



**ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΤΙΤΛΟΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
B. Riemann**

**ΕΠΙΒΛΕΠΩΝ: K.F.GAUSS**



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης  
Σχολή Θετικών Επιστημών  
Τμήμα Πληροφορικής

**Copyright ©All rights reserved Riemann, 2020.**

Με την επιφύλαξη παντός δικαιώματος. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.

### **Υπεύθυνη Δήλωση**

Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας, και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Πληροφορικής του Αριστοτέλειου Πανεπιστημίου Θεσσαλονίκης.

(Υπογραφή) .....

**Riemann**

### **Abstract**

Σκοπός της παρούσας πτυχιακής εργασίας είναι η παρουσίαση και υλοποίηση του αλγορίθμου του Gauss Elimination modulo 2 ...

**Λέξεις Κλειδιά.** Γραμμική άλγεβρα, Γραμμικά Συστήματα, ..., CUDA, C

# ABSTRACT

The purpose of this thesis is to ....

**Key Words.** Linear Algebra, Linear Systems, ..., CUDA, C

# Contents

<b>1</b>	<b>Turing Machine</b>	<b>5</b>
1.0.1	Enigma . . . . .	5
<b>2</b>	<b>Gauss Reduction</b>	<b>6</b>
<b>3</b>	<b>Background</b>	<b>6</b>
3.1	Linear Systems . . . . .	6
<b>4</b>	<b>Gauss reduction - single core case</b>	<b>7</b>
4.1	Algorithms in L <sup>A</sup> T <sub>E</sub> X . . . . .	7
<b>5</b>	<b>see the tutorlias</b>	<b>9</b>

Introduction bla bla

# 1 Turing Machine

## 1.0.1 Enigma

example of code

```
1 from itertools import imap
2
3 def KSA(key):
4     S = range(256)
5     j = 0
6     for i in range(256):
7         j = (j + S[i] + key[i % len(key)]) % 256
8         S[i], S[j] = S[j], S[i]
9     return S
```

## 2 Gauss Reduction

## 3 Background

bla bla

### 3.1 Linear Systems

$$\begin{cases} x_1 = 2r + s - t \\ x_2 = r \\ x_3 = -2s + 2t \\ x_4 = s \\ x_5 = t \end{cases}$$

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ \vdots \\ a_{k1}x_1 + \cdots + a_{kn}x_n = b_k \\ \vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{cases}$$

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

## 4 Gauss reduction - single core case

### 4.1 Algorithms in L<sup>A</sup>T<sub>E</sub>X

---

**Algorithm 4.1.1** : *Multiplication of Karatsuba*

**input.**  $a, b$  integers

**output.**  $a \cdot b$

```
1 def karatsuba(a, b)
2   if  $a < 100$  or  $b < 100$  then
3     return  $a \cdot b$ 
4   end
5    $m = \max(\log_{10}(a), \log_{10}(b))$ 
6    $m_2 = \text{floor}(m/2)$ 
7    $high(a)$  = take the first  $m_2$  decimal digits of  $a$ 
8    $low(a)$  = take the last  $m_2$  decimal digits of  $a$ 
9    $high(b)$  = take the first  $m_2$  decimal digits of  $b$ 
10  ...
11  print  $(z_2 \cdot 10^{2m_2} + (z_1 - z_2 - z_0) \cdot 10^{m_2} + z_0)$ 
```

---



**Algorithm 4.1.2 :** Enumeration algorithm

**input.** An ordered basis  $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\} \subset \mathbb{Z}^m$  of the lattice  $\mathcal{L}(\mathcal{B})$  and a positive real number  $R$ .

**output.** All the vectors  $\mathbf{x} \in \mathcal{L}(\mathcal{B})$  such that  $\|\mathbf{x}\| \leq R$ .

```

01. Compute  $\{\mu_{ij}\}$  and  $B_i = \|\mathbf{b}_i^*\|^2$ 
02.  $\mathbf{x} = (x_i) \leftarrow \mathbf{0}_n, \mathbf{c} = (c_i) \leftarrow \mathbf{0}_n, \mathbf{e} = (\epsilon_i) \leftarrow \mathbf{0}_n, \text{sum} \leftarrow 0, S = \emptyset, i \leftarrow 1$ 
03. While  $i \leq n$ 
04.    $c_i \leftarrow -\sum_{j=i+1}^n x_j \mu_{ji}$ 
05. ...

19. return S

```

**5 see the tutorlias**

# **Appendix**

## **Installation of Cuda**

bla bla

## Experiments