Universidad de Puerto Rico
Recinto de Mayagüez
Departamento de Ingeniería Eléctrica y Computadoras
Mayagüez, Puerto Rico

# Final Report
# LanPlot

Cristian G. Duque
Gladymar O'Neill Rivas
ICOM4036-066
Prof. Wilson Rivera Gallego

## I. Introduction

LanPlot is a new programming language designed to find and plot the Fourier Transform of a given signal. Fourier Series are used to approximate in a periodic way any signal, and its transform is a decomposition of that same signal into frequencies. The motivation for designing this language was to make it easier to visualize the behavior of different signals used specially in communications. LanPlot is able to take expressions expressed as signals or variables, compute their Fourier Transform, and plot them. A projection for this language is to make possible to plot the square, sawtooth, and triangle Fourier Series approximations of different signals accordingly.

## II. Language Features

The main idea for this language is to make graphs of the Fourier Transform of signals used in communications. It takes a signal as direct input or as a variable and plots its Fourier transform over a specified number of samples. A variable in this program can hold a signal expressed as an exponential, cosine or sine. In other words, a variable can be a function. An important detail is that every variable or function to plot has to be accompanied by a parameter corresponding to the number of samples to plot.

## III. Implementation requirements and tools

The language has the following requirements:

- It is not case sensitive
- Every function has to follow the format "*signal, number_of_samples*"
- The order of instructions is important:
    - "plot" instruction cannot be called without first declaring a signal or valid variable
- The parser read from left to right

The tools to create the programming language or to being able to run the language are:

- Python: python version 2.7 or higher
- PyCharm or Anaconda: possible IDEs to program Python
- Matplotlib: python 2D plot library
- Numpy: fundamental package for scientific computing
- Lex: generates a lexical analyzer for input strings
- Yacc: parser generator
- PLY: implementation of Lex and Yacc for lexical and syntax analyzer

## IV. Language References

Key Words:

- plot
- fouriertransform

Reserved:

- pi

Functions from tokens:

- cos
- sin
- exp

## V. Language Development and Translator Architecture

The programming language was made using python. The lexical analyzer and parser are included in the class lex.py. The intermediate code that contains the executable code for computing and plotting the Fourier Transform is in the class FourierTransform.py. For the program to compile and work properly, the packages and tools previously stated were used.

The first part of the development process consisted of testing the intermediate code, which takes as an input an equation containing a periodic, sinusoidal or exponential function and computes its Fourier transform using the 'fft' functions inside the python package numpy. The output, which is a new window containing the interactive graph, is presented. After making sure that part worked properly, the next step was to construct the lexical analyzer, taking into considerations all the possible tokens, expressions, functions, and keywords needed. After defining every component, the parser was created and tested, making sure the program worked and executed as it was desired.

## VI. Example of a program

LanPlot can take inputs in the following ways:

- PLOT FOURIERTRANSFORM *signal, number_of_samples*
- PLOT FOURIERTRANSFORM *variable*
- *variable = signal, number_of_samples*
- *variable* = FOURIERTRANSFORM *variable*
- PLOT *variable*
- *variable = variable*

An example of a program could be:

```
 (LanPlot) > var = cos(2*pi*50*t), 600
var
(LanPlot) > plot fouriertransform var //output graph is generated
```

## VII. Conclusions

The creation of a new programming language is one of pure creativity. For this programming language, we thought of a way of facilitating the visualization of the Fourier series of different signals. Like us, there are a lot of students that which to have a better understanding of how Fourier series worked, which is something this programming language can be helpful for. Making this project helped us understand the complexity of a programming language and the process a code does to execute what is asked. This project gives us the motivation not only to use our creativity and imagination for solving problems using a programming language, but also to dig deeper in the understanding of how new and existing programming languages work.