How to implement a simple system for fast data analysis

Kamil Rusin
Faculty of Applied Mathematics
Silesian University of Technology
Kaszubska 23, Gliwice, 44-100, Poland
kamirus323@student.polsl.pl

***Abstract*—Data analysis is important in our world. It is worth to get to know how data analysis work and how it can be implemented in a simple way. Fast Fourier Transform is very well known in scientific disciplines related to sound. It will be explain how it does work and which formulas should be used. This transform has various applications that affect our lives,especially analysis of sounds and signals in the world. In this paper, a simple system for fast analysis of frequencies will be implemented with explanatory. Scripts are used to divide the program into tasks that one specified programming language process data better than the other. Python and C# are one of the most popular programming languages. They differ on some issues that will be explained and also good programming practices will be presented.**

***Keywords*—** simple system, fast system, implementing a simple system, processing data, fast fourier transform, use of fast fourier transform, analysing data, fft

I. Introduction

Data analysis takes part in our projects, examinations. We need to use a special designed system for processing data that we get during calculations of our programs. The speed of that analysis is defining how useful our applications are for users. It must be done fast and simple systems are the key to reaching that attribute. Moreover, data analysis is used in a psychological researches. Exploratory data analysis is a strategy, which states that people should have an open mind to alternative possibilities. That attitude is about how data analysis should be carried out[1]. Matthew A. Waller and Stanley E. Fawcett see the increasing popularity of data analysis. Moreover they know that it requires knowledge and skills. They say that data science and big data are also relevant to supply chain research and education. They examine possible applications of it in practice and provide examples of research questions from these applications [2]. The analysis of data is more widely used in data mining, which is used for automated discovery of statistical relationships and schemes in very large databases. It is new discipline that has big potential [3]. The reason why data mining is developing really fast is that the data contains valuable information for owners of many corporations. You can find a large number of hidden dependencies in them, which, for example, can greatly improve the promotion of certain products. Jiawei Han, Jian Pei and Micheline Kamber explain the essence of data analysis [4]. They say why it is used and show what patterns and technologies are currently used. Moreover, they also explain how to analyze, assign and how to properly store and manage data.

As an example in this paper we will be considering how to implement simple system for analysing periodic functions and transforming them into frequency values using *Fast Fourier Transform algorithm*. There will be also explained how this algorithm works and how it can be applied in our purposes.

©2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)

II. Fast Fourier Transform

A. Description

Today known as *Fast Fourier Transform algorithm* was reported by Tukey and Cooley in 1965. This algorithm is for efficiently computing the Discrete Fourier Transform of a time series. It caused significant changes in mathematician and technician world. The calculations can be obtained more economically now. That is why it arouses a lot of interest, as stated in paper [5].

Fast Fourier Transform(FFT) refers to a specific group of algorithms, very similar to one another, in which a smaller number of complex actions is enough to calculate *Discrete Fourier Transform(DFT)*. The algorithm divides it into shorter and simpler calculation and thus shortens the computation time, so users can use larger sampling frequencies. Currently, there are many forms of this algorithm. The easiest and the most popular is a Fast Fourier Transform with a base of 2. There is also her recursive variant in which we divide the problem into sub-problems of a smaller size and them also recursively into even smaller ones until we reach enough smaller problems. Next step is to solve them. The solution to the problem is the sum of the subproblems. Let's consider Decimation In Time FFT algorithm. After specifying the number of samples (N), which must be a power of 2, the following steps are performed:

- A recursive division is made into samples with even and odd indexes until two sets are acquired.
- The N/2 two-point DFT is made.
- Two-piece sets are folded into two times larger, until the whole signal is obtained.

Suppose the following:

$$W_N = e^{-i2\pi/N} \tag{1}$$

Divide DFT equation into two parts by indexes:

$$\begin{aligned} x_k = &\sum_{n=0}^{(N/2)-1} x_{2n} \cdot e^{-i2\pi(2n)k/N} + \\ &\sum_{n=0}^{(N/2)-1} x_{2n-1} \cdot e^{-i2\pi(2n+1)k/N} = \\ &\sum_{n=0}^{(N/2)-1} x_{2n} \cdot W_{N/2}^{nk} + W_N^k \cdot \sum_{n=0}^{(N/2)-1} x_{2n+1} \cdot W_{N/2}^{nk} \end{aligned} \tag{2}$$

Define two additional formulas:

$$y_n = x_{2n} \tag{3}$$

$$z_n = x_{2n+1} \tag{4}$$

CEUR-WS.org/Vol-2468/p6.pdf

For every k belonging to the set $0, 1, 2, ..., N/2 - 1$ there are two dependencies:

$$Y_k = \sum_{r=0}^{(N/2)-1} y_r \cdot W_{N/2}^{kr} \tag{5}$$

$$Z_k = \sum_{r=0}^{(N/2)-1} z_r \cdot W_{N/2}^{kr} \tag{6}$$

Using this formula:

$$W_N^{(N/2)+k} = -W_N^k \tag{7}$$

There is the final formula for DIT FFT:

$$X_{k+N} = \sum_{n=0}^{(N/2)-1} y_n \cdot W_{N/2}^{(k+N/2)n} + W_N^{k+N/2} \cdot \sum_{n=0}^{(N/2)-1} z_n \cdot W_{N/2}^{(k+N/2)n} = Y_k - W_N^k \tag{8}$$

In this way, Discrete Fourier Transform is divided into subsequent DFTs until only two-point DFTs are obtained. Then sets must be folded. Therefore, it is important to assume the number N as the power of two [6].

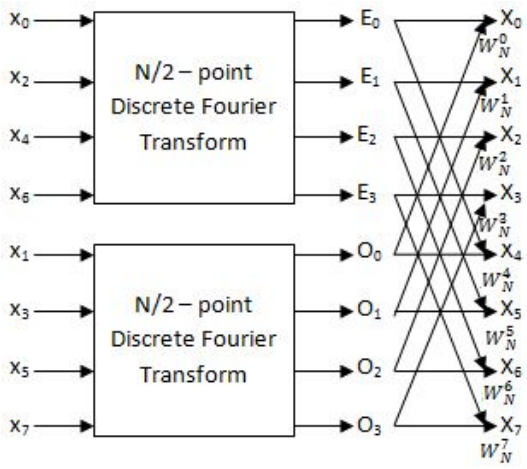

Figure 1: Example graph for 8 sample

B. Use of FFT

Each periodic signal can be presented in the form of a sum of sinusoidal signals with different frequencies and different amplitudes. Thanks to FFT we are able to accurately assess components of a given signal is made of. The FFT result is the spectrum and it allows for fast and precise validation whether the components of the output signal belong to the work of a healthy device or they are a sign of deterioration of the technical condition. In other words it is used to filter the signal, distinguish particular waves. For example, we can conclude only a part of given band in the output signal or define which parts of the band we want to omit. This can also be used to delay the signal, transmit from specific frequency. The greater the sampling frequency, the better the details of the analog signal in digital form are preserved. Another mathematical application of FFT is the rapid multiplication of polynomials. When we have two polynomials of various degrees, we can align them by adding coefficients equal to 0 to the lower degree polynomial. Next step is to reindexate the indicators and the polynomial coefficients are treated as the coordinates of the vectors. The newly created vector is their weave. Using the Fourier transformation definition for a function weave, we can count the FFT of both vectors as well as the inverse Fourier transform of their convolution. Fast Fourier Transform can also be used to solve two- or three-dimensional Poisson equations. The FFT is used in digital audio processing, digital image processing and speech. The signal can be processed in real time. It is also used in compressing MP3 music files. Using the same group of algorithms, Discrete Cosine Transform can be used in order to compress images in .jpg format. This transformation divides the video image into 64-point blocks. Each of these blocks is compressed separately, then by combining them, a distorted picture is created and, as a result, high degradation of video quality is obtained. In FFT, we move in the frequency domain and this is the reason for a different look at the issue of image filtration. Weave function can be calculated more easily. The image after Fourier transform is high-pass filtered, this means zeroing the central part of the image that contains spectrum for low frequencies. The image obtained in the results of this filtration is added to the source image and because of that there is a sharp image.

III. System for analysis

A. Scripts

This system combines several elements whose task is to process the data. The Batch script is the main part that is responsible for support of this program. It coordinates the work of the remaining scripts and at the appropriate moments turns on the C# and Python scripts. The purpose of the program is to calculate the frequency of given functions by using the RADIX-2 FFT algorithm. It allows to create graphs of amplitude. This program processes the function along with the amount of samples using the mentioned algorithm. It performs a specific amount of operations on complex numbers. After all calculations are done, it creates output file and runs Python script that creates statistics of all information and generates HTML page.

The system starts with running batch script, which provides us a simple menu for controlling the program. Menu has several options such as displaying the user manual, making a backup of our files and data that we obtained through the program. This is important to avoid accidental deletion of data. It also starts the second script that calculates FFT frequencies.

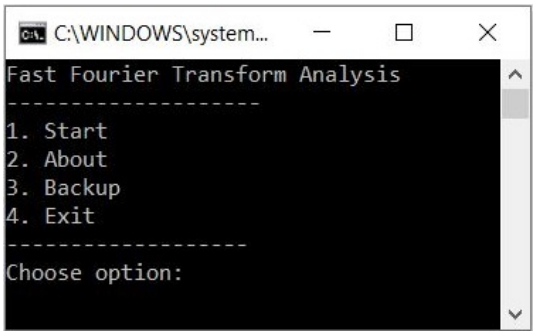

Figure 2: Simple menu

Firstly, C# script checks how many data we provided and looks for text files in an *input* folder. In every two lines of these text files are a formula of function and number of samples to process. The correct function has a variable x. The special mathematics library is responsible for reading it properly. In a *radix-2 Fast Fourier Transform algorithm* number of samples must be a power of two.

This is required by dividing the samples due to the parity. Then all calculations are performed and lists of frequency values are written to output files. To avoid incorrect reading and writing of files during the operation of the system we should ensure that the same type of encoding is set for all files.

After all calculations are done main script automatically turns off C# script and then calls Python script, which creates statistics of processed data written in an output files to HTML file. It also generates a table, which rows contains number of sample and frequency value. Batch script also runs that page, so the whole chain of events happens within seconds. The process is automated. There is also an alternative to calling a Python script. We can use special interpreter in the current C# script and the *Process* class. To know more about that class it is recommended to read official Microsoft documentation.

B. Planning a system

First of all, the structure of a system must be made. This is a framework, which program is based on. Then scripts can be programmed step by step and that does not lead to organizational problems. In analysed example the structure is made from 3 scripts with different objective. First one is to control others and set possible options. The second one is to make all calculations and the third one is to gather the data and analyse it in order to make statistics.

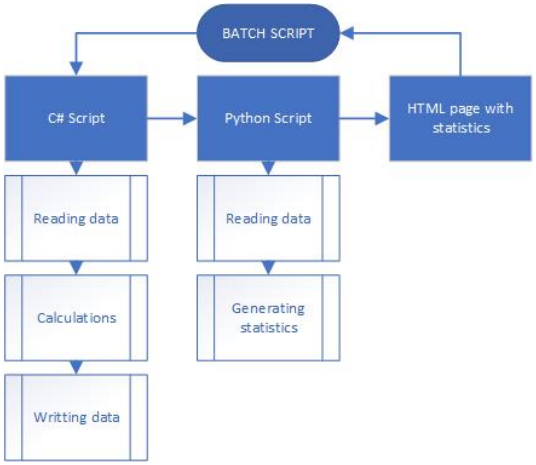

Figure 3: Example of a structure

Batch provides very useful command call that opens us a way to launch second script. It pauses parent batch program until the second one is finished. There is also an option that allows us to call a script with arguments. That is how an options edited in the main script can be forwarded. To know more about call command it is recommended to read official documentation of Batch.

Use of second script depends on user needs. It is written in high-level programming language with high abstraction that allows to program even difficult tasks, complicated mathematical problems or precisely defined ways of processing information. All operations are held here. There are a lot of useful frameworks for every language that can help with calculations and processing.

The statistics are separated from the previous script for easier organization and implementation. In our example it was implemented with Python. It has a various built-in mathematical functions that accelerates our program. In this implementation Python script generates a table with frequency values for a given sample and information about median, mean, various sums or number of processed functions. To view that data in an eye-friendly way this script also generates HTML file that is launched with call command in Batch script after C# and Python scripts are done, so the main script is not longer paused.

Function: 1/5*x*sin(x)+1/2*cos(x)

Number of sample	Frequency
1	1433.29418738018
2	855.739639183371
3	335.493613091777
4	40.0976158998761
5	158.372815848432
6	21.5410110977014
7	13.8926735150989
8	89.4656701060409
9	78.070982833763
10	10.9447169450731

Figure 4: Simple table with values

General statistics

Number of processed files: **16**

Number of processed functions: **16**

Number of processed arguments: **1728**

Sum of all amplitudes: **72404.76075680612**

Average value: **41.900903215744286**

Figure 5: Example of statistics

C. Data and neural network

It is worth considering using a system for data analysis in machine learning. Neural networks process a lot of data [7]–[10]. There are a lot of ready to use datasets for various purposes. For example, our program needs to analyse sounds provided by external library. There can be implemented simple system for data flow. First script can subject data to a Fast Fourier Transform algorithm and save information to a file. Then, second one can read it and process in a way a programmer wants. It has wide range of applications in a voice control for example at smart houses. It should be working on an affordable platforms available at home. The processing time should be relatively fast. When sound is recorded noise is always present. To properly recognize a speech, it should be removed and only those frequencies should be kept, which a person is able to speak in. In this case, Fast Fourier Transform is very helpful. Two filters are most often selected: low-pass and middle-pass [11]. A system can be created that records the sound and then for example sends it to the next program on the workstation that will process the speech. There is an algorithm that allows to quickly identify commands. First of all, an identifier is created for each word. Secondly, the correlation coefficient is calculated. Then the pattern is split into a fixed number of time intervals, which are then subjected to a Fast Fourier Transform. The next step is to calculate totals in intervals and create a command identifier. Finally, the generated identifiers, corresponding to the recorded commands, are compared with patterns, which results in the final decision of the program. This method is written in detail in this paper [12]. Another application of simple system for fast data analysis is in neural networks.

D. Good practices before starting to code

Planning is a first step of every project. It should be determined what the project will accomplish. Once it is done, it allows to avoid rewriting code and keeping the notes of what should be changed.

First, write the basic idea of a project. Then, read it once more and add more sentences describing what the application's functionalities will be. Secondly, think once more and delete unnecessary assumptions. It is recommended to construct logic network diagram

and detailed decomposition of the project to clearly see how the data will be processed, in order to make it efficient.

When the first code is written, the entire functionality should be analysed once again and, if necessary, extended with further functionalities. Logical sequence of activities can be developed deeper.

Moreover, clean code is also an important element in projects and implementations. Following good practices results in minimizing the time that other person has to spend on reading and understanding the code. First thing worth to remember is to set relatively short and meaningful names for classes, methods, functions, variables. It means that a person is able to guess the purpose of using e.g. that method, variable.

If it is hard to name them in a relatively short way or it has a deeper meaning, it is also recommended to use comments. It will help in clarification. They should be also used in explaining the code. Nevertheless comments should not be a way to explain programmers wrong code. To prevent that kind of code there is also a practice, which allows methods or functions to perform only one task, it is called Single responsibility principle introduced by Robert C.Martin. If a one style of writing is chosen, it should be used everywhere. Mixing styles may lead to a complete opposite of the practices that are used.

While planning a structure of our program, the language of all scripts must be chosen. C# and Python were chosen for two scripts in the example project for this paper. The advantages and disadvantages should be considered. First of all, Python is a dynamically-interpreted language, while C# is statically-typed compiled language. Python is also well known for its great libraries. They are open-source and have a wide range of applications. They can work only with a defined Python versions. That may lead to the moment, when two useful libraries use different language versions. Python also has a dynamic typing. It means that every variable is bond only to an object and can be assigned to a different types during the program. On the other hand C# has an automatic garbage collector that manages memory and saves computer resources. C# also allows to extend existing types by adding new methods. That language is backed by producers and supported at various programming aspects.

E. Tests

In order to test the system, the periodic functions were prepared. 2000 files were created that each of them contained 4 functions and an information about number of samples for each function. Every file was analyzed. Math library was used to read a function from a string and do all calculation required to obtain a frequency value for a given argument.

First script was run for 100 files. The time was measure from the point of reading a function to the moment of a last calculation. After each iteration number of analyzed files was increased by 100. As can be seen in a 6, the time (measured in miliseconds) of processing increased linearly. First iteration was done under 1781 miliseconds. The second one analyzed the data faster by 208 miliseconds than first iteration comparing the values after 100 files. Interesting is the fact that counting from iteration number 10, the speed of analyse per 100 files is increasing. The lowest value of analysing the data was 17,81 miliseconds during the first iteration. Data from 2000 files was processed in 24414 miliseconds.

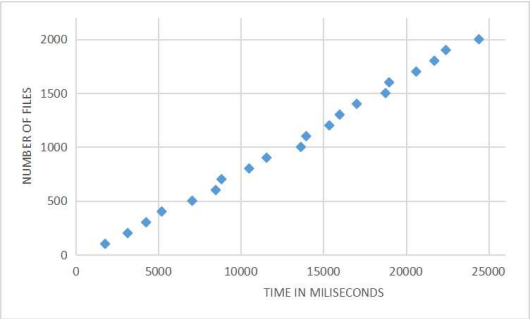

Figure 6: Time of file analysis

IV. Final remarks

Implementing simple structure for data analysis can be done seamlessly. Developers must keep in mind the compatibility of the file encoding and following the scheme. It is worth considering which programming language to choose for a given script, because the volume of built-in functions may affect the delay of the extended program operation. Sound planning of program calculations can significantly increase its efficiency. As introduced in this paper, Fast Fourier Transform has various possible applications ranging from mathematical to implementation in devices in Smart Home and neural networks. It can be used to recognize speech and voice commands. This could be helpful for disabled people [12].

References

[1] C. H. Yu, "Exploratory data analysis," *Methods*, vol. 2, pp. 131–160, 1977.

[2] M. A. Waller and S. E. Fawcett, "Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management," *Journal of Business Logistics*, vol. 34, no. 2, pp. 77–84, 2013.

[3] D. J. Hand, "Data mining," *Encyclopedia of Environmetrics*, vol. 2, 2006.

[4] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[5] W. T. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, D. E. Nelson, C. M. Rader, and P. D. Welch, "What is the fast fourier transform?" *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1664–1674, 1967.

[6] T. P. Zieliński, *Cyfrowe przetwarzanie sygnałów: od teorii do zastosowań*. Wydawnictwa Komunikacji Łączności, 2005.

[7] M. Woźniak, D. Połap, G. Capizzi, G. L. Sciuto, L. Kośmider, and K. Frankiewicz, "Small lung nodules detection based on local variance analysis and probabilistic neural network," *Computer methods and programs in biomedicine*, vol. 161, pp. 173–180, 2018.

[8] M. Wozniak, D. Polap, L. Kosmider, C. Napoli, and E. Tramontana, "A novel approach toward x-ray images classifier," in *2015 IEEE Symposium Series on Computational Intelligence*. IEEE, 2015, pp. 1635–1641.

[9] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals," *Computers in biology and medicine*, vol. 100, pp. 270–278, 2018.

[10] D. Połap and M. Woźniak, "Flexible neural network architecture for handwritten signatures recognition," *International Journal of Electronics and Telecommunications*, vol. 62, no. 2, pp. 197–202, 2016.

[11] P. Walendowski, "Zastosowanie sieci neuronowych typu svm do rozpoznawania mowy," *Praca doktorska, Politechnika Wrocławska*, 2008.

[12] A. Śliwiński and K. Tomczewski, "Badania możliwości rozpoznawania mowy w autonomicznych systemach sterowania," *Poznan University of Technology Academic Journals. Electrical Engineering*, no. 88, pp. 79–88, 2016.