

# ΑΘΑΝΑΣΙΟΣ ΞΑΝΘΟΠΟΥΛΟΣ

## ΑΕΜ: 2392

ΠΡΟΑΙΡΕΤΙΚΗ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ 2019 (ΣΗΜΑΤΑ & ΣΥΣΤΗΜΑΤΑ)

Η εργασία έγινε στην γλώσσα προγραμματισμού Python, έκδοση 3.7.3.

Οι βιβλιοθήκες που είναι απαραίτητες για να τρέχει το πρόγραμμα είναι: **numpy**, **pyfftw**, **soundfile**, **pycuda**, **matplotlib**.

```
1. pip install numpy
2. pip install pyfftw
3. pip install soundfile
4. pip install pycuda
5. pip install matplotlib
```

Επίσης πρέπει να είναι εγκατεστημένο το [CUDA toolkit](#) και το [Visual Studio](#) επειδή γίνεται η χρήση των compilers για Cuda C++ και C++.

Η υλοποίηση της συνάρτησης που πραγματοποιεί την συνέλιξη δύο σημάτων έγινε με δύο τρόπους:

1. Με την χρήση του ορισμού της συνέλιξης (άθροισμα πολλαπλασιασμών):

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n - m],$$

Όνομα συνάρτησης: **myconvolve\_simple**

2. Με την χρήση των ιδιοτήτων του μετασχηματισμού Fourier

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$$

Όνομα συνάρτησης: **myconvolve\_fft**

Σε κάθε υλοποίηση γίνεται padding με μηδενικά στο τέλος κάθε σήματος ώστε να γίνει σωστά η συνέλιξη με τον τρόπο υλοποίησης που επέλεξα.

Στον κώδικα γίνεται χρονομέτρηση και αναπαράσταση των σημάτων.

Ο χρόνος εκτέλεσης της συνέλιξης με την συνάρτηση **myconvolve\_simple** σε CPU είναι υπερβολικός γι' αυτό και ο σχετικός κώδικας είναι commented (γραμμή 95).

Η απλή συνέλιξη με την χρήση των πυρήνων της κάρτας γραφικών (**cuda\_myconvolve\_simple**) είναι μικρός και είναι στα ίδια επίπεδα με τον χρόνο εκτέλεσης της **myconvolve\_fft** σε CPU.

Η παραλληλία βασίζεται στον κάθε πυρήνα της GPU να υπολογίζει το άθροισμα για ένα συγκεκριμένο σημείο=**blockDim.x\*blockIdx.x + threadIdx.x** όπου ουσιαστικά αναφερόμαστε στο thread σε συγκεκριμένο block που υπολογίζει το άθροισμα για το σημείο που του αντιστοιχεί.