



**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**

## Περίληψη

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

# Περιεχόμενα

# ΚΕΦΑΛΑΙΟ 1

## Εισαγωγή

bla bla

## ΚΕΦΑΛΑΙΟ 2

### Αναγωγή Gauss

# ΚΕΦΑΛΑΙΟ 3

## Γραμμική Άλγεβρα

μπλα μπλα

### 3.1 Γραμμικά Συστήματα

μπλα μπλα



## ΚΕΦΑΛΑΙΟ 4

# Υλοποίηση του Gauss με Σειριακό Προγραμματισμό

### 4.1 Σειριακός Προγραμματισμός

Σε ένα σειριακό πρόγραμμα ο υπολογιστής εκτελεί τις εντολές του, τη μία πίσω από την άλλη,

### 4.2 Γλώσσα C

Η γλώσσα C επινοήθηκε το 1972 από τον Dennis Ritchie στα εργαστήρια Bell. Δημιουργήθηκε για να εξυπηρετήσει το λειτουργικό σύστημα Unix, το οποίο έως τότε ήταν γραμμένο σε assembly.

---

**Αλγόριθμος 4.2.1 :** Πολλαπλασιασμός του Karatsuba

**Είσοδος.**  $a, b$  ακέραιοι

**Έξοδος.**  $a \cdot b$

---

```
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  ...
12  print  $(z_2 \cdot 10^{2m_2} + (z_1 - z_2 - z_0) \cdot 10^{m_2} + z_0)$ 
```

---

**Αλγόριθμος 4.2.2 :** Αλγόριθμος Απαρίθμησης (*KFP enumeration algorithm*)

**Είσοδος.** Μια διατεταγμένη βάση  $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\} \subset \mathbb{Z}^m$  του πλέγματος  $\mathcal{L}(\mathcal{B})$  και ένα θετικό αριθμό  $R$ .

**Έξοδος.** Όλα τα διανύσματα  $\mathbf{x} \in \mathbb{L}$  με  $\|\mathbf{x}\| \leq R$ .

```

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

```

```

19. return S

```

## ΚΕΦΑΛΑΙΟ 5

### Υλοποίηση Gauss με Παράλληλο Προγραμματισμό

# ΚΕΦΑΛΑΙΟ 6

## Παράλληλος Προγραμματισμός

Στον παράλληλο προγραμματισμό εκμεταλλευόμαστε την ύπαρξη πολλών επεξεργαστών και διαθέσιμων thread, για να πετύχουμε την αύξηση των υπολογιστικών επιδόσεων και τη μείωση του απαιτούμενου χρόνου εκτέλεσης του προγράμματος.

## ΚΕΦΑΛΑΙΟ 7

# Υλοποίηση Gauss με Παράλληλο Προγραμματισμό στη CPU (C/POSIX Threads)

### 7.1 Εισαγωγή

...

#### 7.1.1 Μετάβαση από τον σειριακό στον παράλληλο προγραμματισμό.

Κατά τη διάρκεια των 70ς, 80ς και ένα μέρος των 90ς, ο σειριακός προγραμματισμός ...

## ΚΕΦΑΛΑΙΟ 8

# Υλοποίηση Gauss με Παράλληλο Προγραμματισμό στη GPU (CUDA/C)

### 8.1 Εισαγωγή

#### 8.1.1 Η Εμφάνιση της GPU/ GPGPU

μπλα μπλα

Παραρτήματα

## ΚΕΦΑΛΑΙΟ 9

# Εγκατάσταση του CUDA για Ubuntu linux

Πριν φτάσουμε στην εγκατάσταση του ΎΔΑ πρέπει πρώτα να ελέγξουμε κάποια πράγματα.

1. Ελέγχουμε αν η ΓΠΥ που έχουμε υποστηρίζει ΎΔΑ. Αυτό μπορεί κάποιος να το ελέγξει από το σιτε<sup>1</sup> της NVIDIA που περιέχει τη λίστα με τις GPU που υποστηρίζουν CUDA.

---

<sup>1</sup> [https://www.geforce.com/hardware/technology/cuda/supported-gpus?field\\_gpu\\_type\\_value=All](https://www.geforce.com/hardware/technology/cuda/supported-gpus?field_gpu_type_value=All)



# ΚΕΦΑΛΑΙΟ 10

## Εκτέλεση του κώδικα

Η σειρά εκτέλεσης της πτυχιακής με τις κατάλληλες εντολές είναι η εξής:

1. Πρώτα εκτελούμε το πρόγραμμα με όνομα `ArrayProduction.c` για να παράξει έναν τυχαίο πίνακα από 0 και 1. Οι εντολές είναι a) `gcc ArrayProduction.c -o ArrayProduction (-std=c99 για cc=1.x)` b) `./ArrayProduction 10` (όπου 10 είναι το μέγεθος του πίνακα, για να δημιουργηθεί ένα πίνακας  $10 \times 11$ )

# Αναφορές

- [1] J. F. Grcar, “Mathematitians of gauss elimination,” *Notices of the American Mathematical Society*, vol. 58, no. 6, pp. 782–792, 2011.
- [2] C. Gauss, “Presentació del volum gauss,” pp. 11–14, Facultat de Matem tiques i Estadística (ed.), Barchelona, Spain, 2006. Conferències FME: volum III. Curs Gauss, 2005-2006.
- [3] Y. M. . L. L. W. Kyi, “Performance comparison of gauss elimination and guass-jordan elimination,” *International Journal of Computer Science and Network Security*, vol. 2, no. 2, pp. 67–71, 2014.
- [4] Γ.Δ. ΑΚΡΙΒΗΣ & Β.Α. ΔΟΥΓΑΛΗΣ, *Εισαγωγή Στην Αριθμητική Ανάλυση*. Πανεπιστημιακές Εκδόσεις Κρήτης, ΗΡΑΚΛΕΙΟ, ΚΡΗΤΗ, 4th ed., 2013.
- [5] M. McCool, A. D. Robison, and J. Reinders, “Chapter 3 - patterns,” in *Structured Parallel Programming* (M. McCool, A. D. Robison, and J. Reinders, eds.), pp. 79 – 119, Boston: Morgan Kaufmann, 2012.
- [6] Tolga Soyata, *GPU Parallel Program Development Using CUDA*. Chapman & Hall/CRC Computational Science, Chapman and Hall/CRC, 1st ed., 2018.
- [7] “Nvida cuda c programming guide version 8.0.” "[https://docs.nvidia.com/cuda/archive/8.0/pdf/CUDA\\_C\\_Programming\\_Guide.pdf](https://docs.nvidia.com/cuda/archive/8.0/pdf/CUDA_C_Programming_Guide.pdf)", June 2017.