Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Computer Science and Information Theory

Simulation of quantum walks on a classical computer

Scientific Students' Association Report

Author:

Viktória Nemkin

Advisor:

dr. Katalin Friedl

2021

Contents

Notes

Kivonat

Az utóbbi években egyre nagyobb figyelem összpontosul a kvantuminformatikára. Olyan globális vállalatok, mint az IBM, a Google, a Microsoft és az Amazon jelentős összegeket fektetnek kutatásba, hardver- és szoftverfejlesztésekbe ezen a területen, míg az Európai Unió és Magyarország számos olyan programot indított, melyek a kvantuminformatikai kutatások fellendítését célozzák meg.

A jelenlegi kvantumszámítógépekben elérhető qubitek (kvantumbitek) mennyisége még csekély, de sokan úgy vélik hogy a jövőben ez a szám növekedni fog. Az első olyan, a gyakorlatban is hasznos kvantumalgoritmusok, amiket ezeken a processzorokon futtatni tudunk majd, várhatóan azok lesznek, melyek takarékosan bánnak a rendelkezésre álló qubitekkel. A kvantumséta, mely a klasszikus véletlen bolyongás általánosítása kvantumos esetben, pontosan ilyen algoritmus. Mivel a qubit igénye a gráf csúcsszámában logaritmikus, így ez egy érdekes módszernek ígérkezik akár a közeljövőre nézve is. A kvantumséták erejét bizonyítja, hogy a Grover keresés (mely több kvantumalgoritmus alapját képezi) is értelmezhető ezek egy speciális fajtájának.

Dolgozatomban leírom a kvantumséták matematikai alapjait, részletezve a megvalósítás szempontjából fontos pontokat, melyek a szakirodalomban kisebb hangsúllyal szerepelnek. Ismertetem az általam írt szimulátor program architekturális felépítését és működését, továbbá a futtatott szimulációim eredményeit.

A szimulátor programot Python 3 nyelven írtam, a Stratégia tervezési minta alapján. A szakirodalomban tipikusan használt gráfokat beépítetten támogatja, melyek kombinálásával tetszőlegesen bonyolult reguláris gráf előállítható, ez az előállítás képezi a kvantumséta alapját is. A szoftver reguláris gráfokon történő kvantum és klasszikus séták szimulációját teszi lehetővé, az eredményekről pedig egy részletes report fájlt generál. A kvantumos séták esetében a séta tulajdonságai a valószínűségek generálásához felhasznált érmétől is függnek, melyet többféleképpen is lehet definiálni. A program beépítetten tartalmazza az Hadamard-, a Grover- és a Fourier-érméket, de felépítéséből adódóan könnyen bővíthető tetszőleges érmével is.

Szimulációim segítségével összehasonlítottam a klasszikus és a kvantum séták viselkedését, továbbá kimutattam az elméleti szakirodalom alapján elvárt kvantumos jellegzetességeket, az Hadamard-séta ballisztikus természetét és a kvantumséták ciklikus tulajdonságát.

Abstract

In recent years, there has been an increasing focus on quantum informatics. Influential global companies such as IBM, Google, Microsoft, and Amazon have invested significant amounts into studying and developing hardware and software for this sector, while the European Union and Hungary have launched several programs to accelerate quantum research.

Current technology is yet to produce a significant number of qubits (quantum bits) in a quantum processor, but many believe the amount will increase over the years. The first practical quantum algorithms to be run on these processors are likely to be the ones that use qubits sparingly. Quantum walking, the generalized version of classical random walking, is exactly this kind of algorithm. The number of qubits required to run a quantum walk on a graph is logarithmic in the number of vertices, making it a promising technique for the near future. Furthermore, Grover's search algorithm (a basis for many quantum algorithms) can be viewed as a special case of quantum walks, which illustrates the potential power of this method.

In my dissertation, I present the mathematical framework for quantum walks, detailing the points critical for implementation, which are given less emphasis in the literature. I describe the architecture and capabilities of the simulator program I have written and the conclusions of the simulations I have run.

I developed the software using Python 3, based on the Strategy design pattern. It supports graphs commonly found in the literature while also providing a method for combining them, facilitating experimentation on several kinds of regular graphs. This composition is also the foundation of the quantum walk. It can simulate classical and quantum walks on the same graphs and produce a report file detailing the results. In the quantum case, the characteristics of the walk are also dependent on the type of coin used to generate the probabilities, which can be defined in several ways. The program includes the Hadamard, Grover, and Fourier coins and can easily be extended with others.

Running several simulations, I compared the behavior of classical and quantum walks and demonstrated the quantum characteristics expected from the theoretical literature, the ballistic nature of the Hadamard walk, and the cyclic property of quantum walks.

Chapter 1

Introduction

The following chapter contains the analysis of the TDK dissertation topic, historic background, motivation for the task: why it is important, relevant, useful, etc, the solutions currently available and then compare and contrast with the student's solution. Then, it concludes with the structure of the dissertation, describing the remaining chapters.

The following sections are based on:

Feng Xia, Senior Member, IEEE, Jiaying Liu, Hansong Nie, Yonghao Fu, Liangtian Wan, Member, IEEE, Xiangjie Kong, Senior Member, IEEE - Random Walks: A Review of Algorithms and Applications, from IEEE Transactions on emerging topics in computational intelligence, on May 2019.

1.1 Classical random walks on graphs

Classical random walks are used to describe stochastic processes and many real life sciences rely on these methods. For example stock price movement prediction in finance, natural language processing algorithms, describing brownian motion in engineering physics, and various applications in biology and bioinformatics, such as DNA evolution models, population dynamics, modeling disease outbreaks, epidemic modeling ...etc. There are also algorithms, where stochastic processes are not directly present, but introduced as a way to deal with large quantities of data. Most notably, Google's Page Rank algorithm utilizes classical random walks on the large sized graph of the internet, to score documents on how good of an information source they are or how well they link to other information sources. Other algorithms include Markov chain Monte Carlo, for sampling from a probability distribution, difficult to directly model. Instead, they construct a stohastic process which's equilibrium distribution is the desired one and sample this by repeatedly recording states from the chain. The Metropolis Hashtings algorithm is similar to this, and is used to approximate the distribution or compute an integral (e.g expected value).

Itt az lenne a célom hogy eladjam a véletlen sétákat, hogy ezek nagyon fontos algoritmusok. A Google PageRank egy nagyon jó példa.

Random walks are effective in link prediction and recommendation system, computer vision, semi-supervised learning, network embedding, and complex social network analysis. There are also some literature illustrating the applications of random walks on graphs, text analysis, science of science, and knowledge discovery.

In conclusion, there are many important applications for classical random walks.

1.2 From classical to quantum

The scalable quantum computer is a topical issue so that approaches of quantum computation are popular topics nowadays. Quantum walks are the corresponding part of classical random walks in quantum mechanics. The main difference between them is that quantum walks don't converge to some limiting distributions. Due to the quantum interference, quantum walks can spread significantly faster or slower than classical random walks.

There are many properties of quantum walks that make them differ from their classical counterparts. While in classical walks, in N steps the walks reaches a distance of $O(\sqrt{N})$, in quantum, it is a lot faster, $O(N)$. Quantum walks have inherent interferences, where some amplitudes can amplify, while others can be destrutive to each other and diminish. This makes them behave very differently from classical random walks and a good reason to study them.

Quantum walks are often used to accelerate classical algorithms. It can be used to decision trees, search problems, and element distinctness.

There are many algorithms based on Quantum Walks. Two types of models are present: continuous time and discrete time walks. Quantum decision tree algorithm is one of the continuous time quantum walk based algorithm. In this algorithm, you systematically explore the decision tree as a graph. The decision tree nodes are quantum states in a Hilbert space.

Based on Quantum Walks

- Quantum Decision Tree
- Quantum PageRank
- Grover Search Algorithm

1.3 Applications

- Collaborative Filtering
- Recommender System
- Link Prediction
- Computer Vision
- Semi-supervised
- Learning
- Network Embedding
- Element Distinctness

1.4 From classical to quantum walks

Classical random walks describe a stohastic process

Tool in mathematics for modeling discrete time and discrete space stochastic processes.

Classical random walks on graphs can be defined using Markov-chains. Markov-chains are well explained in Leo Breiman's book on Probability[1].

Definition 1. Markov-chain A Markov-chain is a process consisting of random variables

1.5 Tenzorszorzás

Definition 2. Tenzorszorzat Az $r \times s$ méretű A és $t \times u$ méretű B mátrixok $rt \times su$ méretű $A \otimes B$ tenzorszorzata a következőképpen definiált:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1s} \\ a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1} & a_{r2} & \ddots & a_{rs} \end{pmatrix} \text{ és } B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1u} \\ b_{21} & b_{22} & \dots & b_{2u} \\ \vdots & \vdots & \ddots & \vdots \\ b_{t1} & b_{t2} & \ddots & b_{tu} \end{pmatrix} \text{ esetén}$$

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1s}B \\ a_{21}B & a_{22}B & \dots & a_{2s}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1}B & a_{r2}B & \ddots & a_{rs}B \end{pmatrix}$$

1.5.1 Azonosságai

Theorem 1 (Asszociativitás).

$$(A \otimes B) \otimes C = A \otimes (B \otimes C) \tag{1.1}$$

Theorem 2 (Disztibutivitás a mátrixszorzás felett). Ha A és C, illetve B és D kompatibilisek, akkor:

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{1.2}$$

Theorem 3 (Lemma: Egységmátrixok beolvadása).

$$(A_n \otimes I_m)(I_n \otimes B_m) = A_n \otimes B_m \tag{1.3}$$

1.6 Unitér mátrixok

Ezt a Wikipedia-ról szedtem, hivatkozni: https://en.wikipedia.org/wiki/Unitary_matrix

Definition 3. Unitary matrix A complex square matrix U is unitary if its conjugate transpose U* is also its inverse, as in:

$$U * U = UU* = I \tag{1.4}$$

1.7 Kvantumséta a rácson

A következőkben a rácson, mint 4-reguláris gráfon vett bolyongást fogjuk részletesen megvizsgálni.

Legyen a rács $N = n \times n$ -es.

Pozíció vektor: $|P\rangle = (p_0, \dots, p_N)$, ahol $p_i \in \mathbb{C}$ ahol $0 \leq i < N$, $\forall i$-re és $\sum_{i=0}^{N} |p_i|^2 = 1$.

$$|P_0\rangle = (0, \ldots, 0, 1, 0, \ldots, 0) \tag{1.5}$$

$$C = (\frac{1}{\sqrt{2}}, \frac{i}{\sqrt{2}}) \otimes (\frac{1}{\sqrt{2}}, \frac{i}{\sqrt{2}}) \tag{1.6}$$

$$S_{\text{left}} = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & \end{pmatrix} \tag{1.7}$$

Definition 4 (Kronecker delta). Olyan kétváltozós függvény ami akkor 1, ha a két változója egyenlő, egyébként 0.

$$\delta_{ij} = \begin{cases} 0 & \text{ha } i \neq j \\ 1 & \text{ha } i = j \end{cases} \tag{1.8}$$

▪

Definition 5 (Upper cyclic shift matrix). Az U_k mátrix egy olyan $(k \times k)$-s mátrix, aminek a felső mellékátlóján és a bal alsó sarokban 1-esek vannak, máshol 0-k. Az index számozása moduló k történik. Ennek az i. sor, j. oszlopában:

$$U_k[i, j] = \delta_{(i+1),j} \tag{1.9}$$

▪

Definition 6 (Lower cyclic shift matrix). Az L_k mátrix egy olyan $(k \times k)$-s mátrix, aminek az alsó mellékátlóján és a jobb felső sarokban 1-esek vannak, máshol 0-k. Az index számozása moduló k történik. Ennek az i. sor, j. oszlopában:

$$L_k[i, j] = \delta_{i,j+1} \tag{1.10}$$

▪

$$L = U^T \tag{1.11}$$

- Rács = két körnek a tenzorszorzata.
- Fel/le = mindig 1 sorral felette/alatta lévő (modulóval) van összekötve = n-el van elcsúsztatva.
- Függőlegesen számozva ugyanígy csak fordított elnevezések.
- Jobbra-balra = 1 távolság, kivéve sor végén, ahol a sor elejére ugrik = kis körök.
- Tenzorszorzás tulajdonságait előre kiszedni, műveletek, használatkor hivatkozni.
- Többsoros képleteknek csak 1 száma legyen.

1.8 Példa

$$n = 3 \tag{1.12}$$

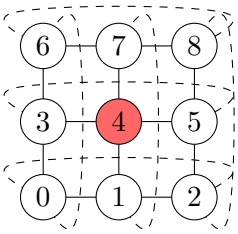

Figure 1.1: n=3-ös rács

$$N = n^2 = 9 \tag{1.13}$$

$$S_{\text{left}} = I_3 \otimes U_3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{1.14}$$

$$S_{\text{right}} = I_3 \otimes L_3 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{1.15}$$

$$S_{\text{down}} = U_3^3 = U_3 \otimes I_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{1.16}$$

$$S_{\text{up}} = L_3^3 = L_3 \otimes I_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{1.17}$$

$$X_{\text{head}} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{1.18}$$

$$X_{\text{tail}} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{1.19}$$

$$S = (S_{\text{up}} S_{\text{left}}) \otimes (X_{\text{head}} \otimes X_{\text{head}}) + \tag{1.20}$$
$$(S_{\text{up}} S_{\text{right}}) \otimes (X_{\text{head}} \otimes X_{\text{tail}}) + \tag{1.21}$$
$$(S_{\text{down}} S_{\text{left}}) \otimes (X_{\text{tail}} \otimes X_{\text{head}}) + \tag{1.22}$$
$$(S_{\text{down}} S_{\text{right}}) \otimes (X_{\text{tail}} \otimes X_{\text{tail}}) \tag{1.23}$$

$$C_4 = H^{\otimes 2} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \tag{1.24}$$

$$U = S(I_9 \otimes C_4) \tag{1.25}$$

$$U = ((S_{\text{up}}S_{\text{left}}) \otimes (X_{\text{head}} \otimes X_{\text{head}}) + \tag{1.26}$$
$$(S_{\text{up}}S_{\text{right}}) \otimes (X_{\text{head}} \otimes X_{\text{tail}}) + \tag{1.27}$$
$$(S_{\text{down}}S_{\text{left}}) \otimes (X_{\text{tail}} \otimes X_{\text{head}}) + \tag{1.28}$$
$$(S_{\text{down}}S_{\text{right}}) \otimes (X_{\text{tail}} \otimes X_{\text{tail}}))(I_9 \otimes C_4) \tag{1.29}$$

Definition 7 (Tenzorszorzás azonosság: mátrix szorzással disztibutív). Bal oldalt 9*9-es, jobb oldalt 4*4-es mátrixok vannak.

$$U = (S_{\text{up}}S_{\text{left}}) \otimes ((X_{\text{head}} \otimes X_{\text{head}})C_4) + \tag{1.30}$$
$$(S_{\text{up}}S_{\text{right}}) \otimes ((X_{\text{head}} \otimes X_{\text{tail}})C_4) + \tag{1.31}$$
$$(S_{\text{down}}S_{\text{left}}) \otimes ((X_{\text{tail}} \otimes X_{\text{head}})C_4) + \tag{1.32}$$
$$(S_{\text{down}}S_{\text{right}}) \otimes ((X_{\text{tail}} \otimes X_{\text{tail}})C_4) \tag{1.33}$$

$$C_4 = C_2 \otimes C_2 \tag{1.34}$$

$$U = (S_{\text{up}}S_{\text{left}}) \otimes ((X_{\text{head}} \otimes X_{\text{head}})(C_2 \otimes C_2)) + \tag{1.35}$$
$$(S_{\text{up}}S_{\text{right}}) \otimes ((X_{\text{head}} \otimes X_{\text{tail}})(C_2 \otimes C_2)) + \tag{1.36}$$
$$(S_{\text{down}}S_{\text{left}}) \otimes ((X_{\text{tail}} \otimes X_{\text{head}})(C_2 \otimes C_2)) + \tag{1.37}$$
$$(S_{\text{down}}S_{\text{right}}) \otimes ((X_{\text{tail}} \otimes X_{\text{tail}})(C_2 \otimes C_2)) \tag{1.38}$$

Definition 8 (Tenzorszorzás azonosság: mátrix szorzással disztibutív).

$$U = (S_{\text{up}}S_{\text{left}}) \otimes (((X_{\text{head}}C_2) \otimes (X_{\text{head}}C_2))) + \tag{1.39}$$
$$(S_{\text{up}}S_{\text{right}}) \otimes (((X_{\text{head}}C_2) \otimes (X_{\text{tail}}C_2))) + \tag{1.40}$$
$$(S_{\text{down}}S_{\text{left}}) \otimes (((X_{\text{tail}}C_2) \otimes (X_{\text{head}}C_2))) + \tag{1.41}$$
$$(S_{\text{down}}S_{\text{right}}) \otimes (((X_{\text{tail}}C_2) \otimes (X_{\text{tail}}C_2))) \tag{1.42}$$

Definition 9 (Tenzorszorzás azonosság: asszociatív).

$$U = (S_{\text{up}}S_{\text{left}}) \otimes (X_{\text{head}}C_2) \otimes (X_{\text{head}}C_2) + \tag{1.43}$$
$$(S_{\text{up}}S_{\text{right}}) \otimes (X_{\text{head}}C_2) \otimes (X_{\text{tail}}C_2) + \tag{1.44}$$
$$(S_{\text{down}}S_{\text{left}}) \otimes (X_{\text{tail}}C_2) \otimes (X_{\text{head}}C_2) + \tag{1.45}$$
$$(S_{\text{down}}S_{\text{right}}) \otimes (X_{\text{tail}}C_2) \otimes (X_{\text{tail}}C_2) \tag{1.46}$$

Definition 10 (Tenzorszorzás Lemma?).

$$S = ((S_{\text{up}} \otimes X_{\text{head}} \otimes I) + (S_{\text{down}} \otimes X_{\text{tail}} \otimes I))((S_{\text{left}} \otimes I \otimes X_{\text{head}}) + (S_{\text{right}} \otimes I \otimes X_{\text{tail}})) \tag{1.47}$$

1.9 Tervezési megfontolások

Az első fázisban a szomszédossági mátrixot generáltam le teljesen. Ez nagyon sok memóriát használt, az unitér mátrix meg mégtöbbet, nagyon rosszul skálázódott. Lecseréltem spare mátrixokra, az már egy fokkal jobb.

Mérés összehasonlítás: Mátrix, Spare mátrix, Orákulum grafikon.

Ezután kipróbáltam a spare mátrixokat, de azok is picit sokak voltak, ugyan a gráfban sok a 0, de azért így is nagyon skálázódott.

A legjobb megoldás az úgynevezett gráf orákulum volt. A gráf orákulum lényege, hogy nem tároljuk a szomszédossági mátrixot, helyette biztosítunk egy függvényt, mely on-the-fly a gráf adott csúcsindexét odaadva neki visszaadja hogy annak mik a szomszédai. A coin kis méretű, azt lehet tárolni (max fokszám * max fokszám), ezért azzal nem bajlódtam, bár éppen lehetne. A

Acknowledgements

Ez nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

Bibliography

[1] Leo Breiman. *Probability*. Number 7 in Classics in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1992. ISBN 9780898712964.

[2] Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Kar. Diplomaterv portál (2011. február 26.). http://diplomaterv.vik.bme.hu/.

[3] James C. Candy. Decimation for sigma delta modulation. 34(1):72–76, 01 1986. DOI: 10.1109/TCOM.1986.1096432.

[4] P. Erdős, D. Miklós, V.T. Sós, T. Szőnyi, and Bolyai János Matematikai Társulat. *Combinatorics, Paul Erdős is Eighty*. Number v. 2 in Bolyai Society mathematical studies. János Bolyai Mathematical Society, 1993. ISBN 9789638022752.

[5] Paul Erdős, D. Miklós, Vera T. Sós, T. Szőnyi, and Bolyai János Matematikai Társulat, editors. *Combinatorics, Paul Erdős is eighty*. Number 1-[2] in Bolyai Society mathematical studies. János Bolyai Mathematical Society, Budapest, Hungary, 1993. ISBN 9789638022738 9789638022745 9789638022752.

[6] Gábor Jeney. Hogyan néz ki egy igényes dokumentum? Néhány szóban az alapvető tipográfiai szabályokról, 2014. http://www.mcl.hu/~jeneyg/kinezet.pdf.

[7] Peter Kiss. Adaptive digital compensation of analog circuit imperfections for cascaded delta-sigma analog-to-digital converters, 04 2000.

[8] Wai L. Lee and Charles G. Sodini. A topology for higher order interpolative coders. In *Proc. of the IEEE International Symposium on Circuits and Systems*, pages 459–462, 05 4–7 1987.

[9] Kabódi László. Kvantumalgoritmusok véletlen bolyongással. 2014.

[10] Alexey Mkrtychev. Models for the logic of proofs. In Sergei Adian and Anil Nerode, editors, *Logical Foundations of Computer Science*, volume 1234 of *Lecture Notes in Computer Science*, pages 266–275. Springer Berlin Heidelberg, 1997. ISBN 978-3-540-63045-6. DOI: 10.1007/3-540-63045-7_27. URL http://dx.doi.org/10.1007/3-540-63045-7_27.

[11] Richard Schreier. *The Delta-Sigma Toolbox v5.2*. Oregon State University, 01 2000. http://www.mathworks.com/matlabcentral/fileexchange/.

[12] Nemkin Viktória. Kvantum gráfbolyongások. 2021.

[13] Ferenc Wettl, Gyula Mayer, and Péter Szabó. *LaTeX kézikönyv*. Panem Könyvkiadó, 2004.

Appendix

A.1 Első függelék

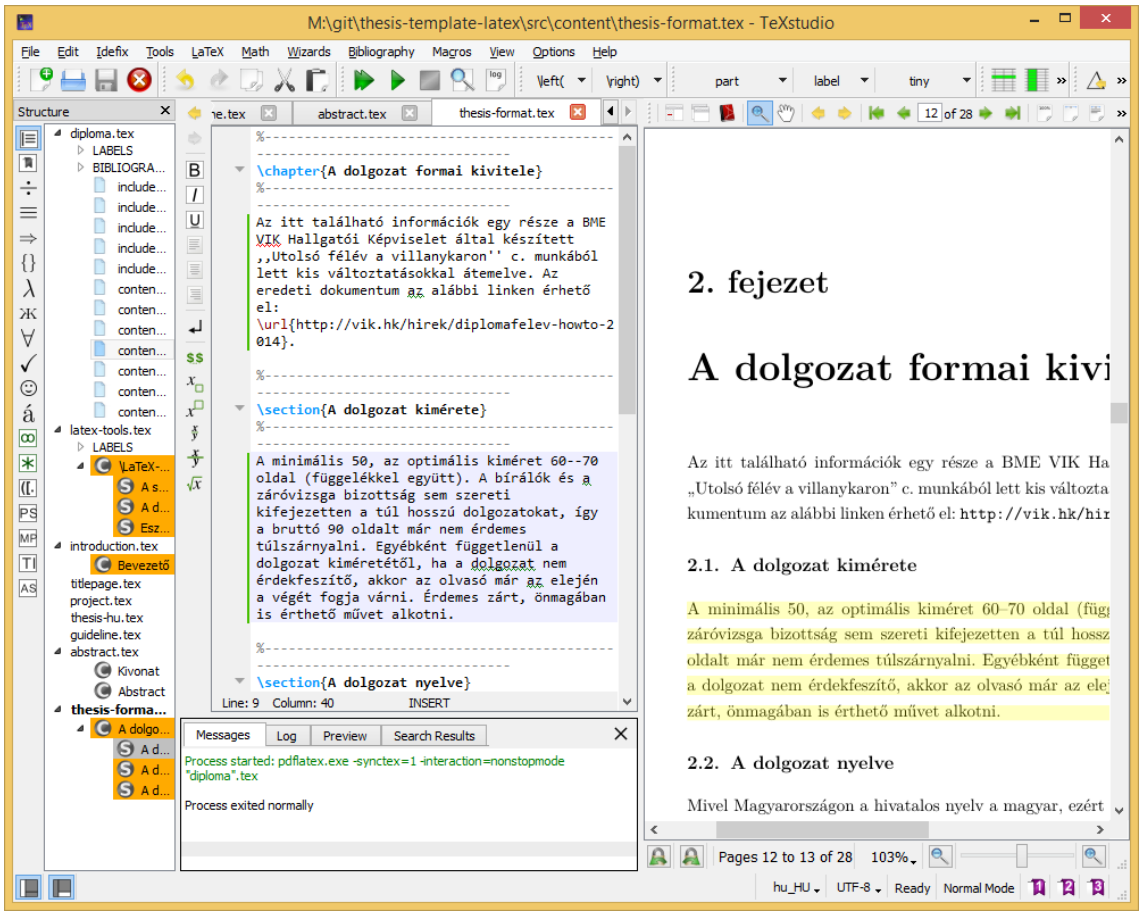

Figure A.1.1: A TeXstudio LaTeX-szerkesztő.

A.2 Második függelék

A Pitagorasz-tételből levezetve

$$c^2 = a^2 + b^2 = 42. \tag{A.2.1}$$

A Faraday-indukciós törvényből levezetve

$$\operatorname{rot} E = -\frac{dB}{dt} \qquad \longrightarrow \qquad U_i = \oint\limits_{\mathbf{L}} \mathbf{Edl} = -\frac{d}{dt} \int\limits_{A} \mathbf{Bda} = 42. \tag{A.2.2}$$