500} = 0.152992$.

N	k	m	$\lambda_{\min}$	Absolute error	$\lambda_{\min}$	Absolute error
			$RMC_{(MT)}$	$RMC_{(MT)}$	$RQMC_{(S)}$	$RQMC_{(S)}$
$512 * 2^0$	5	5	0.1585637	0.0055717	0.1561046	0.0031126
	5	10	0.1585163	0.0055243	0.1524884	0.0005036
	10	5	0.1556132	0.0026212	0.1560595	0.0030675
$512 * 2^1$	5	5	0.158395	0.005403	0.1596503	0.0066583
	5	10	0.1585518	0.0055598	0.1595946	0.0066026
	10	5	0.1580794	0.0050874	0.1596616	0.0066696
$512 * 2^5$	5	5	0.1572588	0.0042668	0.1591139	0.0061219
	5	10	0.1566524	0.0036604	0.1585388	0.0055468
	10	5	0.156875	0.003883	0.1591292	0.0061372
$512 * 2^6$	5	5	0.157129	0.004137	0.1589384	0.0059464
	5	10	0.1565796	0.0035876	0.1584563	0.0054643
	10	5	0.1570216	0.0040296	0.158961	0.005969

This leads to the accumulation of variances in both the numerator and the denominator of the rational formula for estimating eigenvalues, which affects the required number of realizations. Therefore, an effective balance requires selecting N so that the stochastic error is of the same order as the prescribed systematic error ε_{sys} .

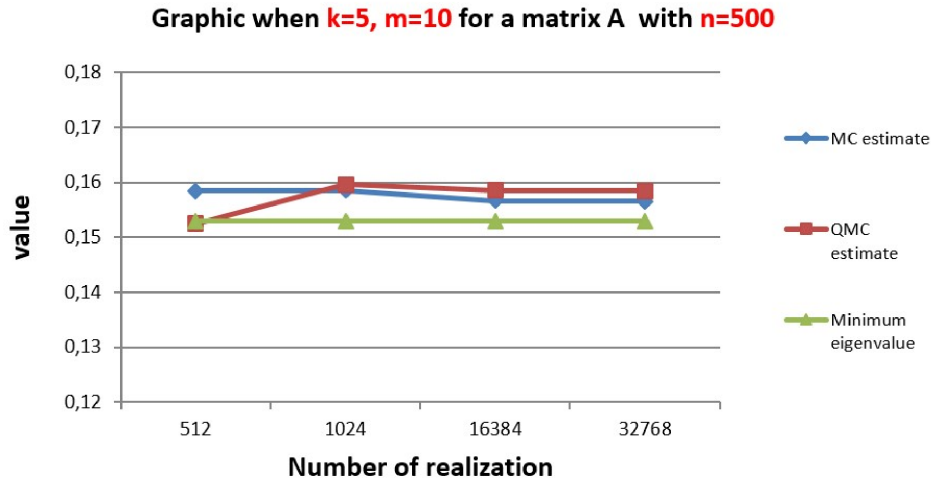


Figure 2.1: Convergence to the exact minimum eigenvalue of the matrix A using MC and QMC at $k = 5, m = 10$ for different values of N

The results obtained for $n = 500$ confirm the theoretical conclusions. With an appropriate choice of parameters (m, k) , which guarantee a limitation of the systematic error below $\mathcal{O}(10^{-3})$, high precision is achieved even with a relatively small number of realizations N , comparable to the dimensionality of the matrix. The best accuracy is obtained at $k = 5, m = 10$, and $N = 512$, where the absolute error is of the same order as the previously specified systematic error. Increasing N (for example, to $N = 16384$ or $N = 32768$, see Figure 2.1 and Table 2.1) does not lead to a monotonic decrease in absolute error.

Saturation or oscillating behavior is observed, which is characteristic of situations in which the systematic error already determines the lower limit of accuracy. Additional influence is exerted by numerical effects associated with the increase in binomial coefficients at larger values of m and k , which increase the sensitivity of the fractional-rational formula to small fluctuations in estimates of $\theta^{(i)}$. Therefore, increasing N does not automatically guarantee improved accuracy when the systematic component or the numerical sensitivity of the method becomes dominant.

2.4.2 Case (ii): Matrix size $n=1000$

Table 2.2 presents numerical results for estimating the minimum eigenvalue of a symmetric matrix of dimension $n = 1000$, obtained by the *almost optimal* algorithms $RM C_{(MT)}$ and $RQMC_{(S)}$, together with the corresponding absolute errors. In these experiments, the acceleration parameter is fixed at $q = -0.2302458$, and its absolute value is higher compared to the case $n = 500$. This increases the parameter $t = |q||\lambda_1|$ and requires, for the same values of k and m , that the systematic error be fixed at a higher level $\varepsilon_{sys} = 0.01$ to satisfy the theoretical conditions to estimate the tail of the series.

Table 2.2: Numerical results for the matrix $A \in \mathbb{R}^{1000 \times 1000}$ for different values of N, k and m with acceleration parameter $q = -0.230245$. The exact minimum eigenvalue $\lambda_{\min} = \lambda_{1000} = 0.151911$.

N	k	m	$\lambda_{\min}$	Absolute error	$\lambda_{\min}$	Absolute error
			$RM C_{(MT)}$	$RM C_{(MT)}$	$RQMC_{(S)}$	$RQMC_{(S)}$
$512 * 2^0$	5	5	0.1556059	0.0045949	0.1705751	0.0186641
	5	10	0.168251	0.01634	0.1886959	0.0367849
	10	5	0.1593428	0.0074318	0.1701307	0.0182197
$512 * 2^1$	5	5	0.1504153	0.0014957	0.1533797	0.0014687
	5	10	0.1417805	0.0101305	0.1499974	0.0019136
	10	5	0.156739	0.004828	0.1534383	0.0015273
$512 * 2^5$	5	5	0.1550015	0.0030905	0.1573842	0.0054732
	5	10	0.1495483	0.0023627	0.1539357	0.0020247
	10	5	0.1559548	0.0040438	0.1575454	0.0056344
$512 * 2^6$	5	5	0.1557509	0.0038399	0.1576339	0.0057229
	5	10	0.1537008	0.0017898	0.1562911	0.0043801
	10	5	0.1562905	0.0043795	0.15764225	0.0057312

The results (see Table 2.2 and Figure 2.2) show that the best balance between the systematic and stochastic components of the error is achieved at $k = 5, m = 5$, and $N = 1024$. For this combination, the absolute error is minimal and of the same order as the prescribed systematic error. This confirms the theoretical conclusion that the parameters m and k , determined based on a fixed level of systematic error, ensure the required accuracy of the approximation of the resolvent matrix, while the number of realizations N should be chosen so that the stochastic error is comparable to the systematic one. For higher values of m, k and N , a certain improvement of the absolute error is observed in terms of the constant factor, but without a change in order, while the computational complexity increases. Increasing N (for example, to $N = 16384$ or $N = 32768$, see Figure 2.2 and Table 2.2) does not lead to a monotonic decrease in the error. The reason is due to the increased sensitivity of the rational formula to small fluctuations in the estimates of the random variables $\theta^{(i)}$, as well as the dominant role of the systematic error.

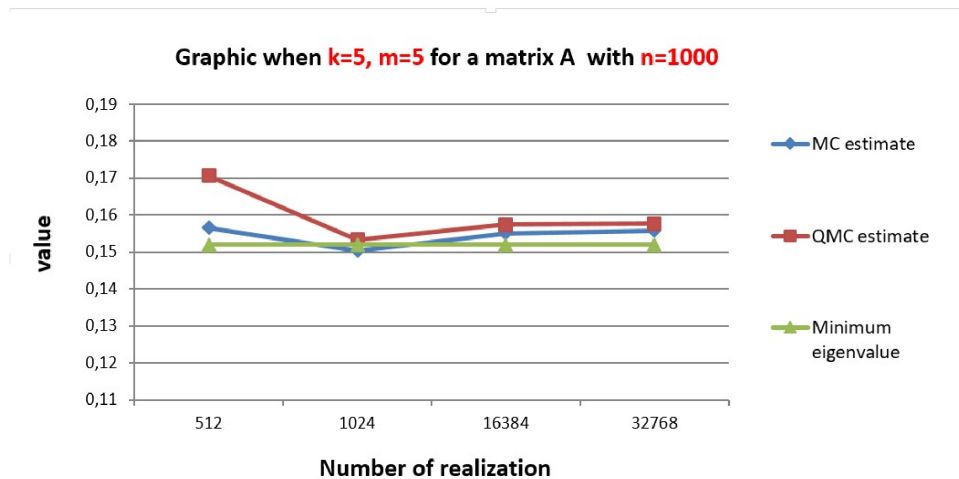


Figure 2.2: Convergence to the exact minimum eigenvalue of the matrix A using MC and QMC at $k = 5, m = 5$ for different values N

In practice, the results indicate that for matrices of dimension on the order of several thousand, an optimal balance is achieved when m and k are chosen according to theoretical estimates at a pre-fixed level of systematic

error, and N is of the same order as the matrix dimension. Too small N leads to a dominant stochastic error, while too large N increases the computational cost without a significant improvement in accuracy.

2.5 Summary

Estimating the minimum eigenvalue $\lambda_{\min}$ of a symmetric matrix using stochastic power algorithms requires the computation of the inverse matrix A^{-1} , which is computationally expensive for large dimensions. As an efficient alternative, the Stochastic Resolvent method is investigated, which eliminates the need to explicitly calculate A^{-1} by using the resolvent matrix $R_q = (\mathbf{I} - qA)^{-1}$ and its powers. The sign of the acceleration parameter q determines whether the minimum or maximum eigenvalue is estimated.

Theoretical analysis shows that the convergence of the Resolvent Power method is essentially dependent on the parameter q and the power m . The systematic error decreases as m increases, while the ratio of the extreme eigenvalues of the matrix determines the convergence rate. The truncation of the binomial series through the parameter k introduces an additional systematic error, from which analytical estimates are derived. A linear dependence between m and k is established, enabling the selection of minimal values at a fixed level of systematic error and thereby reducing computational complexity.

In stochastic implementation, a stochastic error also arises. *Almost optimal* RMC and *almost optimal* RQMC algorithms are proposed, in which mathematical expectations are approximated by sample averages of N realizations of a truncated Markov chain with k transitions. In the MC approach, the error is of order $\mathcal{O}(N^{-1/2})$. In contrast, QMC methods use the Koksma-Hlawka inequality together with low-discrepancy sequences (e.g., Sobol sequences), providing faster convergence. The general accuracy is determined by the balance between systematic error (due to truncation of the series) and stochastic error (due to the finite number of realizations), which requires a balanced choice of the parameters m , k , and N .

The theoretical estimates obtained provide practical guidelines for selecting m , k , and N so that both types of errors are of comparable order with respect to the prescribed accuracy. Numerical experiments for matrices of dimensions $n = 500$ and $n = 1000$ confirm these conclusions: the best efficiency is achieved for parameter choices corresponding to the theoretically predicted balance, and increasing N beyond a certain threshold does not lead to significant improvement. The *almost optimal* RQMC algorithm demonstrates faster convergence compared to the MC approach, and the hybrid GPU/CPU implementation enables efficient application to high-dimensional problems. Part of the results presented in this chapter are published in the articles [35, 36].

3. Stochastic methods for assessing the market risk of investment portfolios

This chapter presents an application of **Algorithm 1** to a real-world problem in financial mathematics: assessing market risk in an investment portfolio. The aim is to demonstrate the practical applicability of stochastic power algorithms by analyzing the spectral characteristics of the correlation matrix of asset returns. Estimating extreme eigenvalues enables the identification of dominant risk factors and the examination of the portfolio's sensitivity to market fluctuations. In this way, theoretical algorithms are employed as tools for quantitative risk analysis and to support investment decision-making.

3.1 Problem Formulation

The main task addressed in this chapter is to estimate the maximum eigenvalue of correlation matrices and to analyze the accuracy of the estimates obtained by applying *almost optimal* Power Monte Carlo (PMC) and *almost optimal* Power Quasi-Monte Carlo (PQMC) algorithms, presented in Chapter 1. These algorithms are used to estimate the largest eigenvalue of the matrices numerically and to investigate the approximation error. Within the PQMC algorithm, two additional operators are introduced: (1) a skipping operator and (2) a leaping operator. Their use is motivated by the need to reduce the influence of initial irregularities inherent in randomized low-discrepancy sequences.

3.2 Methods and Data

The financial data used to construct the correlation matrices are extracted from the *Portfolio Visualizer* and *Yahoo Finance* platforms, with access to the latter provided through the *yFinance* library. The estimation of the largest eigenvalue of the correlation matrices is performed using the algorithms specified in Chapter 1 and the balance between systematic and stochastic error is analyzed. For interpretation of the results in the context of financial analysis, the *Fraction of Variance Explained* (FVE) indicator is used, which measures the share of the explained variation corresponding to the maximum eigenvalue and characterizes the degree of influence of the dominant market factor.

3.3 General Formulation of Covariance and Correlation Matrices

Covariance and the correlation coefficient are fundamental characteristics for describing the linear dependence between random variables. The covariance between two random variables X and Y is defined as

$$\text{Cov}(X, Y) = E[(X - \mu_X)(Y - \mu_Y)], \quad (3.1)$$

where $\mu_X = E[X]$ and $\mu_Y = E[Y]$ [77, 91]. It measures the direction of joint variability and can take both positive and negative values, but it depends on the scale of the variables.

The correlation coefficient represents a normalized form of covariance and is given by:

$$\text{Corr}(X, Y) = \rho_{XY} = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}, \quad (3.2)$$

where σ_X and σ_Y are the standard deviations [77, 91]. It takes values in the interval $[-1, 1]$ and allows for the comparison of linear dependencies independently of the measurement units.

For a random vector $\mathbf{X} = (X_1, \dots, X_n)^T$, the covariance matrix is defined as

$$\Sigma = \text{Cov}(\mathbf{X}) = E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T], \quad (3.3)$$

where the elements $\Sigma_{ij} = \text{Cov}(X_i, X_j)$ and the diagonal elements $\Sigma_{ii} = \sigma_i^2$ represent the variances. Accordingly, the correlation matrix is defined as:

$$\hat{R} = [\rho_{ij}]_{i,j=1}^n, \quad \rho_{ij} = \text{Corr}(X_i, X_j), \quad i, j = 1, \dots, n, \quad (3.4)$$

with $\rho_{ii} = 1$ [42, 57].

In the context of portfolio analysis, the correlation matrix describes the relationships between asset returns, while its largest eigenvalue indicates the dominant market factor [42]. High values of this characteristic are associated with increased market synchronization and limited diversification opportunities. The covariance matrix, in turn, provides complete information about the interdependencies between assets and is a fundamental tool in portfolio optimization and risk management.

3.3.1 Construction of Covariance and Correlation Matrices from Empirical Financial Data

In empirical studies of financial markets, data are typically represented as a matrix $B \in \mathbb{R}^{r \times n}$, where the rows correspond to time points and the columns correspond to the returns of different assets. Each row $\mathbf{b}_i$ describes the joint market state at a given time, allowing analysis of interdependencies between assets. The matrix B is defined as

$$B = [b_{ij}]_{i,j=1}^{r,n} = (\mathbf{b}_1, \dots, \mathbf{b}_i, \dots, \mathbf{b}_r)^T, \quad (3.5)$$

where each row $\mathbf{b}_i$ represents a vector of observed returns for all assets at time i . This representation forms the basis for statistical analysis and the construction of covariance and correlation structures.

Data Centering and Sample Covariance and Correlation Matrices

Let $\bar{\mathbf{b}}$ be the n -dimensional vector of sample column means of matrix B . The centered matrix is defined as follows:

$$B_c = B - \mathbf{e}\bar{\mathbf{b}}^T, \quad (3.6)$$

where $\mathbf{e} \in \mathbb{R}^r$ is a vector of units. The sample covariance matrix is defined as:

$$\Sigma = \frac{1}{r} B_c^T B_c. \quad (3.7)$$

For small values of r in formula (3.7), the division is performed by $r - 1$. The elements Σ_{ij} measure the joint variability between asset returns, and the diagonal elements Σ_{ii} correspond to their variations. The risk of a portfolio Π with weights $\mathbf{w}$ is given by the quadratic form:

$$Var(\Pi) = \mathbf{w}^T \Sigma \mathbf{w}, \quad (3.8)$$

which emphasizes the role of both individual risks and interdependencies between assets. The correlation matrix is obtained by normalizing the covariance matrix with the standard deviations:

$$\hat{R} = D_\sigma^{-1/2} \Sigma D_\sigma^{-1/2}, \quad \rho_{ij} = \frac{\Sigma_{ij}}{\sigma_i \sigma_j}, \quad (3.9)$$

where $\rho_{ij} \in [-1, 1]$ and $\rho_{ii} = 1$. This matrix provides a scale-independent measure of linear relationships between assets and is key to the analysis of diversification [42].

3.3.2 Fraction of Variance Explained

The Fraction of Variance Explained (FVE) is an important characteristic related to the spectral properties of covariance or correlation matrices [47]. Let A be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$. Then FVE for the largest eigenvalue is defined as:

$$FVE_1 = \frac{\lambda_1}{\sum_{j=1}^n \lambda_j} = \frac{\lambda_1}{\text{trace}(A)}, \quad (3.10)$$

and for the first k components:

$$FVE_k = \frac{\sum_{i=1}^k \lambda_i}{\text{trace}(A)}. \quad (3.11)$$

In the special case of a correlation matrix $\hat{R}$, for which $\text{trace}(\hat{R}) = n$, we obtain:

$$FVE_1 = \frac{\lambda_1}{n}. \quad (3.12)$$

This quantity measures the relative contribution of the dominant factor to the total variance of the system. In

the context of financial markets, high values of FVE_1 indicate the presence of a strong common market factor and a high level of systematic risk, which limits the efficiency of diversification. In contrast, lower values suggest a more balanced distribution of risk between independent factors and more favorable conditions for portfolio optimization. Therefore, estimating the largest eigenvalue of the correlation matrix and the corresponding fraction of variance explained is a key step in quantitative market risk analysis.

3.3.3 Stochastic Algorithms for Estimation of the Maximum Eigenvalue of a Correlation Matrix

When estimating the largest eigenvalue of the considered correlation matrices, the *almost optimal* Power Monte Carlo (PMC) and Power Quasi-Monte Carlo (PQMC) algorithms developed in Chapter 1 are applied. These algorithms demonstrate significant superiority over their *classical* counterparts, providing higher numerical accuracy and more stable convergence of estimates at comparable computational costs.

The *almost optimal* PQMC algorithm (**Algorithm 1**) uses low-discrepancy scrambled sequences to construct a Markov chain over a finite number of steps. In particular, in Step 4, Halton and Sobol sequences are generated using the corresponding pseudocodes **Algorithm 3** and **Algorithm 4**.

Algorithm 3 Pseudocode for generating a scrambled Halton Sequence with *skip* and *leap* parameters

- 1: CREATE FUNCTION: `generateHaltonSequence` ($N, k+1$)
 - 2: GENERATE: $p \leftarrow \text{haltonset}(k, \text{'skip'} = s_1, \text{'leap'} = l_1)$
 - 3: GENERATE: $p \leftarrow \text{scramble}(p, \text{'RR2'})$
 - 4: GENERATE: $\text{points} \leftarrow \text{net}(p, N)$
 - 5: OUTPUT: Return points
-

Algorithm 4 Pseudocode for generating a scrambled Sobol Sequence with *skip* and *leap* parameters

- 1: CREATE FUNCTION: `generateSobolSequence` ($N, k+1$)
 - 2: GENERATE: $p \leftarrow \text{sobolset}(k, \text{'skip'} = s_1, \text{'leap'} = l_1)$
 - 3: GENERATE: $p \leftarrow \text{scramble}(p, \text{'MatousekAffineOwen'})$
 - 4: GENERATE: $\text{points} \leftarrow \text{net}(p, N)$
 - 5: OUTPUT: Return points
-

When $s_1 = 0$ and $l_1 = 0$ are chosen, standard (*default*) versions of the sequences are obtained, respectively denoted by $PQMC_{(H)}^{(d)}$ and $PQMC_{(S)}^{(d)}$. For non-zero values of the parameters, the operations *skip* and *leap* are applied, which allow control over the initial elements and the thinning of the sequences, and the corresponding algorithms are denoted by $PQMC_{(H)}^{(s,l)}$ and $PQMC_{(S)}^{(s,l)}$.

For the implementation of the *almost optimal* PMC algorithm (**Algorithm 1a**), the pseudorandom number generators *Mersenne Twister* (MT) and *Middle Square* (MS) are used, with the corresponding variants denoted by $PMC_{(MT)}$ and $PMC_{(MS)}$. All algorithms are implemented in the MATLAB environment (version 9.10, R2021a), and the syntax used for the `haltonset`, `sobolset`, and `scramble` functions in **Algorithm 3** and **Algorithm 4** is consistent with the official documentation of MATLAB [93, 94].

3.4 Numerical Results and Discussion

This section presents numerical results for estimating the maximum eigenvalue of two correlation matrices using the *almost optimal* PMC and PQMC algorithms. The study was conducted on two asset portfolios with different dimensions, which allows for analysis of the behavior of the algorithms at different levels of complexity and information saturation of the data.

The first correlation matrix is constructed from the monthly returns of $n = 32$ global assets for the period January 2007-March 2024, with dimensions $r = 207$ and $n = 32$. The second matrix is constructed from the returns of $n = 490$ companies in the *S&P500* index for the period 2021-2025, with $r = 5$. By applying centering (i.e., formula (3.6)), estimation of the covariance matrix (3.7) and normalization (3.9), correlation matrices with corresponding maximum eigenvalues and FVE are obtained:

- for $n = 32$: $\lambda_{max} = 18.14714$, $FVE_1 = 56.71\%$;
- for $n = 490$: $\lambda_{max} = 201.090512$, $FVE_1 = 41.04\%$.

These values are calculated using the built-in functions of MATLAB and serve as a reference for assessing the accuracy of the algorithms.

In the numerical experiments, the vectors $\mathbf{h}$ and $\mathbf{f}$, used in the construction of $\theta^{(k)}$ and $\theta^{(k-1)}$, are chosen with a L_1 -norm:

$$\|\mathbf{h}\| = \|\mathbf{f}\| = \sum_{i=1}^n \frac{1}{n} = 1.$$

which provides normalization and stability of the stochastic estimates. To ensure convergence, the matrix $A = q\hat{R}$ is considered, with the parameter q chosen such that condition $\|A\| < 1$ is satisfied:

$$q = \frac{1}{|\lambda_1| + \delta}, \quad \delta > 0.$$

or in the absence of a prior estimate for λ_1 :

$$q = \frac{1}{\|\hat{R}\|_1 + \delta}, \quad \|\hat{R}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |\rho_{ij}| \quad \text{and} \quad \delta > 0.$$

This approach ensures the correct applicability of stochastic methods and guarantees their numerical robustness in estimating the maximum eigenvalue.

3.4.1 Case (i): correlation matrix with dimension $n = 32$

Table 3.1 presents the numerical results for estimating the largest eigenvalue of the correlation matrix $\hat{R}$ with dimension $n = 32$. The numerical results obtained are implemented by the *almost optimal* PMC algorithm (**Algorithm 1a**), using two pseudorandom number generators: *Mersenne Twister* (MT) and *Middle Square* (MS), respectively, denoted by $PMC_{(MT)}$ and $PMC_{(MS)}$. The estimation of the maximum eigenvalue $\lambda_1 = \lambda_{\max}$ was performed for different values of the number of transitions k in the Markov chain and the number of realizations N of the random variables $\theta^{(k)}$ and $\theta^{(k-1)}$, and the corresponding absolute errors are presented in Table 3.1. The analysis shows that for a fixed N , increasing k leads to a decrease in error. It stabilizes around $k = 8$, indicating that a practically optimal balance has been reached between the systematic and stochastic components of the error. On the other hand, with an increase in N , a monotonic decrease in absolute error is observed, which is in accordance with the *classical* properties of Monte Carlo methods.

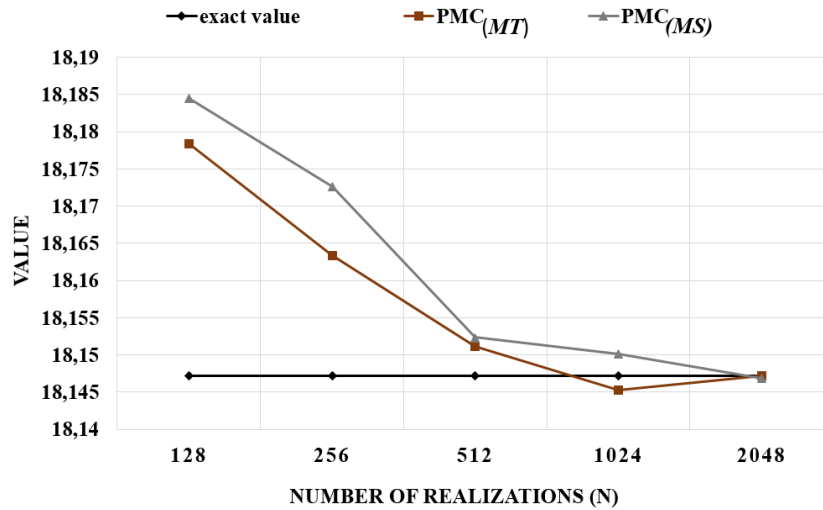
The minimum absolute errors for the two algorithms are noted in Table 3.1, and the results show a comparable order of accuracy for both generators. The visualization in Figure 3.1 illustrates the convergence of the estimates $\lambda_1^{(k)}$ to the exact value λ_1 for a fixed $k = 8$ and an increasing number of realizations N . It is observed that both algorithms tend to the reference value, but $PMC_{(MT)}$ demonstrates higher accuracy and more stable convergence, especially for smaller N . In practice, to achieve accuracy in the second or third decimal place, simulations of $N = 512$ or $N = 1024$ are sufficient, which is fully adequate for applications in portfolio analysis. Using formula (3.12) for a correlation matrix of dimension $n = 32$ and estimates at $N = 1024$, we obtain values $FVE_1 \approx 56.70\%$ for $PMC_{(MT)}$ and $FVE_1 \approx 56.72\%$ for $PMC_{(MS)}$. This confirms that approximations of λ_1 with precision on the order of $\mathcal{O}(10^{-3})$ enable a reliable calculation of the fraction of explained variation, a key indicator for assessing systematic risk and the collective dynamics of assets. Therefore, the *almost optimal* PMC algorithm is an effective tool for analyzing correlation structures of moderate dimension at relatively low computational costs.

Table 3.2 presents the results of the application of the *almost optimal* PQMC algorithm for the same correlation matrix ($n = 32$), implemented through four variants: $PQMC_{(H)}^{(d)}$, $PQMC_{(S)}^{(d)}$, $PQMC_{(H)}^{(s,l)}$, and $PQMC_{(S)}^{(s,l)}$. The variants with index (d) correspond to the choice of $s_1 = 0$ and $l_1 = 0$ in **Algorithm 3** and **Algorithm 4**, i.e., using scrambled Halton and Sobol sequences with default settings. For non-zero values of the parameters *skip* and *leap*, the (s, l) variants are implemented, which allow finer control over the uniformity of the generated points and potentially improve the convergence of the estimates.

The numerical results for the *almost optimal* PQMC algorithms $PQMC_{(H)}^{(s,l)}$ and $PQMC_{(S)}^{(s,l)}$ were obtained by choosing the parameters $s_1 = 1024$ and $l_1 = 128$ in **Algorithm 3** and **Algorithm 4**, which corresponds to applying the *skip* and *leap* operations when generating the scrambled Halton and Sobol sequences. The maximum eigenvalue $\lambda_1 = \lambda_{\max}$ of the correlation matrix was estimated for different values of the number of transitions k and the number of realizations N , and the corresponding absolute errors are presented in Table 3.2. The results show that the parameters k and N significantly affect the balance between systematic and stochastic errors, as well as the computational complexity of the algorithms. At a fixed N , increasing k leads to a decrease

Table 3.1: Results for the maximum values of $\lambda_{\max}^{(k)}$ and the absolute error for $PMC_{(MT)}$ and $PMC_{(MS)}$ at different N and k when $\lambda_{\max} = 18.14714$

N	k	$\lambda_1^{(k)}$ using $PMC_{(MT)}$	Absolute Error $PMC_{(MT)}$	$\lambda_1^{(k)}$ using $PMC_{(MS)}$	Absolute Error $PMC_{(MS)}$
128	5	17.809541	0.3376	17.766635	0.3805
	6	17.955782	0.1914	17.931153	0.2160
	7	18.031908	0.1152	18.016774	0.1304
	8	18.178382	0.0312	18.184488	0.0373
	9	18.217756	0.0706	18.234182	0.0870
	10	18.259830	0.1127	18.278015	0.1309
	11	18.293246	0.1461	18.336586	0.1894
256	5	17.886432	0.2607	17.880110	0.2670
	6	18.013889	0.1333	18.025125	0.1220
	7	18.098671	0.0485	18.090287	0.0569
	8	18.163375	0.0162	18.172589	0.0254
	9	18.214595	0.0675	18.220052	0.0729
	10	18.265757	0.1186	18.270934	0.1238
	11	18.307841	0.1607	18.317397	0.1703
512	5	17.894326	0.2528	17.893265	0.2539
	6	18.019977	0.1272	18.014737	0.1324
	7	18.099165	0.0480	18.094732	0.0524
	8	18.151116	0.0040	18.152321	0.0052
	9	18.194868	0.0477	18.218433	0.0713
	10	18.231143	0.0840	18.244924	0.0978
	11	18.275571	0.1284	18.284115	0.1370
1024	5	17.977295	0.1698	17.947696	0.1994
	6	18.102537	0.0446	18.038992	0.1081
	7	18.118865	0.0283	18.098850	0.0483
	8	18.145221	0.0019	18.150097	0.0030
	9	18.176356	0.0292	18.214874	0.0677
	10	18.202606	0.0555	18.276008	0.1289
	11	18.250397	0.1033	18.294888	0.1477
2048	5	18.007899	0.1392	17.981049	0.1661
	6	18.106753	0.0404	18.056203	0.0909
	7	18.126731	0.0204	18.104529	0.0426
	8	18.147226	0.0001	18.146851	0.0003
	9	18.164272	0.0171	18.172111	0.0250
	10	18.193708	0.0466	18.215959	0.0688
	11	18.218186	0.0710	18.260127	0.1130

*The highlighted rows represent the optimal k values for each N .Figure 3.1: Comparison between the exact maximum eigenvalue and the maximum approximate eigenvalue $\lambda_1^{(k)}$ for $PMC_{(MT)}$, $PMC_{(MS)}$ obtained in the balancing cases for different values of N and $k = 8$

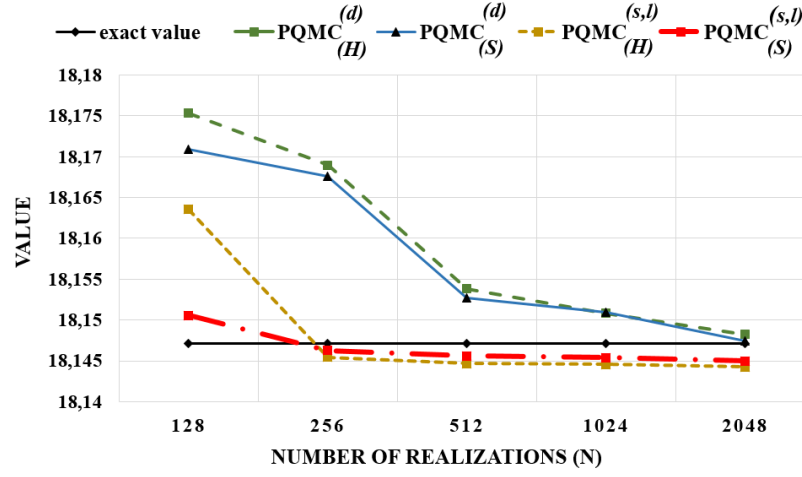


Figure 3.2: $(PQMC_{(H)}^{(d)}, PQMC_{(S)}^{(d)})$ and $(PQMC_{(H)}^{(s,l)}, PQMC_{(S)}^{(s,l)})$. Comparison of the maximum eigenvalue $\lambda_{\max}$ of the correlation matrix with the estimated values $\lambda_1^{(k)}$ obtained when $k = 11$ in the default case and when $k = 13$ in the skip and leap case.

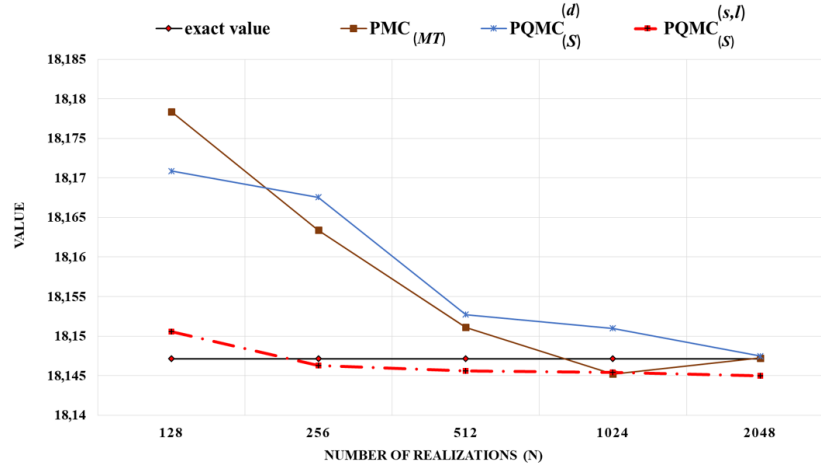


Figure 3.3: Comparison between maximum eigenvalue $\lambda_{\max}$ and approximate eigenvalues $\lambda_1^{(k)}$ obtained at $k = 8$ for $PMC_{(MT)}$, $k = 11$ for $PQMC_{(S)}^{(d)}$ and $k = 13$ for $PQMC_{(S)}^{(s,l)}$ at different values of N .

in absolute error, which stabilizes around $k = 11$ for the default variants $PQMC_{(H)}^{(d)}$ and $PQMC_{(S)}^{(d)}$, and around $k = 13$ for the variants with parameters $PQMC_{(H)}^{(s,l)}$ and $PQMC_{(S)}^{(s,l)}$, indicating reaching a practically optimal balance between the error components. At a fixed k and an increasing N , the algorithms $PQMC_{(H)}^{(d)}$ and $PQMC_{(S)}^{(d)}$ demonstrate a monotonic decrease in the errors and stable convergence. In contrast, $PQMC_{(H)}^{(s,l)}$ and $PQMC_{(S)}^{(s,l)}$ show significant improvement for small samples ($N = 128, 256$), but for larger N the error remains almost constant or even increases, which is consistent with the observations in [52] regarding the sensitivity to the parameters *skip* and *leap*.

The graphical comparison in Figure 3.2 confirms the conclusions drawn by illustrating the convergence of the estimates $\lambda_1^{(k)}$ to the exact value λ_1 with an increasing number of realizations N . The algorithms $PQMC_{(H)}^{(d)}$ and $PQMC_{(S)}^{(d)}$ demonstrate stable and smooth convergence, while the variants $PQMC_{(H)}^{(s,l)}$ and $PQMC_{(S)}^{(s,l)}$ show better accuracy for small samples N , but limited improvement for larger samples.

In Figure 3.3, the $PMC_{(MT)}$ and $PQMC_{(S)}^{(d)}$ algorithms stand out with higher accuracy and stability for large N , while for small samples ($N \leq 512$) the use of PQMC algorithms with parameters *skip* and *leap* is more efficient.

Table 3.3 presents estimates of FVE_1 obtained by $PMC_{(MT)}$, $PQMC_{(S)}^{(d)}$ and $PQMC_{(S)}^{(s,l)}$ for the balancing values of k and $N = 256, 2048$, with the reference value being $FVE_1 = 56.71\%$. The results show that for small

Table 3.2: Results for maximum eigenvalues for $\lambda_{max}^{(k)}$ and absolute error by default scenario and scenario using *skip* and *leap* for different N and k when $\lambda_{max} = 18.14714$

N	k	$\lambda_1^{(k)}$ using $PQMC_{(H)}^{(d)}$	Absolute Error $PQMC_{(H)}^{(d)}$	$\lambda_1^{(k)}$ using $PQMC_{(S)}^{(d)}$	Absolute Error $PQMC_{(S)}^{(d)}$	$\lambda_1^{(k)}$ using $PQMC_{(H)}^{(s,l)}$	Absolute Error $PQMC_{(H)}^{(s,l)}$	$\lambda_1^{(k)}$ using $PQMC_{(S)}^{(s,l)}$	Absolute Error $PQMC_{(S)}^{(s,l)}$
128	8	17.757547	0.3896	17.761379	0.3858	17.672733	0.4744	17.706637	0.4405
	9	17.888100	0.2590	17.888185	0.2590	17.787523	0.3596	17.792723	0.3544
	10	18.016669	0.1305	18.038214	0.1089	17.856741	0.2904	17.859089	0.2881
	11	18.175307	0.0282	18.170859	0.0237	17.903166	0.2440	17.914975	0.2322
	12	18.298811	0.1517	18.264832	0.1177	18.036970	0.1102	18.097564	0.0496
	13	18.479146	0.3320	18.474626	0.3275	18.145462	0.0017	18.146253	0.0009
	14	18.643362	0.4962	18.639031	0.4919	18.194240	0.0471	18.191902	0.0448
	15	18.712889	0.5657	18.706247	0.5591	18.290732	0.1436	18.290924	0.1438
256	8	17.882593	0.2645	17.883812	0.2633	17.761315	0.3858	17.763949	0.3832
	9	17.994408	0.1527	18.016913	0.1302	17.854804	0.2923	17.867712	0.2794
	10	18.061271	0.0859	18.062487	0.0847	17.936785	0.2104	17.921160	0.2260
	11	18.168944	0.0218	18.167556	0.0204	18.036010	0.1111	18.043959	0.1032
	12	18.284126	0.1370	18.251619	0.1045	18.097978	0.0492	18.098070	0.0491
	13	18.482373	0.3352	18.469322	0.3222	18.145462	0.0017	18.146253	0.0009
	14	18.545036	0.3979	18.516617	0.3695	18.209384	0.0622	18.185109	0.0380
	15	18.608838	0.4617	18.607262	0.4601	18.289967	0.1428	18.270105	0.1230
512	8	17.927397	0.2197	17.962654	0.1845	17.774098	0.3730	17.778972	0.3682
	9	18.020330	0.1268	18.021557	0.1256	17.846070	0.3011	17.867473	0.2797
	10	18.081105	0.0660	18.081198	0.0659	17.938077	0.2091	17.949995	0.1971
	11	18.153831	0.0067	18.152703	0.0056	18.009669	0.1375	18.014807	0.1323
	12	18.263301	0.1162	18.248667	0.1015	18.038078	0.1091	18.038550	0.1086
	13	18.425061	0.2779	18.411903	0.2648	18.144697	0.0024	18.145602	0.0015
	14	18.506259	0.3591	18.504204	0.3571	18.204814	0.0577	18.191490	0.0444
	15	18.551586	0.4044	18.546687	0.3995	18.249184	0.1020	18.242947	0.0958
1024	8	17.962214	0.1849	17.978111	0.1690	17.776881	0.3703	17.792702	0.3544
	9	18.035437	0.1117	18.042592	0.1045	17.867701	0.2794	17.887652	0.2595
	10	18.103422	0.0437	18.109582	0.0376	17.980078	0.1671	17.980502	0.1666
	11	18.150822	0.0037	18.150986	0.0038	18.010761	0.1364	18.037651	0.1095
	12	18.294487	0.1473	18.284447	0.1373	18.098692	0.0484	18.098718	0.0484
	13	18.385875	0.2387	18.384413	0.2373	18.144277	0.0029	18.144964	0.0022
	14	18.446470	0.2993	18.443825	0.2967	18.202841	0.0557	18.203207	0.0561
	15	18.530500	0.3834	18.521017	0.3739	18.236167	0.0890	18.235788	0.0886
2048	8	18.000431	0.1467	18.000619	0.1465	17.787254	0.3599	17.788274	0.3589
	9	18.061446	0.0857	18.064399	0.0827	17.879783	0.2674	17.887915	0.2592
	10	18.108451	0.0387	18.110469	0.0367	17.980078	0.1671	17.980502	0.1666
	11	18.148212	0.0011	18.147480	0.0003	18.010923	0.1362	18.040106	0.1070
	12	18.226453	0.0793	18.221684	0.0745	18.118819	0.0283	18.120939	0.0262
	13	18.385443	0.2383	18.379872	0.2327	18.144277	0.0029	18.144964	0.0022
	14	18.443228	0.2961	18.435573	0.2884	18.202381	0.0552	18.203555	0.0564
	15	18.509033	0.3619	18.492101	0.3450	18.247648	0.1005	18.243565	0.0964

*The highlighted rows represent the optimal k values for each N Table 3.3: Approximate values for the fraction of variance explained (FVE_1) obtained at $k = 8$ for $PMC_{(MT)}$, $k = 11$ for $PQMC_{(S)}^{(d)}$ and $k = 13$ for $PQMC_{(S)}^{(s,l)}$ for the correlation matrix $\hat{R}$ with dimension $n = 32$. The reference value is $FVE_1 = 56.71\%$

N	$PMC_{(MT)}$	$PQMC_{(S)}^{(d)}$	$PQMC_{(S)}^{(s,l)}$
256	56.76%	56.77%	56.70%
2048	56.71%	56.71%	56.70%

samples, $PQMC_{(S)}^{(s,l)}$ gives better estimates, while for large N , $PMC_{(MT)}$ and $PQMC_{(S)}^{(d)}$ are more efficient, achieving higher accuracy at lower computational complexity. Finally, the high value of FVE_1 for $n = 32$ indicates the presence of a dominant factor. This means a higher concentration of systematic risk and limited diversification opportunities, as assets respond similarly to market shocks.

3.4.2 Case (ii): correlation matrix with dimension $n = 490$

Table 3.4 presents the numerical results for estimating the maximum eigenvalue $\lambda_1 = \lambda_{max}$ of a correlation matrix $\hat{R}$ with a high dimension $n = 490$, constructed based on the returns of companies from the index $S\&P500$ for the period 2021–2025 (see the Appendix). Due to incomplete data, 10 companies were excluded from the analysis. Estimation was performed using the algorithms $PMC_{(MT)}$, $PQMC_{(H)}^{(d)}$, and $PQMC_{(S)}^{(d)}$, for different values of the number of transitions k and the number of realizations N , and the corresponding absolute errors were also analyzed. The estimation was performed using the *almost optimal* PMC algorithm $PMC_{(MT)}$

and the *almost optimal* PPMC algorithms $PPMC_{(H)}^{(d)}$ and $PPMC_{(S)}^{(d)}$, using Halton and Sobol sequences with default settings. The results show that the parameters k and N have a significant impact on the precision of the estimations, with the balancing values of the number of transitions being $k = 8$ for $PPMC_{(MT)}$ and $k = 11$ for $PPMC_{(H)}^{(d)}$ and $PPMC_{(S)}^{(d)}$, despite the significantly larger dimensionality compared to the case $n = 32$.

Table 3.4: Results for maximum eigenvalues for $\lambda_{max}^{(k)}$ and absolute error for $PPMC_{(MT)}$, $PPMC_{(H)}^{(d)}$ and $PPMC_{(S)}^{(d)}$ at different N and k when $\lambda_{max} = 201.090512$

N	k	$\lambda_1^{(k)}$ using $PPMC_{(MT)}$	Absolute Error $PPMC_{(MT)}$	k	$\lambda_1^{(k)}$ using $PPMC_{(H)}^{(d)}$	Absolute Error $PPMC_{(H)}^{(d)}$	$\lambda_1^{(k)}$ using $PPMC_{(S)}^{(d)}$	Absolute Error $PPMC_{(S)}^{(d)}$
512	6	191.261222	9.8293	8	192.052259	9.0383	192.056796	9.0337
	7	197.764391	3.3261	9	193.305250	7.7853	194.4876692	6.6028
	8	203.030089	1.9396	10	196.545330	4.5452	196.561636	4.5289
	9	206.001210	4.9107	11	202.016871	0.9264	201.830255	0.7397
	10	209.202619	8.1121	12	207.216480	6.1260	206.994716	5.9042
1024	6	194.244715	6.8458	8	191.604757	9.4858	191.662089	9.4284
	7	199.671479	1.4190	9	194.355045	6.7355	194.443329	6.6472
	8	202.103664	1.0132	10	198.362863	2.7276	196.623242	4.4673
	9	204.706063	3.6156	11	201.526659	0.4361	201.467671	0.3772
	10	211.940453	10.8499	12	204.493635	3.4031	204.718275	3.6278
2048	6	196.531375	4.5591	8	191.522259	9.5683	191.737033	9.3535
	7	199.092487	1.9980	9	194.216871	6.8736	194.383618	6.7069
	8	201.462168	0.3717	10	198.552479	2.5380	198.866204	2.2243
	9	204.787620	3.6971	11	201.267671	0.1772	201.199260	0.1087
	10	206.264686	5.1942	12	207.590516	6.5000	206.670141	5.5796
4096	6	195.762474	5.3280	8	194.293635	6.7969	195.846272	6.2442
	7	197.402088	3.6884	9	198.498616	2.5919	198.498988	2.5915
	8	201.154640	0.0641	10	199.459470	1.6310	199.802438	1.2881
	9	202.887637	1.7971	11	201.083321	0.0072	201.091681	0.0012
	10	207.343376	6.2529	12	206.448996	5.3585	205.624451	4.5339

*The highlighted rows represent the optimal k values for each N

The results show that the optimal balance between systematic and stochastic error is achieved at $k = 8$ for $PPMC_{(MT)}$ and $k = 11$ for $PPMC_{(H)}^{(d)}$ and $PPMC_{(S)}^{(d)}$, regardless of the larger dimensionality of the matrix. At a fixed k , a monotonic decrease in the absolute error is observed with increasing number of realizations N , with the minimum errors noted in Table 3.4. Figure 3.4 illustrates that all algorithms demonstrate stable convergence to the reference value λ_{max} , with the PPMC variants $PPMC_{(H)}^{(d)}$ and $PPMC_{(S)}^{(d)}$ showing faster and smoother convergence compared to $PPMC_{(MT)}$. The estimates obtained for λ_1 allow for a precise calculation of FVE₁ using formula (3.12), the results are presented in Table 3.5.

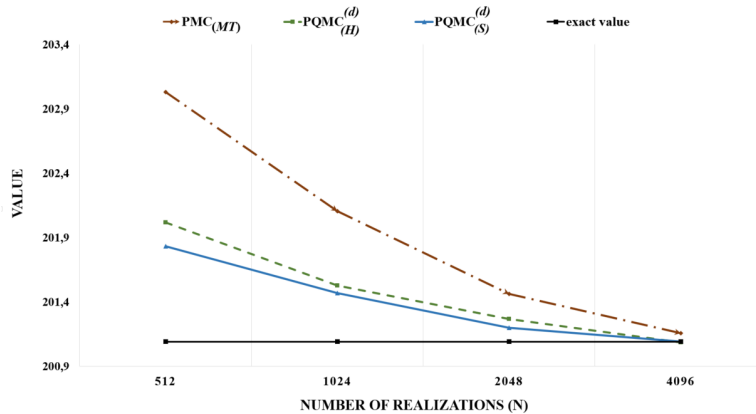


Figure 3.4: Comparison between the exact maximum eigenvalue (λ_{max}) and the maximum approximate eigenvalue ($\lambda_1^{(k)}$) for $PPMC_{(MT)}$, $PPMC_{(H)}^{(d)}$, $PPMC_{(S)}^{(d)}$, obtained when $k = 8$ in the case of $PPMC_{(MT)}$ and $k = 11$ in the case of $PPMC_{(*)}^{(d)}$ and different values of N

Table 3.5: Approximate values for the fraction of variance explained (FVE_1) obtained at $k = 8$ for $\text{PMC}_{(MT)}$ and at $k = 11$ for $\text{PQMC}_{(H)}^{(d)}$ and $\text{PQMC}_{(S)}^{(d)}$ for the correlation matrix $\hat{R}$ with dimension $n = 490$. The reference value is $\text{FVE}_1 = 41.04\%$

N	$\text{PMC}_{(MT)}$	$\text{PQMC}_{(H)}^{(d)}$	$\text{PQMC}_{(S)}^{(d)}$
1024	41.25%	41.13%	41.12%
2048	41.11%	41.08%	41.06%
4096	41.05%	41.04%	41.04%

The estimates obtained for λ_1 by $\text{PQMC}_{(H)}^{(d)}$ and $\text{PQMC}_{(S)}^{(d)}$ allow a more precise calculation of the fraction of the explained variation FVE_1 by formula (3.12). Table 3.5 presents the corresponding results obtained at the balancing values $k = 8$ and $k = 11$, with the reference value being $\text{FVE}_1 = 41.04\%$. Data show that when N increases, the use of the PQMC algorithms $\text{PQMC}_{(H)}^{(d)}$ and $\text{PQMC}_{(S)}^{(d)}$ is preferable to $\text{PMC}_{(MT)}$ due to higher precision at comparable computational complexity.

The value $\text{FVE}_1 \approx 41\%$ for the correlation matrix with dimension $n = 490$ shows that the variation is not concentrated in the first principal component but is distributed among multiple factors. This indicates a more complex structure of dependencies between assets and suggests a higher potential for diversification. At the same time, this result requires the use of multifactor models to adequately describe market risk, since single-factor approaches would be insufficient to capture the observed dynamics.

3.5 Summary

The results confirm the practical applicability and computational efficiency of the *almost optimal* stochastic algorithms for estimating the maximum eigenvalue of correlation matrices in market risk analysis. The analyzed matrices, with dimensions $n = 32$ and $n = 490$, are constructed based on data from *Portfolio Visualizer* and *Yahoo Finance* (see Appendix), where the maximum eigenvalue is interpreted as an indicator of the dominant market factor. Analysis of the correlation matrices shows that the largest eigenvalue, λ_1 , is clearly interpreted as a measure of the presence and strength of a dominant common market factor. Through the corresponding fraction of variance explained FVE_1 , a quantitative assessment is obtained for both the level of systematic risk and the limitations of effective portfolio diversification.

Numerical experiments for the case $n = 32$ show that, for *almost optimal* PMC algorithms, the balance between stochastic and systematic error is achieved at $k = 8$. For *almost optimal* PQMC algorithms that use scrambled Sobol and Halton sequences with default settings, this balance is reached at $k = 11$. In the case when applying the *skip* and *leap* operators, it is achieved at $k = 13$. Furthermore, as the number of realizations N increases, the absolute error decreases monotonically for both PMC and PQMC algorithms with default settings, confirming their stable convergence behaviour.

A significant result is that, for high-dimensional matrices ($n = 490$), the *almost optimal* PQMC algorithms with default settings demonstrate better convergence and higher accuracy compared to the *almost optimal* PMC algorithms. At the same time, the use of *skip* and *leap* operators is effective mainly for small sample sizes N . In contrast, for larger samples, their advantage diminishes and does not guarantee a further improvement in accuracy. Some of the results presented have been published in [38].

In conclusion, the developed *almost optimal* stochastic power algorithms represent a reliable tool for the numerical estimation of spectral characteristics of correlation matrices. This enables a deeper analysis of systematic risk, collective asset dynamics, and diversification efficiency, while also providing a foundation for future applications in financial modeling, portfolio management, and decision support in high-dimensional problems.

Conclusion

This dissertation is devoted to the development, analysis, and application of stochastic numerical methods, namely Monte Carlo (MC) and randomized Quasi-Monte Carlo (QMC) methods for estimating extreme eigenvalues of symmetric matrices and their application to practical problems related to risk assessment and investment portfolio optimization. The main objective is to construct a stable, computationally efficient methodological framework that maintains a controllable balance between systematic and stochastic errors and applies to high-dimensional problems.

Within the scope of the research, the Power (Resolvent) Monte Carlo method and the randomized Power (Resolvent) Quasi-Monte Carlo method for estimating extreme eigenvalues are theoretically justified and investigated. A significant result is the refinement of the construction of random variables when using an almost optimal transition density in the Markov chain, as well as the identification of classes of symmetric matrices for which this construction is simplified. On this basis, almost optimal algorithms (Power and Resolvent MC and QMC) are developed and presented in pseudocode, along with estimates of their computational complexity and conditions for balancing statistical and systematic errors. Theoretical arguments and experimental results confirm that the proposed almost optimal constructions lead to reduced variance and improved accuracy at equal or lower computational cost compared to classical stochastic analogues.

The experimental validation is performed using both pseudo-random number generators (MT and MS) and low-discrepancy sequences (Sobol and Halton sequences), including their scrambled variants. Numerical experiments demonstrate a measurable impact of the choice of generator and sequence type on the accuracy and stability of the estimates. In the tests considered, the MT pseudorandom number generator outperforms the MS generator, while scrambled Sobol sequences outperform scrambled Halton sequences. It is also established that, in order to achieve a comparable balance of errors, the randomized Power (Resolvent) QMC algorithm may require a slightly larger number of transitions in the Markov chain, which is compensated by improved convergence under limited computational resources.

The developed algorithms are applied to a financial mathematics problem related to managing the risk of an investment portfolio. The maximum eigenvalue of the asset correlation matrix is used as an indicator of risk concentration, and its estimation is performed using the proposed MC and QMC algorithms, including additional randomization techniques such as *skip* and *leap* applied to scrambled low-discrepancy sequences (Sobol and Halton sequences).

In conclusion, the dissertation demonstrates that applying the proposed algorithms for spectral estimation, combined with targeted error balancing, yields higher accuracy, stability, and computational efficiency. Future research directions are also outlined, including extensions to robust models, dynamic covariance structures, and further computational optimizations for large-scale sparse symmetric matrices with dimensions on the order of millions.

List of Publications on the Dissertation

The following publications present results from the dissertation:

1. **Gurova, S.-M.**, Karaivanova, A., *Quasi-Monte Carlo Algorithms for Eigenvalue Problems, 4th International Conference Numerical Methods for Scientific Computations and Advanced Applications, NM-SCAA'24*, 18-22, (**2024**) (short communication)
<https://parallel.bas.bg/~elilkova/nmscaa24.pdf>
2. **Gurova, S.-M.**, Atanassov, E., Karaivanova, A., *A Resolvent Quasi-Monte Carlo Method for Estimating the Minimum Eigenvalues Using the Error Balancing*, In: Lirkov, I., Margenov, S. (eds), *Large-Scale Scientific Computing, LSSC'23, LNCS*, vol. **13952**, 394-403, (**2024**), (SJR:0.352, 2024)
https://doi.org/10.1007/978-3-031-56208-2_40
3. **Gurova, S.-M.**, Gurov, T., Karaivanova, A., *On the Stochastic Power Algorithms for Estimating the Maximum Eigenvalue of Symmetric Matrices*, **Contemp. Math.**, Universal Wiser Publisher Pte. Ltd, **6(5)**: 7223-49, (**2025**) doi.org/10.37256/cm.6520257781 (**IF:2.5**, 2024, **WoS: Q1**, <https://wos-journal.info/journalid/23187>)
4. **Gurova, S.-M.**, Gurov, T., Karaivanova, A., *Eigenvalue Estimation in Portfolio Risk: The Role of Skipping and Leaping in Sobol and Halton Sequences*, In: Lirkov, I., Margenov, S. (eds), *Large-Scale Scientific Computing, LSSC'25, LNCS*, **16061**, (accepted)

Approbation of the Results

The results included in the dissertation have been reported at seminars of the section "Scalable Algorithms and Cybersecurity with the Center for High Performance Computing" at IICT-BAS, as well as at international conferences. The following reports related to the topic of the dissertation have been presented:

1. Gurova, S.-M., Monte Carlo Method for Estimating Eigenvalues Using Error Balancing, The 13th International Conference on Large-Scale Scientific Computations, LSSC'21, 07-11.06.2021, Sozopol, Bulgaria. <https://parallel.bas.bg/Conferences/SciCom21/program.html>
2. Gurova, S.-M., A Quasi-Monte Carlo method for estimation of eigenvalues using error balancing, The 15th International Conference on Monte Carlo and Quasi-Monte Carlo methods in Scientific Computing, MCQMC'22, 17-22.07.2022, Linz, Austria. https://www.ricam.oeaw.ac.at/events/conferences/mcqm2022/schedule/MCQMC2022_book_final_version.pdf
3. Gurova, S.-M., Power Monte Carlo method using randomized low discrepancy sequences, the 17th Annual Meeting of the Bulgarian Section of SIAM, 20-22.12.2022, BGSIAM'22, Sofia, Bulgaria. http://www.math.bas.bg/bgsiam/docs/bgsiam_2022_program.pdf
4. Gurova, S.-M., A Resolvent Quasi-Monte Carlo Method for Estimating the Minimum Eigenvalues Using the Error Balancing, The 14th International Conference on Large-Scale Scientific Computations, LSSC'23, 05-09.06.2023, Sozopol, Bulgaria. <https://parallel.bas.bg/~elilkova/lssc23.pdf#page=17>
5. Gurova, S.-M., Quasi-Monte Carlo Algorithms for Eigenvalue Problems, Fourth International Conference Numerical Methods for Scientific Computations and Advanced Applications, NMSCAA'24, 17-21.06.2024, Sozopol, Bulgaria. <https://parallel.bas.bg/Conferences/nmscaa24/program.html>

The articles present results obtained within the framework of one international and three national scientific projects, one of which is divided into two phases, namely:

1. International Scientific Project

- New Scalable Algorithms Using Monte Carlo and Quasi-Monte Carlo Methods and Their Application in the Field of Machine Learning, funded by CAF America, 2021-2024, head: Prof. Emanuil Atanasov.

2. National Scientific Projects

- "Center of Excellence in Informatics and Information and Communication Technologies" (phase 1) under contract BG05M2OP001-1.001-0003, funded by the Operational Program "Science and Education for Smart Growth" 2014-2020, co-financed by the European Union through the European Structural and Investment Funds, head: Corr. Mem. Svetozar Margenov.
- "National Geoinformation Center (NGIC) for monitoring, assessment and forecasting of natural and anthropogenic risks and disasters" object of the "National Roadmap for Scientific Infrastructure 2020-2027", funded by the Ministry of Education and Science under the contracts: D01-321/30.11.2023 and D01-164/28.07.2022, head: Assoc. Prof. Petya Trifonova.
- "National Center for High-Performance and Distributed Computing" (NCHDC) object of the "National Roadmap for Scientific Infrastructure 2020-2027", funded by the Ministry of Education and Science under the contracts: D01-98/26.06.2025, D01-325/01.12.2023, head: Prof. Aneta Karaivanova.
- "Center for Excellence in Informatics and Information and Communication Technologies" (phase 2) under contract BG16RFPR002-1.014-0018-C01, funded under the Research, Innovation and Digitalisation for Smart Transformation Programme 2021-2027 and co-financed by the European Union, head: Corr. Mem. Svetozar Margenov.

Major Scientific and Applied Contributions

The obtained scientific and applied contributions fully correspond to the stated objectives and tasks of the dissertation. They directly reflect the theoretical justification, development, and experimental investigation of Power MC/QMC algorithms and Resolvent MC/QMC algorithms for estimating extreme eigenvalues of symmetric matrices. Furthermore, the contributions are validated through a real-world application in financial mathematics, namely the assessment of market risk in a portfolio of global assets.

1. The main scientific contributions of the dissertation are:

- The Power (Resolvent) Monte Carlo method and the randomized Power (Resolvent) Quasi-Monte Carlo method for estimating extreme eigenvalues of symmetric matrices are theoretically justified and investigated. The construction of the random variables $\theta^{(k)}$ and $\theta^{(k-1)}$ is refined when using an almost optimal transition density in the Markov chain. Classes of symmetric matrices are identified for which this construction is simplified. Theoretical estimates are derived for the relationship between the degree of the resolvent matrix and the length of the Markov chain, as a function of the acceleration parameter and the matrix norm.
- The almost optimal Power MC/QMC and almost optimal Resolvent MC/QMC algorithms are developed based on the proposed refined construction of the random variables. These algorithms are presented in pseudocode form. Estimates of their computational complexity are provided, along with conditions for balancing stochastic and systematic errors. It is demonstrated that the proposed algorithms lead to variance reduction compared to classical stochastic approaches.
- A practical problem from financial mathematics is proposed, related to the assessment of market risk in investment portfolios. The constructed correlation matrix of the portfolio is used to estimate the maximum eigenvalue, which serves as an indicator of risk concentration. The estimation is performed using the almost optimal Power MC and Power QMC algorithms developed in Chapter 1, with the QMC algorithm further enhanced by the parameters *skip* and *leap*.

2. The main applied contributions of the dissertation are:

- The proposed almost optimal Power MC algorithms are implemented using MT and MS pseudo-random number generators, while the Power QMC algorithms are implemented using scrambled Sobol and Halton sequences. Numerical experiments for estimating the maximum eigenvalue of large dense symmetric matrices demonstrate the superiority of the almost optimal algorithms over their classical counterparts in terms of both accuracy and computational efficiency.
- The balance between stochastic and systematic error is numerically investigated and shown to depend significantly on the number of transitions k in the Markov chain and the number of realizations N of the random variables $\theta^{(k)}$ and $\theta^{(k-1)}$. The results indicate that QMC algorithms require approximately two or three additional steps in the Markov chain compared to MC algorithms. The choice of pseudo-random number generator or low-discrepancy sequence has a measurable effect on accuracy and stability; in the experiments conducted, the MT generator outperforms MS, while Sobol sequences outperform Halton sequences.
- The almost optimal Resolvent MC algorithm is implemented using the MT pseudo-random number generator, while the almost optimal Resolvent QMC algorithm is implemented using a scrambled Sobol sequence via the BRODA Sobol RSG generator. Numerical experiments conducted on an HPC cluster with NVIDIA Tesla V100 (32 GB) GPUs to estimate the minimum eigenvalue show that the balance between stochastic and systematic error depends on the number of Markov chain transitions k , the number of realizations N , the degree m of the resolvent matrix and the acceleration parameter q . The results obtained using randomized Sobol sequences and the MT generator demonstrate the same order of convergence. A key factor is the ability to determine a constant that separates the minimum eigenvalue from zero.
- Numerical experiments are conducted to estimate the maximum eigenvalue of two correlation matrices used as indicators of market risk concentration. The first matrix is constructed based on the returns of 32 global assets, while the second uses returns from 490 companies from the *S&P500* index over specific time periods (see Appendix). The estimation is performed using the almost optimal Power MC algorithm with MT/MS generators and the almost optimal Power QMC algorithm, both in default settings and with additional tuning using the *skip* and *leap* parameters for scrambled Sobol and Halton sequences.

Appendix

The numerical experiments are carried out using generated symmetric matrices (Chapter 1, Chapter 2) and real financial data (Chapter 3). The symmetric matrices of sizes 100 and 500 used in Chapter 1 are generated using the MATLAB software environment, while those of sizes 500 and 1000 in Chapter 2 are part of an algorithmic implementation developed in the C++ programming language.

The financial data used to construct the correlation matrix of dimension $n = 32$ in Chapter 3 are obtained from the web platform *Portfolio Visualizer* (www.portfoliovisualizer.com). The studied time horizon covers the period from January 2007 to March 2024 and is based on real market indicators. Based on this empirical data, an investment portfolio is constructed. The portfolio consists of thirty-two global assets, classified according to their investment type as follows:

1. **Eighteen equities** – “US Stock Market”, “US Large Cap Value”, “US Large Cap Growth”, “US Mid Cap”, “US Mid Cap Value”, “US Mid Cap Growth”, “US Small Cap Value”, “US Small Cap Growth”, “US Large Cap”, “US Small Cap”, “US Micro Cap”, “Global ex-US Stock Market”, “European Stocks”, “Pacific Stocks”, “Emerging Markets”, “Intl Developed ex-US Market”, “International ex-US Small Cap”, “International ex-US Value”;
2. **Nine bonds** – “Short Term Treasury”, “Intermediate Term Treasury”, “10-year Treasury”, “Long Term Treasury”, “Total US Bond Market”, “Global Bonds (USD Hedged)”, “Short-Term Investment Grade”, “TIPS (Treasury Inflation-Protected Securities)”, “Corporate Bonds”;
3. **Four alternative investments** – “REIT (Real Estate Investment Trusts)”, “Gold”, “Precious Metals”, “Commodities”;
4. **One cash asset & equivalents** – “Cash”.

The financial data used to construct the correlation matrix of dimension $n = 490$ in Chapter 3 consist of the returns of 490 companies from the *S&P500* (*Standard&Poor's 500*), an index measuring the performance of the U.S. stock market, over 5 years from 2021 to 2025. The return data were downloaded from *Yahoo Finance* (<https://finance.yahoo.com/>) using the *yFinance* library, available via the *PyPI* repository (<https://pypi.org/project/yfinance/>). This library enables automated retrieval of historical price series and market indicators through an interface to the *Yahoo Finance* website, allowing programmatic data extraction for specific financial instruments over a given time interval and their subsequent processing in an analytical environment.

1. The generated symmetric matrices of sizes 100 and 500 used in Chapter 1 are published on the **GitHub** platform and are available at:
 - https://github.com/smgurova/Symmetric-matrices_chapter-1.git
2. The financial report provided by *Portfolio Visualizer* is available on **GitHub** at:
 - <https://github.com/smgurova/AssetAllocation.git>
3. The correlation matrix of the thirty-two global assets used in the numerical experiments in Chapter 3 is available on **GitHub** at:
 - https://github.com/smgurova/correlation-matrix_chapter-4.git
4. The return data of 490 companies from the *S&P500* (*Standard&Poor's 500*) for the period 2021–2025, along with the resulting correlation matrix used in the numerical experiments in Chapter 3, are available on **GitHub** at:
 - <https://github.com/smgurova/Data-S-P500.git>

Bibliography

- [1] Alexandrov, V., Atanassov, E., Dimov, I., Branford, S., Thandavan, A., Weihrauch, C., Parallel Hybrid Monte Carlo Algorithms for Matrix Computations. In: Sunderam, V.S., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) Computational Science – ICCS 2005. LNCS, vol. 3516, Springer, 752-759, (2005), https://doi.org/10.1007/11428862_102
- [2] Alexandrov, V., Davila, D., Esquivel-Flores, O., Karaivanova, A., Gurov, T., Atanassov, E., On Monte Carlo and quasi-Monte Carlo Methods for Matrix Computations, Lecture Notes in Computer Science, **10665**, Springer-Verlag, 249-257, (2018), DOI: https://doi.org/10.1007/978-3-319-73441-5_26
- [3] Alexandrov, V., Esquivel-Flores, O., Ivanovska, S., Karaivanova, A., On the Preconditioned Quasi-Monte Carlo Algorithm for Matrix Computations. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds.) Large-Scale Scientific Computing. LSSC 2015. Lecture Notes in Computer Science, vol. **9374**, Springer, Cham., 163-171, (2015), https://doi.org/10.1007/978-3-319-26520-9_17
- [4] Alexandrov, V., Esquivel-Flores, O.A., Towards Monte Carlo preconditioning approach and hybrid Monte Carlo algorithms for matrix computations, Computers & Mathematics with Applications, vol. **70**(11), 2709-2718, (2015), <https://doi.org/10.1016/j.camwa.2015.08.035>
- [5] Almulihi A.H. H., High-Performance Quasi-Monte Carlo Integration and Applications, Dissertations. 3452, Western Michigan University, (2019), <https://scholarworks.wmich.edu/dissertations/3452>, (Last accessed: January 2026)
- [6] Branford, S., Hybrid Monte Carlo Methods for Linear Algebra Problems, Ph.D. dissertation, School of Systems Engineering, The University of Reading, (2009)
- [7] BRODA's Sobol RSG: broda.co.uk/software.html (Last accessed: 2026/01/11)
- [8] Brouwer, A.E., Haemers, W.H., Spectra of Graphs, Springer, (2012) <https://doi.org/10.1007/978-1-4614-1939-6>
- [9] Caflisch, R. E., Monte Carlo and quasi-Monte Carlo methods, Acta Numerica, Cambridge University Press, 1-49, (1998) <https://www.math.ucla.edu/~caflisch/Pubs/Pubs1995-1999/actaNumerica1998.pdf> (Last accessed: 2026/01/11)
- [10] Chi, H. Scrambled Quasirandom Sequences and Their Applications, Florida State University, (2004) http://purl.flvc.org/fsu/fd/FSU_migr_etd-3823, (Last accessed: 2026/01/11)
- [11] Chi, H, Beerli, P., Evans, D. W., Mascagni, M., On the Scrambled Sobol Sequence, LNCS 3516, Springer, 775-782, (2005)
- [12] Chung, F., Spectral Graph Theory, CBMS Regional Conference Series in Mathematics 92, AMS, (1997).
- [13] Dick, J., Pillichshammer, F., Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration, Cambridge University Press (2010), <https://doi.org/10.1017/CB09780511761188>
- [14] Dekking, M., A Modern Introduction to Probability and Statistics. Springer, 181–190, (2005), ISBN: 9781852338961
- [15] Dimov, I., Monte Carlo algorithms for linear problems, Pliska Stud. Math. Bulgar., **13**, 57-77, (2000)
- [16] Dimov, I., Dimov, T., Gurov, T., A new iterative Monte Carlo approach for inverse matrix problem, Journal of Computational and Applied Mathematics, **92**(1), 15-35, (1998)
- [17] Dimov, I., Tonev, O., Monte Carlo algorithms: performance analysis for some computer architectures, Journal of Computational and Applied Mathematics, **48**, 253-277, (1993)
- [18] Dimov, I., Alexandrov, V., Karaivanova, A, Implementation of Monte Carlo Algorithms for Eigenvalue Problem Using MPI, PVM/MPI'98, LNCS, **1497**, 346-353, (1998)
- [19] Dimov, I., Philippe, B., Karaivanova, A., Weihrauch, C., Robustness and applicability of Markov chain Monte Carlo algorithms for eigenvalue problems, Applied Mathematical Modelling, **32**(8), 1511-1529, (2008)

- [20] Dimov, I., Alexandrov, V., Karaivanova, A., Parallel resolvent Monte Carlo algorithms for linear algebra problems, *Mathematics and Computers in Simulation*, **55**(1), 25-35, Published by Elsevier BV., (2005)
- [21] Dimov, I., Karaivanova, A.: A Power Method with Monte Carlo Iterations, *Proceedings Recent Advances in Numerical Methods and Applications II*, World Scientific, 239-247, (1999), https://doi.org/10.1142/9789814291071_0022
- [22] Dimov, I., *Monte Carlo Methods for Applied Scientists*, World Scientific Publishing Co. Pte. Ltd., (2007), ISBN: 13 978-981-02-2329-8
- [23] Dimov, I., Karaivanova, A., Iterative Monte Carlo algorithms for linear algebra problems, *Springer Lecture Notes in Computer Science*, **1196**, 66-77, (1996)
- [24] Dimov, I., Karaivanova, A., Parallel computations of eigenvalues based on a Monte Carlo approach, *Monte Carlo Methods and Applications*, Vol. 4, Num. 1: 33-52, 1998
- [25] Dimov, I.T., et al.: Monte Carlo Algorithms for Calculating Eigenvalues. In: Niederreiter, H., et.al. (eds.) *Monte Carlo and Quasi-Monte Carlo Methods*, 1996, *Lecture Notes in Statistics*, vol. **127**, Springer, New York, NY., (1998)
- [26] Erkan, U., A precise and stable machine learning algorithm: eigenvalue classification (EigenClass), *Neural Comput & Applic*, **33**, 5381-5392, (2021), <https://doi.org/10.1007/s00521-020-05343-2>
- [27] Fathi, B., Liu, B., Alexandrov, V., Mixed MC parallel algorithms for Matrix Computation, *LNCS*, **2330**, 609-618, (2002)
- [28] Faure H., Lemieux Ch.: Generalized Halton sequences in 2008: A comparative study. *ACM Transactions on Modeling and Comp. Sim*, Vol. **19**, Issue 4, 1-31, (2009), <https://doi.org/10.1145/1596519.1596520>
- [29] Faure, H., Discrepance de suites associées a un système de numération (en dimension s), *Acta Arithmetica*, Vol. **41**, 337-351, (1982)
- [30] Faure, H., Good permutations for extreme discrepancy, *Journal of Number Theory*, Vol. **42**, 47-56, (1992)
- [31] Fischer, H., A History of the Central Limit Theorem: From Classical to Modern Probability Theory, *Sources and Studies in the History of Mathematics and Physical Sciences*. New York: Springer, (2011), <http://dx.doi.org/10.1007/978-0-387-87857-7>
- [32] Fishman, G.S., Multiplicative congruential random number generators, *Math. Comp.*, **54**, 331-344, (1990)
- [33] Golub, G., Loan, Ch. F. Van, *Matrix computation*, 4th edition, The Johns Hopkins University Press, (2013), ISBN: 13-978-1-4214-0794-4, <https://math.ecnu.edu.cn/~jypan/Teaching/books/2013%20Matrix%20Computations%204th.pdf>
- [34] Gilboa, G., *Nonlinear Eigenproblems in Image Processing and Computer Vision*, Publisher: Springer Cham, ed.number 1, (2018), eBook ISBN: 978-3-319-75847-3
- [35] Gurova, S.-M., Atanassov, E., Karaivanova, A., A Resolvent Quasi-Monte Carlo Method for Estimating the Minimum Eigenvalues Using the Error Balancing, In: Lirkov, I., Margenov, S. (eds.), *Large-Scale Scientific Computing, LSSC 2023*, *Lecture Notes in Computer Science*, vol. **13952**, 394-403, (2024)
- [36] Gurova, S.-M., Karaivanova, A., Quasi-Monte Carlo Algorithms for Eigenvalue Problems, *Fourth International Conference Numerical Methods for Scientific Computations and Advanced Applications, NM-SCAA'24*, 17-21, (2024) (short communication) <https://parallel.bas.bg/~elilkova/nmscaa24.pdf>
- [37] Gurova, S.-M., Gurov, T., Karaivanova, A., On the Stochastic Power Algorithms for Estimating the Maximum Eigenvalue of Symmetric Matrices, *Contemp. Math.*, Universal Wiser Publisher Pte. Ltd., **6**(5): 7223-7249, (2025) <https://doi.org/10.37256/cm.6520257781>
- [38] Gurova, S.-M., Gurov, T., Karaivanova, A., Eigenvalue Estimation in Portfolio Risk: The Role of Skipping and Leaping in Sobol and Halton Sequences, In: Lirkov, I., Margenov, S. (eds), *Large-Scale Scientific Computing, LSSC'25*, *LNCS*, **16061**, (accepted for publication)
- [39] Halton, J., On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numerische Mathematik*, **2**, 84-90, (1960), <https://doi.org/10.1007/BF01386213>

- [40] Halton, J., Smith, GB, Algorithm 247: Radical-Inverse Quasi-Random Point Sequence, Communications of the ACM, Vol. **7**, 701-702, (1964)
- [41] Hammersley, J.M., Handscomb, D.C., Random, Pseudorandom, and Quasirandom Numbers. In: Monte Carlo Methods. Monographs on Applied Probability and Statistics. Springer, Dordrecht, (1964), https://doi.org/10.1007/978-94-009-5819-7_3
- [42] Higham, N. J., Computing the nearest correlation matrix—a problem from finance, IMA Journal of Numerical Analysis, **22**(3): 329–343, (2002), <https://doi.org/10.1093/imanum/22.3.329>
- [43] Isaacson, E., Keller, H. B., Analysis of Numerical Methods, Dover Publications, Inc., N.Y., Wiley, (1996), ISBN: 0-486-68029-0
- [44] Ivanovska, S., Karaivanova, A., Manev, N., Numerical Integration Using Sequences Generating Permutations. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds) Large-Scale Scientific Computing. LSSC 2011. Lecture Notes in Computer Science, vol. **7116**, Springer, Berlin, Heidelberg, 455-463, (2011), https://doi.org/10.1007/978-3-642-29843-1_51
- [45] Joe, S., Sloan, I., Implementation of a lattice method for numerical multiple integration, ACM Transactions on Mathematical Software (TOMS), Vol. **19**, Issue 4, 523–545, (1993), <https://doi.org/10.1145/168173.168425>
- [46] Joe S., Kuo F.Y.: Constructing Sobol Sequences with Better Two-Dimensional Projections, SIAM Journal on Scientific Computing, **30**(5): 2635-2654, (2008), <https://doi.org/10.1137/070709359>
- [47] Johnstone, I. M., On the distribution of the largest eigenvalue in principal components analysis, Ann. Statist. **29**(2), 295-327, (2001), <https://doi.org/10.1214/aos/1009210544>
- [48] Karaivanova, A., Quasi-Monte Carlo Methods for Some Linear Algebra Problems, Convergence and complexity, Serdica J. Computing, **4**, 57-72, (2010)
- [49] Kolda, T. and Bader, B., Tensor Decompositions and Applications, SIAM Review, **51**(3), 455-500, (2009) <https://doi.org/10.1137/07070111X>
- [50] Kroese, D.P., Taimre, T., Botev, Z. I., Handbook of Monte Carlo Methods, John Wiley & Sons Inc., New York, N.Y., (2011), ISBN:9780470177938
- [51] Laloux, L., Cizeau, P., Bouchaud, J.-P., Potters, M., Noise Dressing of Financial Correlation Matrices, Phys. Rev. Lett., **83**(7), 1467-1470, American Physical Society, (1999), <https://doi.org/10.1103/PhysRevLett.83.1467>
- [52] Lemieux, C., Monte Carlo and Quasi-Monte Carlo Sampling. Springer, (2009), <https://doi.org/10.1007/978-0-387-78165-5>
- [53] L'Ecuyer, P. Uniform random number generation. Ann Oper Res, **53**, 77–120, (1994), <https://doi.org/10.1007/BF02136827>
- [54] L'Ecuyer, P., Efficient and portable combined random number generators, Communications of the ACM, vol. **31**, Issue 6, 742-751, (1988), <https://doi.org/10.1145/62959.62969>
- [55] Lobo, P.D.C., Mikhailov, M., Ozisik, M.N., On the complex eigenvalues of Luikov system of equations, Drying Technology, **5**(2), 273-286, (1987), <https://doi.org/10.1080/07373938708916540>
- [56] Loperfido, N., Skewness-based projection pursuit: A computational approach, Computational Statistics and Data Analysis, **120**, 42-57, (2018), <https://doi.org/10.1016/j.csda.2017.11.001>
- [57] Macdonald, R.R., Correlation and Covariance Matrices. In Wiley StatsRef: Statistics Reference Online (eds. N. Balakrishnan, T. Colton, B. Everitt, W. Piegorsch, F. Ruggeri and J.L. Teugels), (2014), <https://doi.org/10.1002/9781118445112.stat06481>
- [58] Mascagni, M., Karaivanova, A., A Parallel Quasi-Monte Carlo Method for Solving Systems of Linear Equations, P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS **2330**, Springer-Verlag Berlin Heidelberg, 598-608, (2002)
- [59] Mascagni, M., Karaivanova, A., A Parallel Quasi-Monte Carlo Method for Computing Extremal Eigenvalues, K.-T. Fang et al. (eds.), Monte Carlo and Quasi-Monte Carlo Methods, Springer-Verlag Berlin Heidelberg, 369 – 380, (2000)

- [60] Mascagni, M., Karaivanova, A., Matrix Computations Using Quasirandom Sequences, Lecture Notes in Computer Science, (Wulkov, Yalamov, Wasniewsky eds.), Vol. **1988**, Springer, 552-559, (2001)
- [61] Mascagni, M., Karaivanova, A., A Monte Carlo Approach for Finding More than One Eigenpair. In: Dimov, I., Lirkov, I., Margenov, S., Zlatev, Z. (eds) Numerical Methods and Applications - NMA 2002, Lecture Notes in Computer Science, vol.**2542**, Springer, 123-131, (2003), https://doi.org/10.1007/3-540-36487-0_13
- [62] Matsumoto, M., Nishimura, T., Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-Random Number Generator, ACM Transactions on Modeling and Computer Simulation, vol. 8(1), 3-30, (1998), <https://doi.org/10.1145/272991.272995>
- [63] Mehrdoust, F., Algorithm for Estimating the Condition Number of Matrices, Math. Reports, **15(65)**(3), 203-210, (2013)
- [64] Muminov M.I., Rasulov T.H.: Embedded Eigenvalues of a Hamiltonian in Bosonic Fock Space. Commun. Math. Anal., **17**(1), 1 - 22, (2014)
- [65] Mushtaq, A., Noreen, A., Olaussen, K., Numerical Solutions of Quantum Mechanical Eigenvalue Problems, Front. Phys., **8**:390, (2020), <https://doi.org/10.3389/fphy.2020.00390>
- [66] Niederreiter, H., Random Number Generation and Quasi-Monte Carlo Methods, SIAM (1992), <https://doi.org/10.1137/1.9781611970081>
- [67] Owen, A., Randomly Permuted (t,m,s)-Nets and (t, s)-Sequences. In: Niederreiter, H., Shiue, P.J.S. (eds.) Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing. Lecture Notes in Statistics, vol. **106**, Springer, New York, 299-317, (1995), https://doi.org/10.1007/978-1-4612-2552-2_19
- [68] Owen, A., Monte Carlo variance of scrambled net quadrature, SIAM Journal on Numerical Analysis, Vol. **34**, No. 5, 1884-1910, (1997), <https://www.jstor.org/stable/2952022>
- [69] Owen, A., Variance with alternative scramblings of digital nets, ACM Transactions on Modeling and Computer Simulations, **13**(4), 363-378, (2003)
- [70] Padányi, V., Herendi, T., Generalized Middle-Square Method, Annales Mathematicae et Informaticae, **56**, 95-108, (2022), <https://doi.org/10.33039/ami.2022.12.003>
- [71] Park, Stephen K.; Miller, Keith W., Random Number Generators: Good Ones Are Hard To Find, Communications of the ACM, vol. **31**(10), 1192-1201, (1988), doi:10.1145/63039.63042
- [72] Rometsch, M., Quasi-Monte Carlo Methods in Finance with Application to Optimal Asset Allocation, Druck Diplomica Verlag GmbH, Hamburg (2008).
- [73] Rotter, I., A non-Hermitian Hamilton operator and the physics of open quantum systems, J. Phys. A: Math. Theor., Vol. **42**, No. 15, 153001, (2009), <https://doi.org/10.1088/1751-8113/42/15/153001>
- [74] Sobol I., Shukhman B., Integration with quasirandom sequences: Numerical experience., Internat. J. Modern Phys. C, **6**(2), 263-275 (1995)
- [75] Sobol I., Asotsky D., Kreinin A., Kucherenko S., Construction and Comparison of High-Dimensional Sobol' Generator, WILMOTT magazine, 64-79, (2012)
- [76] Sobol I., Uniformly distributed sequences with an additional property of uniformity, USSR Computational Mathematics and Mathematical Physics, **16**(5), 236-242, (1976), [https://doi.org/10.1016/0041-5553\(76\)90154-3](https://doi.org/10.1016/0041-5553(76)90154-3)
- [77] Spiegel, M. R., Theory and Problems of Probability and Statistics, 2nd ed. New York: McGraw-Hill, (1992).
- [78] Straßburg J, Alexandrov, V., Enhancing Monte Carlo preconditioning methods for matrix computations, Procedia Computer Science, **29**, 1580-1589, (2014), <https://doi.org/10.1016/j.procs.2014.05.143>
- [79] Straßburg, J., On hybrid and resilient Monte Carlo methods for linear algebra problems, Ph.D. dissertation, The University of Reading, (2014)
- [80] Sturmfels, B., Tensors and Their Eigenvectors, Notices of the AMS, **63**(6), 604-606, (2016) <http://dx.doi.org/10.1090/noti1389>

- [81] Sugita, H., Monte Carlo Method, Random Number, and Pseudorandom Number, MSJ Memoirs, **25**: 133, (2011), <https://doi.org/10.2969/msjmemoirs/025010000>
- [82] Taft K., Vajargah, F., MC method for solving system of linear algebraic equations with minimum markov chains, International Conference PDPTA'2000, Las Vegas, (2000)
- [83] Vajargah F., et al., Parallel Monte Carlo algorithms for matrix computations, Ph.D. dissertation, The University of Reading, (2001)
- [84] Vajargah, F., Hassanzadeh, Z., Improvements on the hybrid Monte Carlo algorithms for matrix computations, Sadhana, Indian Academy of Science, **44**(1), <https://doi.org/10.1007/s12046-018-0983-y>
- [85] Van der Corput, J., Verteilungsfunktionen (Erste Mitteilung), Proceedings of the Koninklijke Akademie van Wetenschappen te Amsterdam (in German), **38**: 813–821, (1935), Zbl 0012.34705
- [86] Wang, X., Hickernell, F., Randomized Halton sequences, Mathematical and Computer Modelling, **32**(7–8), 887–899, (2000) [https://doi.org/10.1016/S0895-7177\(00\)00178-3](https://doi.org/10.1016/S0895-7177(00)00178-3)
- [87] Боянов, Б., Лекции по Числени методи, изд. Дарба София, (2008), ISBN: 954-90126-1
- [88] Димитров, Б., Янев, Н., Вероятности и статистика, Софтех, София, (2007)
- [89] Караиванова, А., Стохастични числени методи и симулации, изд. Деметра ЕООД, (2012), ISBN: 978-954-9526-78-3
- [90] Сендов, Б., Попов, В., Числени методи, Университетско издание „Св. Климент Охридски“, първа част, (1996), ISSN: 954-07-0994-6
- [91] Стоянов, Й., Миразчийски, И., Игнатов, Ц., Танушев, М., Ръководство за упражнения по теория на вероятностите. Издателство „Наука и изкуство“, София, (1976).
- [92] Соболев, И. М., Численные методы Монте-Карло, изд. Москва, Наука, (1973)
- [93] MathWorks: haltonset - Create a Halton point set. <https://www.mathworks.com/help/stats/haltonset.html>, (Last accessed: 2026/01/11)
- [94] MathWorks: sobolset - Create a Sobol point set. <https://www.mathworks.com/help/stats/sobolset.html>, (Last accessed: 2026/01/11)



БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ

АВТОРЕФЕРАТ НА ДИСЕРТАЦИЯ

за присъждане на образователна и научна степен “доктор” по докторска програма „Информатика”

СТОХАСТИЧНИ ЧИСЛЕНИ МЕТОДИ ЗА ОЦЕНКА НА СОБСТВЕНИ СТОЙНОСТИ

Силви-Мария Тодорова Гюрова

Ръководител: Проф. Анета Караиванова

Научно жури:

Проф. Михаил Тодоров
Проф. Надя Златева,
Доц. Иван Христов
Доц. София Ивановска
Проф. Иван Георгиев



**Институт по информационни и комуникационни
технологии**

**Секция „Скалируеми алгоритми и киберсигурност
с Център по високопроизводителни
пресмятания”**

Обща характеристика на дисертационния труд

Актуалност на темата

Задачата за намиране и оценяване на екстремалните собствени стойности на квадратни, в частност симетрични матрици, заема централно място в линейната алгебра поради ключовата си роля в широк кръг научни и приложни области. Такива задачи възникват в квантовата механика, при уравнението на Шрьодингер [65], собствените стойности на Хамилтоновия оператор определят енергийния спектър на системата, като екстремалните стойности дават информация за основното и възбудените състояния [64, 73]. Във финансовата математика анализът на екстремалните собствени стойности на ковариационните матрици е съществен за моделиране на риска и оптимизация на портфейли, включително за оценка на систематичния пазарен риск и възможностите за диверсификация [51, 72]. В спектралната теория на графите тези стойности са свързани със структурните свойства като свързаност, разширение и устойчивост, които са определящи за ефективността на алгоритми в машинното обучение и мрежовия анализ [8, 12, 26]. В обработката на изображения и компютърното зрение собствените стойности играят ключова роля при анализа на главните компоненти, редукцията на размерността, потискането на шума и спектралната сегментация [34, 47]. В тензорния анализ и обработката на многомерни данни, максималната собствена стойност и спектралните норми определят поведението на итеративните методи и техниките за регуляризация, с приложения в обработката на сигнали и изкуствения интелект [49, 56, 80].

Съществуват различни числени подходи за изчисляване на собствени стойности и вектори - точни (директни), итерационни и стохастични. Към директните методи спадат QR декомпозицията и решаването на характеристичния полином при матрици с малка размерност. Те намират решението за краен брой стъпки и имат изчислителна сложност от порядък $\mathcal{O}(n^3)$, където n е размерността на матрицата. Основен недостатък на точните методи е, че при матрици с големи размерности, изчисленията с крайна точност водят до закръгляване и натрупване на грешки, което може да наруши точността на собствените стойности. [87, 90].

Итерационните методи, като например Степенния метод, Обратния Степенен метод, метод на Якоби, метод на Арнолди и метод на Ланцош [33, 43], имат по-малка изчислителна сложност $\mathcal{O}(n^2k)$, където n е размерността на матрицата, а k е броят на итерациите. Този тип методи често се характеризират с бавна сходимост, силна зависимост от началното приближение, значителни изисквания към паметта и ограничена ефективност при паралелизация.

Стохастичните числени методи предлагат ефективна алтернатива, като осигуряват линейна зависимост между размерността на задачата и необходимата памет, както и естествен паралелизъм и относителна лекота на реализация. Макар и с по-бавна сходимост и по-малка прецизност в сравнение с детерминистичните методи, те позволяват решаването на задачи с много големи размерности и са особено подходящи за оценяване на екстремални собствени стойности на симетрични матрици.

В условията на бързото развитие на високопроизводителните изчислителни системи (англ. *High-Performance Computing*, HPC) и навлизането на нови хардуерни архитектури нараства необходимостта от разработване и внедряване на софтуерни решения, базирани на ефективни стохастични алгоритми за задачи с висока размерност, включително за оценяване на екстремални собствени стойности [18, 24, 59]. Това обуславя засилен изследователски интерес към повишаване на точността, ускоряване на сходимостта и разширяване на приложимостта на стохастичните методи в контекста на научните изчисления, машинното обучение и анализа на данни, съобразно съвременните изчислителни възможности.

Обзор на основните резултати в областта

Първите публикации, свързани с Монте Карло (МК) методите в линейната алгебра, датират от 50-те и 60-те години на XX век, като първоначално не е разглеждан пряко проблемът за приближено изчисляване на собствени стойности. Задачата за намиране на собствените стойности и собствените вектори на матрици обикновено възниква при решаване на системи от линейни алгебрични уравнения (СЛАУ) и стохастични диференциални уравнения (СДУ). В практиката, за матрици с големи размерности обикновено е необходимо да се оценят само екстремалните собствени стойности, които определят числото на обусловеност и ключови свойства на системата [63, 87, 90]. Такива задачи възникват в обработката на изображения и сигнали, системите за управление, машинното обучение и квантовата механика, където например енергийните нива на квантовите системи се задават от собствените стойности на Хамилтоновия оператор [64, 73]. Наред с това са широко използвани детерминистични методи като Степенния метод, итерацията на коефициента на Релей, Резолвентния Степенен метод и алгоритъма на Ланцош [33, 43].

Монте Карло методът за оценка на доминиращата собствена стойност е предложен от Собол през 1973 г. [92], а за най-малката собствена стойност - от Михайлов през 1987 г. [55]. Значим принос в развитието на Степенния Монте Карло (СМК) метод имат Димов и Караиванова в периода 1996–2000 г. [15, 18, 21, 23, 24, 25]. През следващите десетилетия са разработени итеративни МК и Квази-Монте Карло (КМК) алгоритми за намиране на екстремални собствени стойности, базирани на Степенния и Резолвентния Степенен метод [3, 19, 21, 25, 60]. Използването на резолвентната матрица за оценка на най-малката собствена стойност е въведено от Димов и Караиванова през 1998 г. [24].

Първите Квази-Монте Карло методи за намиране на екстремални собствени стойности са предложени от Караиванова и Маскани в периода 2001–2003 г. [58, 59, 60, 61]. В редица изследвания са използвани КМК методи, базирани на редици с малък дискрепанс, включително редици на Собол [74, 75, 76], Холтън [39, 40] и Фор [28, 29, 30], които осигуряват по-равномерно и детерминистично покритие на пробното пространство [2, 3, 10, 11, 44, 48]. В сравнение със стандартните МК подходи, КМК методите се отличават с по-добра гъвкавост, приложимост към матрици с големи размерности, относителна простота и висока паралелна ефективност [16, 22, 61].

През последните две десетилетия са разработени и анализирани различни стратегии за паралелизация на Монте Карло алгоритми [4, 5, 6, 78, 79, 83]. Паралелни МК методи за решаване на СЛАУ, базирани на итерацията на Якоби, са представени от Димов [16], както и подходи с минимални вериги на Марков и минимални комуникации [27, 82]. Основна идея в тези разработки е балансирането на стохастичната и систематичната грешка [16]. Разработени са и хибридни Монте Карло алгоритми за матрични пресмятания [1, 84], включително паралелни реализации за оценка на най-голямата и най-малката собствена стойност на матрици, предложени от Димов, Маскани и Караиванова [20, 24, 59].

Цели и задачи на дисертация

Основната цел на дисертационния труд е да се предложат и изследват стохастични числени методи за оценяване на собствени стойности, като се разработят и анализират Монте Карло и рандомизирани Квази-Монте Карло алгоритми за приближено пресмятане на екстремните собствени стойности на симетрични квадратни матрици. За постигане на основната цел са формулирани следните задачи:

1. Да се разгледат и изучат Степенен (Резолвентен) Монте Карло метод и рандомизиран Степенен (Резолвентен) Квази-Монте Карло метод за оценяване на екстремалните собствени стойности на симетрични матрици, както и да се анализира баланса между възникващите грешки - стохастична и систематична.
2. Да се разработят и изследват ефективни Монте Карло и рандомизирани Квази-Монте Карло алгоритми, базирани на тези стохастични методи за числено пресмятане на екстремалните собствени стойности на симетрични квадратни матрици, като се използват генератори на псевдослучайни числа и генератори на редици с малък дискрепанс.
3. Да се проведат числени експерименти с разработените алгоритми, като се обоснове изборът на специализиран софтуер за генериране на редици с малък дискрепанс, в частност редици на Собол и Холтън, и числено да се изследва баланса между стохастичната и систематичната грешка в зависимост от определящите ги параметри.
4. Да се формулира и изследва задача от областта на финансовата математика, свързана с управлението на пазарния риск на портфейл от активи, при който оценката на риска се основава на определянето на максималната собствена стойност на корелационна матрица. Да се приложат разработените в Глава 1 Степенен Монте Карло и рандомизиран Степенен Квази-Монте Карло алгоритми за числено оценяване на най-голямата собствена стойност и да се анализира тяхната ефективност при оценяване на пазарния риск.

Методология на изследването

Методологията на изследването в дисертационния труд се основава на фундаментални научни резултати от следните области:

- Теория на вероятностите и математическа статистика - вероятностни пространства; случайни величини и техни характеристики като функции на разпределение, функция на плътност, математическо очакване, дисперсия; закон за големите числа; централна гранична теорема; вероятностна грешка,

доверителен интервал, верига на Марков; извадки на случайни величини; статистически тестове за изследване на генератори на псевдослучайни величини.

- Математически анализ - редици и редове, сходимост на редове, критерии за сходимост, систематична грешка, норми и метрични пространства.
- Числени методи, линейна и изчислителна алгебра - итерационни методи за СЛАУ и собствени стойности; матрични норми; оптимизационни методи; алгоритми за генериране на псевдослучайни числа, стохастични числени методи (Монте Карло и Квази-Монте Карло методи).
- Теория на числата и по-точно раздела за дискретна и аналитична теория на числата - равномерно разпределени редици, разбъркване (англ. *discrepancy*), радикално-инверсни функции, разлагане по прости основи.
- Информатика/компютърни науки - алгоритми и структури от данни; програмни кодове, написани на МАТЛАВ, C++ с използване на генератори на псевдослучайни числа като MT и MS; библиотеки за рандомизирани редици на Собол и Холтън.

Част от числените експерименти са извършени на високопроизводителния клъстър, който се състои от 12 сървъра Fujitsu Primergy RX 2540 M4. Всеки сървър е конфигуриран с GPU карта от вида NVIDIA Tesla V100 32GB и CPU 2 x Intel Xeon Gold 5118 2.30GHz 24 cores, всеки с 128 GB RAM. Редиците на Собол са получени чрез BRODA's Sobol Randomized Sequence Generator (RSG) [7].

В следващите секции сме формулирали основната задача за оценяване на собствени стойности на симетрична матрица и сме описали МК и КМК подходите за числено решаване на многомерни задачи и грешките, с които се характеризират тези методи.

Формулиране на задачата

Нека разгледаме матрицата $A = \{a_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$, $a_{ij} = a_{ji}$, $i, j = 1, \dots, n$. Задачата за намиране на собствени стойности $\lambda(A)$ се формулира по следния начин:

$$Ax = \lambda x, \quad x \in \mathbb{R}^n \text{ е произволен ненулев вектор.} \quad (1)$$

Всички собствени стойности на симетричните матрици са реални числа. Без ограничение на общността предполагаме следната наредба на собствените стойности на A :

$$\lambda_{\min} = |\lambda_n| \leq |\lambda_{n-1}| \leq \dots \leq |\lambda_2| < |\lambda_1| = \lambda_{\max}. \quad (2)$$

Матрицата A може да се представи векторно и по следния начин:

$$A = \{a_{ij}\}_{i,j=1}^n = (\mathbf{a}_1, \dots, \mathbf{a}_i, \dots, \mathbf{a}_n)^T, \text{ където } \mathbf{a}_i = (a_{i1}, \dots, a_{in}), \quad i = 1, \dots, n.$$

Символът T означава операцията *транспониране*. За нуждите на изследването се използват следните векторни и матрични норми:

$$\|\mathbf{h}\| = \|\mathbf{h}\|_1 = \sum_{i=1}^n |h_i|, \quad \|\mathbf{a}_i\| = \|\mathbf{a}_i\|_1 = \sum_{j=1}^n |a_{ij}|, \quad \|A\| = \|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|, \quad j = 1, \dots, n.$$

Прилагането на стохастични числени методи за оценка на екстремалните собствени стойности $\lambda_{\min}$ и $\lambda_{\max}$ на така дефинираната задача, включва разработването и изследването на Степенни (Резолвентни) Монте Карло и рандомизирани Степенни (Резолвентни) Квази-Монте Карло алгоритми за тяхното приближено пресмятане. Под стохастични числени методи по-нататък ще разбираме Монте Карло методите и тяхната разновидност, рандомизирани Квази-Монте Карло методи.

Монте Карло методи

Монте Карло (МК) методите се основават на конструиране на случайна величина θ , която всъщност представлява измерима функция $\theta = \theta(\omega)$, дефинирана във вероятностното пространство $(\Omega, \mathfrak{A}, \mathbb{P})$ [88], с дефиниционна област Ω и с област на стойностите $\mathbb{R}$ (множеството на реалните числа). Условието при конструирането на случайната величина θ е свързано с математическото ѝ очакване, като изискването е

математическото очакване да съществува и да е равно на търсеното решение на дадена задача или на функционал от решението на изходната задача.

$$E[\theta] = I, \quad (3)$$

където I е търсеното решение (или стойността на функционал от решението). По определение $E[\theta]$ съществува тогава и само тогава, когато съществува $E|\theta|$.

В зависимост от конкретната задача, случайната величина θ може да бъде дискретна или непрекъсната. В дискретния случай математическото очакване се задава чрез:

$$E[\theta(\omega)] = \sum_i \mathbb{P}\{\theta(\omega) = x_i\} x_i = \sum_i p_i x_i = I, \text{ където } \sum_i p_i = 1,$$

а в непрекъснатия случай:

$$E[\theta] = \int_{-\infty}^{\infty} x p(x) dx, \text{ където } \int_{-\infty}^{\infty} p(x) dx = 1,$$

където $p(x) \geq 0$ е вероятностна плътност на разпределение. Функцията на разпределение $F_\theta(x) = \mathbb{P}\{\theta < x\}$ описва натрупаната вероятност и е свързана с $p(x)$ чрез стандартните диференциални зависимости [88, 91]. При реализация на МК алгоритъм се генерира извадка $\theta_1, \theta_2, \dots, \theta_N, \dots$ от независими и еднакво разпределени реализации на случайната величина θ , като приближената стойност на I се определя чрез средната стойност:

$$\bar{\theta}_N = \frac{1}{N} \sum_{i=1}^N \theta_i. \quad (4)$$

Сходимость и оценка на грешката при МК методите

Теоретичната обосновка на МК методите се базира на закона за големите числа (ЗГЧ). Когато математическото очакване е крайно, редицата от средни стойности $\bar{\theta}_N$ (при $N \rightarrow \infty$) е сходяща по вероятност към I (слаб закон на големите числа, теорема на Хинчин) [14, 88]. Необходимото и достатъчно условие, за да бъде сходимостьта почти сигурна (усилен закон на големите числа, теорема на Колмогоров), е да има математическо очакване.

Предполагаме, че случайната величина θ има крайна дисперсия:

$$Var[\theta] = E[\theta - E\theta]^2 = E[\theta^2] - (E[\theta])^2 < \infty.$$

Тогава централната гранична теорема (ЦГТ) осигурява асимптотично нормално разпределение на грешката [31, 88]. В резултат се получава вероятностната оценка:

$$\lim_{N \rightarrow \infty} \mathbb{P} \left\{ \left| \frac{1}{N} \sum_{i=1}^N (\theta_i - I) \right| < x \sigma N^{-1/2} \right\} = \Phi(x),$$

където $\sigma = \sqrt{Var[\theta]}$, а $\Phi(x) = \frac{2}{\sqrt{2\pi}} \int_0^x e^{-\frac{t^2}{2}} dt$ се нарича вероятностен интеграл и се определя от функцията на разпределение на стандартна нормална случайна величина в интервала $(0, 1)$ [92].

Величината

$$r_N = x_\beta \sigma N^{-\frac{1}{2}} \quad (5)$$

се нарича вероятностна грешка, като параметърът β определя нивото на доверие. При $\beta = 0.5$ се получава т.нар. вероятна грешка, която характеризира типичната скорост на сходимость на МК методите от порядъка $\mathcal{O}(N^{-1/2})$.

Дисперсията на случайната величина θ в практическите приложения обикновено се оценява емпирично чрез формулата:

$$Var[\theta] \approx \frac{1}{N} \sum_{i=1}^N (\theta_i)^2 - (\bar{\theta}_N)^2. \quad (6)$$

При неголяма размерност на извадката N от стойности на случайната величина, обикновено се използва

по-точната формула [92]:

$$\text{Var}[\theta] \approx \frac{1}{(N-1)} \sum_{i=1}^N (\theta_i)^2 - \frac{N}{N-1} (\bar{\theta}_N)^2. \quad (7)$$

Разграничават се два типа МК методи, които се използват за намаляване на грешката [22, 89, 92]:

- Ефективни МК методи, при които се цели да се минимизира дисперсията чрез техники за редуциране (отделяне на главна част, симетризация на подинтегралната функция, метод на съществената извадка, метод на разделяне по важност и други), без да се променя асимптотичният порядък на грешката.
- Суперсходящи МК методи, при които чрез симулиране на стойностите на случайна величина в равномерни непокриващи се подобласти и при допълнителни условия за гладкост на подинтегралната функция се постига по-висок порядък на сходимост $r_N = \mathcal{O}(N^{-1/2-\varepsilon(k)})$, където $\varepsilon(k) > 0$ зависи от размерността на задачата.

За симулиране на стойностите на случайните величини се използват генератори на псевдослучайни числа.

Генератори на псевдослучайни числа

Практическата реализация на МК методите изисква използването на генератори на псевдослучайни числа (PRNGs), които генерират равномерно разпределени стойности в $(0, 1)$ [53, 81]. Основните изисквания към добрите PRNGs включват статистическа случайност, дълъг период на повтаряемост, възпроизводимост, ефективност и възможност за паралелно генериране [50, 53].

Сред най-често използваните генератори са линейните и комбиниранияте конгруентни генератори [32, 54, 71], генераторите с изместени регистри (например, Mersenne Twister [62]), както и генератори на базата на други математически конструкции (например, Middle Square PRNG [70]). В дисертационните изследвания за реализацията на стохастичните алгоритми за оценяване на екстремални собствени стойности са използвани Mersenne Twister (MT) и Middle Square (MS) генератори на псевдослучайни числа.

Квази-Монте Карло методи

Квази-Монте Карло (КМК) методите са детерминистична алтернатива на МК методите и използват редици с малък дискрепанс вместо псевдослучайни точки [39, 66, 74]. Те осигуряват по-равномерно покритие на единичния хиперкуб $[0, 1]^k$ и водят до по-добри скорости на сходимост от порядък $\mathcal{O}(N^{-1}(\log N)^k)$ при достатъчна гладкост на подинтегралната функция [9, 52]. При оценяване на многомерни интегрални приближеното решение има вида:

$$\int_{[0,1]^k} f(x) dx \approx \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i), \quad (8)$$

където x_i са елементи на редица с малък дискрепанс.

Редици с малък дискрепанс и оценка на грешката

Ключовата характеристика на тези редици е дискрепанс-звездата $D_N^*(S)$, която измерва отклонението от равномерно разпределение:

$$D_N^*(S) = \sup_{\mathbf{z} \in [0,1]^k} \left| \frac{1}{N} \sum_{i=1}^N \mathbf{1}_Z(\mathbf{x}_i) - \prod_{j=1}^k z_j \right|. \quad (9)$$

Редицата е с малък дискрепанс, ако

$$D_N^*(S) = \mathcal{O}(N^{-1}(\log N)^k) \text{ или } D_N^*(S) \leq C_k \frac{(\log N)^k}{N}, \quad (10)$$

където константата C_k в (10) зависи само от размерността k .

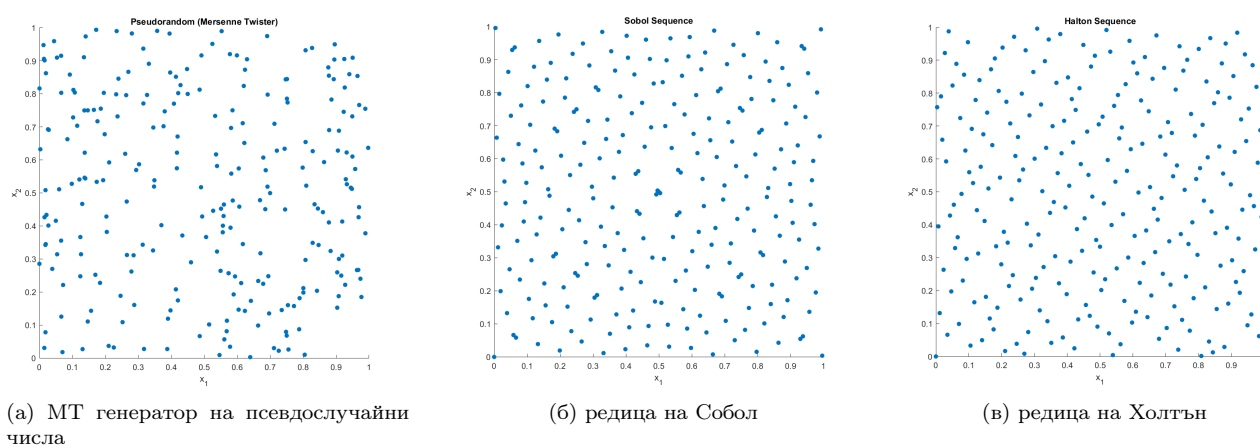
Грешката при КМК методите се оценява чрез неравенството на Коксма-Хлавка [9, 72]:

$$\left| I(f) - \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \right| \leq V(f) D_N^*(S), \quad (11)$$

където $D_N^*(S)$ е дискрепанс-звезда на редицата от точки $S = \{\mathbf{x}_i\}_{i=1}^\infty \in [0, 1]^k$. $V(f)$ е вариацията на функцията в смисъл на Харди-Краус. Това неравенство показва, че грешката зависи както от гладкостта на подинтегралната функция, така и от равномерността на използваната точкова редица. При достатъчно гладки функции се достига асимптотичен порядък на грешката $\mathcal{O}(N^{-1}(\log N)^k)$, който е теоретично по-добър от този на класическите МК методи.

Първите конструкции на редици с малък дискрепанс произлизат от редицата на Ван дер Корпут [85]. Холтън разширява тази идея към многомерния случай чрез използване на радикална инверсия в различни прости основи по координати [39]. Редиците на Холтън са концептуално прости, но при по-високи размерности могат да проявят корелации, което налага използването на модификации [41, 86]. Редиците на Соболев, въведени през 1976 г. [76], представляват цифрови (t, k) -редици при основа 2. Те се конструират чрез направляващи числа, получени от примитивни полиноми над крайното поле $\mathbb{F}_2 = \{0, 1\}$ (събиране по модул 2), което осигурява висока равномерност и добри проекционни свойства [74, 89]. Поради тези свойства редиците на Соболев са сред най-широко използваните в КМК приложения.

Фигура (1) представя сравнение между двумерна псевдослучайна редица, генерирана с Mersenne Twister (MT) и двумерни проекции на 16-мерни редици на Соболев и Холтън. Наблюдава се ясно по-равномерно покритие на единичния квадрат при редиците с малък дискрепанс в сравнение с псевдослучайната редица.

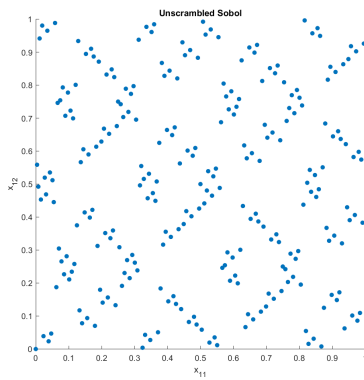


Фигура 1: Сравнение на двумерна псевдослучайна редица, получена с MT PRNG, с двумерни проекции на 16-мерни редици на Соболев и Холтън. Генерираните точки са 256, а проекцията е направена по първите две координати.

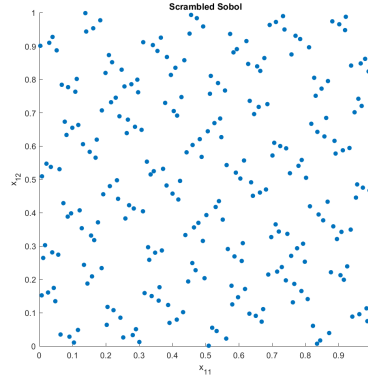
Рандомизирани редици с малък дискрепанс

Детерминистичният характер на класическите редици с малък дискрепанс води до липса на естествена вероятностна рамка за оценка на грешката и до риск от корелации при високи размерности. Тези недостатъци се премахват чрез въвеждане на рандомизирани редици с малък дискрепанс, при които чрез контролирана стохастичност се запазва равномерното покритие на единичния хиперкуб, като едновременно с това се преодоляват ограниченията на фиксираната геометрия [13]. Допълнително към разбърканите редици с малък дискрепанс могат да се прилагат техники за пропускане на начални членове на редицата, чрез използване на оператор за прескачане (англ. *skipping operator*) и оператор за прескачане през стъпка (англ. *leaping operator*) на редицата. Така се запазват свойствата на редиците с малък дискрепанс, като едновременно се въвежда контролирана стохастичност, позволяваща статистическа оценка на грешката. Фигура (2) и Фигура (3) показват наличието на корелации при неразбъркана редица на Соболев и Холтън в по-високи координати и ефекта от прилагането на метода на разбъркване (англ. *scrambling*). Разбъркването води до значително подобряване на равномерността на точковото разпределение, както е показано на Фигура (2б) и Фигура (3б). На Фигура (2в) и Фигура (3в) е показан и допълнителният ефект от комбиниране на метода на разбъркване чрез параметрите *skip* и *leap*. Тези техники редуцират корелациите и подобряват практическите свойства на редиците при задачи с по-висока размерност.

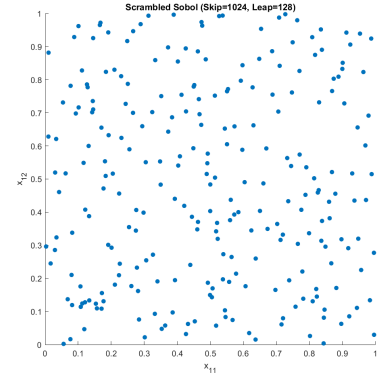
Съвременните модификации на редиците на Холтън и Соболев включва рандомизиране, целящо намаляване на корелациите и подобряване на сходимостта [52, 66] за конкретни задачи. Рандомизацията се реализира чрез прилагане на специално конструирани трансформации (разбърквания) върху вече изградени редици с малък дискрепанс. Съществуват два основни подхода за разбъркване (англ. *scrambling*). Първият се основава на *побитово (цифрово) случайно пермутиране*, при което цифрите на координатите в основа b



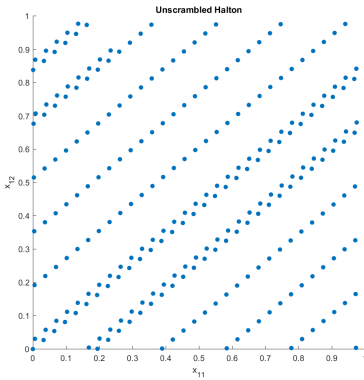
(а) неразбъркана редица на Собо



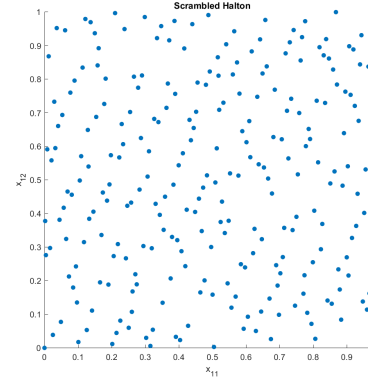
(б) разбъркана редица на Собо

(в) разбъркана редица на Собо с параметри *skip*, *leap*

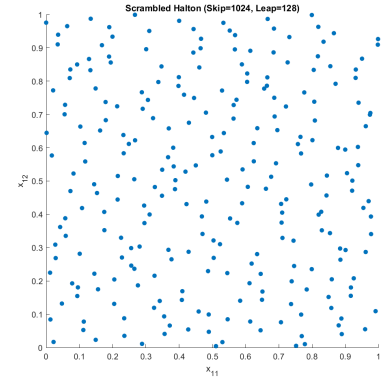
Фигура 2: Сравнение между двумерна проекция с наличие на корелация при 11-та и 12-та координата на редицата на Собо и съответните двумерни проекции на рандомизирани редици на Собо без и с допълнително включени параметри *skip* = 1024 и *leap* = 128.



(а) неразбъркана редица на Холтън



(б) разбъркана редица на Холтън

(в) разбъркана редица на Холтън с параметри *skip*, *leap*

Фигура 3: Сравнение между двумерна проекция с наличие на корелация при 11-та и 12-та координата на редицата на Холтън и съответните двумерни проекции на рандомизирани редици на Холтън без и с допълнително включени параметри *skip* = 1024 и *leap* = 128.

(в частност при основа 2 при двоично представяне) се подлагат на случайни пермутации [67]. Вторият подход се основава на *събиране по модул 1* със случайно число или случаен вектор, генерирани чрез псевдослучаен генератор. Този метод е известен още като *случайно изместване* (англ. *shift*) по модул 1 и се характеризира с проста и ефективна реализация [45, 46]. На практика двата подхода често се комбинират, което води до множество техни варианти и комбинации [68, 69].

Стохастични методи с вериги на Марков

Монте Карло (МК) и Квази-Монте Карло (КМК) методите условно се класифицират в две основни групи: директни и итерационни.

Директните МК/КМК методи се характеризират с това, че полученото приближено решение е повлияно единствено от стохастичната (статистическа) грешка. При класическия МК подход тази грешка е от порядък $\mathcal{O}(N^{-1/2})$ и се оценява чрез неравенство (5), докато при КМК методите тя се описва чрез неравенството на Коксма-Хлавка (11) и има асимптотичен порядък $\mathcal{O}(N^{-1}(\log N)^k)$ при използване на редици с малък дискрепанс, като редиците на Собо и Холтън. В дисертационния труд се прилагат рандомизирани редици с малък дискрепанс, които съчетават по-равномерно покритие на единичния куб с контролирана стохастичност. Поради това грешките, оценявани чрез (5) и (11), се интерпретират като стохастична грешка, зависеща от размера на извадката N .

Итерационните МК/КМК методи, освен стохастична, съдържат и систематична грешка, тъй като оце-

няват не директно точното решение, а крайна итерация на сходящ итерационен процес. Аналогично на класическите числени методи в линейната алгебра, като метода на простата итерация и метода на Гаус–Зайдел [90], систематичната грешка зависи от свойствата на итерационната схема и от нормата на матрицата, участваща в процеса.

В стохастичните итерационни методи за задачи от линейната алгебра се конструира случайна величина $\theta^{(k)}$ чрез верига на Марков с k прехода [16, 22, 92]. Броят на преходите може да бъде фиксиран или случаен с математическо очакване, равно на k . Математическото очакване на случайната величина $\theta^{(k)}$ апроксимира търсеното решение чрез функционал, включващ матрично-векторни произведения. В този контекст систематичната грешка се определя от броя на итерациите k , докато стохастичната грешка зависи от размера на извадката N . Следователно, балансирането между систематичната и стохастичната грешка, както и редукцията на дисперсията, представляват основни изследователски задачи при разработването на ефективни стохастични итерационни методи и алгоритми за линейна алгебра.

Структура на дисертацията

Дисертационният труд се състои от увод, три глави, заключение, приложение и библиография. Дисертацията съдържа 118 страници, включва 15 фигури и 16 таблици, библиография от 121 заглавия, както и списък с математически означения и използвани абривиатури.

- В **Увода** е представена актуалността на тематиката, направен е обзор на основните научни резултати в областта, поставени са целите и задачите на дисертационния труд и подробно е описана методологията на изследването.
- **Глава 1** „Стохастичен Степенен метод: разработване и изследване на ефективни алгоритми за оценяване на максимална собствена стойност“ е посветена на разработването на ефективни стохастични степенни алгоритми за намиране на максималната собствена стойност на плътни симетрични матрици. Дефинирана е задачата и са разгледани методите, които се използват за нейното решаване, а именно, детерминистичния Степенен метод и стохастичния Степенен метод, в неговите два варианта - Степенен Монте Карло (СМК) и Степенен Квази-Монте Карло (СКМК) метод. Подробно е описана констукцията на случайната величина с използване на верига на Марков, като се акцентира на използването на почти оптимални вероятности. Представен е псевдокод на разработените почти оптимални СМК и СКМК алгоритми и са определени условията за балансиране на стохастичната и систематичната грешка. Главата завършва с дискусия върху направените числени експерименти за матрици с различна размерност.
- **Глава 2** „Стохастичен Резолвентен метод: разработване и изследване на ефективни алгоритми за оценяване на минимална собствена стойност“ разглежда задачата за оценяване минималната собствена стойност на симетрични квадратни матрици, чрез използване на нейната резолвентна матрица. Представен е Резолвентният Степенен метод и неговите стохастични реализации чрез Резолвентен Монте Карло (РМК) и Резолвентен Квази-Монте Карло (РКМК) подход. В главата се изследва ролята на параметрите, влияещи върху сходимостта на безкрайния ред, чрез който се представя резолвентната матрица. Конструирани са почти оптимални РМК и РКМК алгоритми и се изследва балансът на стохастичната и систематична грешка. Този баланс зависи от броя реализации на случайните величини, дължината на веригата на Марков, степента на резолвентната матрица и ускоряващия параметър, включен в представянето на резолвентната матрица като степенен ред. В последната секция са представени проведените числени експерименти, получени при оценяването на минималната собствена стойност на тестови симетрични матрици.
- **Глава 3** „Стохастични методи за оценяване на пазарния риск на инвестиционни портфейли“ е свързана с прилагането на разработените алгоритми от Глава 1 към реална задача от областта на финансовата математика, свързана с оценяване на пазарния риск на инвестиционен портфейл. Оценяването му се извършва чрез максималната собствена стойност на корелационната матрица на портфейла. Намирането на най-голямата собствена стойност се извършва чрез почти оптималните СМК и СКМК алгоритми. При прилагането на почти оптималния СКМК алгоритъм са разгледани два варианта: (1) с разбъркани редици на Собол и Холтън с настройка по „подразбиране“; (2) с разбъркани редици на Собол и Холтън с използване на параметри `skip` и `leap`. Проведени са числени експерименти с реални финансови данни, които демонстрират приложимостта и ефективността на предложените алгоритми.

В заключителната част са включени: списък на публикациите, апробация, научните и научно-приложните приноси и благодарности.

1. Стохастичен Степенен метод: разработване и изследване на ефективни алгоритми за оценяване на максимална собствена стойност

1.1 Постановка на задачата

Нека $A = \{a_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$ е дадена несингулярна симетрична матрица. Собствените стойности (спектърът) на матрицата A се дефинират като реалните стойности на параметъра $\lambda(A)$, за които уравнението:

$$A\mathbf{x} = \lambda(A)\mathbf{x}, \quad \text{където } \mathbf{x} \in \mathbb{R}^n, \quad (1.1)$$

има ненулево решение, като съответните ненулеви вектори $\mathbf{x}$ се наричат собствени вектори на A . Необходимо и достатъчно условие за съществуването на ненулево решение е параметърът $\lambda(A)$ да бъде корен на характеристичното уравнение $|A - \lambda(A)\mathbf{I}| = 0$, където $\mathbf{I}$ е единичната n -мерна матрица [90].

Предполага се, че собствените стойности на A са подредени по абсолютна стойност и удовлетворяват неравенствата:

$$\lambda_{\min} = |\lambda_n| \leq |\lambda_{n-1}| \leq |\lambda_{n-2}| \leq \dots \leq |\lambda_2| < |\lambda_1| = \lambda_{\max}. \quad (1.2)$$

Разглеждаме задачата за разработване и изследване на ефективни алгоритми на базата на Стохастичния Степенен метод за оценяване на $\lambda_{\max}$, която съгласно (1.2) е единствена.

1.2 Методи за изчисляване на максималната собствена стойност

1.2.1 Степенен метод

Степенният метод представлява класически итеративен числен алгоритъм за апроксимация на максималната собствена стойност $\lambda_{\max}$ на квадратни матрици [33, 43], базиран на коефициента на Рейли [43]:

$$r(\mathbf{x}) = \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}},$$

Алгоритъмът се реализира чрез инициализация с нормализиран начален вектор, итеративно обновяване на вектора и оценяване на собствената стойност след k итерации чрез:

$$\lambda_{\max} \approx \lambda_{\max}^{(k)} = r(\mathbf{x}_k),$$

като при условие (1.2) полученото приближение съответства на доминиращата собствена стойност.

Систематична грешка: В случая на симетрична матрица грешката от прекъсване на Степенния метод при представяне на векторите $\mathbf{x}^T$ и $\mathbf{x}$ като линейни комбинации на ортонормиран базис е от порядъка:

$$|\lambda_{\max}^{(k)} - \lambda_{\max}| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right), \quad (1.3)$$

което показва силна зависимост от спектралното разстояние между първите две собствени стойности [43].

Сходимост: Скоростта на сходимост на итерационния процес при симетрични матрици се определя от коефициента $\left|\frac{\lambda_2}{\lambda_1}\right|^2$. При близки по модул доминиращи собствени стойности, сходимостта се забавя и може да изисква прилагане на техники за предварителна обработка.

Изчислителна сложност: Изчислителната сложност на Степенния метод е $\mathcal{O}(kn^2)$, където k е броят на итерациите, а n - размерността на матрицата A , което го прави подходящ за големи, но структурирани симетрични задачи.

1.2.2 Стохастичен Степенен метод

Стохастичният Степенен метод, изследван в дисертационния труд, е разработен от Караиванова и Димов [21, 22]. Той представлява стохастична реализация на *класическия* Степенен метод за задача за собствени стойности (1.1) при условие (1.2). Основата на метода е представянето на най-голямата собствена стойност

като граница на коефициента на Рейли:

$$\lambda_{max} = \lim_{k \rightarrow \infty} \frac{(\mathbf{h}, A^k \mathbf{f})}{(\mathbf{h}, A^{k-1} \mathbf{f})}, \quad (1.4)$$

където $\mathbf{f}$ и $\mathbf{h}$ са произволно избрани и нормирани вектори. Ако k е произволно голямо естествено число, тогава

$$\lambda_{max}^{(k)} = \frac{(\mathbf{h}, A^k \mathbf{f})}{(\mathbf{h}, A^{k-1} \mathbf{f})} \quad (1.5)$$

е приближение на λ_{max} с определена систематична грешка.

Стохастичният Степенен метод се реализира чрез конструиране на случайни величини, чиито математически очаквания съвпадат със скаларните произведения в числителя и знаменателя на горното отношение. За оценяване на $(\mathbf{h}, A^k \mathbf{f})$ се дефинира случайна величина $\theta^{(k)}$, базирана на дискретна крайна верига на Марков със състояния:

$$l_0 \rightarrow \dots \rightarrow l_t \rightarrow \dots \rightarrow l_k \quad (1 \leq l_t \leq n, \quad 0 \leq t \leq k), \quad (1.6)$$

с начален вектор на плътността $\mathbf{p} = \{p_i\}_{i=1}^n \in \mathbb{R}^n$ и матрица на преходната плътност $P = \{p_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$. Конструкцията включва итеративно изчисляване на теглото W_k :

$$W_0 = 1, \quad W_t = W_{t-1} \frac{a_{l_{t-1}l_t}}{p_{l_{t-1}l_t}}, \quad t = 1, \dots, k, \quad (1.7)$$

а случайната величина $\theta^{(k)}$ се определя чрез:

$$\theta^{(k)} = \frac{h_{l_0}}{p_{l_0}} W_k f_{l_k}. \quad (1.8)$$

Случайната величина $\theta^{(k)}$, дефинираната чрез (1.8), има математическо очакване равно на скаларното произведение $(\mathbf{h}, A^k \mathbf{f})$, т.е. $E[\theta^{(k)}] = (\mathbf{h}, A^k \mathbf{f})$ [17].

Дефиниция 1.2.1. Началният вектор на плътността $\mathbf{p} = \{p_i\}_{i=1}^n \in \mathbb{R}^n$ се нарича допустим за вектор $\mathbf{h} = \{h_i\}_{i=1}^n \in \mathbb{R}^n$, т.е. $\mathbf{p} \in \mathfrak{P}_{\mathbf{h}}$, ако $p_i > 0$, когато $h_i \neq 0$ и $p_i = 0$, когато $h_i = 0$ за $i = 1, \dots, n$.

Дефиниция 1.2.2. Матрицата на преходната плътност $P = \{p_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$ се нарича допустима за симетричната матрица $A = \{a_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$, т.е. $P \in \mathfrak{P}_A$, ако $p_{ij} > 0$, когато $a_{ij} \neq 0$ и $p_{ij} = 0$, когато $a_{ij} = 0$ за $i, j = 1, \dots, n$.

Въведените понятия за допустим начален вектор на плътността $\mathbf{p} \in \mathfrak{P}_{\mathbf{h}}$ и допустима матрица на преходната плътност $P \in \mathfrak{P}_A$ гарантират коректното конструиране на веригата на Марков. За извадка от N реализации на случайната величина $\theta^{(k)}$, средната стойност се представя чрез:

$$\bar{\theta}_N^{(k)} = \frac{1}{N} \sum_{s=1}^N (\theta^{(k)})_s, \quad (1.9)$$

където s -тата реализация на случайната величина $\theta^{(k)}$ е означена с $(\theta^{(k)})_s$ за $1 \leq s \leq N$. Тя представлява стохастична апроксимация на $(\mathbf{h}, A^k \mathbf{f})$. При използване на генератори на псевдослучайни числа вероятностната грешка за Монте Карло метода е:

$$R_N^{(k)} = |(\mathbf{h}, A^k \mathbf{f}) - \bar{\theta}_N^{(k)}| < c_1 \sigma(\theta^{(k)}) N^{-1/2}, \quad (1.10)$$

където c_1 е константа, а $\sigma(\theta^{(k)})$ е стандартно отклонение [17]. При използването на редици с малък дискрепанс (Собол, Холтън), $\bar{\theta}_N^{(k)}$ е Квази-Монте Карло апроксимация с грешка:

$$QR_N^{(k)} = |(\mathbf{h}, A^k \mathbf{f}) - \bar{\theta}_N^{(k)}| < c_2 \frac{(\log N)^{k+1}}{N}, \quad (1.11)$$

получена чрез неравенството на Коксма-Хлавка, където $(k+1)$ е размерността на използваните редици.

Теорема 1.2.1. ¹ Нека се разгледат случайните величини $\theta^{(k)}$ и $\theta^{(k-1)}$ при фиксирани стъпки k и

¹Доказателството на Теорема 1.2.1 е разписано в дисертационния труд.

$k - 1$ на веригата на Марков (1.6). Тогава:

$$\frac{E[\theta^{(k)}]}{E[\theta^{(k-1)}]} = \lambda_{max}^{(k)}. \quad (1.12)$$

Забележка 1. Стохастичният Степенен метод се нарича Степенен Монте Карло (СМК) метод, когато извадките за двете случайни величини $\theta^{(k)}$ и $\theta^{(k-1)}$ се генерират чрез генератори на случайни числа за конструиране на веригата на Марков (1.6).

По този начин стохастичната грешка на СМК метода е функция на вероятностната грешка, дефинирана от (1.10).

Забележка 2. Когато за конструирането на веригата на Марков при генериране на извадките на $\theta^{(k)}$ и $\theta^{(k-1)}$ се използват разбъркани редици с малък дискрепанс, като например разбърканите редици на Соболев [75], Стохастичният Степенен метод се нарича Степенен Квази-Монте Карло (СКМК) метод.

Стохастичната грешка на СКМК метода е функция на грешката, дефинирана чрез (1.11).

Преформулирането на задачата в термините на интегралните уравнения с подходящи избрани ядра и области позволява използването на неравенството на Коксма-Хлавка при анализиране на сходимостта на разглежданите алгоритми. В общия случай, използването на редици с малък дискрепанс (т. нар. квази-редици) подобрява скоростта на сходимост на съответните КМК алгоритми, но не дава възможност за апостериорна оценка на грешката. Разбъркването (рандомизирането) на квазиредицата позволява избягването на този недостатък.

Теоретичното сравнение на грешката при СМК и СКМК метод показва, че и в двата случая грешката е произведение от два множителя. Единият, зависещ от редицата, а другият от функцията или случайната величина. При това, неравенството на Коксма-Хлавка е граница в най-лошия случай, докато грешката при СМК подход има вероятностен характер. Изчислителният опит показва, че вариацията $V(f)$ в неравенството на Коксма-Хлавка обикновено е свръхоценка, докато вторият множител дискрепансът на редицата показва действителния порядък на грешката.

1.3 Почти оптимални СМК и СКМК методи и алгоритми

1.3.1 Прецизиране на конструкцията на $\theta^{(k)}$ и $\theta^{(k-1)}$ с почти оптимални вероятности за веригата на Марков: балансиране на грешките

Нека симетричната матрица A се представя чрез редовете си $\mathbf{a}_i$, като се въвеждат стандартни L_1 -норми за вектори и матрици:

$$\|\mathbf{h}\| = \|\mathbf{h}\|_1 = \sum_{i=1}^n |h_i|, \quad \|\mathbf{a}_i\| = \|\mathbf{a}_i\|_1 = \sum_{j=1}^n |a_{ij}|, \quad \|A\| = \|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|, \quad j = 1, \dots, n.$$

Вероятностната грешка на МК апроксимациите, съгласно (1.10), зависи от стандартното отклонение $\sigma(\theta^{(k)})$ и броя реализации N . При фиксирано N тя се определя основно от дисперсията на случайната величина $\theta^{(k)}$. Дисперсията зависи от избора на допустимия начален вектор на плътност $\mathbf{p} \in \mathfrak{P}_h$ и матрицата на преходната плътност $P \in \mathfrak{P}_A$, като всички такива избори могат да се използват за конструиране на оценки за $\lambda_{max}^{(k)}$, при условие че дисперсиите са крайни.

Най-често използваният подход за конструиране на верига на Марков за задачи на линейната алгебра се основава на равномерното разпределение. В този случай координатите на вектора на плътност $\mathbf{p}$ и елементите на матрицата на плътност на прехода P се задават равни на $\frac{1}{n}$, т.е.:

$$\mathbf{p} = \{p_i\}_{i=1}^n = \frac{1}{n} \text{ и } P = \{p_{ij}\}_{i,j=1}^n = \frac{1}{n}, \quad i, j = 1, \dots, n. \quad (1.13)$$

Когато изборът (1.13) се използва при конструирането на веригата на Марков, Степенният МК и Степенният КМК методи се наричат *класически*. Алтернативни избори на $\mathbf{p}$ и P позволяват конструиране на по-ефективни алгоритми, водещи до редуциране на дисперсията, намаляване на изчислителната сложност и повишаване на точността спрямо *класическия* случай. Съществува *почти оптимален* избор, представящ се чрез формулите:

$$\mathbf{p} = \{p_i\}_{i=1}^n, \quad p_i = \frac{|h_i|}{\|\mathbf{h}\|} \text{ и } P = \{p_{ij}\}_{i,j=1}^n, \quad p_{ij} = \frac{|a_{ij}|}{\|\mathbf{a}_i\|}, \quad i, j = 1, \dots, n. \quad (1.14)$$

който води до значително редуциране на дисперсията, като при този избор конструкцията на случайната величина $\theta^{(k)}$ се различава съществено от *класическия* метод.

Теорема 1.3.1.² Нека $\mathbf{p}$ и P са почти оптималният начален вектор на плътност и почти оптималната матрица на плътност (виж (1.14)). Тогава средната стойност (1.9) може да се представи като:

$$\bar{\theta}_N^{(k)} = \frac{1}{N} \sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \|\mathbf{h}\| \left\{ \prod_{t=1}^{t=k} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \|\mathbf{a}_{l_{t-1}^{(s)}}\| \right\} f_{l_k^{(s)}}. \quad (1.15)$$

От Теорема 1.2.1 и 1.3.1 следват следните следствия:

Следствие 1. Нека $\mathbf{p}$ и P са почти оптималният начален вектор на плътност и почти оптималната матрица на плътност. Тогава Стохастичният Степенен метод за оценяване на $\lambda_{max}^{(k)}$ се дефинира като съотношение на средните стойности на случайните величини $\bar{\theta}_N^{(k)}$ и $\bar{\theta}_N^{(k-1)}$ по формула (1.15), т.е.

$$\lambda_{max}^{(k)} \approx \frac{\sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \left\{ \prod_{t=1}^{t=k} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \|\mathbf{a}_{l_{t-1}^{(s)}}\| \right\} f_{l_k^{(s)}}}{\sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \left\{ \prod_{t=1}^{t=k-1} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \|\mathbf{a}_{l_{t-1}^{(s)}}\| \right\} f_{l_{k-1}^{(s)}}}. \quad (1.16)$$

Следствие 2. Да предположим, че L_1 -нормата на всички вектори-редове на симетричната матрица A е равна на константата a , т.е.

$$\|\mathbf{a}_i\|_1 = \|\mathbf{a}_i\| = \sum_{j=1}^n |a_{ij}| = a, \quad i = 1, \dots, n. \quad (1.17)$$

Тогава оценката за $\lambda_{max}^{(k)}$ по формула (1.16) може да се пренапише по по-прост начин:

$$\lambda_{max}^{(k)} \approx \frac{\sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \left\{ \prod_{t=1}^{t=k} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \right\} a f_{l_k^{(s)}}}{\sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \left\{ \prod_{t=1}^{t=k-1} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \right\} f_{l_{k-1}^{(s)}}}. \quad (1.18)$$

Следствие 3. Нека L_1 -нормата на всички вектори-редове на симетричната матрица A удовлетворява условието (1.17). Предполага се, че следното условие е валидно за всички елементи на A :

$$\text{sign}(h_{l_0^{(s)}}) \prod_{t=1}^{t=k-1} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) = 1, \quad s = 1, \dots, N. \quad (1.19)$$

Тогава $\lambda_{max}^{(k)}$ може да се оцени по по-проста формула, както следва:

$$\lambda_{max}^{(k)} \approx \frac{\sum_{s=1}^N \text{sign}(a_{l_{k-1}^{(s)} l_k^{(s)}}) a f_{l_k^{(s)}}}{\sum_{s=1}^N f_{l_{k-1}^{(s)}}}. \quad (1.20)$$

Чрез Следствие (2) и Следствие (3) се конструират опростени оценки за приближено пресмятане на $\lambda_{max}^{(k)}$ за определени класове симетрични матрици, като съгласно Забележка 3 тези класове не са празно множество.

Забележка 3. Множеството от симетрични матрици, които удовлетворяват условията на Следствие 3, е непразно. Например, нека се разгледа симетрична матрица A с положителни елементи, за която всички елементи по главния диагонал са равни, т.е. $a_{ii} = b > 0$, $i = 1, \dots, n$, и всички извън диагонални елементи също са равни, т.е. $a_{ij} = d > 0$, $i \neq j$, $i, j = 1, \dots, n$.

Тъй като систематичната грешка ε_1 зависи от броя k на преходите във веригата на Марков, а стохастичната грешка ε_2 зависи от броя на реализациите N , се получава оценка за съответните грешки:

$$\left| \lambda_{max} - \frac{\bar{\theta}_N^{(k)}}{\bar{\theta}_N^{(k-1)}} \right| = \left| \lambda_{max} - \lambda_{max}^{(k)} + \lambda_{max}^{(k)} - \frac{\bar{\theta}_N^{(k)}}{\bar{\theta}_N^{(k-1)}} \right| \leq \left| \lambda_{max} - \lambda_{max}^{(k)} \right| + \left| \lambda_{max}^{(k)} - \frac{\bar{\theta}_N^{(k)}}{\bar{\theta}_N^{(k-1)}} \right| < \varepsilon_1 + \varepsilon_2, \quad (1.21)$$

²Доказателството на Теорема 1.3.1 е разписано в дисертационния труд.

като оптимален баланс между двата вида грешки (систематичната и стохастичната грешка) се постига при условие $\epsilon_1 = \epsilon_2 = \epsilon$.

Дефиниция 1.3.1. *СМК и СКМК методи за оценка на максималната собствена стойност на симетрична матрица при избор на начален вектор на плътност $\mathbf{p}$ и матрица на плътност на прехода P съгласно формула (1.14) се наричат почти оптимални СМК и СКМК методи.*

1.3.2 Почти оптимален СМК и почти оптимален СКМК алгоритми: изчислителна сложност

Псевдокодът в **Алгоритъм 1** описва процедура за оценяване на максималната собствена стойност на симетрична плътна матрица въз основа на формула (1.16) и Следствие 1. Алгоритъмът се определя като *почти оптимален* СКМК алгоритъм при използване на редици с малък дискрепанс (Собол или Холтърн) в стъпка 4, а при прилагане на генератори на псевдослучайни числа - като *почти оптимален* СМК алгоритъм, означаван с **Алгоритъм 1а**. При изпълнение на условията от Следствие 2 и Следствие 3 формулите в стъпка 6 могат да бъдат заменени с по-опростените изрази (1.18) или (1.20).

Algorithm 1 Pseudocode for computing the *Almost Optimal* PQMC algorithm

- 1: INPUT: matrix $A = \{a_{ij}\} \in \mathbb{R}^{n \times n}$; vectors $\mathbf{h}, \mathbf{f} \in \mathbb{R}^n$ and positive integers n, N, k
- 2: COMPUTE: $\|\mathbf{a}_i\| = \sum_{j=1}^n |a_{ij}|$, $1 \leq i \leq n$ and $\|\mathbf{h}\| = \sum_{i=1}^n |h_i|$
- 3: COMPUTE: The coordinates of a permissible density vector $\mathbf{p}$ and the elements of a permissible density matrix P

$$p_i = \frac{|h_i|}{\|\mathbf{h}\|} \quad \text{and} \quad p_{ij} = \frac{|a_{ij}|}{\|\mathbf{a}_i\|}, \quad i, j = 1, \dots, n.$$

- 4: GENERATE: N elements of the $(k+1)$ - dimensional Sobol (or Halton) sequence
- 5: CONSTRUCT: N realizations of the Markov chain with integer elements

$$l_0^{(s)} \rightarrow \dots \rightarrow l_t^{(s)} \rightarrow \dots \rightarrow l_{k-1}^{(s)} \rightarrow l_k^{(s)}, \quad 1 \leq l_t^{(s)} \leq n, \quad 0 \leq t \leq k, \quad 1 \leq s \leq N$$

- 6: COMPUTE:

$$\begin{aligned} \bar{\theta}_N^{(k)} &= \frac{1}{N} \sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \|\mathbf{h}\| \left\{ \prod_{t=1}^{t=k} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \|\mathbf{a}_{l_{t-1}^{(s)}}\| \right\} f_{l_k^{(s)}}; \\ \bar{\theta}_N^{(k-1)} &= \frac{1}{N} \sum_{s=1}^N \text{sign}(h_{l_0^{(s)}}) \|\mathbf{h}\| \left\{ \prod_{t=1}^{t=k-1} \text{sign}(a_{l_{t-1}^{(s)} l_t^{(s)}}) \|\mathbf{a}_{l_{t-1}^{(s)}}\| \right\} f_{l_{k-1}^{(s)}}; \end{aligned}$$

- 7: OUTPUT:

$$\lambda_{max}^{(k)} := \frac{\bar{\theta}_N^{(k)}}{\bar{\theta}_N^{(k-1)}}$$

Изчислителната сложност на **Алгоритъм 1** се определя от броя реализации N на веригата на Марков, средното време за изчисляване на една реализация на случайните величини $\theta^{(k)}$ и $\theta^{(k-1)}$, както и от предварителното конструиране на матрицата на преходната плътност P . Стъпки 2 и 3 изискват $\mathcal{O}(n^2)$ операции и се разглеждат като предварителна обработка, докато основната изчислителна тежест е съсредоточена в стъпки 4–6. За фиксирано k времето за една реализация е $\tau_k = \mathcal{O}(nk)$, поради което общата изчислителна сложност на алгоритъма е:

$$\text{Cost}(\text{Alg1}) = \mathcal{O}(Nnk) + \mathcal{O}(n^2) = \mathcal{O}(Nnk + n^2),$$

която при $N > n$ се свежда до $\mathcal{O}(Nnk)$. Числените експерименти показват, че за плътни симетрични матрици балансът между систематичната и стохастичната грешка се постига при $N = \mathcal{O}(n)$, като при разредени матрици може да се очаква допълнително намаляване на изчислителните разходи.

1.4 Числени резултати и дискусия

В числените експерименти **Алгоритъм 1а** е реализиран с два генератора на псевдослучайни числа: Mersenne Twister (MT) и Middle Square (MS), като съответните реализации са означени като $PMC_{(MT)}$ и $PMC_{(MS)}$. Получените резултати са сравнени с тези на *класическия* СМК алгоритъм, реализиран чрез

стандартния избор на началния вектор и матрицата на преходната плътност по формула (1.14), означени като $PMC_{(MT)}^{(n)}$ и $PMC_{(MS)}^{(n)}$. **Алгоритъм 1** е реализиран с редици с малък дискрепанс - Соболев и Холтърн, означени като $PQMC_{(S)}^{(n)}$ и $PQMC_{(H)}^{(n)}$, и е сравнен с *класическия* СКМК подход и неговите варианти $PQMC_{(S)}^{(n)}$ и $PQMC_{(H)}^{(n)}$. За намаляване на корелациите при големи размерности, редиците са подложени на разбъркване: "RR2" (*reverse-radix*) за Холтърн и "MatousekAffineOwen" за Соболев. Всички генератори и редици са внедрени в MATLAB 9.10 (R2021a).

Числените експерименти са проведени върху две симетрични плътни матрици с размерности $n = 100$ и $n = 500$, със случайно генерирани елементи и със собствени стойности съответно $(\lambda_1, \lambda_2) = (50.0408, 4.0522)$ и $(\lambda_1, \lambda_2) = (250.2454, 9.0721)$. Векторите $\mathbf{h}$ и $\mathbf{f}$ са избрани с L_1 -норма. За да се осигури сходимостта, матрицата A е мащабирана с параметъра:

$$q = \frac{1}{|\lambda_1| + \delta_1}, \quad \delta_1 > 0,$$

или алтернативно:

$$q = \frac{1}{\max_i \|\mathbf{a}_i\| + \delta}, \quad \delta > 0, \quad i = 1, \dots, n,$$

където $\|\mathbf{a}_i\|$ е L_1 -нормата на i -тия ред на A . Подборът на добре разделени собствени стойности гарантира бърза сходимост и позволява експериментално изследване на баланса между систематичната и стохастичната грешка. Анализирано е влиянието на броя преходи k и броя реализации N върху точността, както и ефектът от редуцираната дисперсия при *почти оптималните* СКМК/СКМК алгоритми при фиксирана изчислителна сложност.

1.4.1 Случай (i): Размер на матрицата $n = 100$

В Таблица 1.1 са представени резултатите за оценяване на най-голямата собствена стойност на симетрична матрица ($n = 100$, $\lambda_1 = 50.0408$) с *почти оптималния* $PMC_{(MT)}^{(n)}$ и *класическия* $PMC_{(MT)}^{(n)}$ алгоритъм. Оптималният баланс между систематичната и стохастичната грешка се достига при $k = 8$ или $k = 9$, като минималната абсолютна грешка за $PMC_{(MT)}^{(n)}$ е 5×10^{-5} при $k = 8$, $N = 512$, докато за *класическия* $PMC_{(MT)}^{(n)}$ се наблюдава грешка 1.5×10^{-3} при $k = 8$, $N = 2048$, което показва значително превъзходство на *почти оптималния* вариант.

Таблица 1.1: Резултати от $PMC_{(MT)}^{(n)}$ и $PMC_{(MT)}^{(n)}$ при различни N и k за матрица A с размерност $n = 100$, за която $\lambda_2/\lambda_1 \approx 0.081$, където $\lambda_1 = 50.0408$ и $\lambda_2 = 4.0522$.

N	k	$\lambda_1^{(k)}$ using $PMC_{(MT)}^{(n)}$	Absolute Error $PMC_{(MT)}^{(n)}$	$\lambda_1^{(k)}$ using $PMC_{(MT)}^{(n)}$	Absolute Error $PMC_{(MT)}^{(n)}$
128	8	50.039901	0.00090	50.019883	0.0209
	9	50.040709	0.00009	50.037795	0.0030
	10	50.045295	0.00450	50.066104	0.0253
256	7	50.023923	0.01688	50.012158	0.0286
	8	50.040628	0.00017	50.039528	0.0013
	9	50.044896	0.00410	50.050593	0.0098
512	7	50.035045	0.00576	49.996040	0.0448
	8	50.040750	0.00005	50.039264	0.0015
	9	50.052027	0.01123	50.050158	0.0094
1024	7	50.029828	0.01097	50.019435	0.0214
	8	50.040720	0.00008	50.039735	0.0011
	9	50.045403	0.00460	50.054351	0.0136
2048	7	50.036291	0.00451	50.029563	0.0112
	8	50.041086	0.00029	50.041728	0.0009
	9	50.043850	0.00305	50.057179	0.0164

*Маркираните редове представляват оптималните k стойности за всяко N .

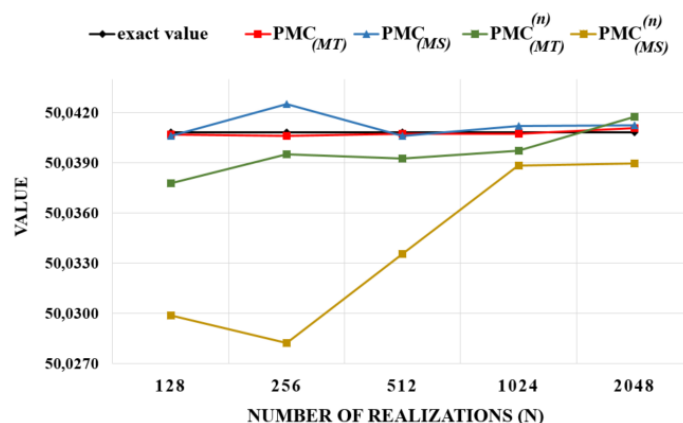
Таблица 1.2 показва аналогичните резултати за $PMC_{(MS)}^{(n)}$ и $PMC_{(MS)}^{(n)}$. Най-малката абсолютна грешка за *почти оптималния* $PMC_{(MS)}^{(n)}$ алгоритъм също се наблюдава при $k = 8$, $N = 512$ и е 1.8×10^{-4} . При *класическия* $PMC_{(MS)}^{(n)}$ алгоритъм тя е 1.8×10^{-3} при $k = 8$, $N = 2048$. Това потвърждава тенденцията, че *почти оптималните* алгоритми редуцират дисперсията и водят до по-точни оценки в сравнение с

класическите, независимо от използвания генератор на псевдослучайни числа.

Таблица 1.2: Резултати от $PMC_{(MS)}$ и $PMC_{(MS)}^{(n)}$ при различни N и k за матрицата A с размерност $n = 100$, за която $\lambda_2/\lambda_1 \approx 0.081$, където $\lambda_1 = 50.0408$ и $\lambda_2 = 4.0522$.

N	k	$\lambda_1^{(k)}$ using $PMC_{(MS)}$	Absolute Error $PMC_{(MS)}$	$\lambda_1^{(k)}$ using $PMC_{(MS)}^{(n)}$	Absolute Error $PMC_{(MS)}^{(n)}$
128	8	50.039440	0.00136	50.019388	0.0214
	9	50.040595	0.00020	50.029883	0.0109
	10	50.051262	0.01046	50.060923	0.0201
256	7	50.031044	0.00976	49.984883	0.0559
	8	50.042490	0.00169	50.028211	0.0126
	9	50.050490	0.00969	50.051251	0.0105
512	7	50.030518	0.01028	49.997989	0.0428
	8	50.040617	0.00018	50.033570	0.0072
	9	50.058913	0.01811	50.053570	0.0128
	10	50.061048	0.02025	50.091035	0.0502
1024	7	50.031472	0.00933	50.009835	0.0310
	8	50.041209	0.00041	50.038850	0.0020
	9	50.044938	0.00414	50.056396	0.0156
2048	7	50.038998	0.00180	50.022658	0.0181
	8	50.041243	0.00044	50.038951	0.0018
	9	50.043045	0.00225	50.063537	0.0227

*Маркираните редове представляват оптималните k стойности за всяко N .

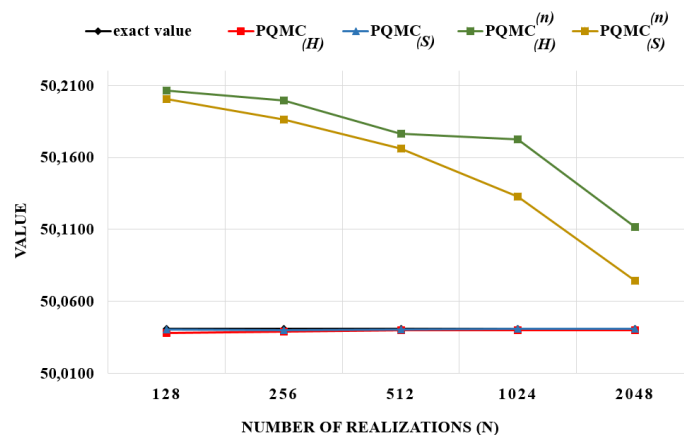


Фигура 1.1: Графично сравняване на приближените резултати за $\lambda_{max}^{(k)}$ с $\lambda_{max} = 50.0408$ при балансиран случай за грешките, т.е. когато $(k = 9, N = 128)$ и $(k = 8, N > 128)$. Приложени алгоритми: $PMC_{(MT)}$, $PMC_{(MS)}$, $PMC_{(MT)}^{(n)}$ и $PMC_{(MS)}^{(n)}$.

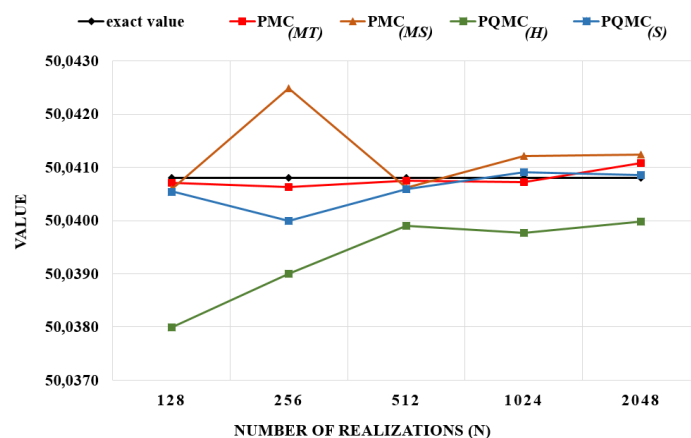
На Фигура 1.1 е показано, че *почти оптималните* версии ($PMC_{(MT)}$ и $PMC_{(MS)}$) следват равномерни траектории близо до точната стойност λ_{max} дори при по-малки N , докато *класическите* варианти, особено $PMC_{(MS)}^{(n)}$, демонстрират по-големи колебания и по-слаба стабилност. Основната причина е намалената дисперсия при конструирането на случайните величини $\theta^{(k)}$ и $\theta^{(k-1)}$ чрез оптималната матрица на преходната плътност, което потвърждава по-високата точност и изчислителна ефективност на *почти оптималните* СКМК алгоритми.

В Таблица 1.3 са представени резултатите за *почти оптималните* и *класическите* СКМК алгоритми, базирани на редиците на Холтън, при оценяване на $\lambda_1 = 50.0408$ за матрица с размер $n = 100$. Наблюдава се монотонно намаляване на абсолютната грешка с увеличаване на броя преходи k , като оптималната сходимост се постига при $k = 11$ – 12 . При $N = 2048$ алгоритъмът $PQMC_{(H)}$ достига абсолютна грешка 8.2×10^{-4} , докато *класическият* $PQMC_{(H)}^{(n)}$ остава значително по-неточен с грешка 0.0707, което ясно демонстрира предимството на *почти оптималната* конструкция.

Таблица 1.4 съдържа аналогичните резултати за алгоритмите, базирани на редиците на Соболев. *Почти оптималният* алгоритъм $PQMC_{(S)}$ систематично превъзхожда както своя *класически* аналог, така и варианта с редици на Холтън. Например, при $N = 2048$ и оптимален брой преходи, $PQMC_{(S)}$ постига



Фигура 1.2: Графично сравняване на приближените резултати за $\lambda_{max}^{(k)}$ с $\lambda_{max} = 50.0408$ при балансирания случай за грешките, т.е. когато $(k = 12, N = 128)$ и $(k = 11, N > 128)$. Приложени алгоритми: $PQMC_{(S)}$, $PQMC_{(H)}$, $PQMC_{(S)}^{(n)}$ и $PQMC_{(H)}^{(n)}$.



Фигура 1.3: Графично сравнение на λ_{max} на симетричната матрица A ($n = 100$) с $\lambda_{max}^{(k)}$ в балансиращите случаи. Използвани са *почти оптималните* СМК и СКМК алгоритми.

Таблица 1.3: Резултати от $PQMC_{(H)}$ и $PQMC_{(H)}^{(n)}$ при различни N и k за матрица A с размерност $n = 100$, за която $\lambda_2/\lambda_1 \approx 0.081$, където $\lambda_1 = 50.0408$ и $\lambda_2 = 4.0522$.

N	k	$\lambda_1^{(k)}$ using $PQMC_{(H)}$	Absolute Error $PQMC_{(H)}$	$\lambda_1^{(k)}$ using $PQMC_{(H)}^{(n)}$	Absolute Error $PQMC_{(H)}^{(n)}$
128	11	49.801798	0.23900	49.574856	0.4659
	12	50.037989	0.00281	50.206208	0.1654
	13	50.152986	0.11219	50.314121	0.2733
256	10	50.001834	0.03897	49.739820	0.3010
	11	50.038996	0.00180	50.199525	0.1587
	12	50.067986	0.02719	50.229280	0.1885
512	10	50.017989	0.02281	49.749688	0.2911
	11	50.039899	0.00090	50.176724	0.1359
	12	50.178512	0.13771	50.196566	0.1558
1024	10	49.976987	0.06381	49.806135	0.2347
	11	50.039772	0.00103	50.172345	0.1315
	12	50.107985	0.06719	50.295866	0.2551
2048	10	49.973344	0.06746	49.907160	0.1336
	11	50.039982	0.00082	50.111519	0.0707
	12	50.183245	0.14245	50.275169	0.2344

*Маркираните редове представляват оптималните k стойности за всяко N .

абсолютна грешка 5×10^{-5} , докато $PQMC_{(S)}^{(n)}$ достига 0.0336. Дори при по-малки стойности на N , като $N = 256$ и $k = 11$, разликата е съществена - 8.1×10^{-4} срещу 0.1455 в полза на *почти оптималния* алгоритъм.

Таблица 1.4: Резултати от $PQMC_{(S)}$ и $PQMC_{(S)}^{(n)}$ при различни N и k за матрица A с размерност $n = 100$, за която $\lambda_2/\lambda_1 \approx 0.081$, където $\lambda_1 = 50.0408$ и $\lambda_2 = 4.0522$.

N	k	$\lambda_1^{(k)}$ using $PQMC_{(S)}$	Absolute Error $PQMC_{(S)}$	$\lambda_1^{(k)}$ using $PQMC_{(S)}^{(n)}$	Absolute Error $PQMC_{(S)}^{(n)}$
128	11	49.679383	0.36142	49.624160	0.4166
	12	50.040544	0.00026	50.200481	0.1597
	13	50.091116	0.05032	50.299826	0.2590
256	10	49.994318	0.04648	49.742961	0.2978
	11	50.039989	0.00081	50.186264	0.1455
	12	50.049407	0.00861	50.225854	0.1851
512	10	50.050000	0.00920	49.866152	0.1746
	11	50.040585	0.00021	50.166400	0.1256
	12	50.148019	0.10722	50.223877	0.1831
1024	10	49.986946	0.05385	49.933245	0.1076
	11	50.040913	0.00011	50.132603	0.0918
	12	50.087435	0.04664	50.167252	0.1265
2048	10	49.972883	0.06792	49.957605	0.0832
	11	50.040854	0.00005	50.074364	0.0336
	12	50.155992	0.11519	50.186986	0.1462

*Маркираните редове представляват оптималните k стойности за всяко N .

Визуализацията на Фигура 1.2 потвърждава числените резултати от Таблица 1.3 и Таблица 1.4, като показва по-гладка и стабилна сходимост на алгоритмите $PQMC_{(S)}$ и $PQMC_{(H)}$ към точната стойност λ_{max} за всички стойности на N . За разлика от тях, *класическите* варианти $PQMC_{(S)}^{(n)}$ и $PQMC_{(H)}^{(n)}$ демонстрират по-големи колебания и начално надценяване на собствената стойност, като сходимостта им е значително по-бавна.

На Фигура 1.3 са показани сравненията между *почти оптималните* СМК и СКМК алгоритми за балансираните случаи. При СМК методите $PMC_{(MT)}$ достига минимална грешка 5×10^{-5} при $N = 512$ и $k = 8$, докато $PMC_{(MS)}$ показва по-голяма, но все още малка грешка 1.8×10^{-4} . От страна на СКМК алгоритмите, $PQMC_{(S)}$ е най-точният, с грешка 5×10^{-5} при $N = 2048$ и $k = 11$, докато $PQMC_{(H)}$ достига 8.2×10^{-4} . Тези резултати ясно показват, че използването на разбъркани редици на Соболев и *почти оптималната* преходна матрица води до по-малка дисперсия и оптимален баланс между стохастичната и систематичната грешка спрямо *класическите* им варианти.

1.4.2 Случай (ii): Размер на матрицата $n = 500$

В Таблица 1.5 са представени резултатите за оценяване на максималната собствена стойност λ_{max} на симетрична плътна матрица с размерност $n = 500$, получени с *почти оптималния* алгоритъм $PMC_{(MT)}$ и неговия *класически* аналог $PMC_{(MT)}^{(n)}$. Числените данни показват, че при фиксиран брой реализации N увеличаването на броя преходи k води до намаляване на абсолютната грешка, като оптималният баланс между систематичната и стохастичната грешка се постига при $k = 9$. В този балансиран случай *почти оптималният* алгоритъм достига абсолютна грешка от порядъка на 10^{-4} (например, при $N = 512$), докато при *класическия* вариант грешката остава от порядъка на 10^{-2} (при същите параметри), което ясно демонстрира значително по-висока точност. Аналогични изводи следват от Таблица 1.6, където са представени резултатите за *почти оптималния* алгоритъм $PMC_{(MS)}$ и *класическия* $PMC_{(MS)}^{(n)}$. В този случай оптималната сходимост се наблюдава при $k = 9$, като минималната абсолютна грешка за *почти оптималния* алгоритъм е постигната при $N = 128$ и е от порядъка на 10^{-4} . За разлика от него, *класическият* вариант демонстрира значително по-големи грешки от порядъка на 10^{-2} , което потвърждава предимството на оптимизираната конструкция дори при използване на по-слаб псевдослучаен генератор. Визуализацията на Фигура 1.4 потвърждава резултатите от Таблица 1.5 и Таблица 1.6. *Почти оптималните* СМК алгоритми показват по-гладка и стабилна сходимост към λ_{max} при нарастващ брой реализации N . Това се дължи на значително намалената дисперсия при конструирането на случайните величини $\theta^{(k)}$ и $\theta^{(k-1)}$. *Класическият* алгоритъм $PMC_{(MS)}^{(n)}$ демонстрира колебания и дори надценяване на собствената

Таблица 1.5: Резултати от $PMC_{(MT)}$ и $PMC_{(MT)}^{(n)}$ при различни N и k за матрица A с размерност $n = 500$, за която $\lambda_2/\lambda_1 \approx 0.0363$, където $\lambda_1 = 250.2454$ и $\lambda_2 = 9.0721$.

N	k	$\lambda_1^{(k)}$ using $PMC_{(MT)}$	Absolute Error $PMC_{(MT)}$	$\lambda_1^{(k)}$ using $PMC_{(MT)}^{(n)}$	Absolute Error $PMC_{(MT)}^{(n)}$
128	8	250.222396	0.02300	249.492834	0.7526
	9	250.232288	0.01311	250.032970	0.2124
	10	250.295027	0.04963	250.149293	0.0961
256	8	250.228998	0.01640	250.124994	0.1204
	9	250.250130	0.00473	250.222275	0.0231
	10	250.283657	0.03826	250.311181	0.0658
512	8	250.235302	0.01011	250.115109	0.1303
	9	250.245110	0.00029	250.192305	0.0531
	10	250.268302	0.02290	250.328365	0.0830
1024	8	250.235306	0.01009	250.164200	0.0812
	9	250.245015	0.00038	250.205542	0.0399
	10	250.269913	0.02451	250.300328	0.0549
2048	8	250.226172	0.01923	250.175939	0.0695
	9	250.246938	0.00154	250.224397	0.0210
	10	250.251779	0.00638	250.295497	0.0501

*Маркираните редове представляват оптималните k стойности за всяко N .

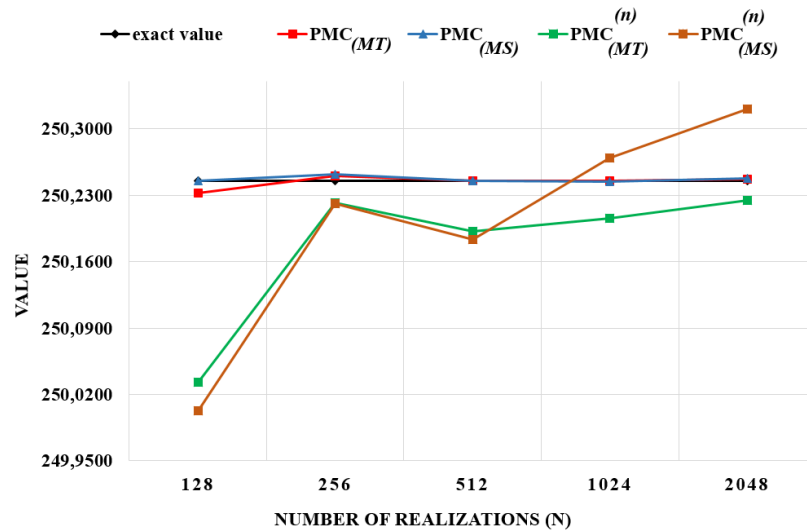
стойност при по-големи стойности на N , докато $PMC_{(MT)}^{(n)}$ се характеризира с по-стабилна сходимост. Това показва, че МТ генераторът на псевдослучайни числа е за предпочитане при задачи с високи размерности.

Таблица 1.6: Резултати от $PMC_{(MS)}$ и $PMC_{(MS)}^{(n)}$ при различни N и k за матрица A с размерност $n = 500$, за която $\lambda_2/\lambda_1 \approx 0.0363$, където $\lambda_1 = 250.2454$ и $\lambda_2 = 9.0721$.

N	k	$\lambda_1^{(k)}$ using $PMC_{(MS)}$	Absolute Error $PMC_{(MS)}$	$\lambda_1^{(k)}$ using $PMC_{(MS)}^{(n)}$	Absolute Error $PMC_{(MS)}^{(n)}$
128	8	250.211445	0.03395	249.462722	0.7827
	9	250.245247	0.00015	250.003068	0.2423
	10	250.279476	0.03408	250.120230	0.1252
256	8	250.203720	0.04168	250.015811	0.2296
	9	250.252044	0.00664	250.221130	0.0243
	10	250.290513	0.04511	250.363882	0.1185
512	8	250.206884	0.03852	250.058060	0.1873
	9	250.244890	0.00051	250.183410	0.0620
	10	250.283279	0.03788	250.356760	0.1114
1024	8	250.238977	0.00642	250.143770	0.1016
	9	250.244751	0.00065	250.268955	0.0236
	10	250.270855	0.02546	250.396782	0.1514
2048	8	250.237612	0.00779	250.103221	0.1422
	9	250.247710	0.00231	250.320873	0.0755
	10	250.275206	0.02981	250.347819	0.1024

*Маркираните редове представляват оптималните k стойности за всяко N .

В Таблица 1.7 са представени резултатите за оценяване на λ_{max} чрез *почти оптималните* и *класически-*те СКМК алгоритми, базирани на разбъркани редица на Холтън. При всички алгоритми абсолютната грешка намалява с увеличаване на броя преходи k , като оптималният баланс се постига при $k = 12$. В този случай *почти оптималният* алгоритъм $PQMC_{(H)}$ достига абсолютна грешка от порядъка на 10^{-3} (например, 4.59×10^{-3} при $N = 2048$), докато *класическият* вариант $PQMC_{(H)}^{(n)}$ демонстрира значително по-големи грешки, достигащи 0.0726 при същия брой реализации. Това ясно показва предимството на оптимизираната конструкция на алгоритъма. В Таблица 1.8 са представени аналогични резултати за алгоритмите, използващи редици на Соболев. *Почти оптималният* алгоритъм $PQMC_{(S)}$ демонстрира най-висока точност, като в балансиран случай при $k = 12$ абсолютната грешка е от порядъка на 10^{-4} (например, 5.4×10^{-4} при $N = 2048$). За разлика от него, *класическият* вариант $PQMC_{(S)}^{(n)}$ достига значително по-големи отклонения, като при $N = 256$ грешката е 0.3172. Това потвърждава по-добрата равномерност на редиците на Соболев и тяхното предимство при по-високи размерности.



Фигура 1.4: Графично сравняване на λ_{max} за матрицата A ($n = 500$) с приближените резултати за $\lambda_{max}^{(k)}$, получени при балансирания случай на грешките, т.е. когато ($k = 9, N \geq 128$). Приложени алгоритми: $PMC_{(MT)}$, $PMC_{(MS)}$, $PMC_{(MT)}^{(n)}$ и $PMC_{(MS)}^{(n)}$.

Таблица 1.7: Резултати от $PQMC_{(H)}$ и $PQMC_{(H)}^{(n)}$ при различни N и k за матрица A с размерност $n = 500$. Абсолютната грешка е за случая $\lambda_2/\lambda_1 \approx 0.0363$.

N	k	$\lambda_1^{(k)}$ using $PQMC_{(H)}$	Absolute Error $PQMC_{(H)}$	$\lambda_1^{(k)}$ using $PQMC_{(H)}^{(n)}$	Absolute Error $PQMC_{(H)}^{(n)}$
128	11	250.167707	0.07769	251.486594	1.2412
	12	250.249391	0.00399	249.309133	0.9363
	13	250.357823	0.11242	251.999063	1.7537
256	11	250.197841	0.04756	249.470953	0.7744
	12	250.239655	0.00574	250.770441	0.5250
	13	250.357573	0.11217	251.711421	1.4660
512	11	250.089218	0.15618	249.591525	0.6539
	12	250.236033	0.00937	250.669970	0.4246
	13	250.369509	0.12411	250.978674	0.7333
1024	11	250.127756	0.11764	249.672942	0.5725
	12	250.250718	0.00532	250.090116	0.1553
	13	250.387049	0.14165	250.829267	0.5839
2048	11	250.166549	0.07885	249.987224	0.2582
	12	250.249986	0.00459	250.172833	0.0726
	13	250.321274	0.07587	250.581459	0.3361

*Маркираните редове представляват оптималните k стойности за всяко N .

Фигура 1.5 илюстрира сходимостта на *почти оптималните* и *класическите* СКМК алгоритми при оптималната стойност $k = 12$. *Почти оптималните* алгоритми $PQMC_{(S)}$ и $PQMC_{(H)}$ следват плътно точната стойност на λ_{max} за всички стойности на N , демонстрирайки по-бърза и по-гладка сходимост. За разлика от тях, *класическите* варианти се характеризират с по-бавна сходимост и по-големи колебания, особено при по-малки стойности на N .

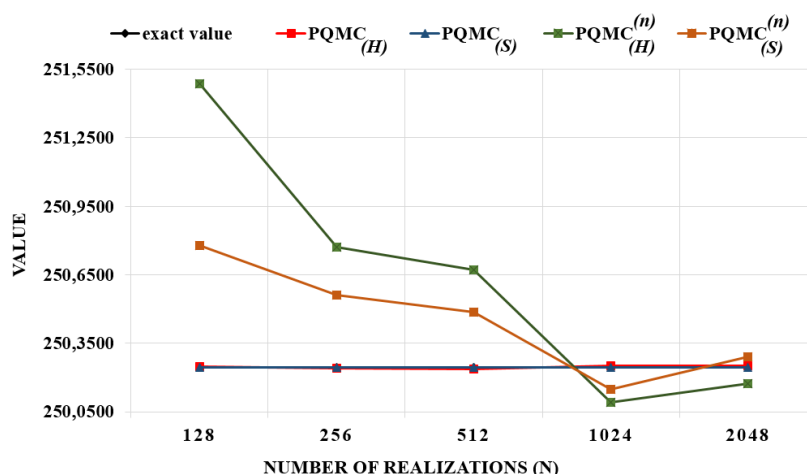
На Фигура 1.6 са показани сравненията между *почти оптималните* СКМК и СКМК алгоритми. *Почти оптималните* СКМК алгоритми $PMC_{(MT)}$ и $PMC_{(MS)}$ достигат минимални грешки съответно 3.8×10^{-4} и 6.5×10^{-4} при $N = 1024$ и $k = 9$, докато *почти оптималният* СКМК алгоритъм $PQMC_{(S)}$ осигурява съпоставима или по-висока точност при $k = 12$.

В Таблица 1.9 са сравнени емпиричните оценки за дисперсиите на случайната величина $\theta^{(k)}$ при *класическите* и *почти оптималните* СКМК алгоритми. Оценките са извършени за матрици с размерности $n = 100$ и $n = 500$. Резултатите показват, че при *почти оптималните* СКМК алгоритми дисперсията е с около два порядъка по-малка в сравнение с *класическите* им варианти. Тази оценка не зависи съществено от използвания генератор или от броя реализации N , което е в съзвучие с теоретичните резултати. Фактът, че дисперсията при *почти оптималните* версии е с два порядъка по-малка от *класическите*

Таблица 1.8: Резултати от $PQMC_{(S)}$ и $PQMC_{(S)}^{(n)}$ при различни N и k за матрица A с размерност $n = 500$. Абсолютната грешка е за случая $\lambda_2/\lambda_1 \approx 0.0363$.

N	k	$\lambda_1^{(k)}$ using $PQMC_{(S)}$	Absolute Error $PQMC_{(S)}$	$\lambda_1^{(k)}$ using $PQMC_{(S)}^{(n)}$	Absolute Error $PQMC_{(S)}^{(n)}$
128	11	250.236415	0.00898	249.415659	0.8297
	12	250.245037	0.00036	250.777310	0.5319
	13	250.330000	0.08460	250.989461	0.7441
256	11	250.199189	0.04621	249.483392	0.7620
	12	250.245119	0.00028	250.562622	0.3172
	13	250.291367	0.04597	250.704555	0.4592
512	11	250.099484	0.14592	249.632500	0.6129
	12	250.244872	0.00053	250.486175	0.2408
	13	250.378047	0.13265	250.679597	0.4342
1024	11	250.140000	0.10540	249.811533	0.4339
	12	250.244996	0.00040	250.148126	0.0973
	13	250.351139	0.10574	250.651883	0.4065
2048	11	250.193656	0.05174	249.937040	0.3084
	12	250.245937	0.00054	250.290864	0.0455
	13	250.272562	0.02716	250.555662	0.3103

*Маркираните редове представляват оптималните k стойности за всяко N .

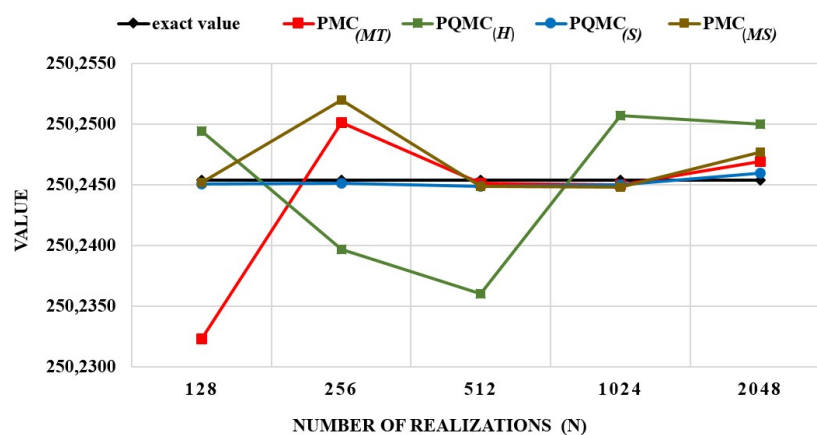


Фигура 1.5: Графично сравняване на λ_{max} за матрицата A ($n = 500$) с приближените резултати за $\lambda_{max}^{(k)}$, получени при балансиран случай на грешките, т.е. когато ($k = 12, N \geq 128$). Приложени алгоритми: $PQMC_{(S)}$, $PQMC_{(H)}$, $PQMC_{(S)}^{(n)}$ и $PQMC_{(H)}^{(n)}$.

Таблица 1.9: Дисперсия на оценките на максималната собствена стойност при почти оптималния и класическия СМК алгоритъм за различни размерности на матрицата, брой реализации (N), брой преходи (k) и използваните генератори на псевдослучайни числа (MT и MS)

matrix size	N	k	$Var(PMC_{(MT)})$	$Var(PMC_{(MT)}^{(n)})$	$Var(PMC_{(MS)})$	$Var(PMC_{(MS)}^{(n)})$
100	512	8	1.169203e-06	1.720495e-04	1.236100e-06	1.781005e-04
	2048	8	1.300635e-06	1.772653e-04	1.326996e-06	1.792424e-04
500	512	9	1.186151e-08	1.229778e-05	1.269925e-08	1.285468e-05
	2048	9	1.189763e-08	1.184918e-05	1.268721e-08	1.299595e-05

версии, потвърждава отново направените изводи, че тези алгоритми осигуряват по-точно оценяване на максималната собствена стойност на симетрични матрици при намалена изчислителна сложност.



Фигура 1.6: Графично сравнение на λ_{max} на симетричната матрица A ($n = 500$) с $\lambda_{max}^{(k)}$ в балансиращите случаи. Използвани са *почти оптималните* СМК и СКМК алгоритми.

1.5 Обобщение

В тази глава е изследвана ефективността на *почти оптималните* СМК и СКМК алгоритми за оценка на най-голямата собствена стойност на симетрични плътни матрици за $n = 100$ и $n = 500$. Числените експерименти показват, че оптимизираният алгоритми значително превъзхождат *класическите* по точност и изчислителна сложност чрез целенасочен избор на преходната матрица и балансиране на систематичната и стохастичната грешка. При СМК алгоритмите оптималният брой преходи е $k = 8$ за $n = 100$ и $k = 9$ за $n = 500$, като генераторът Mersenne Twister (MT) осигурява по-малка дисперсия от Middle Square (MS). При СКМК оптималният брой преходи е $k = 11$ за $n = 100$ и $k = 12$ за $n = 500$, като разбърканите редици на Соболев водят до по-гладка сходимост и по-малки грешки от тези на Холтън. Абсолютните грешки при *почти оптималните* алгоритми са от порядъка на 10^{-4} , докато при *класическите* достигат до 10^{-2} , потвърждавайки значителното намаляване на дисперсията и повишаване на точността.

В заключение, *почти оптималните* СМК и СКМК алгоритми осигуряват по-висока точност при същата изчислителна сложност, което ги прави предпочитан избор за практически приложения. При тяхното конструиране трябва да се съобразяваме и с избора на генераторите на псевдослучайни числа и от избора на редиците с малък дискрепанс. Описаните резултати са публикувани в [36, 37].

2. Стохастичен Резолвентен метод: разработване и изследване на ефективни алгоритми за оценяване на минимална собствена стойност

2.1 Постановка на задачата

Нека $A = \{a_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$ е несингулярна симетрична матрица. Собствените стойности на A се определят като всички реални стойности на параметъра $\lambda(A)$, за които уравнението:

$$A\mathbf{x} = \lambda(A)\mathbf{x}, \quad \text{където } \mathbf{x} \in \mathbb{R}^n, \quad (2.1)$$

има ненулево решение. Приема се, че собствените стойности удовлетворяват наредбата:

$$0 < \lambda_{min} = \lambda_n < \lambda_{n-1} \leq \lambda_{n-2} \leq \dots \leq \lambda_2 < \lambda_1 = \lambda_{max}. \quad (2.2)$$

като минималната собствена стойност λ_{min} е положителна и съгласно (2.2) е единствена.

Разглеждаме задачата за разработване и численото изследване на ефективни стохастични алгоритми за нейното оценяване.

2.2 Методи за изчисляване на минималната собствена стойност

Нека $R = \{r_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$ е несингулярна симетрична матрица, чиито собствени стойности удовлетворяват наредбата:

$$\mu_{min} = |\mu_n| < |\mu_{n-1}| \leq |\mu_{n-2}| \leq \dots \leq |\mu_2| < |\mu_1| = \mu_{max}, \quad (2.3)$$

и за тях уравнението:

$$R\mathbf{y} = \mu(R)\mathbf{y} \quad (2.4)$$

има ненулево решение. От теорията на линейната алгебра е известно, че при определени зависимости между матриците R и A съществуват връзки между техните собствени стойности и вектори. Разглеждат се три основни случая:

- При обратна матрица $R = A^{-1}$ собствените стойности са свързани чрез $\mu = \frac{1}{\lambda}$. Това позволява използване на *Обратния Степенен* метод (англ. *Inverse Power method*) за намиране на минималната собствена стойност на A , но изисква скъпо изчисляване на обратна матрица.
- При отместена матрица $R_q = A - q\mathbf{I}$ е валидно $\mu = \lambda - q$. Това води до *Отместен Степенен* метод (англ. *Shifted Power method*), при който изборът на параметъра q влияе върху оценката на минималната собствена стойност.
- При резолвентна матрица $R_q = (\mathbf{I} - qA)^{-1}$ собствените стойности са свързани чрез $\mu = \frac{1}{1 - q\lambda}$, а съответният метод се нарича *Резолвентен Степенен* метод (англ. *Resolvent Power method*). Матрицата R_q може да се представи като граница от безкраен сходящ ред при условие $|q\lambda| < 1$ или еквивалентно $|q|||A|| < 1$, а именно:

$$R_q = \sum_{i=0}^{\infty} q^i A^i = [\mathbf{I} - qA]^{-1}. \quad (2.5)$$

В зависимост от знака на параметъра q се получават различни връзки за максималната собствена стойност:

$$\mu_{max} = \frac{1}{1 - q\lambda_{min}}, \quad \text{при } q < 0, \quad (2.6)$$

$$\mu_{max} = \frac{1}{1 - q\lambda_{max}}, \quad \text{при } q > 0. \quad (2.7)$$

В рамките на настоящото изследване основен интерес представлява случаят $q < 0$, който позволява ускорено приближение и ефективно оценяване на екстремните собствени стойности.

2.2.1 Резолвентен Степенен метод

В *класическия* Степенен метод най-голямата собствена стойност λ_{max} на симетричната матрица A се представя като граница на коефициента на Рейли (1.4), което позволява практическо приближение чрез крайни степени. На тази основа се въвежда Резолвентният Степенен метод, при който матрицата A се заменя с резолвентната матрица $R_q = (\mathbf{I} - qA)^{-1}$, която съществува при условие $|q|||A|| < 1$. Резолвентната матрица има развитие в Нойманов ред, а нейната степен R_q^m може да се представи чрез обобщен биномен ред:

$$R_q^m = (\mathbf{I} - qA)^{-m} = \sum_{i=0}^{\infty} q^i C_{i+m-1}^i A^i, \quad C_{i+m-1}^i = \frac{(m+i-1)!}{(m-1)!i!}, \quad i = 0, 1, 2, \dots \quad (2.8)$$

Безкрайният ред за R_q^m (q се нарича параметър за ускорение) може да се запише като:

$$R_q^m = (I - qA)^{-m} = \mathbf{I} + mqA + \frac{m(m+1)}{2} q^2 A^2 + \frac{m(m+1)(m+2)}{6} q^3 A^3 + \dots \quad (2.9)$$

Прилагането на Степенния метод към резолвентната матрица (R_q) води до Резолвентния Степенен метод:

$$\lim_{m \rightarrow \infty} \frac{(\mathbf{h}, R_q^m \mathbf{f})}{(\mathbf{h}, R_q^{m-1} \mathbf{f})} = \lim_{m \rightarrow \infty} \frac{(\mathbf{h}, (\mathbf{I} - qA)^{-m} \mathbf{f})}{(\mathbf{h}, (\mathbf{I} - qA)^{-(m-1)} \mathbf{f})} = \mu_{max} = \frac{1}{1 - q\lambda}, \quad (2.10)$$

който позволява намирането на максималната собствена стойност μ_{max} на резолвентната матрица и съответно оценка на екстремалните собствени стойности на A . В зависимост от знака на параметъра q , методът може да се използва за приближено пресмятане както на λ_{max} , така и на λ_{min} , съгласно формулите [19, 20, 22]:

$$\lambda = \frac{(\mathbf{h}, AR_q^m \mathbf{f})}{(\mathbf{h}, R_q^m \mathbf{f})} \approx \lambda_{max}, \quad \text{ако } q > 0, \quad (2.11)$$

$$\frac{(\mathbf{h}, AR_q^m \mathbf{f})}{(\mathbf{h}, R_q^m \mathbf{f})} \approx \frac{1}{q} \left(1 - \frac{1}{\mu^{(m)}}\right) \approx \lambda_{min}, \quad \text{ако } q < 0. \quad (2.12)$$

В практиката резолвентната матрица се апроксимира чрез прекъсване на реда до краен брой членове k :

$$R_{q,k}^m = \sum_{i=0}^k C_{i+m-1}^i q^i A^i, \quad (2.13)$$

което води до систематична грешка, равна на "опашката" на реда:

$$\Delta_k^{(m)}(q) = (I - qA)^{-m} - R_{q,k}^m = \sum_{i=k+1}^{\infty} C_{i+m-1}^i q^i A^i. \quad (2.14)$$

Получени са оценки за тази грешка и условия за избор на параметрите m и k , така че тя да бъде по-малка от зададено ε , т.е.:

$$|\Delta_k^{(m)}(q)| < \varepsilon. \quad (2.15)$$

В сила е следната оценка:

$$|\Delta_k^{(m)}(t)| \leq T_{m,k}(t) = \sum_{i=k+1}^{\infty} C_{i+m-1}^i t^i, \quad t = |q||\lambda_{max}| < 1. \quad (2.16)$$

Съществува линейна връзка между параметрите m и k при условие $t = |q||\lambda_{max}| < 1$. Следните оценки са получени за връзката между k и m :

- Неравенство за точност:

$$T_{m,k}(t) \leq \frac{b_{k+1}}{1 - \alpha} = \frac{\binom{m+k}{k+1} t^{k+1}}{1 - \frac{m+k+1}{k+2} t} < \varepsilon, \quad b_i = C_{i+m-1}^i t^i, \quad \alpha := \frac{m+k+1}{k+2}, \quad t < 1. \quad (2.17)$$

- Линейни зависимости между параметрите:

$$m < \frac{1-t}{t}(k+1) + \frac{1}{t} \quad \text{или} \quad m_{max} = \left\lfloor \frac{(1-t)k+2}{t} - 1 \right\rfloor. \quad (2.18)$$

- Асимптотична връзка (ако m расте, а t е фиксирано):

$$k \gtrsim \frac{t}{1-t}m, \quad \text{за големи стойности на } m. \quad (2.19)$$

- Минимална стойност на k при фиксирано t се определя от формулата:

$$k_{min} = \left\lceil \frac{tm-1}{1-t} - 1 \right\rceil. \quad (2.20)$$

Получените зависимости позволяват ефективен избор на параметри при числена реализация и са основа за построяване на Стохастичен Резолвентен метод, когато се търси баланс между систематична и стохастична грешка.

2.2.2 Стохастичен Резолвентен метод

Стохастичният Резолвентен метод, въведен от Караиванова и Димов [20, 25], представлява обобщение на Резолвентния Степенен метод за оценяване на екстремалните собствени стойности на матрица A . Основен елемент е конструкцията на случайните величини $\theta^{(i)}$, дефинирани чрез верига на Марков, чиито математически очаквания съвпадат със скаларните произведения $(\mathbf{h}, A^i \mathbf{f})$, $i = 1, 2, 3, \dots$, т.е.:

$$E[\theta^{(i)}] = (\mathbf{h}, A^i \mathbf{f}), \quad i = 1, 2, 3, \dots \quad (2.21)$$

При условие за параметъра q , удовлетворяващ $|q \lambda_{max}| < 1$, е доказано в [17], че скаларното произведение с резолвентната матрица може да се представи като безкраен ред:

$$(\mathbf{h}, R_q^m \mathbf{f}) = (\mathbf{h}, (\mathbf{I} - qA)^{-m} \mathbf{f}) = \sum_{i=0}^{\infty} q^i C_{i+m-1}^i E[\theta^{(i)}], \quad (2.22)$$

което позволява стохастично оценяване чрез Монте Карло методи. На практика математическите очаквания се апроксимират чрез средната стойност:

$$\bar{\theta}_N^{(i)} = \frac{1}{N} \sum_{s=1}^N (\theta^{(i)})_s, \quad i = 1, 2, 3, \dots \quad (2.23)$$

като стохастичната грешка е от порядък $\mathcal{O}(N^{-1/2})$. При фиксирана точност ε е решена задачата за определяне на минималния брой реализации N , необходим за гарантиране на грешка под зададен праг и при условие за ограничена дисперсия и $\|A\| < 1$:

$$N \geq \left\{ \frac{c_k \sigma}{\varepsilon} \right\}^2. \quad (2.24)$$

Поради прекъсване на безкрайния ред до краен брой членове k възникват две грешки: (1) стохастична (Монте Карло апроксимация) и (2) систематична (от прекъсването на реда). Изборът на параметрите m, k контролира систематичната грешка, а N - стохастичната. Балансът между тях е решаващ за точността и изчислителната ефективност. Приближената формула за собствена стойност е:

$$\lambda \approx \frac{E[\theta^{(1)}] + qC_m^1 E[\theta^{(2)}] + q^2 C_{m+1}^2 E[\theta^{(3)}] + \dots + q^k C_{k+m-1}^k E[\theta^{(k+1)}]}{1 + qC_m^1 E[\theta^{(1)}] + q^2 C_{m+1}^2 E[\theta^{(2)}] + \dots + q^k C_{k+m-1}^k E[\theta^{(k)}]}, \quad (2.25)$$

като знакът на параметъра q определя дали се оценява максималната (λ_{max} при $q > 0$) или минималната собствена стойност (λ_{min} при $q < 0$). При използване на псевдослучайни числа се получава Резолвентен Монте Карло (РМК) метод, а при редици с малък дискрепанс - Резолвентен Квази-Монте Карло (РКМК). Конструкцията на случайните величини се базира на подходящ избор на начална плътност и матрица на преходните вероятности, като числените резултати показват по-висока точност спрямо *класическите* детерминистични алгоритми.

2.3 Почти оптимален Резолвентен Монте Карло и Резолвентен Квази-Монте Карло алгоритми

Нека матрицата A се представя чрез редовете $\mathbf{a}_i$, като се въвеждат L_1 -норми за вектори и матрици, които служат като основа при конструиране на *почти оптимални* стохастични резолвентни алгоритми. Изборът на начален вектор на плътността и матрица на преходните вероятности се реализира чрез нормализиране на абсолютните стойности на компонентите на $\mathbf{h}$ и редовете на матрицата A , което води до *почти оптимална* структура на веригата на Марков.

Представен е *почти оптимален* Резолвентен Квази-Монте Карло (РКМК) алгоритъм за оценяване на минималната собствена стойност на симетрична матрица. Алгоритъмът използва редици с малък дискрепанс (например, разбъркана редица на Соболю) за генериране на реализации във веригата на Марков и изчисляване на случайните величини $\theta^{(i)}$. При използване на псевдослучайни генератори методът се нарича *почти оптимален* Резолвентен Монте Карло (РМК) алгоритъм. В стъпка 4 на алгоритъма се прилага разбъркана редица на Соболю за изчисляване стойностите на случайните величини $\theta^{(i)}$, $i = 1, 2, \dots, k+1$, чрез верига на Марков.

Алгоритъм 2, представен чрез псевдокод, описва *почти оптималния* Резолвентен Квази-Монте Карло (РКМК) алгоритъм (англ. *almost optimal Resolvent Quasi-Monte Carlo (RQMC) algorithm*) за оценяване на минималната собствена стойност на симетричната матрица A . В случай, че в стъпка 4 се използват генератори на псевдослучайни числа (PRNGs), алгоритъмът се определя като *почти оптимален* Резолвентен Монте Карло (РМК) алгоритъм (англ. *almost optimal Resolvent Monte Carlo (RMC) algorithm*). Псевдокодът, представящ реализацията на *почти оптималния* РМК алгоритъм, ще означаваме с **Алгоритъм 2а**. Двата алгоритъма принадлежат към класа на итерационните стохастични методи и се характеризират с наличието на два вида грешки: стохастична и систематична. Стохастичната грешка при МК методите е от порядък $\mathcal{O}(N^{-1/2})$, докато при КМК методите (базирани на редици с малък дискрепанс) - $\mathcal{O}(N^{-1}(\log N)^k)$. При използване на редици с малък дискрепанс в пространства с висока размерност може да доведе до корелации между координатите на квазислучайните редици, което се преодолява чрез рандомизирането (разбъркването) им.

Algorithm 2 Pseudocode for computing the almost optimal Resolvent QMC algorithm

- 1: INPUT: matrix $A = \{a_{ij}\} \in \mathbb{R}^{n \times n}$; vectors $\mathbf{h}, \mathbf{f} \in \mathbb{R}^n$, positive integers n, N, k, m and the real number q , where $(-1)/\|A\| < q < 0$
- 2: COMPUTE: $p_i = \frac{|h_i|}{\sum_{i=1}^n |h_i|}$; $p_{ij} = \frac{|a_{ij}|}{\sum_{j=1}^n |a_{ij}|}$, $1 \leq i \leq n$, $1 \leq j \leq n$
- 3: COMPUTE: $a_i = \sum_{j=1}^n |a_{ij}|$, $1 \leq i \leq n$, $\|\mathbf{h}\| = \sum_{i=1}^n |h_i|$
- 4: GENERATE: N elements of the $(k+1)$ - dimensional Sobol sequence
- 5: CONSTRUCT: N realizations of the Markov chain with integer elements

$$l_1^{(s)} \rightarrow \dots \rightarrow l_i^{(s)} \rightarrow \dots \rightarrow l_k^{(s)} \rightarrow l_{k+1}^{(s)}, \quad 1 \leq l_i^{(s)} \leq n, \quad 1 \leq i \leq k+1, \quad 1 \leq s \leq N$$

- 6: COMPUTE:

$$\bar{\theta}^{(i)} = \frac{1}{N} \|\mathbf{h}\| \sum_{s=1}^N \text{sign}\{h_{l_1^{(s)}} \prod_{i=1}^{k+1} (a_{l_i^{(s)} l_{i+1}^{(s)}})\} \prod_{i=1}^{k+1} a_{l_i^{(s)}} f_{l_{i+1}^{(s)}}; \quad 1 \leq i \leq k+1$$

- 7: COMPUTE:

$$C_{i+m-1}^i = \frac{(i+m-1)!}{i!(m-1)!}, \text{ where } 1 \leq i \leq k$$

- 8: OUTPUT:

$$\lambda_{\min} \approx \frac{\bar{\theta}^{(1)} + qC_m^1 \bar{\theta}^{(2)} + q^2 C_{m+1}^2 \bar{\theta}^{(3)} + \dots + q^k C_{k+m-1}^k \bar{\theta}^{(k+1)}}{1 + qC_m^1 \bar{\theta}^{(1)} + q^2 C_{m+1}^2 \bar{\theta}^{(2)} + \dots + q^k C_{k+m-1}^k \bar{\theta}^{(k)}}$$

Систематичната грешка произтича от използването на крайна (прекъсната) верига на Марков и е аналогична на прекъсната верига на Нойман при детерминистичните методи. Тя зависи от параметъра на прекъсване k в представянето на резолвентната матрица $R_{q,k}^m$ като граница на безкраен ред при условие $|q|||A|| < 1$. Параметърът m определя степента на резолвентната матрица и влияе върху сходимостта на метода:

$$\mathcal{O}\left(\left|\frac{\mu_2}{\mu_1}\right|^m\right), \quad (2.26)$$

където μ_1, μ_2 са най-големите собствени стойности на резолвентната матрица. Изразено чрез собствените стойности на A :

$$\mathcal{O}\left(\left|\frac{1-q\lambda_1}{1-q\lambda_2}\right|^m\right), \text{ при } q > 0 \text{ и } \mathcal{O}\left(\left|\frac{1-q\lambda_n}{1-q\lambda_{n-1}}\right|^m\right), \text{ при } q < 0. \quad (2.27)$$

Изборът на ускоряващия параметър q определя дали алгоритъмът оценява минималната ($q \in (-\frac{1}{\|A\|}, 0)$) или максималната собствена стойност ($q > 0$). Изчислителният процес включва генериране на N реализации на веригата на Марков, пресмятане на средните стойности при симулиране на случайните величини и използване на биномни коефициенти за оценка на собствената стойност.

Балансът между стохастичната и систематичната грешка е съществен за постигане на предварително зададена точност ε при оптимална изчислителна цена. Този баланс се контролира чрез параметрите N (брой реализации), k (дължина на веригата на Марков), m (степената на резолвентната матрица) и q (ускоряващ параметър). Числените експерименти показват, че е необходимо изборът на тези параметри да бъде съгласуван с получените теоретични оценки за сходимост.

2.4 Числени резултати и дискусия

Числените резултати са постигнати чрез прилагането на *почти оптималния* Резолвентен Квази-Монте Карло (РКМК) алгоритъм (**Алгоритъм 2**) и *почти оптималния* Резолвентен Монте Карло (РМК) алгоритъм (**Алгоритъм 2а**) за оценяване на минималната собствена стойност на случайно генерирани симетрични матрици с размерности $n = 500$ и $n = 1000$. *Почти оптималният* РКМК алгоритъм е реализиран с генератора BRODA's Sobol RSG, който позволява генериране на разбъркани редици на Собол с необходимата размерност при конструиране на веригата на Марков. *Почти оптималният* РМК алгоритъм (**Алгоритъм 2а**) е реализиран с генератора на псевдослучайни числа Mersenne Twister (MT). Двата алгоритъма са имплементирани на C++, като предварително се използва подпрограма за генериране на случайна симетрична матрица A със спектър в единичната окръжност, удовлетворяващ условието:

$$0 < c < \lambda_n < \lambda_{n-1} \leq \dots \leq \lambda_2 < \lambda_1 < 1,$$

където константата c е избрана да бъде $c = 0.15$. Това гарантира положителност и спектрална отделимост, необходими за сходимостта на Резолвентния Степенен метод.

Разгледани са два тестови случая:

- (i) Матрица A с размерност $n = 500$, като двете най-големи и двете най-малки собствени стойности са: $\lambda_1 = 0.620713$, $\lambda_2 = 0.335683$, $\lambda_{499} = 0.15452$ и $\lambda_{500} = 0.152992$.
- (ii) Матрица A с размерност $n = 1000$, като двете най-големи и двете най-малки собствени стойности са: $\lambda_1 = 0.688181$, $\lambda_2 = 0.443078$, $\lambda_{999} = 0.154138$ и $\lambda_{1000} = 0.151911$.

Различието между двете най-големи и двете най-малки собствени стойности осигурява надеждна сходимост на метода при оценяване на λ_{min} .

Числените експерименти са проведени върху високопроизводителен клъстер (ИИКТ-БАН), състоящ се от 12 броя сървъри Fujitsu Primergy RX 2540 M4 като всеки от тях е със следната конфигурация: 128 GB оперативна памет; два броя процесори - CPU 2x Intel Xeon Gold 5118 2.30 GHz с 24 ядра; 2 броя по 800 GB SSD; 36 TB HDD, Графична карта - NVIDIA Tesla V100 с 32 GB оперативна памет. Сървърите са свързани с InfiniBand за бърз обмен на данни с малка латентност. Операционната система е Red Hat Enterprise Linux 7.6 и NVIDIA GRID.

Реализацията на **Алгоритъм 2** е хибридна: генерирането на разбърканите редици на Собол се извършва върху GPU (NVIDIA Tesla V100, 5120 CUDA ядра), докато останалите изчисления се изпълняват върху CPU. **Алгоритъм 2а** е реализиран върху CPU с генератора Mersenne Twister.

Произволните вектори $\mathbf{h}$ и $\mathbf{f}$ са избрани с L_1 -норма по следния начин:

$$\|\mathbf{h}\| = \|\mathbf{f}\| = \sum_{i=1}^n \frac{1}{n} = 1,$$

за да се осигури коректна нормировка при стохастичните оценки. Числените експерименти показват, че при големи стойности на броя реализации N , времето за изпълнение на РКМК алгоритъма ($RQMC_{(S)}$) е

съществено по-малко от това на РМК алгоритъма ($RMC_{(MT)}$), което демонстрира предимството на Квази-Монте Карло подхода в комбинация с GPU ускорение при задачи с големи размерности за оценяване на минималната собствена стойност.

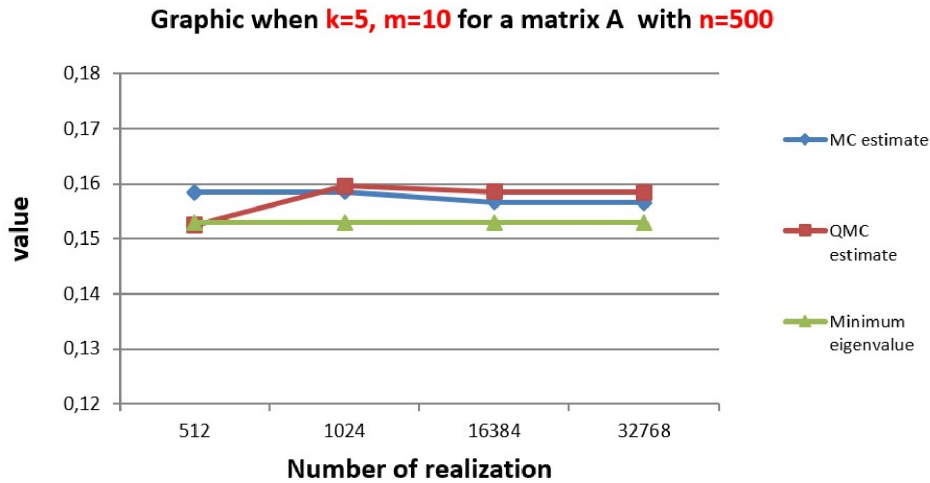
2.4.1 Случай (i): Размер на матрицата $n=500$

В Таблица 2.1 са представени числени резултати за оценяване на минималната собствена стойност на симетрична матрица с размерност $n = 500$, получени чрез *почти оптималните* алгоритми ($RMC_{(MT)}$ и $RQMC_{(S)}$) при различни стойности на параметрите N, k и m . В таблицата са включени и съответните абсолютни грешки на получените оценки. Параметърът на ускорение е фиксиран на $q = -0.157428$, а систематичната грешка е предварително зададена на ниво $\varepsilon_{sys} = 0.001$.

Таблица 2.1: Числени резултати за матрицата $A \in \mathbb{R}^{500 \times 500}$ при различни стойности на N, k и m с параметър на ускорение $q = -0.157428$. Точната минималната собствена стойност $\lambda_{\min} = \lambda_{500} = 0.152992$.

N	k	m	$\lambda_{\min}$ $RMC_{(MT)}$	Absolute error $RMC_{(MT)}$	$\lambda_{\min}$ $RQMC_{(S)}$	Absolute error $RQMC_{(S)}$
$512 * 2^0$	5	5	0.1585637	0.0055717	0.1561046	0.0031126
	5	10	0.1585163	0.0055243	0.1524884	0.0005036
	10	5	0.1556132	0.0026212	0.1560595	0.0030675
$512 * 2^1$	5	5	0.158395	0.005403	0.1596503	0.0066583
	5	10	0.1585518	0.0055598	0.1595946	0.0066026
	10	5	0.1580794	0.0050874	0.1596616	0.0066696
$512 * 2^5$	5	5	0.1572588	0.0042668	0.1591139	0.0061219
	5	10	0.1566524	0.0036604	0.1585388	0.0055468
	10	5	0.156875	0.003883	0.1591292	0.0061372
$512 * 2^6$	5	5	0.157129	0.004137	0.1589384	0.0059464
	5	10	0.1565796	0.0035876	0.1584563	0.0054643
	10	5	0.1570216	0.0040296	0.158961	0.005969

Анализът на резултатите се основава на балансирането на двата основни източника на грешка - систематична и стохастична. Систематичната грешка произтича от прекъсването на Ноймановия ред при $i = k+1$, което съответства на краен брой преходи във веригата на Марков, както и от избора на степента m на резолвентната матрица. Параметрите m и k се избират въз основа на теоретични оценки за опаската на реда $T_{m,k}(t)$, при $t = |q||\lambda_1| < 1$, така че да е изпълнено условието $T_{m,k}(t) \leq \varepsilon_{sys} = 0.001$. По този начин систематичната грешка се контролира предварително и остава от порядък $\mathcal{O}(10^{-3})$.



Фигура 2.1: Сходимост към точната минимална собствена стойност на матрицата A при използване на РМК и РКМК при $k = 5, m = 10$ за различни на брой N

Стохастичната грешка възниква от апроксимацията на математическите очаквания чрез краен брой реализации N . При Монте Карло подхода ($RMC_{(MT)}$) стохастичната грешка е от порядък $\mathcal{O}(N^{-1/2})$ при

фиксирана доверителна вероятност. При Квази-Монте Карло подхода ($RQMC_{(S)}$) се очаква по-бърза сходимост. За оценяване на грешката се използват рандомизирани редици на Соболев. Стохастичната грешка зависи не само от N , но и от параметъра k , тъй като при нарастване на k се увеличава броят на оценяваните случайни величини $\theta^{(i)}$, $i = 1, \dots, k + 1$. Това води до натрупване на дисперсии както в числителя, така и в знаменателя на дробно-рационалната формула за оценяване на собствената стойност, което влияе върху необходимия брой реализации. Оптималният баланс изисква избор на такова N , че стохастичната грешка да бъде от същия порядък като предварително зададената систематична грешка ε_{sys} .

Получените резултати за $n = 500$ потвърждават теоретичните изводи. При подходящ избор на параметрите (m, k) , гарантиращ ограничаване на систематичната грешка под $\mathcal{O}(10^{-3})$, се постига висока точност още при сравнително малък брой реализации N , съпоставим с размерността на матрицата. Най-добра точност е получена при $k = 5, m = 10$ и $N = 512$, където абсолютната грешка е от същия порядък като предварително зададената систематична грешка. Увеличаването на N (например, до $N = 16384$ или $N = 32768$, виж Фигура 2.1 и Таблица 2.1) не води до монотонно намаляване на абсолютната грешка. Наблюдава се насищане или колебливо поведение, което е характерно за ситуации, в които систематичната грешка вече определя долната граница на точността. Допълнително влияние оказват числени ефекти, свързани с нарастването на биномните коефициенти при по-големи стойности на m и k , които увеличават чувствителността на дробно-рационалната формула към малки флуктуации в оценките на $\theta^{(i)}$. Следователно, увеличаването на N не гарантира автоматично повишаване на точността в случаите, когато систематичната компонента е определяща.

2.4.2 Случай (ii): Размер на матрицата $n=1000$

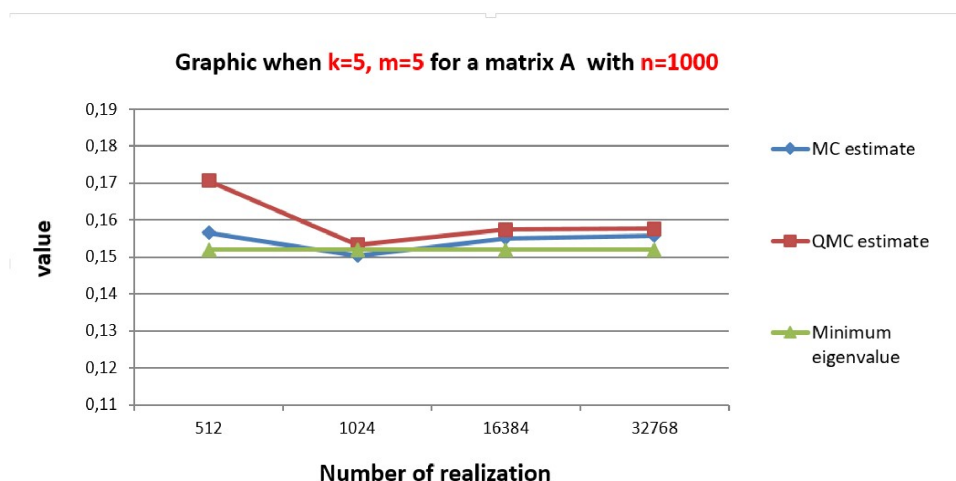
В Таблица 2.2 са представени числени резултати за оценяване на минималната собствена стойност на симетрична матрица с размерност $n = 1000$, получени чрез *почти оптималните* алгоритми $RM C_{(MT)}$ и $RQMC_{(S)}$, заедно със съответните абсолютни грешки. В тези експерименти параметърът на ускорение е фиксиран на $q = -0.2302458$, като абсолютната му стойност е по-голяма в сравнение със случая $n = 500$. Това води до увеличаване на параметъра $t = |q||\lambda_1|$ и налага, при същите стойности на k и m , систематичната грешка да бъде фиксирана на по-високо ниво $\varepsilon_{sys} = 0.01$, за да бъдат изпълнени теоретичните условия за оценка на опашката на реда.

Таблица 2.2: Числени резултати за матрицата $A \in \mathbb{R}^{1000 \times 1000}$ при различни стойности на N, k и m с параметър на ускорение $q = -0.2302458$. Точната минимална стойност $\lambda_{\min} = \lambda_{1000} = 0.151911$.

N	k	m	$\lambda_{\min}$ $RM C_{(MT)}$	Absolute error $RM C_{(MT)}$	$\lambda_{\min}$ $RQMC_{(S)}$	Absolute error $RQMC_{(S)}$
$512 * 2^0$	5	5	0.1556059	0.0045949	0.1705751	0.0186641
	5	10	0.168251	0.01634	0.1886959	0.0367849
	10	5	0.1593428	0.0074318	0.1701307	0.0182197
$512 * 2^1$	5	5	0.1504153	0.0014957	0.1533797	0.0014687
	5	10	0.1417805	0.0101305	0.1499974	0.0019136
	10	5	0.156739	0.004828	0.1534383	0.0015273
$512 * 2^5$	5	5	0.1550015	0.0030905	0.1573842	0.0054732
	5	10	0.1495483	0.0023627	0.1539357	0.0020247
	10	5	0.1559548	0.0040438	0.1575454	0.0056344
$512 * 2^6$	5	5	0.1557509	0.0038399	0.1576339	0.0057229
	5	10	0.1537008	0.0017898	0.1562911	0.0043801
	10	5	0.1562905	0.0043795	0.15764225	0.0057312

Получените резултати (Таблица 2.2 и Фигура 2.2) показват, че най-добрият баланс между систематичната и стохастичната компонента на грешката се постига при $k = 5, m = 5$ и $N = 1024$. При тази комбинация абсолютната грешка е минимална и от същия порядък, като предварително се задава систематичната грешка. Това потвърждава теоретичния извод, че параметрите m и k , определени въз основа на фиксираното ниво на систематичната грешка, осигуряват необходимата точност на апроксимацията на резолвентната матрица. Броят на реализациите N трябва да бъде избран така, че стохастичната грешка да бъде съпоставима по порядък със систематичната. При по-големи стойности на m, k и N се наблюдава известно подобрене на абсолютната грешка по отношение на константния множител, но без промяна в нейния порядък, като същевременно нараства изчислителната сложност. Увеличаването на N (например, до $N = 16384$ или $N = 32768$, виж Фигура 2.2 и Таблица 2.2) не води до монотонно намаляване на грешката. Причината се свързва с повишената чувствителност на дробно-рационалната формула към малки

флуктуации в оценките на случайните величини $\theta^{(i)}$, както и с доминиращата роля на систематичната грешка.



Фигура 2.2: Сходимост към точната минимална собствена стойност на матрицата A при използване на РМК и РКМК при $k = 5, m = 5$ за различни на брой N

Практически резултатите показват, че за матрици с размерност от порядъка на няколко хиляди елемента, оптималният баланс се постига, когато m и k се избират според теоретичните оценки при предварително фиксирано ниво на систематичната грешка, а N е от същия порядък като размерността на матрицата. Прекалено малко N води до доминиране на стохастичната грешка, а прекомерно голямо N увеличава изчислителната цена без съществено подобрение на точността.

2.5 Обобщение

Оценяването на минималната собствена стойност λ_{min} на симетрична матрица чрез степенни стохастични алгоритми изисква изчисляване на обратната матрица A^{-1} , което е изчислително скъпо при големи размерности. Като ефективна алтернатива е изследван Стохастичен Резолвентен метод, който елиминира необходимостта от явно намиране на A^{-1} , използвайки вместо това резолвентната матрица $R_q = (I - qA)^{-1}$ и нейните степени. Знакът на параметъра на ускорение q определя дали се оценява минималната или максималната собствена стойност.

Теоретичният анализ показва, че сходимостта на Резолвентния Степенен метод зависи съществено от параметъра q и степента m . Систематичната грешка намалява при увеличаване на m , а скоростта на сходимост се определя от отношението между крайните собствени стойности на матрицата. Прекъсването на биномния ред чрез параметъра k въвежда допълнителна систематична грешка, за която са получени аналитични оценки. Установена е линейна зависимост между m и k , позволяваща избор на минимални стойности при фиксирано ниво на систематична грешка и съответно намаляване на изчислителната сложност.

В стохастичната реализация възниква и стохастична грешка. Предложени са Резолвентен Монте Карло (РМК) и Резолвентен Квази-Монте Карло (РКМК) алгоритми, при които математическите очаквания се апроксимират чрез средните стойности от N реализации на случайните величини чрез прекъснатата верига на Марков с k прехода. При МК подхода грешката е от порядък $\mathcal{O}(N^{-1/2})$, докато при РКМК се използва неравенството на Коксма-Хлавка и редици с малък дискрепанс (например, редица на Соболев), осигуряваща по-бърза сходимост. Общата точност се определя от баланса между систематичната грешка (поради прекъсване на реда) и стохастичната грешка (поради крайния брой реализации), което налага балансиран избор на параметрите m, k и N .

Получените теоретични оценки дават практически насоки за избор на m, k и N така, че двата вида грешки да бъдат съпоставими по порядък спрямо зададена точност. Числените експерименти за матрици с размерности $n = 500$ и $n = 1000$ потвърждават тези изводи. Най-добрата ефективност се постига при параметри, съответстващи на теоретично предсказания баланс, като увеличаването на N над определена граница не води до съществено подобрение. РКМК алгоритъм показва по-бърза сходимост спрямо РМК подхода, а хибридната GPU/CPU реализация позволява ефективно приложение при задачи с голяма размерност. Част от получените резултати, описани в тази глава, са публикувани в статиите [35, 36].

3. Стохастични методи за оценяване на пазарния риск на инвестиционни портфейли

В тази глава представяме приложение на разработения **Алгоритъм 1** към реален проблем от финансовата математика - оценяване на пазарния риск на инвестиционен портфейл. Целта е да се демонстрира практическата приложимост на стохастичните степенни алгоритми чрез анализ на спектралните характеристики на корелационната матрица на доходностите. Оценяването на екстремални собствени стойности позволява идентифициране на доминиращи рискови фактори и изследване на чувствителността на портфейла към пазарни колебания. По този начин теоретичните алгоритми се използват като инструмент за количествен анализ на риска и за подпомагане на инвестиционни решения.

3.1 Постановка на задачата

Основната задача, която си поставяме, е да се оцени максималната собствена стойност на корелационни матрици и да се анализира точността на получените оценки чрез прилагането на разработените *почти оптимални* Степенен Монте Карло (СМК) и Степенен Квази-Монте Карло (СКМК) алгоритми, представени в Глава 1. В рамките на СКМК алгоритъма се въвеждат допълнително два оператора: (1) оператор за пропускане (англ. *skipping operator*) и (2) оператор за прескачане през стъпка (англ. *leaping operator*). Тяхното използване е мотивирано от необходимостта да се намали влиянието на началните неравномерности, характерни за рандомизираните редици с малък дискрепанс.

3.2 Методи и данни

Финансовите данни, използвани за конструирането на корелационните матрици, са извлечени от платформите *Portfolio Visualizer* и *Yahoo Finance*, като достъпът до последната е осъществен чрез библиотеката *yFinance*. Оценяването на най-голямата собствена стойност на корелационните матрици се извършва с посочените алгоритми в Глава 1 и се анализира балансът между систематичната и стохастичната грешка. За интерпретация на резултатите в контекста на финансовия анализ се използва показателят, наречен *дял на обяснената вариация* (англ. *Fraction of Variance Explained*, FVE). Той измерва частта на вариацията, съответстваща на максималната собствена стойност и характеризира степента на влияние на доминиращия пазарен фактор.

3.3 Обща формулировка на ковариационна и корелационна матрица

Ковариацията и коефициентът на корелация са основни характеристики за описване на линейната зависимост между случайни величини. Ковариацията между две случайни величини X и Y се дефинира чрез:

$$\text{Cov}(X, Y) = E[(X - \mu_X)(Y - \mu_Y)], \quad (3.1)$$

където $\mu_X = E[X]$ и $\mu_Y = E[Y]$ [77, 91]. Тя измерва посоката на съвместната изменчивост, като може да приема както положителни, така и отрицателни стойности, но зависи от мащаба на променливите. Коефициентът на корелация представлява нормализирана форма на ковариацията и се задава чрез:

$$\text{Corr}(X, Y) = \rho_{XY} = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}, \quad (3.2)$$

където σ_X и σ_Y са стандартните отклонения [77, 91]. Той приема стойности в интервала $[-1, 1]$ и позволява сравнение на линейните зависимости, независимо от мерните единици.

За случаен вектор $\mathbf{X} = (X_1, \dots, X_n)^T$ ковариационната матрица се дефинира като:

$$\Sigma = \text{Cov}(\mathbf{X}) = E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T], \quad (3.3)$$

където елементите $\Sigma_{ij} = \text{Cov}(X_i, X_j)$ се пресмятат по формула (3.1), а диагоналните елементи $\Sigma_{ii} = \sigma_i^2$ представляват дисперсиите на съответните компоненти на случайния вектор. Съответно, корелационната

матрица е:

$$\hat{R} = [\rho_{ij}]_{i,j=1}^n, \quad \rho_{ij} = \text{Corr}(X_i, X_j), \quad i, j = 1, \dots, n, \quad (3.4)$$

като $\rho_{ii} = 1$ [42, 57].

В контекста на портфейлния анализ, корелационната матрица описва зависимостите между доходностите на активите, а най-голямата ѝ собствена стойност служи като индикатор за доминиращ пазарен фактор [42]. Високи стойности на тази характеристика са свързани със засилена пазарна синхронизация и ограничени възможности за диверсификация. Ковариационната матрица предоставя пълна информация за взаимозависимостите между активите и е фундаментален инструмент в задачите по оптимизация и управление на портфейли.

3.3.1 Конструирание на ковариационна и корелационна матрица от емпирични данни от финансови пазари

В емпиричните изследвания на финансовите пазари данните обикновено се представят чрез матрицата $B \in \mathbb{R}^{r \times n}$, където редовете r съответстват на наблюденията на доходностите във времето, а колоните n - на броя на различни активи. Всеки ред $\mathbf{b}_i$ описва съвместното състояние на пазара в даден момент, което позволява анализ на взаимозависимостите между активите. Матрицата B се задава чрез:

$$B = [b_{ij}]_{i,j=1}^{r,n} = (\mathbf{b}_1, \dots, \mathbf{b}_i, \dots, \mathbf{b}_r)^T, \quad (3.5)$$

където $\mathbf{b}_i = (b_{i1}, \dots, b_{in})$ представлява n -мерен вектор от наблюдавани доходности за всички активи в момента i ($i = 1, \dots, r$).

Центриране на данните и извадкова ковариационна и корелационна матрица

Нека $\bar{\mathbf{b}}$ е n -мерният вектор на извадковите средни стойности на колоните на матрицата B . Центрираната матрица се дефинира по следния начин:

$$B_c = B - \mathbf{e}\bar{\mathbf{b}}^T, \quad (3.6)$$

където $\mathbf{e} \in \mathbb{R}^r$ е вектор от единици. Извадковата ковариационна матрица се определя чрез:

$$\Sigma = \frac{1}{r} B_c^T B_c. \quad (3.7)$$

При малки стойности на r във формула (3.7) делението се извършва с $r - 1$. Елементите Σ_{ij} измерват съвместната изменчивост между доходностите на активите, а диагоналните елементи Σ_{ii} съответстват на техните вариации. Рискът на портфейл Π с тегла $\mathbf{w}$ се задава чрез квадратичната форма:

$$\text{Var}(\Pi) = \mathbf{w}^T \Sigma \mathbf{w}, \quad (3.8)$$

което подчертава ролята както на индивидуалните рискове, така и на взаимните зависимости между активите. Корелационната матрица се получава чрез нормализиране на ковариационната матрица със стандартните отклонения:

$$\hat{R} = D_\sigma^{-1/2} \Sigma D_\sigma^{-1/2}, \quad \rho_{ij} = \frac{\Sigma_{ij}}{\sigma_i \sigma_j}, \quad (3.9)$$

като $\rho_{ij} \in [-1, 1]$ и $\rho_{ii} = 1$. Тази матрица предоставя мащабно независима мярка за линейните зависимости между активите и е ключова за анализа на диверсификацията [42].

3.3.2 Дял на обяснената вариация

Дялът на обяснената вариация (FVE) е важна характеристика, свързана със спектралните свойства на ковариационната или корелационната матрица [47]. За симетрична матрица A със собствени стойности $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$, дялът на обяснената вариация (FVE), съответстващ на най-голямата собствена стойност λ_1 , се дефинира като:

$$\text{FVE}_1 = \frac{\lambda_1}{\sum_{j=1}^n \lambda_j} = \frac{\lambda_1}{\text{trace}(A)}, \quad (3.10)$$

а за първите k компоненти:

$$\text{FVE}_k = \frac{\sum_{i=1}^k \lambda_i}{\text{trace}(A)}. \quad (3.11)$$

В частния случай на корелационна матрица $\hat{R}$, за която $\text{trace}(\hat{R}) = n$, се получава:

$$FVE_1 = \frac{\lambda_1}{n}. \quad (3.12)$$

Тази величина измерва относителния принос на доминиращия фактор в общата вариация на системата. В контекста на финансовите пазари големите стойности на FVE_1 показват наличие на силен общ пазарен фактор и високо ниво на систематичен риск, което ограничава ефективността на диверсификацията. Обратно, по-малките стойности предполагат по-добра разпределеност на риска между независими фактори и по-благоприятни условия за оптимизация на портфейла. Следователно, оценяването на най-голямата собствена стойност на корелационната матрица и съответния дял на обяснената вариация представлява ключов етап при количествения анализ на пазарния риск.

3.3.3 Стохастични алгоритми за оценяване на максималната собствена стойност на корелационна матрица

При оценяването на най-голямата собствена стойност на разглежданите корелационни матрици се прилагат разработените в Глава 1 *почти оптимални* Степенен Монте Карло (СМК) и Степенен Квази-Монте Карло (СКМК) алгоритми. За тези алгоритми беше показано, че превъзхождат *класическите* им аналози, тъй като при съпоставими изчислителни разходи осигуряват по-висока числена точност и по-стабилна сходимост на оценките. *Почти оптималният* СКМК алгоритъм, означен като **Алгоритъм 1** в Глава 1, използва разбъркани редици с малък дискрепанс за конструиране на верига на Марков. Ако в стъпка 4 се генерират редици на Холтърн или Соболев, то те се реализират съответно чрез псевдокодовете **Алгоритъм 3** и **Алгоритъм 4**.

Algorithm 3 Pseudocode for generating a scrambled Halton Sequence with *skip* and *leap* parameters

- 1: CREATE FUNCTION: `generateHaltonSequence` ($N, k+1$)
 - 2: GENERATE: $p \leftarrow \text{haltonset}(k, \text{'skip'} = s_1, \text{'leap'} = l_1)$
 - 3: GENERATE: $p \leftarrow \text{scramble}(p, \text{'RR2'})$
 - 4: GENERATE: $\text{points} \leftarrow \text{net}(p, N)$
 - 5: OUTPUT: Return points
-

Algorithm 4 Pseudocode for generating a scrambled Sobol Sequence with *skip* and *leap* parameters

- 1: CREATE FUNCTION: `generateSobolSequence` ($N, k+1$)
 - 2: GENERATE: $p \leftarrow \text{sobolset}(k, \text{'skip'} = s_1, \text{'leap'} = l_1)$
 - 3: GENERATE: $p \leftarrow \text{scramble}(p, \text{'MatousekAffineOwen'})$
 - 4: GENERATE: $\text{points} \leftarrow \text{net}(p, N)$
 - 5: OUTPUT: Return points
-

При избор $s_1 = 0$ и $l_1 = 0$ се получават стандартни (*default*) версии на редиците съответно означени с $PQMC_{(H)}^{(d)}$ и $PQMC_{(S)}^{(d)}$. При ненулеви стойности на параметрите се прилагат операциите *skip* и *leap*, които позволяват контрол върху началните елементи и допълнително разбъркване на редиците, като съответните алгоритми се означават с $PQMC_{(H)}^{(s,l)}$ и $PQMC_{(S)}^{(s,l)}$.

За реализацията на *почти оптималния* СМК алгоритъм (**Алгоритъм 1a**) се използват генераторите на псевдослучайни числа *Mersenne Twister* (MT) и *Middle Square* (MS), като съответните варианти се означават с $PMC_{(MT)}$ и $PMC_{(MS)}$. Всички алгоритми са реализирани в среда MATLAB (версия 9.10, R2021a), като използваният синтаксис на функциите `haltonset`, `sobolset` и `scramble` в **Алгоритъм 3** и **Алгоритъм 4** е съобразен с официалната документация на MATLAB [93, 94].

3.4 Числени резултати и дискусия

В настоящата секция са представени числени резултати за оценяване на максималната собствена стойност на две корелационни матрици чрез *почти оптималните* СМК и СКМК алгоритми. Изследването е проведено върху два портфейла от активи с различна размерност, което позволява анализ на поведението на алгоритмите при различна степен на сложност и информационна наситеност на данните.

Първата корелационна матрица е конструирана на базата на месечните доходности на $n = 32$ глобални актива за периода от януари 2007 г. до март 2024 г., със съответни размерности $r = 207$ и $n = 32$. Втората

матрица е изградена от доходностите на $n = 490$ компании от индекса $S\&P500$ за периода 2021-2025 г., при $r = 5$ и $n = 490$. Чрез прилагане на центриране (т.е. формула (3.6)), оценка на ковариационната матрица (3.7) и нормализация (3.9) се получават корелационни матрици със съответни максимални собствени стойности и дялове на обяснена вариация:

- за $n = 32$: $\lambda_{max} = 18.14714$, $FVE_1 = 56.71\%$;
- за $n = 490$: $\lambda_{max} = 201.090512$, $FVE_1 = 41.04\%$.

Тези стойности са изчислени чрез вградените функции на МАТЛАВ и служат като референтни стойности при оценка на точността на алгоритмите.

В числените експерименти векторите $\mathbf{h}$ и $\mathbf{f}$, използвани при конструирането на случайните величини $\theta^{(k)}$ и $\theta^{(k-1)}$, са избрани с L_1 -норма:

$$\|\mathbf{h}\| = \|\mathbf{f}\| = \sum_{i=1}^n \frac{1}{n} = 1,$$

което осигурява нормализация и стабилност на стохастичните оценки. За осигуряване на сходимост се разглежда матрицата $A = q\hat{R}$, където параметърът q се избира така, че да е изпълнено условието $\|A\| < 1$:

$$q = \frac{1}{|\lambda_1| + \delta}, \quad \delta > 0,$$

или при липса на предварителна оценка за λ_1 :

$$q = \frac{1}{\|\hat{R}\|_1 + \delta}, \quad \|\hat{R}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |\rho_{ij}| \quad \text{и} \quad \delta > 0.$$

Този подход осигурява коректна приложимост на стохастичните методи и гарантира тяхната числена устойчивост при оценяването на максималната собствена стойност.

3.4.1 Случай (i): корелационна матрица с размерност $n = 32$

В Таблица 3.1 са представени числените резултати за оценяване на най-голямата собствена стойност на корелационната матрица $\hat{R}$ при размерност $n = 32$. Получените числени резултати са реализирани чрез *почти оптималния* СМК алгоритъм (**Алгоритъм 1a**). Използват се два генератора на псевдослучайни числа - *Mersenne Twister* (MT) и *Middle Square* (MS), съответно означени с $PMC_{(MT)}$ и $PMC_{(MS)}$. Оценяването на максималната собствена стойност $\lambda_1 = \lambda_{max}$ е извършено при различни стойности на броя преходи k във веригата на Марков и броя реализации N на случайните величини $\theta^{(k)}$ и $\theta^{(k-1)}$, като в Таблица 3.1 са представени и съответните абсолютни грешки. Анализът показва, че при фиксирано N увеличаването на k води до намаляване на грешката, което се стабилизира около $k = 8$ и се достига практическият оптимален баланс между систематичната и стохастичната компонента на грешката. От друга страна, с увеличаване на N се наблюдава монотонно намаляване на абсолютната грешка, което е в съответствие с класическите свойства на Монте Карло методите.

Минималните абсолютни грешки за двата алгоритъма са отбелязани в Таблица 3.1, като резултатите показват съпоставим порядък на точност при двата генератора. Визуализацията на Фигура 3.1 илюстрира сходимостта на оценките $\lambda_1^{(k)}$ към точната стойност λ_1 при фиксирано $k = 8$ и нарастващ брой реализации N . Наблюдава се, че и двата алгоритъма се стремят към референтната стойност, но $PMC_{(MT)}$ демонстрира по-висока точност и по-устойчива сходимост, особено при по-малки N . Практически, за постигане на точност до втория или третия знак след десетичната запетая са достатъчни $N = 512$ или $N = 1024$ симулации, което е напълно адекватно за приложения в портфейлния анализ.

Чрез използване на формула (3.12) за корелационна матрица с размерност $n = 32$ и оценките при $N = 1024$, се получават стойности $FVE_1 \approx 56.70\%$ за $PMC_{(MT)}$ и $FVE_1 \approx 56.72\%$ за $PMC_{(MS)}$. Това потвърждава, че приближенията на λ_1 с точност от порядъка $O(10^{-3})$ позволяват надеждно изчисляване на дяла на обяснената вариация, който е ключов показател за оценка на систематичния риск и колективната динамика на активите. Следователно, *почти оптималният* СМК алгоритъм представлява ефективен подход за анализ на корелационни матрици с неголеми размерности при относително малки изчислителни разходи.

В Таблица 3.2 са представени резултати от прилагането на *почти оптималния* СКМК алгоритъм за корелационната матрица ($n = 32$), реализиран чрез четири варианта: $PQMC_{(H)}^{(d)}$, $PQMC_{(S)}^{(d)}$, $PQMC_{(H)}^{(s,l)}$

и $PQMC_{(S)}^{(s,l)}$. Вариантите с индекс (d) съответстват при избор $s_1 = 0$ и $l_1 = 0$ в **Алгоритъм 3** и **Алгоритъм 4**, когато са използвани разбъркани редици на Холтън и Соболев с настройки по подразбиране. При ненулеви стойности на параметрите $skip$ и $leap$ се реализират вариантите (s, l) , които позволяват по-фин контрол върху равномерността на генерираните точки и потенциално подобряват сходимостта на оценките.

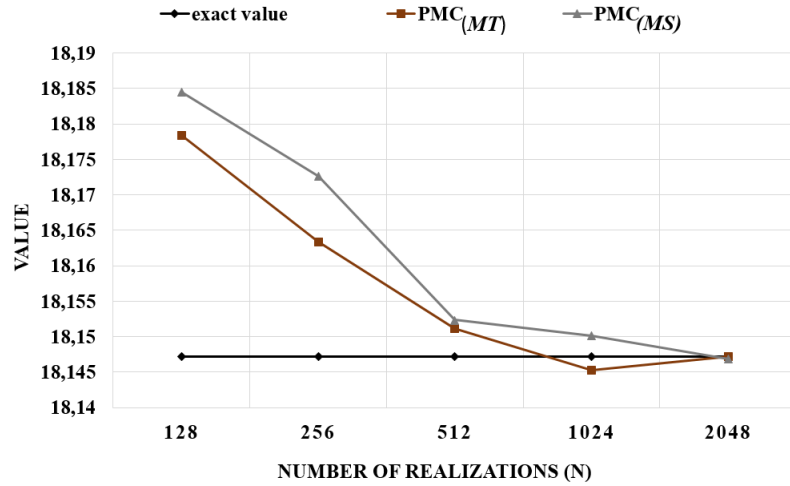
Таблица 3.1: Резултати за максималните стойности на $\lambda_{max}^{(k)}$ и абсолютната грешка за $PMC_{(MT)}$ и $PMC_{(MS)}$ при различни N и k , когато $\lambda_{max} = 18.14714$

N	k	$\lambda_1^{(k)}$ using $PMC_{(MT)}$	Absolute Error $PMC_{(MT)}$	$\lambda_1^{(k)}$ using $PMC_{(MS)}$	Absolute Error $PMC_{(MS)}$
128	5	17.809541	0.3376	17.766635	0.3805
	6	17.955782	0.1914	17.931153	0.2160
	7	18.031908	0.1152	18.016774	0.1304
	8	18.178382	0.0312	18.184488	0.0373
	9	18.217756	0.0706	18.234182	0.0870
	10	18.259830	0.1127	18.278015	0.1309
	11	18.293246	0.1461	18.336586	0.1894
256	5	17.886432	0.2607	17.880110	0.2670
	6	18.013889	0.1333	18.025125	0.1220
	7	18.098671	0.0485	18.090287	0.0569
	8	18.163375	0.0162	18.172589	0.0254
	9	18.214595	0.0675	18.220052	0.0729
	10	18.265757	0.1186	18.270934	0.1238
	11	18.307841	0.1607	18.317397	0.1703
512	5	17.894326	0.2528	17.893265	0.2539
	6	18.019977	0.1272	18.014737	0.1324
	7	18.099165	0.0480	18.094732	0.0524
	8	18.151116	0.0040	18.152321	0.0052
	9	18.194868	0.0477	18.218433	0.0713
	10	18.231143	0.0840	18.244924	0.0978
	11	18.275571	0.1284	18.284115	0.1370
1024	5	17.977295	0.1698	17.947696	0.1994
	6	18.102537	0.0446	18.038992	0.1081
	7	18.118865	0.0283	18.098850	0.0483
	8	18.145221	0.0019	18.150097	0.0030
	9	18.176356	0.0292	18.214874	0.0677
	10	18.202606	0.0555	18.276008	0.1289
	11	18.250397	0.1033	18.294888	0.1477
2048	5	18.007899	0.1392	17.981049	0.1661
	6	18.106753	0.0404	18.056203	0.0909
	7	18.126731	0.0204	18.104529	0.0426
	8	18.147226	0.0001	18.146851	0.0003
	9	18.164272	0.0171	18.172111	0.0250
	10	18.193708	0.0466	18.215959	0.0688
	11	18.218186	0.0710	18.260127	0.1130

*Маркираните редове представляват оптималните k стойности за всяко N .

Числените резултати за *почти оптималните* СКМК алгоритми ($PQMC_{(H)}^{(s,l)}$ и $PQMC_{(S)}^{(s,l)}$) са получени при избор на параметри $s_1 = 1024$ и $l_1 = 128$ в **Алгоритъм 3** и **Алгоритъм 4**, което съответства на прилагане на операциите $skip$ и $leap$ при генерирането на разбърканите редици на Холтън и Соболев. Максималната собствена стойност $\lambda_1 = \lambda_{max}$ на корелационната матрица е оценена при различни стойности на броя преходи k и броя реализации N , като в Таблица 3.2 са представени и съответните абсолютни грешки. Получените резултати показват, че параметрите k и N оказват съществено влияние върху баланса между систематичната и стохастичната грешка, както и върху изчислителната сложност на алгоритмите. При фиксирано N увеличаването на k води до намаляване на абсолютната грешка, което се стабилизира около $k = 11$ за вариантите по подразбиране $PQMC_{(H)}^{(d)}$ и $PQMC_{(S)}^{(d)}$, и около $k = 13$ за вариантите с параметри $PQMC_{(H)}^{(s,l)}$ и $PQMC_{(S)}^{(s,l)}$, което показва достигане на практически оптимален баланс между компонентите на грешката. При фиксирано k и нарастващо N , алгоритмите $PQMC_{(H)}^{(d)}$ и $PQMC_{(S)}^{(d)}$ демонстрират монотонно намаляване на грешката и стабилна сходимост. За разлика от тях, при $PQMC_{(H)}^{(s,l)}$ и $PQMC_{(S)}^{(s,l)}$ се наблюдава значително подобрение при малки извадки ($N = 128, 256$), но при по-големи N грешката остава почти постоянна или дори нараства, което съответства на наблюденията в [52] относно

чувствителността към параметрите *skip* и *leap*.



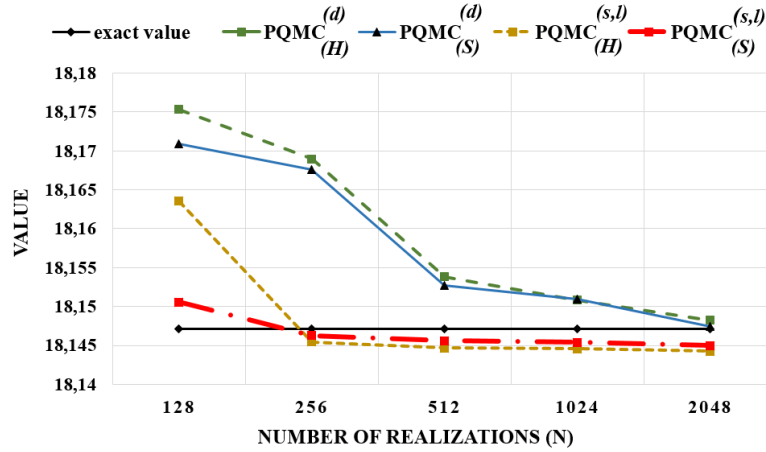
Фигура 3.1: Сравнение между точната максимална собствена стойност λ_1 и приближена собствена стойност $\lambda_1^{(k)}$ за $PMC_{(MT)}$, $PMC_{(MS)}$, получени в балансиращите случаи при различни стойности на N и $k = 8$

Таблица 3.2: Резултати за максимални собствени стойности за $\lambda_{max}^{(k)}$ и абсолютна грешка чрез сценарий по подразбиране и сценарий, използващ *skip* и *leap* при различни N и k , когато $\lambda_{max} = 18.14714$

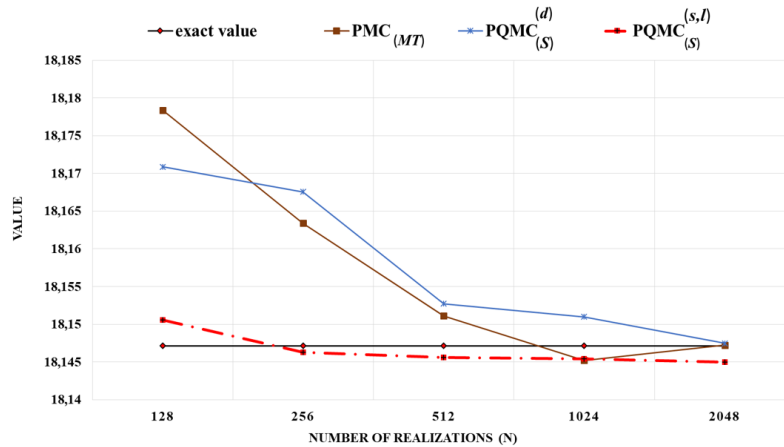
N	k	$\lambda_1^{(k)}$ using $PQMC_{(H)}^{(d)}$	Absolute Error $PQMC_{(H)}^{(d)}$	$\lambda_1^{(k)}$ using $PQMC_{(S)}^{(d)}$	Absolute Error $PQMC_{(S)}^{(d)}$	$\lambda_1^{(k)}$ using $PQMC_{(H)}^{(s,l)}$	Absolute Error $PQMC_{(H)}^{(s,l)}$	$\lambda_1^{(k)}$ using $PQMC_{(S)}^{(s,l)}$	Absolute Error $PQMC_{(S)}^{(s,l)}$
128	8	17.757547	0.3896	17.761379	0.3858	17.672733	0.4744	17.706637	0.4405
	9	17.888100	0.2590	17.888185	0.2590	17.787523	0.3596	17.792723	0.3544
	10	18.016669	0.1305	18.038214	0.1089	17.856741	0.2904	17.859089	0.2881
	11	18.175307	0.0282	18.170859	0.0237	17.903166	0.2440	17.914975	0.2322
	12	18.298811	0.1517	18.264832	0.1177	18.036970	0.1102	18.097564	0.0496
	13	18.479146	0.3320	18.474626	0.3275	18.145462	0.0017	18.146253	0.0009
	14	18.643362	0.4962	18.639031	0.4919	18.194240	0.0471	18.191902	0.0448
	15	18.712889	0.5657	18.706247	0.5591	18.290732	0.1436	18.290924	0.1438
256	8	17.882593	0.2645	17.883812	0.2633	17.761315	0.3858	17.763949	0.3832
	9	17.994408	0.1527	18.016913	0.1302	17.854804	0.2923	17.867712	0.2794
	10	18.061271	0.0859	18.062487	0.0847	17.936785	0.2104	17.921160	0.2260
	11	18.168944	0.0218	18.167556	0.0204	18.036010	0.1111	18.043959	0.1032
	12	18.284126	0.1370	18.251619	0.1045	18.097978	0.0492	18.098070	0.0491
	13	18.482373	0.3352	18.469322	0.3222	18.145462	0.0017	18.146253	0.0009
	14	18.545036	0.3979	18.516617	0.3695	18.209384	0.0622	18.185109	0.0380
	15	18.608838	0.4617	18.607262	0.4601	18.289967	0.1428	18.270105	0.1230
512	8	17.927397	0.2197	17.962654	0.1845	17.774098	0.3730	17.778972	0.3682
	9	18.020330	0.1268	18.021557	0.1256	17.846070	0.3011	17.867473	0.2797
	10	18.081105	0.0660	18.081198	0.0659	17.938077	0.2091	17.949995	0.1971
	11	18.153831	0.0067	18.152703	0.0056	18.009669	0.1375	18.014807	0.1323
	12	18.263301	0.1162	18.248667	0.1015	18.038078	0.1091	18.038550	0.1086
	13	18.425061	0.2779	18.411903	0.2648	18.144697	0.0024	18.145602	0.0015
	14	18.506259	0.3591	18.504204	0.3571	18.204814	0.0577	18.191490	0.0444
	15	18.551586	0.4044	18.546687	0.3995	18.249184	0.1020	18.242947	0.0958
1024	8	17.962214	0.1849	17.978111	0.1690	17.776881	0.3703	17.792702	0.3544
	9	18.035437	0.1117	18.042592	0.1045	17.867701	0.2794	17.887652	0.2595
	10	18.103422	0.0437	18.109582	0.0376	17.980078	0.1671	17.980502	0.1666
	11	18.150822	0.0037	18.150986	0.0038	18.010761	0.1364	18.037651	0.1095
	12	18.294487	0.1473	18.284447	0.1373	18.098692	0.0484	18.098718	0.0484
	13	18.385875	0.2387	18.384413	0.2373	18.144277	0.0029	18.144964	0.0022
	14	18.446470	0.2993	18.443825	0.2967	18.202841	0.0557	18.203207	0.0561
	15	18.530500	0.3834	18.521017	0.3739	18.236167	0.0890	18.235788	0.0886
2048	8	18.000431	0.1467	18.000619	0.1465	17.787254	0.3599	17.788274	0.3589
	9	18.061446	0.0857	18.064399	0.0827	17.879783	0.2674	17.887915	0.2592
	10	18.108451	0.0387	18.110469	0.0367	17.980078	0.1671	17.980502	0.1666
	11	18.148212	0.0011	18.147480	0.0003	18.010923	0.1362	18.040106	0.1070
	12	18.226453	0.0793	18.221684	0.0745	18.118819	0.0283	18.120939	0.0262
	13	18.385443	0.2383	18.379872	0.2327	18.144277	0.0029	18.144964	0.0022
	14	18.443228	0.2961	18.435573	0.2884	18.202381	0.0552	18.203555	0.0564
	15	18.509033	0.3619	18.492101	0.3450	18.247648	0.1005	18.243565	0.0964

*Маркираните редове представляват оптималните k стойности за всяко N .

Графичното сравнение на Фигура 3.2 потвърждава направените изводи, като илюстрира сходимостта на оценките $\lambda_1^{(k)}$ към точната стойност λ_1 при нарастващ брой реализации N . Алгоритмите $PQMC_{(S)}^{(d)}$ и $PQMC_{(H)}^{(d)}$ демонстрират устойчива и гладка сходимост, докато вариантите $PQMC_{(H)}^{(s,l)}$ и $PQMC_{(S)}^{(s,l)}$ показват по-добра точност при малки N , но ограничено подобрение при по-големи извадки. На Фигура 3.3 алгоритмите $PMC_{(MT)}$ и $PQMC_{(S)}^{(d)}$ се открояват с по-висока точност и стабилност при големи N , докато при малки извадки ($N \leq 512$) използването на СКМК алгоритми с параметри *skip* и *leap* е по-ефективно.



Фигура 3.2: $(PQMC_{(H)}^{(d)}, PQMC_{(S)}^{(d)})$ и $(PQMC_{(H)}^{(s,l)}, PQMC_{(S)}^{(s,l)})$. Сравнение между максималната собствена стойност $\lambda_{\max}$ на корелационната матрица и оценените стойности $\lambda_1^{(k)}$, получени при $k = 11$ в случая по подразбиране и $k = 13$ в случая с пропускане и прескачане и различни стойности на N .



Фигура 3.3: Сравнение между максимална собствена стойност $\lambda_{\max}$ и приближените собствени стойности $\lambda_1^{(k)}$, получени при $k = 8$ за $PMC_{(MT)}$, $k = 11$ за $PQMC_{(S)}^{(d)}$ и $k = 13$ за $PQMC_{(S)}^{(s,l)}$ при различни стойности на N .

Таблица 3.3: Приблизени стойности за дяла на обяснената вариация (FVE_1), получени при $k = 8$ за $PMC_{(MT)}$, $k = 11$ за $PQMC_{(S)}^{(d)}$ и $k = 13$ за $PQMC_{(S)}^{(s,l)}$ за корелационната матрица $\hat{R}$ с размерност $n = 32$. Референтната стойност е $FVE_1 = 56.71\%$.

N	$PMC_{(MT)}$	$PQMC_{(S)}^{(d)}$	$PQMC_{(S)}^{(s,l)}$
256	56.76%	56.77%	56.70%
2048	56.71%	56.71%	56.70%

В Таблица 3.3 са представени оценки за FVE_1 , получени чрез $PMC_{(MT)}$, $PQMC_{(S)}^{(d)}$ и $PQMC_{(S)}^{(s,l)}$, при

балансиращи стойности на k и $N = 256, 2048$, като референтната стойност е $FVE_1 = 56.71\%$. Резултатите показват, че при малки извадки $PQMC_{(S)}^{(s,l)}$ дава по-добри оценки, докато при големи N по-ефективни са $PMC_{(MT)}$ и $PQMC_{(S)}^{(d)}$, които постигат по-висока точност при по-малка изчислителна сложност. Високата стойност на FVE_1 за $n = 32$ показва наличие на доминиращ фактор, което означава по-висока концентрация на систематичен риск и ограничени възможности за диверсификация, тъй като активите реагират сходно на пазарни шокове.

3.4.2 Случай (ii): корелационна матрица с размерност $n = 490$

В Таблица 3.4 са представени числените резултати за оценяване на максималната собствена стойност $\lambda_1 = \lambda_{\max}$ на корелационна матрица $\hat{R}$ с размерност $n = 490$, конструирана на базата на доходностите на компании от индекса $S\&P500$ за периода 2021–2025 г. (виж Приложение). Поради липса на пълни данни, десет компании са изключени от анализа. Оценяването е извършено чрез алгоритмите $PMC_{(MT)}$, $PQMC_{(H)}^{(d)}$ и $PQMC_{(S)}^{(d)}$, при различни стойности на броя преходи k и броя реализации N , като са анализирани и съответните абсолютни грешки. Оценяването е извършено чрез *почти оптималния* СКМ алгоритъм $PMC_{(MT)}$ и *почти оптималните* СКМК алгоритми ($PQMC_{(H)}^{(d)}$ и $PQMC_{(S)}^{(d)}$), използващи разбъркани редици на Холтън и Собол с настройки по *подразбиране*. Резултатите показват, че параметрите k и N оказват съществено влияние върху точността на оценките, като балансиращите стойности на броя преходи са $k = 8$ за $PMC_{(MT)}$ и $k = 11$ за $PQMC_{(H)}^{(d)}$ и $PQMC_{(S)}^{(d)}$, независимо от значително по-голямата размерност спрямо случая $n = 32$.

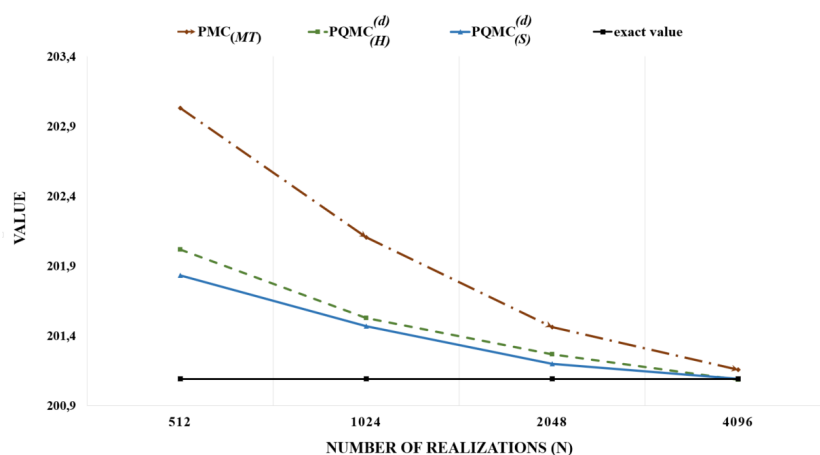
Таблица 3.4: Резултати за максимални собствени стойности за $\lambda_{\max}^{(k)}$ и абсолютна грешка при $PMC_{(MT)}$, $PQMC_{(H)}^{(d)}$ и $PQMC_{(S)}^{(d)}$ при различни N и k , когато $\lambda_{\max} = 201,090512$.

N	k	$\lambda_1^{(k)}$ using $PMC_{(MT)}$	Absolute Error $PMC_{(MT)}$	k	$\lambda_1^{(k)}$ using $PQMC_{(H)}^{(d)}$	Absolute Error $PQMC_{(H)}^{(d)}$	$\lambda_1^{(k)}$ using $PQMC_{(S)}^{(d)}$	Absolute Error $PQMC_{(S)}^{(d)}$
512	6	191.261222	9.8293	8	192.052259	9.0383	192.056796	9.0337
	7	197.764391	3.3261	9	193.305250	7.7853	194.4876692	6.6028
	8	203.030089	1.9396	10	196.545330	4.5452	196.561636	4.5289
	9	206.001210	4.9107	11	202.016871	0.9264	201.830255	0.7397
	10	209.202619	8.1121	12	207.216480	6.1260	206.994716	5.9042
1024	6	194.244715	6.8458	8	191.604757	9.4858	191.662089	9.4284
	7	199.671479	1.4190	9	194.355045	6.7355	194.443329	6.6472
	8	202.103664	1.0132	10	198.362863	2.7276	196.623242	4.4673
	9	204.706063	3.6156	11	201.526659	0.4361	201.467671	0.3772
	10	211.940453	10.8499	12	204.493635	3.4031	204.718275	3.6278
2048	6	196.531375	4.5591	8	191.522259	9.5683	191.737033	9.3535
	7	199.092487	1.9980	9	194.216871	6.8736	194.383618	6.7069
	8	201.462168	0.3717	10	198.552479	2.5380	198.866204	2.2243
	9	204.787620	3.6971	11	201.267671	0.1772	201.199260	0.1087
	10	206.264686	5.1942	12	207.590516	6.5000	206.670141	5.5796
4096	6	195.762474	5.3280	8	194.293635	6.7969	195.846272	6.2442
	7	197.402088	3.6884	9	198.498616	2.5919	198.498988	2.5915
	8	201.154640	0.0641	10	199.459470	1.6310	199.802438	1.2881
	9	202.887637	1.7971	11	201.083321	0.0072	201.091681	0.0012
	10	207.343376	6.2529	12	206.448996	5.3585	205.624451	4.5339

*Маркираните редове представляват оптималните k стойности за всяко N .

При фиксирано k се наблюдава монотонно намаляване на абсолютната грешка с увеличаване на броя реализации N , като минималните грешки са отбелязани в Таблица 3.4. Фигура 3.4 илюстрира, че всички алгоритми демонстрират устойчива сходимост към референтната стойност $\lambda_{\max}$, като вариантите на СКМК, $PQMC_{(H)}^{(d)}$ и $PQMC_{(S)}^{(d)}$, показват по-бърза и по-гладка сходимост спрямо $PMC_{(MT)}$. Получените оценки за λ_1 чрез $PQMC_{(H)}^{(d)}$ и $PQMC_{(S)}^{(d)}$ позволяват по-прецизно изчисляване на дяла на обяснената вариация FVE_1 чрез формула (3.12).

В Таблица 3.5 са представени съответните резултати, получени при балансиращите стойности $k = 8$ и $k = 11$, като референтната стойност е $FVE_1 = 41.04\%$. Данните показват, че при увеличаване на N използването на СКМК алгоритмите ($PQMC_{(H)}^{(d)}$ и $PQMC_{(S)}^{(d)}$) са за предпочитане пред $PMC_{(MT)}$ поради по-високата точност и при съпоставима изчислителна сложност.



Фигура 3.4: Сравнение между точната максимална собствена стойност ($\lambda_{\max}$) и максималната приближена собствена стойност ($\lambda_1^{(k)}$) за $PMC_{(MT)}$, $PQMC_{(H)}^{(d)}$, $PQMC_{(S)}^{(d)}$, получени когато $k = 8$ в случая за $PMC_{(MT)}$ и $k = 11$ в случая за $PQMC_{(*)}^{(d)}$ и различни стойности на N

Таблица 3.5: Приблизени стойности за дяла на обяснената вариация (FVE_1), получени при $k = 8$ за $PMC_{(MT)}$ и при $k = 11$ за $PQMC_{(H)}^{(d)}$ и $PQMC_{(S)}^{(d)}$ за корелационната матрица $\hat{R}$ с размерност $n = 490$. Референтната стойност е $FVE_1 = 41.04\%$.

N	$PMC_{(MT)}$	$PQMC_{(H)}^{(d)}$	$PQMC_{(S)}^{(d)}$
1024	41.25%	41.13%	41.12%
2048	41.11%	41.08%	41.06%
4096	41.05%	41.04%	41.04%

Стойността $FVE_1 \approx 41\%$ за корелационната матрица с размерност $n = 490$ показва, че вариацията не е концентрирана в първата главна компонента, а е разпределена между множество фактори. Това свидетелства за по-сложна структура на зависимостите между активите и предполага по-висок потенциал за диверсификация. В същото време, този резултат налага използването на многофакторни модели за адекватно описание на пазарния риск, тъй като еднофакторните подходи биха били недостатъчни за улавяне на наблюдаваната динамика.

3.5 Обобщение

Получените резултати потвърждават практическата приложимост и числената ефективност на *почти оптималните* стохастични алгоритми за оценяване на максималната собствена стойност на корелационни матрици при анализ на пазарния риск. Изследваните матрици, с размерности $n = 32$ и $n = 490$, са конструирани на базата на данни от *Portfolio Visualizer* и *Yahoo Finance* (виж Приложение). Анализът на корелационните матрици с размерности $n = 32$ и $n = 490$ показва, че максималната собствена стойност λ_1 има ясна интерпретация като индикатор за наличието и интензитета на доминиращ общ пазарен фактор. Чрез съответния дял на обяснената вариация FVE_1 се получава количествена оценка както за нивото на систематичния риск, така и за ограниченията пред ефективната диверсификация на портфейла.

Числените експерименти ($n = 32$) показват, че при *почти оптималните* СМК алгоритми балансът между стохастичната и систематичната грешка се постига при $k = 8$. При *почти оптималните* СКМК алгоритми с разбъркани редици на Собол и Холтън с настройки по *подразбиране* този баланс се достига при $k = 11$, а при използване на операторите *skip* и *leap* - при $k = 13$. Освен това, с увеличаване на броя реализации N се наблюдава монотонно намаляване на абсолютната грешка при СМК и СКМК алгоритмите с настройки по *подразбиране*, което потвърждава тяхната стабилна сходимост.

При оценяване на максималната собствена стойност на матрица с размерност ($n = 490$) *почти оптималните* СКМК алгоритми с настройки по *подразбиране* демонстрират по-добра сходимост и по-висока точност спрямо *почти оптималните* СМК алгоритми. Използването на операторите *skip* и *leap* е ефективно предимно при малък брой симулации N , като при по-големи извадки тяхното предимство намалява и не гарантира допълнително подобрене на точността. Част от представените резултати са публикувани

в [38].

В обобщение, разработените стохастични степенни алгоритми могат да бъдат разглеждани като надежден подход за числено оценяване на спектрални характеристики на корелационни матрици. Това от своя страна позволява задълбочен анализ на систематичния риск, колективната динамика на активите и ефективността на диверсификацията. В бъдещи изследвания тези алгоритми могат да бъдат прилагани в задачи, свързани с финансово моделиране, управление на портфейли и подпомагане на инвестиционни решения в условия на нарастваща размерност.

Заклучение

Настоящата дисертация е посветена на разработването, анализа и приложението на стохастични числени методи от типа МК и рандомизирани КМК методи за оценяване на екстремални собствени стойности на симетрични матрици и използването им при практически задачи, свързани с оценка на риска на инвестиционни портфейли. Основната цел е изграждането на стабилна и изчислително ефективна методологична рамка, която осигурява управляем баланс между систематичната и стохастичната грешка и е приложима при задачи с висока размерност.

В рамките на изследването са обосновани и изследвани Степенният (Резолвентният) Монте Карло метод и рандомизираният Степенен (Резолвентен) Квази-Монте Карло метод за оценяване на екстремални собствени стойности. Съществен резултат е прецизирането на конструкцията на случайните величини при използването на почти оптимална преходна плътност във веригата на Марков, както и идентифицирането на класове симетрични матрици, при които тази конструкция се опростява. На тази основа са разработени почти оптимални алгоритми (Степенен и Резолвентен МК и КМК), описани чрез псевдокод, за които са дадени оценки за изчислителната сложност и условия за балансиране на статистическата и систематичната грешка. Теоретичните аргументи и експерименталните резултати потвърждават, че предложените почти оптимални конструкции водят до намаляване на дисперсията и повишаване на точността при същия или по-малък изчислителен разход спрямо класическите стохастични аналози.

Експерименталната верификация е проведена както с генератори на псевдослучайни числа (МТ и MS), така и с редици с малък дискрепанс (редици на Соболев и Холтърн) и техни разбъркани варианти. Числените изследвания показват измерим ефект от избора на генератор и типа редица върху точността и стабилността на оценките, като при разглежданите тестове генераторът на псевдослучайни числа МТ демонстрира предимство спрямо генератора MS, а редиците на Соболев, спрямо редиците на Холтърн. Установено е също, че за постигане на сравним баланс на грешките рандомизираният Степенен (Резолвентен) КМК алгоритъм може да изисква малко по-голям брой преходи във веригата на Марков, което се компенсира от по-добрата сходимост при ограничени изчислителни ресурси.

Разработените алгоритми са приложени към задача от финансовата математика, свързана с управлението на риска на инвестиционен портфейл. Максималната собствена стойност на корелационната матрица на активи е използвана като индикатор за концентрация на риска, като оценяването ѝ е реализирано чрез предложените CMK и SKMK алгоритми. При реализиране на SKMK алгоритмите с разбъркани редици с малък дискрепанс (редици на Соболев и Холтърн) са изследвани варианти и с използване на операторите *skip* и *leap*.

В обобщение, дисертационният труд демонстрира, че прилагането на разработените алгоритми за получаване на спектрални оценки с целенасочено балансиране на грешките води до по-висока точност, стабилност и изчислителна ефективност. Очертани са и перспективи за бъдещо развитие, включително разширяване към устойчиви (робастни) модели, динамични ковариационни структури и допълнителни изчислителни оптимизации за разреждени симетрични матрици с големи размерности от порядъка на милиони.

Списък на публикациите по дисертацията

За написването на дисертацията са използвани следните публикации:

1. **Gurova, S.-M.**, Karaivanova, A., *Quasi-Monte Carlo Algorithms for Eigenvalue Problems, 4th International Conference Numerical Methods for Scientific Computations and Advanced Applications, NMSCAA'24*, 18-22, (2024) (short communication)
<https://parallel.bas.bg/~elilkova/nmscaa24.pdf>
2. **Gurova, S.-M.**, Atanassov, E., Karaivanova, A., *A Resolvent Quasi-Monte Carlo Method for Estimating the Minimum Eigenvalues Using the Error Balancing*, In: Lirkov, I., Margenov, S. (eds), *Large-Scale Scientific Computing, LSSC'23, LNCS*, vol. **13952**, 394-403, (2024), (SJR:0.352, Q2, 2024)
https://doi.org/10.1007/978-3-031-56208-2_40
3. **Gurova, S.-M.**, Gurov, T., Karaivanova, A., *On the Stochastic Power Algorithms for Estimating the Maximum Eigenvalue of Symmetric Matrices*, **Contemp. Math.**, Universal Wiser Publisher Pte. Ltd, **6(5):7223-7249**, (2025) doi.org/10.37256/cm.6520257781 (IF:2.5, 2024, WoS: Q1, <https://wos-journal.info/journalid/23187>)

4. Gurova, S.-M., Gurov, T., Karaivanova, A., *Eigenvalue Estimation in Portfolio Risk: The Role of Skipping and Leaping in Sobol and Halton Sequences*, In: Lirkov, I., Margenov, S. (eds), Large-Scale Scientific Computing, LSSC'25, LNCS, 16061, (accepted)

Апробация на резултатите

Резултатите, които са включени в дисертацията, са докладвани на семинари на секцията "Скалируеми алгоритми и киберсигурност с Център по високопроизводителни пресмятания" към ИИКТ-БАН, както и на международни конференции. Изнесени са следните доклади, които са свързани с темата на дисертацията:

1. Gurova, S.-M., Monte Carlo Method for Estimating Eigenvalues Using Error Balancing, The 13th International Conference on Large-Scale Scientific Computations, LSSC'21, 07-11.06.2021, Sozopol, Bulgaria. <https://parallel.bas.bg/Conferences/SciCom21/program.html>
2. Gurova, S.-M., A Quasi-Monte Carlo method for estimation of eigenvalues using error balancing, The 15th International Conference on Monte Carlo and Quasi-Monte Carlo methods in Scientific Computing, MCQMC'22, 17-22.07.2022, Linz, Austria. https://www.ricam.oeaw.ac.at/events/conferences/mcqm2022/schedule/MCQMC2022_book_final_version.pdf
3. Gurova, S.-M., Power Monte Carlo method using randomized low discrepancy sequences, the 17th Annual Meeting of the Bulgarian Section of SIAM, 20-22.12.2022, BGSIAM'22, Sofia, Bulgaria. http://www.math.bas.bg/bgsiam/docs/bgsiam_2022_program.pdf
4. Gurova, S.-M., A Resolvent Quasi-Monte Carlo Method for Estimating the Minimum Eigenvalues Using the Error Balancing, The 14th International Conference on Large-Scale Scientific Computations, LSSC'23, 05-09.06.2023, Sozopol, Bulgaria. <https://parallel.bas.bg/~elilkova/lssc23.pdf#page=17>
5. Gurova, S.-M., Quasi-Monte Carlo Algorithms for Eigenvalue Problems, Fourth International Conference Numerical Methods for Scientific Computations and Advanced Applications, NMSCAA'24, 17-21.06.2024, Sozopol, Bulgaria. <https://parallel.bas.bg/Conferences/nmscaa24/program.html>

Статиите представят резултати, получени в рамките на един международен и три национални научни проекта, като единият от тях е разделен на две фази, а именно:

1. Международен научен проект

- Нови скалируеми алгоритми с използване на Монте Карло и Квази-Монте Карло методи и тяхното приложение в областта на машинното обучение, финансиран от CAF America, 2021-2024, ръководител: проф. д-р Емануил Атанасов.

2. Национални научни проекти

- "Център за върхови постижения по Информатика и информационни и комуникационни технологии" (фаза 1) по договор BG05M2OP001-1.001-0003, финансиран от оперативна програма "Наука и образование за интелигентен растеж" 2014-2020, съфинансиран от Европейския съюз чрез Европейските структурни и инвестиционни фондове, ръководител: чл.-кор. д.м.н. Светозар Маргенов.
- "Национален геоинформационен център (НГИЦ) за мониторинг, оценка и прогнозиране на природни и антропогенни рискове и бедствия" обект от "Националната пътна карта за научна инфраструктура 2020-2027" (договори: Д01-321/30.11.2023, Д01-164/28.07.2022), финансиран от Министерство на образованието и науката, ръководител: доц. д-р Петя Трифонова.
- "Национален център за високопроизводителни и разпределени пресмятания" (НЦВРП) обект от "Националната пътна карта за научна инфраструктура 2020-2027" (договори: Д01-98/26.06.2025, Д01-325/01.12.2023), финансиран от Министерство на образованието и науката, ръководител: проф. д-р Анета Караиванова.
- "Център за върхови постижения в областта на информатиката и информационните и комуникационни технологии" (фаза 2) по договор BG16RFPR002-1.014-0018-C01, финансиран по програмата за научни изследвания, иновации и дигитализация за интелигентна трансформация 2021-2027 и съфинансиран от Европейския съюз, ръководител: чл.-кор. д.м.н. Светозар Маргенов.

Основни научни и научно-приложни приноси

Получените научни и научно-приложни приноси напълно съответстват на поставените цели и задачи на дисертационния труд. Те пряко отразяват обосноваването, разработването и експерименталното изследване на Степенен МК/КМК алгоритми и Резолвентен МК/КМК алгоритми за оценка на екстремни собствени стойности на симетрични матрици. Освен това, приносите са потвърдени и в реална задача от финансовата математика чрез оценяване на пазарния риск на портфейл от глобални активи.

1. Основните научни приноси на настоящата дисертация са:

- Обосновани и изследвани са Степенният (Резолвентният) Монте Карло метод и рандомизираният Степенен (Резолвентен) Квази-Монте Карло метод за оценяване на екстремални собствени стойности на симетрични матрици. Прецизирана е конструкцията на случайните величини $\theta^{(k)}$ и $\theta^{(k-1)}$ при използване на почти оптимална преходна плътност във веригата на Марков. Формулирани са класове симетрични матрици, за които се опростява конструкцията на случайните величини в стохастичните степенни алгоритми. Получени са теоретични оценки за зависимостта между степента на резолвентната матрица и дължината на веригата на Марков в зависимост от ускоряващия параметър и нормата на матрицата.
- Разработени са почти оптимални СМК и СКМК алгоритми, както и почти оптимални РМК и РКМК алгоритми, базирани на предложената прецизирана конструкция на случайните величини, които са представени чрез псевдокод. Дадени са оценки за изчислителната сложност на алгоритмите и условията за баланс между стохастичната и систематична грешка. Аргументирано е, че предложените алгоритми водят до намаляване на дисперсията спрямо класическите аналози.
- Предложена е практическа задача от областта на финансовата математика за оценяване на пазарния риск на инвестиционни портфейли. Конструираната корелационна матрица за съответния портфейл е използвана за оценяването на максималната собствена стойност, която е индикатор за концентрация на пазарния риск. Оценяването се извършва чрез разработените в Глава 1 почти оптимален Степенен МК и Степенен КМК алгоритми, като допълнително Степенният КМК алгоритъм е модифициран чрез използване на параметрите *skip* и *leap*.

2. Основните научно-приложни приноси на настоящата дисертация са:

- Предложените почти оптимални СМК алгоритми са реализирани с МТ и MS генератори на псевдослучайни числа, а СКМК алгоритмите, с разбъркани редици на Соболев и Холтън. Числените експерименти за оценяване на максималната собствена стойност на плътни симетрични матрици с големи размерности демонстрират превъзходството на почти оптималните алгоритми над класическите им аналози както по точност, така и по изчислителна сложност.
- Числено е изследван балансът между стохастичната и систематичната грешка, който зависи съществено от броя на преходите k във веригата на Марков и от броя реализации N на случайните величини $\theta^{(k)}$ и $\theta^{(k-1)}$. Резултатите показват, че СКМК алгоритмите се нуждаят от 2-3 стъпки повече във веригата на Марков в сравнение със СМК алгоритмите. Изборът на генератор на псевдослучайни числа или на разбърканата редица с малък дискрепанс има измерим ефект върху точността и стабилността. В разгледаните експерименти МТ е предпочитан пред MS, а редиците на Соболев пред редиците на Холтън.
- Почти оптималният РМК алгоритъм е реализиран с МТ генератор на псевдослучайни числа, а почти оптималният РКМК алгоритъм с разбъркана редица на Соболев чрез BRODA's Sobol RSG генератор. Числените експерименти, проведени на HPC клъстер с NVIDIA Tesla V100 (32 GB) GPU карти, за оценяване на минималната собствена стойност, показват, че балансът между стохастичната и систематичната грешка зависи от броя на преходите k във веригата на Марков, броя реализации N на случайните величини $\theta^{(k)}$ и $\theta^{(k-1)}$, степента m на резолвентната матрица и параметъра за ускорение q . Резултатите, получени с рандомизирана редица на Соболев и с МТ генератор за оценяване на минималната собствена стойност на симетрични матрици, демонстрират един и същ порядък. От съществено значение е възможността да се намери константа, която отделя минималната собствена стойност от нулата.
- Проведени са числени експерименти за оценяване на максималната собствена стойност на две корелационни матрици, която е индикатор за концентрация на пазарния риск. Първата корелационна матрица е конструирана въз основа на доходите на 32 глобални актива, докато

при втората са използвани доходите на 490 компании от *S&P500*, за конкретни времеви периоди (виж Приложение). Оценяването е извършено чрез почти оптималния СМК алгоритъм с MT/MS генератори на псевдослучайни числа и чрез почти оптималния СКМК алгоритъм, както без допълнителни настройки (*default* вариант), така и с използване на параметрите *skip* и *leap* при разбъркани редици на Соболев и Холтън.

Приложение

За провеждането на числените експерименти са използвани генерирани симетрични матрици (Глава 1, Глава 2) и реални финансови данни (Глава 3). Симетричните матрици с размерности 100 и 500, използвани за числените експерименти в Глава 1, са генерирани в софтуерната среда MATLAB, докато тези с размерности 500 и 1000 от Глава 2 са част от алгоритмичната реализация, разработена на програмния език C++.

Финансовите данни, използвани за конструирането на корелационната матрица с размерност $n = 32$ в Глава 3, са извлечени от уеб платформата *Portfolio Visualizer* (www.portfoliovisualizer.com). Изследваният времеви обхват обхваща периода от януари 2007 г. до март 2024 г., като са използвани реални пазарни показатели. Въз основа на тази емпирична информация е конструиран инвестиционен портфейл. Портфейлът се състои от тридесет и два глобални актива, класифицирани според техния инвестиционен тип, както следва:

1. **Осемнадесет акции** - „US Stock Market“, „US Large Cap Value“, „US Large Cap Growth“, „US Mid Cap“, „US Mid Cap Value“, „US Mid Cap Growth“, „US Small Cap Value“, „US Small Cap Growth“, „US Large Cap“, „US Small Cap“, „US Micro Cap“, „Global ex-US Stock Market“, „European Stocks“, „Pacific Stocks“, „Emerging Markets“, „Intl Developed ex-US Market“, „International ex-US Small Cap“, „International ex-US Value“;
2. **Девет облигации** - „Short Term Treasury“, „Intermediate Term Treasury“, „10-year Treasury“, „Long Term Treasury“, „Total US Bond Market“, „Global Bonds (USD Hedged)“, „Short-Term Investment Grade“, „TIPS (Treasury Inflation-Protected Securities)“, „Corporate Bonds“;
3. **Четири алтернативни инвестиции** - „REIT (Real Estate Investment Trusts)“, „Gold“, „Precious Metals“, „Commodities“;
4. **Един паричен актив и парични еквиваленти** - „Cash“.

Финансовите данни, използвани за конструирането на корелационната матрица с размерност $n = 490$ в Глава 3, са доходите на 490 компании от *S&P500* (*Standard&Poor's 500*), индекс за оценка на състоянието на фондовия пазар в САЩ, за 5-годишен период от 2021 до 2025 година. Данните за тяхната доходност са изтеглени от *Yahoo Finance* (<https://finance.yahoo.com/>) посредством програмната библиотека *yFinance*, достъпна чрез хранилището на *PyPI* (<https://pypi.org/project/yfinance/>). Чрез тази библиотека се осъществява автоматизирано извличане на исторически ценови серии и пазарни индикатори чрез интерфейс към уеб страницата на *Yahoo Finance*, което позволява програмно изтегляне на данни за конкретни финансови инструменти в зададен времеви интервал и последващата им обработка в аналитична среда.

1. Генерираните симетрични матрици с размерности 100, 500, използвани за числените експерименти в Глава 1, са публикувани в платформата **GitHub** и са достъпни на следния адрес:
 - https://github.com/smgurova/Symmetric-matrices_chapter-1.git
2. Финансовият отчет, предоставен от *Portfolio Visualizer*, е наличен в платформата **GitHub** на следния адрес:
 - <https://github.com/smgurova/AssetAllocation.git>
3. Корелационната матрица на тридесетте и два глобални актива, използвана за числените експерименти в Глава 3, е достъпна в платформата **GitHub** на следния адрес:
 - https://github.com/smgurova/correlation-matrix_chapter-4.git
4. Данните за доходността на 490 компании от *S&P500* (*Standard&Poor's 500*) за времеви период от 2021 до 2025 година и образуваната от тях корелационна матрица, използвана за числените експерименти в Глава 3, са достъпни в платформата **GitHub** на следния адрес:
 - <https://github.com/smgurova/Data-S-P500.git>

Библиография

- [1] Alexandrov, V., Atanassov, E., Dimov, I., Branford, S., Thandavan, A., Weihrauch, C., Parallel Hybrid Monte Carlo Algorithms for Matrix Computations. In: Sunderam, V.S., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) Computational Science – ICCS 2005. LNCS, vol. 3516, Springer, 752-759, (2005), https://doi.org/10.1007/11428862_102
- [2] Alexandrov, V., Davila, D., Esquivel-Flores, O., Karaivanova, A., Gurov, T., Atanassov, E., On Monte Carlo and quasi-Monte Carlo Methods for Matrix Computations, Lecture Notes in Computer Science, **10665**, Springer-Verlag, 249-257, (2018), DOI: https://doi.org/10.1007/978-3-319-73441-5_26
- [3] Alexandrov, V., Esquivel-Flores, O., Ivanovska, S., Karaivanova, A., On the Preconditioned Quasi-Monte Carlo Algorithm for Matrix Computations. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds.) Large-Scale Scientific Computing. LSSC 2015. Lecture Notes in Computer Science, vol. **9374**, Springer, Cham., 163-171, (2015), https://doi.org/10.1007/978-3-319-26520-9_17
- [4] Alexandrov, V., Esquivel-Flores, O.A., Towards Monte Carlo preconditioning approach and hybrid Monte Carlo algorithms for matrix computations, Computers & Mathematics with Applications, vol. **70**(11), 2709-2718, (2015), <https://doi.org/10.1016/j.camwa.2015.08.035>
- [5] Almulihi A.H. H., High-Performance Quasi-Monte Carlo Integration and Applications, Dissertations. 3452, Western Michigan University, (2019), <https://scholarworks.wmich.edu/dissertations/3452>, (Last accessed: January 2026)
- [6] Branford, S., Hybrid Monte Carlo Methods for Linear Algebra Problems, Ph.D. dissertation, School of Systems Engineering, The University of Reading, (2009)
- [7] BRODA's Sobol RSG: broda.co.uk/software.html (Last accessed: 2026/01/11)
- [8] Brouwer, A.E., Haemers, W.H., Spectra of Graphs, Springer, (2012) <https://doi.org/10.1007/978-1-4614-1939-6>
- [9] Caflisch, R. E., Monte Carlo and quasi-Monte Carlo methods, Acta Numerica, Cambridge University Press, 1-49, (1998) <https://www.math.ucla.edu/~caflisch/Pubs/Pubs1995-1999/actaNumerica1998.pdf> (Last accessed: 2026/01/11)
- [10] Chi, H. Scrambled Quasirandom Sequences and Their Applications, Florida State University, (2004) http://purl.flvc.org/fsu/fd/FSU_migr_etd-3823, (Last accessed: 2026/01/11)
- [11] Chi, H, Beerli, P., Evans, D. W., Mascagni, M., On the Scrambled Sobol Sequence, LNCS 3516, Springer, 775-782, (2005)
- [12] Chung, F., Spectral Graph Theory, CBMS Regional Conference Series in Mathematics 92, AMS, (1997).
- [13] Dick, J., Pillichshammer, F., Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration, Cambridge University Press (2010), <https://doi.org/10.1017/CB09780511761188>
- [14] Dekking, M., A Modern Introduction to Probability and Statistics. Springer, 181–190, (2005), ISBN: 9781852338961
- [15] Dimov, I., Monte Carlo algorithms for linear problems, Pliska Stud. Math. Bulgar., **13**, 57-77, (2000)
- [16] Dimov, I., Dimov, T., Gurov, T., A new iterative Monte Carlo approach for inverse matrix problem, Journal of Computational and Applied Mathematics, **92**(1), 15-35, (1998)
- [17] Dimov, I., Tonev, O., Monte Carlo algorithms: performance analysis for some computer architectures, Journal of Computational and Applied Mathematics, **48**, 253-277, (1993)
- [18] Dimov, I., Alexandrov, V., Karaivanova, A., Implementation of Monte Carlo Algorithms for Eigenvalue Problem Using MPI, PVM/MPI'98, LNCS, **1497**, 346-353, (1998)
- [19] Dimov, I., Philippe, B., Karaivanova, A., Weihrauch, C., Robustness and applicability of Markov chain Monte Carlo algorithms for eigenvalue problems, Applied Mathematical Modelling, **32**(8), 1511-1529, (2008)

- [20] Dimov, I., Alexandrov, V., Karaivanova, A., Parallel resolvent Monte Carlo algorithms for linear algebra problems, *Mathematics and Computers in Simulation*, **55**(1), 25-35, Published by Elsevier BV., (2005)
- [21] Dimov, I., Karaivanova, A.: A Power Method with Monte Carlo Iterations, *Proceedings Recent Advances in Numerical Methods and Applications II*, World Scientific, 239-247, (1999), https://doi.org/10.1142/9789814291071_0022
- [22] Dimov, I., *Monte Carlo Methods for Applied Scientists*, World Scientific Publishing Co. Pte. Ltd., (2007), ISBN: 13 978-981-02-2329-8
- [23] Dimov, I., Karaivanova, A., Iterative Monte Carlo algorithms for linear algebra problems, *Springer Lecture Notes in Computer Science*, **1196**, 66-77, (1996)
- [24] Dimov, I., Karaivanova, A., Parallel computations of eigenvalues based on a Monte Carlo approach, *Monte Carlo Methods and Applications*, Vol. **4**, Num. 1: 33-52, 1998
- [25] Dimov, I.T., et al.: Monte Carlo Algorithms for Calculating Eigenvalues. In: Niederreiter, H., et.al. (eds.) *Monte Carlo and Quasi-Monte Carlo Methods*, 1996, *Lecture Notes in Statistics*, vol. **127**, Springer, New York, NY., (1998)
- [26] Erkan, U., A precise and stable machine learning algorithm: eigenvalue classification (EigenClass), *Neural Comput & Applic*, **33**, 5381-5392, (2021), <https://doi.org/10.1007/s00521-020-05343-2>
- [27] Fathi, B., Liu, B., Alexandrov, V., Mixed MC parallel algorithms for Matrix Computation, *LNCS*, **2330**, 609-618, (2002)
- [28] Faure H., Lemieux Ch.: Generalized Halton sequences in 2008: A comparative study. *ACM Transactions on Modeling and Comp. Sim*, Vol. **19**, Issue 4, 1-31, (2009), <https://doi.org/10.1145/1596519.1596520>
- [29] Faure, H., Discrepance de suites associées a un système de numération (en dimension s), *Acta Arithmetica*, Vol. **41**, 337-351, (1982)
- [30] Faure, H., Good permutations for extreme discrepancy, *Journal of Number Theory*, Vol. **42**, 47-56, (1992)
- [31] Fischer, H., A History of the Central Limit Theorem: From Classical to Modern Probability Theory, *Sources and Studies in the History of Mathematics and Physical Sciences*. New York: Springer, (2011), <http://dx.doi.org/10.1007/978-0-387-87857-7>
- [32] Fishman, G.S., Multiplicative congruential random number generators, *Math. Comp.*, **54**, 331-344, (1990)
- [33] Golub, G., Loan, Ch. F. Van, *Matrix computation*, 4th edition, The Johns Hopkins University Press, (2013), ISBN: 13-978-1-4214-0794-4, <https://math.ecnu.edu.cn/~jypan/Teaching/books/2013%20Matrix%20Computations%204th.pdf>
- [34] Gilboa, G., *Nonlinear Eigenproblems in Image Processing and Computer Vision*, Publisher: Springer Cham, ed.number 1, (2018), eBook ISBN: 978-3-319-75847-3
- [35] Gurova, S.-M., Atanassov, E., Karaivanova, A., A Resolvent Quasi-Monte Carlo Method for Estimating the Minimum Eigenvalues Using the Error Balancing, In: Lirkov, I., Margenov, S. (eds.), *Large-Scale Scientific Computing, LSSC 2023*, *Lecture Notes in Computer Science*, vol. **13952**, 394-403, (2024)
- [36] Gurova, S.-M., Karaivanova, A., Quasi-Monte Carlo Algorithms for Eigenvalue Problems, *Fourth International Conference Numerical Methods for Scientific Computations and Advanced Applications, NMSCAA'24*, 17-21, (2024) (short communication) <https://parallel.bas.bg/~elilkova/nmscaa24.pdf>
- [37] Gurova, S.-M., Gurov, T., Karaivanova, A., On the Stochastic Power Algorithms for Estimating the Maximum Eigenvalue of Symmetric Matrices, *Contemp. Math.*, Universal Wiser Publisher Pte. Ltd., **6**(5): 7223-7249, (2025) <https://doi.org/10.37256/cm.6520257781>
- [38] Gurova, S.-M., Gurov, T., Karaivanova, A., Eigenvalue Estimation in Portfolio Risk: The Role of Skipping and Leaping in Sobol and Halton Sequences, In: Lirkov, I., Margenov, S. (eds), *Large-Scale Scientific Computing, LSSC'25*, *LNCS*, **16061**, (accepted for publication)
- [39] Halton, J., On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numerische Mathematik*, **2**, 84-90, (1960), <https://doi.org/10.1007/BF01386213>

- [40] Halton, J., Smith, GB, Algorithm 247: Radical-Inverse Quasi-Random Point Sequence, Communications of the ACM, Vol. **7**, 701-702, (1964)
- [41] Hammersley, J.M., Handscomb, D.C., Random, Pseudorandom, and Quasirandom Numbers. In: Monte Carlo Methods. Monographs on Applied Probability and Statistics. Springer, Dordrecht, (1964), https://doi.org/10.1007/978-94-009-5819-7_3
- [42] Higham, N. J., Computing the nearest correlation matrix—a problem from finance, IMA Journal of Numerical Analysis, **22**(3): 329–343, (2002), <https://doi.org/10.1093/imanum/22.3.329>
- [43] Isaacson, E., Keller, H. B., Analysis of Numerical Methods, Dover Publications, Inc., N.Y., Wiley, (1996), ISBN: 0-486-68029-0
- [44] Ivanovska, S., Karaivanova, A., Manev, N., Numerical Integration Using Sequences Generating Permutations. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds) Large-Scale Scientific Computing. LSSC 2011. Lecture Notes in Computer Science, vol. **7116**, Springer, Berlin, Heidelberg, 455-463, (2011), https://doi.org/10.1007/978-3-642-29843-1_51
- [45] Joe, S., Sloan, I., Implementation of a lattice method for numerical multiple integration, ACM Transactions on Mathematical Software (TOMS), Vol. **19**, Issue 4, 523–545, (1993), <https://doi.org/10.1145/168173.168425>
- [46] Joe S., Kuo F.Y.: Constructing Sobol Sequences with Better Two-Dimensional Projections, SIAM Journal on Scientific Computing, **30**(5): 2635-2654, (2008), <https://doi.org/10.1137/070709359>
- [47] Johnstone, I. M., On the distribution of the largest eigenvalue in principal components analysis, Ann. Statist. **29**(2), 295-327, (2001), <https://doi.org/10.1214/aos/1009210544>
- [48] Karaivanova, A., Quasi-Monte Carlo Methods for Some Linear Algebra Problems, Convergence and complexity, Serdica J. Computing, **4**, 57-72, (2010)
- [49] Kolda, T. and Bader, B., Tensor Decompositions and Applications, SIAM Review, **51**(3), 455-500, (2009) <https://doi.org/10.1137/07070111X>
- [50] Kroese, D.P., Taimre, T., Botev, Z. I., Handbook of Monte Carlo Methods, John Wiley & Sons Inc., New York, N.Y., (2011), ISBN:9780470177938
- [51] Laloux, L., Cizeau, P., Bouchaud, J.-P., Potters, M., Noise Dressing of Financial Correlation Matrices, Phys. Rev. Lett., **83**(7), 1467-1470, American Physical Society, (1999), <https://doi.org/10.1103/PhysRevLett.83.1467>
- [52] Lemieux, C., Monte Carlo and Quasi-Monte Carlo Sampling. Springer, (2009), <https://doi.org/10.1007/978-0-387-78165-5>
- [53] L'Ecuyer, P. Uniform random number generation. Ann Oper Res, **53**, 77–120, (1994), <https://doi.org/10.1007/BF02136827>
- [54] L'Ecuyer, P., Efficient and portable combined random number generators, Communications of the ACM, vol. **31**, Issue 6, 742-751, (1988), <https://doi.org/10.1145/62959.62969>
- [55] Lobo, P.D.C., Mikhailov, M., Ozisik, M.N., On the complex eigenvalues of Luikov system of equations, Drying Technology, **5**(2), 273-286, (1987), <https://doi.org/10.1080/07373938708916540>
- [56] Loperfido, N., Skewness-based projection pursuit: A computational approach, Computational Statistics and Data Analysis, **120**, 42-57, (2018), <https://doi.org/10.1016/j.csda.2017.11.001>
- [57] Macdonald, R.R., Correlation and Covariance Matrices. In Wiley StatsRef: Statistics Reference Online (eds. N. Balakrishnan, T. Colton, B. Everitt, W. Piegorsch, F. Ruggeri and J.L. Teugels), (2014), <https://doi.org/10.1002/9781118445112.stat06481>
- [58] Mascagni, M., Karaivanova, A., A Parallel Quasi-Monte Carlo Method for Solving Systems of Linear Equations, P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS **2330**, Springer-Verlag Berlin Heidelberg, 598-608, (2002)
- [59] Mascagni, M., Karaivanova, A., A Parallel Quasi-Monte Carlo Method for Computing Extremal Eigenvalues, K.-T. Fang et al. (eds.), Monte Carlo and Quasi-Monte Carlo Methods, Springer-Verlag Berlin Heidelberg, 369 – 380, (2000)

- [60] Mascagni, M., Karaivanova, A., Matrix Computations Using Quasirandom Sequences, Lecture Notes in Computer Science, (Wulkov, Yalamov, Wasniewsky eds.), Vol. **1988**, Springer, 552-559, (2001)
- [61] Mascagni, M., Karaivanova, A., A Monte Carlo Approach for Finding More than One Eigenpair. In: Dimov, I., Lirkov, I., Margenov, S., Zlatev, Z. (eds) Numerical Methods and Applications - NMA 2002, Lecture Notes in Computer Science, vol.**2542**, Springer, 123-131, (2003), https://doi.org/10.1007/3-540-36487-0_13
- [62] Matsumoto, M., Nishimura, T., Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-Random Number Generator, ACM Transactions on Modeling and Computer Simulation, vol. 8(1), 3-30, (1998), <https://doi.org/10.1145/272991.272995>
- [63] Mehrdoust, F., Algorithm for Estimating the Condition Number of Matrices, Math. Reports, **15(65)**(3), 203-210, (2013)
- [64] Muminov M.I., Rasulov T.H.: Embedded Eigenvalues of a Hamiltonian in Bosonic Fock Space. Commun. Math. Anal., **17**(1), 1 - 22, (2014)
- [65] Mushtaq, A., Noreen, A., Olaussen, K., Numerical Solutions of Quantum Mechanical Eigenvalue Problems, Front. Phys., **8**:390, (2020), <https://doi.org/10.3389/fphy.2020.00390>
- [66] Niederreiter, H., Random Number Generation and Quasi-Monte Carlo Methods, SIAM (1992), <https://doi.org/10.1137/1.9781611970081>
- [67] Owen, A., Randomly Permuted (t,m,s)-Nets and (t, s)-Sequences. In: Niederreiter, H., Shiue, P.J.S. (eds.) Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing. Lecture Notes in Statistics, vol. **106**, Springer, New York, 299-317, (1995), https://doi.org/10.1007/978-1-4612-2552-2_19
- [68] Owen, A., Monte Carlo variance of scrambled net quadrature, SIAM Journal on Numerical Analysis, Vol. **34**, No. 5, 1884-1910, (1997), <https://www.jstor.org/stable/2952022>
- [69] Owen, A., Variance with alternative scramblings of digital nets, ACM Transactions on Modeling and Computer Simulations, **13**(4), 363-378, (2003)
- [70] Padányi, V., Herendi, T., Generalized Middle-Square Method, Annales Mathematicae et Informaticae, **56**, 95-108, (2022), <https://doi.org/10.33039/ami.2022.12.003>
- [71] Park, Stephen K.; Miller, Keith W., Random Number Generators: Good Ones Are Hard To Find, Communications of the ACM, vol. **31**(10), 1192-1201, (1988), doi:10.1145/63039.63042
- [72] Rometsch, M., Quasi-Monte Carlo Methods in Finance with Application to Optimal Asset Allocation, Druck Diplomica Verlag GmbH, Hamburg (2008).
- [73] Rotter, I., A non-Hermitian Hamilton operator and the physics of open quantum systems, J. Phys. A: Math. Theor., Vol. **42**, No. 15, 153001, (2009), <https://doi.org/10.1088/1751-8113/42/15/153001>
- [74] Sobol I., Shukhman B., Integration with quasirandom sequences: Numerical experience., Internat. J. Modern Phys. C, **6**(2), 263-275 (1995)
- [75] Sobol I., Asotsky D., Kreinin A., Kucherenko S., Construction and Comparison of High-Dimensional Sobol' Generator, WILMOTT magazine, 64-79, (2012)
- [76] Sobol I., Uniformly distributed sequences with an additional property of uniformity, USSR Computational Mathematics and Mathematical Physics, **16**(5), 236-242, (1976), [https://doi.org/10.1016/0041-5553\(76\)90154-3](https://doi.org/10.1016/0041-5553(76)90154-3)
- [77] Spiegel, M. R., Theory and Problems of Probability and Statistics, 2nd ed. New York: McGraw-Hill, (1992).
- [78] Straßburg J, Alexandrov, V., Enhancing Monte Carlo preconditioning methods for matrix computations, Procedia Computer Science, **29**, 1580-1589, (2014), <https://doi.org/10.1016/j.procs.2014.05.143>
- [79] Straßburg, J., On hybrid and resilient Monte Carlo methods for linear algebra problems, Ph.D. dissertation, The University of Reading, (2014)
- [80] Sturmfels, B., Tensors and Their Eigenvectors, Notices of the AMS, **63**(6), 604-606, (2016) <http://dx.doi.org/10.1090/noti1389>

- [81] Sugita, H., Monte Carlo Method, Random Number, and Pseudorandom Number, MSJ Memoirs, **25**: 133, (2011), <https://doi.org/10.2969/msjmemoirs/025010000>
- [82] Taft K., Vajargah, F., MC method for solving system of linear algebraic equations with minimum markov chains, International Conference PDPTA'2000, Las Vegas, (2000)
- [83] Vajargah F., et al., Parallel Monte Carlo algorithms for matrix computations, Ph.D. dissertation, The University of Reading, (2001)
- [84] Vajargah, F., Hassanzadeh, Z., Improvements on the hybrid Monte Carlo algorithms for matrix computations, Sadhana, Indian Academy of Science, **44**(1), <https://doi.org/10.1007/s12046-018-0983-y>
- [85] Van der Corput, J., Verteilungsfunktionen (Erste Mitteilung), Proceedings of the Koninklijke Akademie van Wetenschappen te Amsterdam (in German), **38**: 813–821, (1935), Zbl 0012.34705
- [86] Wang, X., Hickernell, F., Randomized Halton sequences, Mathematical and Computer Modelling, **32**(7–8), 887–899, (2000) [https://doi.org/10.1016/S0895-7177\(00\)00178-3](https://doi.org/10.1016/S0895-7177(00)00178-3)
- [87] Боянов, Б., Лекции по Числени методи, изд. Дарба София, (2008), ISBN: 954-90126-1
- [88] Димитров, Б., Янев, Н., Вероятности и статистика, Софтех, София, (2007)
- [89] Караиванова, А., Стохастични числени методи и симулации, изд. Деметра ЕООД, (2012), ISBN: 978-954-9526-78-3
- [90] Сендов, Б., Попов, В., Числени методи, Университетско издание „Св. Климент Охридски“, първа част, (1996), ISSN: 954-07-0994-6
- [91] Стоянов, Й., Миразчийски, И., Игнатов, Ц., Танушев, М., Ръководство за упражнения по теория на вероятностите. Издателство „Наука и изкуство“, София, (1976).
- [92] Соболев, И. М., Численные методы Монте-Карло, изд. Москва, Наука, (1973)
- [93] MathWorks: haltonset - Create a Halton point set. <https://www.mathworks.com/help/stats/haltonset.html>, (Last accessed: 2026/01/11)
- [94] MathWorks: sobolset - Create a Sobol point set. <https://www.mathworks.com/help/stats/sobolset.html>, (Last accessed: 2026/01/11)

Abstracts of Dissertations

Number 1, 2026

INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES
BULGARIAN ACADEMY OF SCIENCES

БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ

ИНСТИТУТ ПО ИНФОРМАЦИОННИ И КОМУНИКАЦИОННИ ТЕХНОЛОГИИ

Брой 1, 2026

Автореферати на дисертации