New Efficient FFT with Fewer Operations

Hakob Sarukhanyan
Institute for Informatics and Automation Problems of NAS of Armenia
Yerevan, Armenia
e-mail: hakop@ipia.sci.am

Rafayel Barseghyan
Institute for Informatics and Automation Problems of NAS of Armenia
Yerevan, Armenia
e-mail: barseghyan@gmail.com

ABSTRACT

In this paper we suggest a simple recursive modification of 2^p-point Fast Fourier transform algorithm with fewer arithmetic operations.Required number of operations are the best up to 2^{20}. Algorithm for real-data and real-symmetric (discrete cosine) transforms with fewer arithmetic operations can be easily derive from our algorithm.

Keywords

FFT, DFT, complexity,flops.

1. INTRODUCTION

Applications of linear transforms, such as Fourier, Hadamard, Cosine and Sine transforms in signal and image processing are numerous [1]. Cooley and Tukey published their historic paper on the computation of the Fourier transform in 1965. Overnight, in universities and laboratories around the world, scientists and engineers began developing computer programs and electronic circuits to implement the FFT. The FFT is a brilliant technique for computing the discrete Fourier (DFT) transform quickly. By recognizing that the Fourier transform of a sequence can be derived from the Fourier transforms of two half length sequences more economically than if the whole sequence is transformed directly and by carrying this concept through to its logical conclusion of evaluating only the direct transform of sequences of two terms, Cooley and Tukey showed that the FFT required only $O(N\log N)$ operations while the direct form took $O(N^2)$ operations.Any improvement in FFT algorithms appears to rely on reducing the exact number or cost of these operations rather than their asymptotic functional form [2]. For many years, the time to perform an FFT was dominated by real-number arithmetic, and so considerable effort was devoted towards proving and achieving lower bounds on the exact count of arithmetic operations (real additions and multiplications),called "flops" (floating-point operations), required for a DFT of a given size [3],[4]. Although the performance of FFTs on recent computer hardware is determined by many factors besides pure arithmetic counts, there still remains an intriguing unsolved mathematical question: what is the smallest number of flops required to compute a DFT of a given size N?

2. 1D 2^P-POINT FFT

2.1 Conventional Case

Let $x=\{x_0,x_1,\dots,x_{N-1}\}^T$ be a complex valued column-vector of length N ($N=2^p$). The forward and inverse 1D DFT of this vector are defined as

$$\begin{aligned} X[n] &= \frac{1}{N}\sum_{k=0}^{N-1} x[k]W_N^{nk}, \\ x[k] &= \sum_{n=0}^{N-1} X[n]W_N^{-nk}, \quad n=\overline{0,N-1}, \end{aligned} \tag{1}$$

where $W_N^n = \exp\left(-j\frac{2\pi}{N}n\right) = \cos(\frac{2\pi}{N}n) - j\sin(\frac{2\pi}{N}n)$, $j=\sqrt{-1}$.

Represent the forward transform as follows (here and later the coefficient 1/N is omitted)

$$X[n] = \sum_{k=0}^{N/2-1} x[2k]W_{N/2}^{nk} + W_N^n \sum_{k=0}^{N/2-1} x[2k+1]W_{N/2}^{nk}, \tag{2}$$

where $n=\overline{0,N-1}$.

Introduce the notations:

$$\begin{aligned} Y_0[n] &= \sum_{k=0}^{N/2-1} x[2k]W_{N/2}^{nk}, \\ Y_1[n] &= \sum_{k=0}^{N/2-1} x[2k+1]W_{N/2}^{nk}, \quad n=\overline{0,N/2-1}. \end{aligned} \tag{3}$$

Note that $Y_0[n]$ and $Y_1[n]$ are $N/2-$point forward DFT.

Hence, the equation (2) can be represented as follows

$$\begin{aligned} X[n] &= Y_0[n] + W_N^n Y_1[n], \\ X[n+N/2] &= Y_0[n] - W_N^n Y_1[n], n=\overline{0,N/2-1}. \end{aligned} \tag{4}$$

It is easy to show that

$$\begin{aligned} W_N^0 &= 1, \qquad W_N^{N/8} = \tfrac{\sqrt{2}}{2}(1-j), \\ W_N^{N/4} &= -j, \quad W_N^{3N/8} = -\tfrac{\sqrt{2}}{2}(1+j). \end{aligned}$$

Therefore, the realization of $W_N^n Y_1[n]$, for all n needs only $N-4$ real addition and $2N-12$ real multiplication operations. Storing this results we can calculate the necessary operations for $N-$point DFT given in equation (4), i.e. we obtain

$$\begin{aligned} C_N^+ &= 3N-4+2C_{N/2}^+, \\ C_N^\times &= 2N-12+2C_{N/2}^\times, \end{aligned} \tag{5}$$

where C_N^+ and $C_N^\times$ denotes the number of additions and multiplications of $N-$point DFT, respectively.

Finally, from relations (5) we can obtain

$$\begin{aligned} C_N^{+} &= 3N\log_2 N - 3N + 4, \\ C_N^{\times} &= 2N\log_2 N - 7N + 12, \quad N \geq 8. \end{aligned} \tag{6}$$

Note that $C_2^+ = 4, C_2^\times = 0,\ C_4^+ = 16, C_4^\times = 0$. In the next table some numerical results are given

Table 1

N	Add	Mul	Total
2	4	0	4
4	16	0	16
8	52	4	56
16	148	28	176
32	388	108	496
64	964	332	1296
128	2308	908	3216
256	5380	2316	7696
512	12292	5644	17936
1024	27652	13324	40976
2048	61444	30732	92176
4096	135172	69644	204816
8192	294916	155660	450576
16384	638980	344076	983056
32768	1376260	753676	2129936

3. MODIFIED FFT

3.1 Conventional Case

Let $x = \{x_0, x_1, \ldots, x_{N-1}\}^T$ be a complex valued column-vector of length N $(N = 2^p)$. The DFT of this vector can be represented as (the coefficient $1/N$ is omitted)

$$\begin{aligned} X[n] = \sum_{k=0}^{\frac{N}{2}-1} x[2k] W_{\frac{N}{2}}^{nk} + W_N^n \sum_{k=0}^{\frac{N}{4}-1} x[4k+1] W_{\frac{N}{4}}^{nk} \\ + W_N^{3n} \sum_{k=0}^{\frac{N}{4}-1} x[4k+3] W_{\frac{N}{4}}^{nk}, \end{aligned} \tag{7}$$

where $n = \overline{0, N-1}$.

It is not difficult to show that with assumption
$x[-1] = x[N-1]$ we have

$$W_N^{3n} \sum_{k=0}^{\frac{N}{4}-1} x[4k+3] W_{\frac{N}{4}}^{nk} = W_N^{-n} \sum_{k=0}^{\frac{N}{4}-1} x[4k-1] W_{\frac{N}{4}}^{nk}.$$

Therefore the equation (7) we can rewrite as following

$$\begin{aligned} X[n] = \sum_{k=0}^{\frac{N}{2}-1} x[4k] W_{\frac{N}{2}}^{nk} + W_N^n \sum_{k=0}^{\frac{N}{4}-1} x[4k+1] W_{\frac{N}{4}}^{nk} \\ + W_N^{-n} \sum_{k=0}^{\frac{N}{4}-1} x[4k-1] W_{\frac{N}{4}}^{nk}, \end{aligned} \tag{8}$$

where $n = \overline{0, N-1}$.

Introduce the following notations:

$$\begin{aligned} A_N^n &= W_N^n Y_1[n] + W_N^{-n} Y_2[n], \\ S_N^n &= W_N^n Y_1[n] - W_N^{-n} Y_2[n], && n = \overline{0, N/4-1}; \\ Y_0[n] &= \sum_{k=0}^{N/2-1} x[4k] W_{N/2}^{nk}, && n = \overline{0, N/2-1}; \\ Y_1[n] &= \sum_{k=0}^{N/4-1} x[4k+1] W_{N/4}^{nk}, && , \\ Y_2[n] &= \sum_{k=0}^{N/4-1} x[4k-1] W_{N/4}^{nk}, && n = \overline{0, N/4-1}. \end{aligned} \tag{9}$$

Hence, $N = 2^p-$point DFT can be computed by the following formulae

$$\begin{aligned} X[n] &= Y_0[n] + A_N^n, \\ X[n + \tfrac{N}{4}] &= Y_0[n + \tfrac{N}{4}] - jS_N^n, \\ X[n + \tfrac{2N}{4}] &= Y_0[n] - A_N^n, \\ X[n + \tfrac{3N}{4}] &= Y_0[n + \tfrac{N}{4}] + jS_N^n, \ n = \overline{0, N/4-1}. \end{aligned} \tag{10}$$

3.2 Complexity Evaluation

Now we calculate the necessary operations for DFT which presented in (10). At first using the properties of exponential function W we have

$$W_N^0 = 1, \quad W_N^{N/8} = \frac{\sqrt{2}}{2}(1-j).$$

Therefore, the realization of A_N^n required $\frac{3}{2}N-4$ and $2N-12$ addition and multiplication operations, respectively. The realization of S_N^n required only $N/2$ additions.

Thus, the necessary operations for realization $N-$point DFT presented in (10) can be obtained from following formulae

$$\begin{aligned} C_N^+ &= 4N - 4 + C_{N/2}^+ + 2C_{N/4}^+, \\ C_N^\times &= 2N - 12 + C_{N/2}^\times + 2C_{N/4}^\times, \quad N \geq 8. \end{aligned} \tag{11}$$

Using theory of difference equations [5] we obtain

$$\begin{aligned} C_N^+ &= \tfrac{8}{3}N\log_2 N - \tfrac{16}{9}N - \tfrac{2}{9}(-1)^{\log_2 N} + 2, \\ C_N^\times &= \tfrac{4}{3}N\log_2 N - \tfrac{38}{9}N + \tfrac{2}{9}(-1)^{\log_2 N} + 6. \end{aligned} \tag{12}$$

In the Table 2 some numerical results are given

Table 2

N	Add	Mul	Total
2	4	0	4
4	16	0	16
8	52	4	56
16	144	24	168
32	372	84	456
64	912	248	1160
128	2164	660	2824
256	5008	1656	6664
512	11380	3988	15368
1024	25488	9336	34824
2048	56436	21396	77832
4096	123792	48248	172040
8192	269428	107412	376840
16384	582544	236664	819208
32768	1252468	517012	1769480

4. NEW FFT ALGORITHM WITH FEWER FLOPS

4.1 Efficient Implementation of FFT

We will perform DFT by two step. At first we introduce some notations:

$$\begin{aligned} T_{N,n} &= \left[1 - j\tan\tfrac{2\pi}{N}n\right], \\ P_{N/4}^n &= W_N^n \cos(\tfrac{2\pi}{N/4}n) Y_1[n] + W_N^{-n} \cos(\tfrac{2\pi}{N/4}n) Y_2[n], \\ Q_{N/4}^n &= W_N^n \cos(\tfrac{2\pi}{N/4}n) Y_1[n] - W_N^{-n} \cos(\tfrac{2\pi}{N/4}n) Y_2[n], \end{aligned} \tag{13}$$

where $Y_1[n]$, $Y_2[n]$ given in (9). Note that

$$W_N^n = T_{N,n} \cos\frac{2\pi}{N}n.$$

Now Using this notations now we represent $N = 2^p-$point DFT from (10) by the following two steps

Step 1: $n = 0, 1, \ldots, N/4-1$.

$$\begin{aligned} X[n] &= Y_0[n] + P^n_{N/4}, \\ X[n+\tfrac{N}{4}] &= Y_0[n+\tfrac{N}{4}] - jQ^n_{N/4}, \\ X[n+\tfrac{2N}{4}] &= Y_0[n] - P^n_{N/4}, \\ X[n+\tfrac{3N}{4}] &= Y_0[n+\tfrac{N}{4}] + jQ^n_{N/4}; \end{aligned} \tag{14}$$

Step 2: $n = 0, 1, \ldots, N/16-1$.

$$\begin{aligned} Y_1[n] &= Y_{10}[n]/\cos\tfrac{2\pi}{N/4}n \\ &+(T_{N/4,n}Y_{11}[n] + T^*_{N/4,n}Y_{12}[n]), \\ Y_1[n+\tfrac{N}{16}] &= Y_{10}[n+\tfrac{N}{16}]/\cos\tfrac{2\pi}{N/4}n \\ &-j(T_{N/4,n}Y_{11}[n] - T^*_{N/4,n}Y_{12}[n]), \\ Y_1[n+\tfrac{2N}{16}] &= Y_{10}[n]\cos\tfrac{2\pi}{N/4}n \\ &-(T_{N/4,n}Y_{11}[n] + T^*_{N/4,n}Y_{12}[n]), \\ Y_1[n+\tfrac{3N}{16}] &= Y_{10}[n+\tfrac{N}{16}]/\cos\tfrac{2\pi}{N/4}n \\ &+j(T_{N/4,n}Y_{11}[n] - T^*_{N/4,n}Y_{12}[n]), \end{aligned} \tag{15}$$

$$\begin{aligned} Y_2[n] &= Y_{20}[n]/\cos\tfrac{2\pi}{N/4}n \\ &+(T_{N/4,n}Y_{21}[n] + T^*_{N/4,n}Y_{22}[n]), \\ Y_2[n+\tfrac{N}{16}] &= Y_{20}[n+\tfrac{N}{16}]/\cos\tfrac{2\pi}{N/4}n \\ &-j(T_{N/4,n}Y_{21}[n] - T^*_{N/4,n}Y_{22}[n]), \\ Y_2[n+\tfrac{2N}{16}] &= Y_{20}[n]\cos\tfrac{2\pi}{N/4}n \\ &-(T_{N/4,n}Y_{21}[n] + T^*_{N/4,n}Y_{22}[n]), \\ Y_2[n+\tfrac{3N}{16}] &= Y_{20}[n+\tfrac{N}{16}]/\cos\tfrac{2\pi}{N/4}n \\ &+j(T_{N/4,n}Y_{21}[n] - T^*_{N/4,n}Y_{22}[n]), \end{aligned} \tag{16}$$

4.2 Complexity evaluation

Now we calculate the necessary operations for DFT which presented in (14)-(16). At first using the properties of cosine and exponential functions we obtain

$$W^n_N \cos\frac{2\pi}{N/4}n = \begin{cases} 1, & \text{if} \quad n=0 \\ -\frac{\sqrt{2}}{2}(1-j), & \text{if} \quad n=n/8, \\ 0, & \text{if} \quad n=N/16, \\ 0, & \text{if} \quad n=3N/16, \end{cases} \tag{17}$$

where $n = 0, 1, \ldots, N/4-1$.

For $n = 0, 1, \ldots, N/16-1$ we have

$$\begin{aligned} \cos\tfrac{2\pi}{N/4}n &= \begin{cases} 1, & \text{if} \quad n=0, \\ \frac{\sqrt{2}}{2}, & \text{if} \quad n=N/32, \end{cases} \\ T_{N/4,n} &= \begin{cases} 1, & \text{if} \quad n=0, \\ 1-j, & \text{if} \quad n=N/32. \end{cases} \end{aligned} \tag{18}$$

Using the results of equations (17) and (18) we can define the necessary real operations for computing the terms $P^n_{N/4}$, $Q^n_{N/4}$, $W^n_N\cos\frac{2\pi}{N/4}nY_1[n]$, and $W^{-n}_N\cos\frac{2\pi}{N/4}nY_2[n]$ for all $n = \overline{0, N/4-1}$ without taking the operations for $Y_1[n]$ and $Y_2[n]$ (see Table 3).

Table 3

Expression	Add	Mul
$W^n_N\cos\frac{2\pi}{N/4}nY_1[n]$	$\frac{1}{2}N-6$	$N-14$
$W^{-n}_N\cos\frac{2\pi}{N/4}nY_2[n]$	$\frac{1}{2}N-6$	$N-14$
$P^n_{N/4}$	$\frac{3}{2}N-16$	$2N-28$
$Q^n_{N/4}$	$\frac{1}{2}N-4$	0

Therefore, again without taking the operations for $Y_0[n]$, $Y_1[n]$, and $Y_2[n]$, and using the results of Table 3 we can calculate the necessary real operations for computing the terms $X[n]$, $X[n+\frac{N}{4}]$, $X[n+\frac{2N}{4}]$, and $X[n+\frac{3N}{4}]$ from equation (14) for all $n = \overline{0, N/4-1}$(see Table 4).

Table 4

Expression	Add	Mul
$X[n]$	$2N-20$	$2N-28$
$X[n+\frac{N}{4}]$	$N-8$	0
$X[n+\frac{2N}{4}]$	$\frac{1}{2}N-4$	0
$X[n+\frac{3N}{4}]$	$\frac{1}{2}N-4$	0

Now we can calculate the number of real operations for computing all component $X[n]$ ($n = \overline{0, N-1}$, $N \geq 16$)

$$\begin{aligned} C^+_X &= 4N - 36 + C^+_{Y_0} + C^+_{Y_1} + C^+_{Y_2}, \\ C^\times_X &= 2N - 28 + C^\times_{Y_0} + C^\times_{Y_1} + C^\times_{Y_2}, \end{aligned} \tag{19}$$

where $C^+_{Y_0}$ and $C^\times_{Y_0}$ are the complexity of $N/2-$point DFT, and $C^+_{Y_1}$, $C^\times_{Y_1}$ and $C^+_{Y_2}$, $C^\times_{Y_2}$ are the complexity of transforms given in (15) and (16), respectively.

Now we define the necessary real operations for the terms

$$T_{N/4,n}Y_{11}[n] \pm T^*_{N/4,n}Y_{12}[n],\ T_{N/4,n}Y_{21}[n] \pm T^*_{N/4,n}Y_{22}[n]$$

without taking the operations for terms $Y_{11}[n]$, $Y_{12}[n]$, $Y_{21}[n]$, and $Y_{22}[n]$ (see (15) and (16).

At first we have

$$T_{N/4,0} = 1, \quad T_{N/4,N/32} = 1-j.$$

Hence, we obtain

Table 5

Expression	Add	Mul
$T_{N/4,n}Y_{11}[n] + T^*_{N/4,n}Y_{12}[n]$	$\frac{3}{8}N-4$	$\frac{1}{4}N-8$
$T_{N/4,n}Y_{11}[n] - T^*_{N/4,n}Y_{12}[n]$	$\frac{1}{8}N$	0
$T_{N/4,n}Y_{21}[n] + T^*_{N/4,n}Y_{22}[n]$	$\frac{3}{8}N-4$	$\frac{1}{4}N-8$
$T_{N/4,n}Y_{21}[n] - T^*_{N/4,n}Y_{22}[n]$	$\frac{1}{8}N$	0

Now using the results of Table 5 without taking the operations for $Y_{i,j}[n]$, $i,j = 0,1,2$ (see (15) and (16) we can define the operations for realization of $Y_1[n], n = \overline{0, N/4-1}$ (see Table below).

Table 6

Expression	Add	Mul
$Y_1[n]$	$\frac{1}{2}N-4$	$\frac{3}{8}N-10$
$Y_1[n+\frac{N}{16}]$	$\frac{1}{4}N$	$\frac{1}{8}N-2$
$Y_1[n+\frac{2N}{16}]$	$\frac{1}{8}N$	0
$Y_1[n+\frac{3N}{16}]$	$\frac{1}{8}N$	0

Note that for $Y_2[n]$ number of required operations is the same as for $Y_1[n]$. Now using the results of Table 6 we can calculate the number of real operations for computing all components of $Y_1[n]$ and $Y_2[n]$ ($n = \overline{0, N/4-1}$, $N \geq 32$)

$$\begin{aligned} C^+_{Y_1} &= N - 4 + C^+_{Y_{10}} + C^+_{Y_{11}} + C^+_{Y_{12}}, \\ C^\times_{Y_1} &= \tfrac{1}{2}N - 12 + C^\times_{Y_{10}} + C^\times_{Y_{11}} + C^\times_{Y_{12}}, \end{aligned} \tag{20}$$

It is not difficult to show that

$$
\begin{aligned}
&C_X^+ = C_N^+,\ C_X^\times = C_N^\times,\\
&C_{Y_0}^+ = C_{N/2}^+,\ C_{Y_0}^\times = C_{N/2}^\times,\\
&C_{Y_{10}}^+ = C_{Y_{20}}^+ = C_{N/8}^+,\\
&C_{Y_{10}}^\times = C_{Y_{20}}^\times = C_{N/8}^\times,\\
&C_{Y_{11}}^+ = C_{Y_{21}}^+ = C_{Y_{12}}^+ = C_{Y_{22}}^+ = C_{N/16}^+,\\
&C_{Y_{11}}^\times = C_{Y_{21}}^\times = C_{Y_{12}}^\times = C_{Y_{22}}^\times = C_{N/16}^\times.
\end{aligned} \tag{21}
$$

Finally using the equations (19),(20) and the identities (21) we obtain the complexity of $N-$point DFT as

$$
\begin{aligned}
C_N^+ &= 6N - 44 + C_{N/2}^+ + 2C_{N/8}^+ + 4C_{N/16}^+,\\
C_N^\times &= 3N - 52 + C_{N/2}^\times + 2C_{N/8}^\times + 4C_{N/16}^\times.
\end{aligned} \tag{22}
$$

Optimization by hand for $N = 16$ has allowed us to save 40-additions and 16-multiplications in comparison with algorithm 3 (see, section 3). Using these results and relations (22) we can obtain

$$
\begin{aligned}
C_N^+ &= \tfrac{8}{3}N\log_2 N - \tfrac{34}{9}N - 2\alpha_N^+\sqrt{N}\\
&\qquad - \tfrac{26}{9}(-1)^{\log_2 N} + \tfrac{22}{3},\\
C_N^\times &= \tfrac{4}{3}N\log_2 N - \tfrac{46}{9}N - \tfrac{8}{9}\alpha_N^\times\sqrt{N}\\
&\qquad - \tfrac{2}{3}(-1)^{\log_2 N} + \tfrac{26}{3}.
\end{aligned} \tag{23}
$$

Values of α_N^+ and $\alpha_N^\times$ are defined below

$\log_2 N (mod\, 4)$	α_N^+	$\alpha_N^\times$
0	$\frac{4}{3}$	1
1	$\sqrt{2}$	$\sqrt{2}$
2	$-\frac{4}{3}$	-1
3	$-\sqrt{2}$	$-\sqrt{2}$

In the Table 7 some numerical results are given.

Table 7

N	Add	Mul	Total
16	104	8	112
32	300	52	352
64	808	200	1008
128	1948	564	2512
256	4456	1416	5872
512	10300	3508	13808
1024	23528	8456	31984
2048	52476	19636	72112
4096	115432	44552	159984
8192	252796	100020	352816
16384	550120	222216	772336
32768	1187452	488116	1675568

5. COMPARISON RESULTS

In 2007 Steven Johnson and Matteo Frigo presented [3] new FFT algorithm which has fewer arithmetic operations than all known FFT algorithms. Algorithm 4 yields savings over their method starting at $N = 2^4$ to $N < 20^{20}$, as summarized in Table 8.

Table 8

N	Add	Add	Mul	Mul	Total	Total
64	808	912	200	240	1008	1152
128	1948	2164	564	628	2512	2792
256	4456	5008	1416	1544	5872	6552
512	10300	11380	3508	3668	13808	15048
1024	23528	25488	8456	8480	31984	33968
2048	52476	56436	19636	19252	72112	75688
4096	115432	123792	44552	43064	159984	166856
8192	252796	269428	100020	95252	352816	364680
16384	550120	582544	222216	208720	772336	791264

Graphical presentation of comparison results for algorithm 4, Johnson-Frigo algorithm and algorithm 3 given below

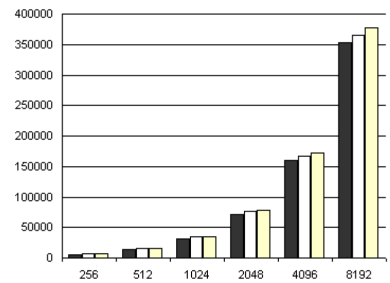

Figure 1: Flop counts of algorithm 4, Johnson-Frigo algorithm, algorithm 3.

6. CONCLUSION

We have introduced a simple, recursive algorithm for the computation of the discrete Fourier transform for $N = 2^p$. The results obtained in Section 3 are the best among the FFT-algorithms ($N < 2^{20}$) for $N = 2^p$. The number of flops can be reduced with manual optimization for higher values of N = 32,64,128,etc.Required flops and comparison with [3] are summarized in Table 8 and Figure 1.Algorithm for real-data and real-symmetric (discrete cosine) transforms with fewer arithmetic operations can be easily derive from our algorithm. In the future we want to reduce number of operations and create framework for parallel computation FFT.

REFERENCES

[1] A.K.Jain. Fundamentals of Digital Image Processing. Prentice-Hall Inc., Englewood Cliffs, 1989.

[2] P. Duhamel and M. Vetterli, Fast Fourier transforms: a tutorial review and a state of the art, Signal Processing - Apr. 1990, vol. 19, pp. 259299.

[3] M. Frigo and S. G. Johnson, "A modified split-radix FFT with fewer arithmetic operations, IEEE TRANS. SIGNAL PROCESSING" - vol. 55, pp. 111-119, 2207.

[4] H.Sarukhanyan, S.Agaian, Conventional, Integer to Integer and Quantized Fast Fourier Transforms, CSIT 2007, pp. 204-207.

[5] Petrovski I.G. Lectures on the theory of ordinary differential equations (in russian),1984.