

РАЗРАБОТКА АЛГОРИТМА БПФ ДЛЯ НЕТРАДИЦИОННОГО ЧИСЛА ТОЧЕК

Просеков О. В.

ФГУП “ЦНИИ “Морфизприбор”, Санкт-Петербург, Россия

Алгоритмы быстрого преобразования Фурье (БПФ) основаны на факторизации матрицы дискретного преобразования Фурье (ДПФ) на сомножители специальной структуры. Такая факторизация определяется далеко не единственным образом. В данной работе указаны простые соображения, позволяющие получить в результате факторизацию, более глубокую, чем обычно, за счет более полного учета симметрии в матрице ДПФ. Она позволяет однозначно и в явном виде записать алгоритм БПФ, обладающий регулярной структурой и высокой вычислительной эффективностью.

Введение

В ряде гидроакустических приложений, в частности, в процедурах формирования характеристик направленности (ФХН), при вычислении свёрток и корреляции, оказывается полезным использование БПФ последовательности данных нетрадиционных длин (не равных степени двух). Разработанный ещё в 1965 году наиболее популярный алгоритм БПФ (Кули-Тьюки) при простоте и наглядности обладает главным недостатком: ограниченным выбором вариантов длин ДПФ (степень двойки). Неблагоприятным следствием популярности алгоритма Кули-Тьюки явилось широкое распространение мнения о том, что ДПФ практически применять лишь при длине преобразования, равной степени двух. Это привело к тому, что алгоритмы БПФ стали диктовать параметры применяемых устройств вместо того, чтобы приложения диктовали выбор подходящего алгоритма БПФ. На самом же деле эффективные алгоритмы БПФ существуют для практически любых длин преобразования. Одним из таких алгоритмов является алгоритм БПФ Винограда (1976-1978). Он более эффективен в вычислительном отношении, чем алгоритм БПФ по основанию два, но примеров практического его использования почти нет вследствие исключительно сложно реализуемой нерегулярной структуры вычислений. В данной работе предложены новые алгоритмы БПФ, которые обладают простотой и наглядностью, и в то же время эффективнее алгоритмов БПФ Винограда. В этой работе продолжается развитие идей изложенных в [2].

Дискретное преобразование Фурье определяется формулой

$$X_k = \sum_{j=0}^{N-1} x_j \omega_N^{-kj}, \quad k = 0, 1, \dots, N-1, \quad (1)$$

где $\omega_N = \exp(2\pi i/N)$, N – длина (порядок) преобразования. Быстрое преобразование Фурье основано на приведении формулы (1) к виду [1, с. 429]

$$X = CBAx. \quad (2)$$

Здесь B – диагональная матрица, ответственная за умножения, A и C – матрицы, элементы которых равны 0, 1 или -1 . Матрицы A и C называются соответственно матрицей предположений и матрицей постсложений.

Факторизация (2) имеет целью минимизировать число умножений. Чтобы минимизировать число сложений, нужно факторизовать матрицы A и C , т. е. перейти к разложению

$$X = C_1 C_2 \dots C_m B A_n A_{n-1} \dots A_1 x, \quad (3)$$

в котором сомножители C_k и A_l обладают следующими свойствами: их элементы по-прежнему равны 0, 1 или -1 , но в каждой строке этих сомножителей содержится минимум отличных от нуля элементов.

Разложение (3) неединственно, поэтому ставятся дополнительные условия. Например, потребуем, чтобы матрицы C_k и A_l были, в основном, симметричными, строки и столбцы линейно независимы, чтобы значение -1 стояло, по возможности, на диагоналях и т. д. Такие дополнительные условия трудно формализовать. Приведем вывод разложений вида (3). При этом будут использоваться только элементарные средства.

Алгоритмы быстрого преобразования Фурье малых порядков

Введем обозначения

$$a_k = \sum_{j=0}^{N-1} x_j \cos\left(\frac{2\pi k}{N} j\right), \quad b_k = -i \sum_{j=1}^{N-1} x_j \sin\left(\frac{2\pi k}{N} j\right) \quad (4)$$

поскольку $a_{N-k} = a_k$ и $b_{N-k} = -b_k$, то формула (1) принимает следующий вид

$$\begin{aligned} X_k &= a_k + b_k, \\ X_{N-k} &= a_k - b_k, \end{aligned} \quad k = 0, 1, \dots, \lfloor N/2 \rfloor. \quad (5)$$

Отметим, что $b_0 = 0$, и при нечетном N формулы (4) можно переписать в виде

$$\begin{aligned} a_0 &= x_0 + \sum_{j=1}^{\lfloor N/2 \rfloor} (x_j + x_{N-j}), \\ a_k &= x_0 + \sum_{j=1}^{\lfloor N/2 \rfloor} (x_j + x_{N-j}) \cos\left(\frac{2\pi k}{N} j\right), \\ b_k &= -i \sum_{j=1}^{\lfloor N/2 \rfloor} (x_j - x_{N-j}) \sin\left(\frac{2\pi k}{N} j\right), \end{aligned} \quad k = 1, \dots, \lfloor N/2 \rfloor. \quad (6)$$

Если N четное, то $b_{N/2} = 0$ и формулы (4) можно переписать в виде

$$\begin{aligned} a_0 &= x_0 + x_{N/2} + \sum_{j=1}^{N/2-1} (x_j + x_{N-j}), \quad a_{N/2} = x_0 + (-1)^{N/2} x_{N/2} + \sum_{j=1}^{N/2-1} (-1)^j (x_j + x_{N-j}), \\ a_k &= x_0 + (-1)^k x_{N/2} + \sum_{j=1}^{N/2-1} (x_j + x_{N-j}) \cos\left(\frac{2\pi k}{N} j\right), \\ b_k &= -i \sum_{j=1}^{N/2-1} (x_j - x_{N-j}) \sin\left(\frac{2\pi k}{N} j\right), \end{aligned} \quad k = 1, \dots, N/2 - 1. \quad (7)$$

Формулы (5) и (6) (или (7)) задают соответственно матрицы C_1 и A_1 в (3). Очевидно, что $C_1 = A_1$. Это простое преобразование приводит матрицу ДПФ к блочно-диагональному виду. Диагональных блоков всего два, и при небольших порядках ДПФ они достаточно легко диагонализируются.

Нам потребуется очевидное соотношение для произведения матриц:

$$A[P, L] \times B[L, Q] = \sum_{l \in L} A[P, l] \times B[l, Q]. \quad (8)$$

Выведем из (8) два следствия. Пусть π – перестановка множества L . Тогда

$$A[P, L] \times B[L, Q] = \sum_{l \in L} A[P, \pi(l)] \times B[\pi(l), Q].$$

Это означает, что произведение AB не изменится, если у матрицы A переставить столбцы и соответствующим образом у матрицы B переставить строки.

Теперь возьмем ненулевые числа c_l при $l \in L$. Имеем

$$A[P, L] \times B[L, Q] = \sum_{l \in L} (A[P, l] \times [c_l^{-1}]) \times ([c_l] \times B[l, Q]).$$

Таким образом, произведение AB не изменится, если l -й столбец матрицы A поделить на c_l , а l -ю строку матрицы B умножить на c_l .

Случай, когда $N = 2$ тривиален и не нуждается в упрощении:

$$\begin{bmatrix} X_0 \\ X_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}.$$

Из формулы видно, что ДПФ порядка 2 задается одной операцией типа “сложить/вычесть”.

Рассмотрим случай $N = 5$. Обозначим $\theta = 2\pi/5$. Согласно (5)-(6) имеем

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & \cos\theta & \cos 2\theta & 0 & 0 \\ 1 & \cos 2\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 0 & i \sin\theta & -i \sin 2\theta \\ 0 & 0 & 0 & -i \sin 2\theta & -i \sin\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (9)$$

Матрица в середине правой части (9) имеет блочно-диагональную структуру. Факторизуем верхний блок. Для этого приведем его к диагональному виду:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \cos \theta & \cos 2\theta \\ 1 & \cos 2\theta & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & \frac{1}{2}(\cos \theta + \cos 2\theta) & 0 \\ 0 & 0 & \frac{1}{2}(\cos \theta - \cos 2\theta) \end{bmatrix}.$$

В правой части получили матрицу блочно-диагонального вида. Верхний блок приводится к диагональному виду следующим образом

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 \\ 1 & \frac{1}{2}(\cos \theta + \cos 2\theta) \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2}(\cos \theta + \cos 2\theta) - 1 \end{bmatrix},$$

тогда

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \cos \theta & \cos 2\theta \\ 1 & \cos 2\theta & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} =$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2}(\cos \theta + \cos 2\theta) - 1 & 0 \\ 0 & 0 & \frac{1}{2}(\cos \theta - \cos 2\theta) \end{bmatrix}.$$

Отсюда следует, что

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & \cos \theta & \cos 2\theta \\ 1 & \cos 2\theta & \cos \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{diag} \begin{bmatrix} 1 \\ \frac{1}{2}(\cos \theta + \cos 2\theta) - 1 \\ \frac{1}{2}(\cos \theta - \cos 2\theta) \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}. \quad (10)$$

Нижний блок факторизуемой матрицы соотношения (9) имеет вид алгоритма комплексного умножения [1, с. 14]

$$\begin{bmatrix} i \sin \theta & -i \sin 2\theta \\ -i \sin 2\theta & -i \sin \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \text{diag} \begin{bmatrix} i(\sin \theta + \sin 2\theta) \\ -i(\sin \theta - \sin 2\theta) \\ -i \sin 2\theta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}. \quad (11)$$

Подставив (10) и (11) в (9), приходим к окончательной формуле

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times$$

$$\times \text{diag} \begin{bmatrix} 1 \\ \frac{1}{2}(\cos \theta + \cos 2\theta) - 1 \\ \frac{1}{2}(\cos \theta - \cos 2\theta) \\ i(\sin \theta + \sin 2\theta) \\ -i(\sin \theta - \sin 2\theta) \\ -i \sin 2\theta \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (12)$$

Разложение (12) можно представить в виде $X = A_1 A_2 A_3^T M A_3 A_2 A_1 x$, где A_1, A_2 – симметричные матрицы.

На Рис. 1 приведена графическая схема алгоритма (12).

Число умножений в полученном алгоритме БПФ порядка 5 равно 5, число сложений равно 17, из них: 6 операций типа “сложить/вычесть”, 2 сложения, не влияющие на размер диагонали, и 3 сложения, которые отвечают за увеличение диагонали.

В Табл. 1 приведены характеристики алгоритмов, построенных по показанному выше методу. Число арифметических операций в новых алгоритмах БПФ совпадает с числом арифметических операций ма-

лых алгоритмов БПФ Винограда, за исключением алгоритма БПФ порядка 9, в котором на 2 сложения меньше, чем у аналогичного алгоритма БПФ Винограда [1, с. 433]. Остается заметить, что при комплексном входе число операций в алгоритме удваивается.

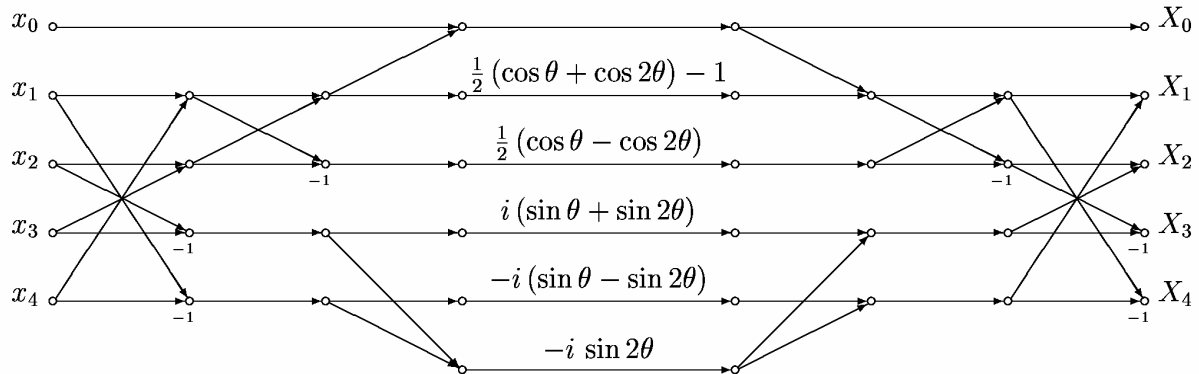


Рис. 1. Графическая схема алгоритма БПФ порядка 5

Таблица 1. Характеристики алгоритмов БПФ малых порядков

Порядок алгоритма БПФ	Матричный вид алгоритма БПФ	Умножения		Сложения			
		Размер диагонали	Число умножений	Число операций типа "сложить вычесть"	Число сложений	Число сложений диагонали	Общее число сложений
2	A_1	2	–	1	–	–	2
3	$A_1 A_2^T M A_2 A_1$	3	2	2	2	–	6
4	$A_1 A_2 M P_1 A_2 A_1$	4	–	4	–	–	8
5	$A_1 A_2 A_3^T M A_3 A_2 A_1$	6	5	6	2	3	17
6	$A_1 A_3^T M A_3 P_1 A_2 A_1$	6	4	7	4	–	18
7	$A_1 A_2 A_3^T M A_3 A_2 A_1$	9	8	10	10	6	36
8	$A_1 A_2 M A_3 P_1 A_2 A_1$	8	2	13	–	–	26
9	$A_1 P_1 A_2 A_4 M A_3 A_2 P_1 A_1$	12	10	12	11	7	42
10	$A_1 P_1 A_3 A_4^T M A_4 A_3 A_2 P_1 A_1$	12	10	17	4	6	44
12	$A_1 A_2 A_4 M A_3 P_1 A_2 A_1$	12	8	20	8	–	48
16	$A_1 A_2 A_3 A_4^T M A_4 P_1 A_3 A_2 A_1$	18	10	34	–	6	74

Алгоритмы быстрого преобразования Фурье больших порядков

Рассмотрим случай, когда $N = n_1 n_2$. Факторизация длины ДПФ позволяет перейти от одномерного ДПФ к двумерному. Одномерная последовательность отображается в $(n_1 \times n_2)$ -таблицу, после выполнения над этой таблицей двумерного ДПФ компоненты преобразования отображаются в выходную последовательность. Способы отображения входных и выходных данных будут описаны ниже.

Двумерное дискретное преобразование Фурье определяется формулой

$$Y[k_1, k_2] = \sum_{j_1=0}^{n_1-1} \sum_{j_2=0}^{n_2-1} y[j_1, j_2] \omega_{n_1}^{-k_1 j_1} \omega_{n_2}^{-k_2 j_2}, \quad \begin{matrix} k_1 = 0, \dots, n_1 - 1 \\ k_2 = 0, \dots, n_2 - 1 \end{matrix} \quad (13)$$

Если рассмотреть двумерные таблицы y и Y в виде одномерных последовательностей \tilde{y} и \tilde{Y} , записывая их в стеки по строкам, то получим (13) в матричном виде [1, с. 268]

$$\tilde{Y} = F_{n_1} \otimes F_{n_2} \tilde{y}, \quad (14)$$

где F_{n_1} и F_{n_2} – матрицы ДПФ порядков n_1 и n_2 соответственно, а символ \otimes обозначает тензорное (кронекеровское) произведение. Матрицы F_{n_1} и F_{n_2} соответствуют алгоритмам БПФ малых порядков, которые получены в виде (3), т. е.

$$\begin{aligned} F_{n_1} &= A_1^{n_1} C_2^{n_1} \dots C_m^{n_1} B^{n_1} A_n^{n_1} A_{n-1}^{n_1} \dots A_1^{n_1}, \\ F_{n_2} &= A_1^{n_2} C_2^{n_2} \dots C_m^{n_2} B^{n_2} A_n^{n_2} A_{n-1}^{n_2} \dots A_1^{n_2}. \end{aligned} \quad (15)$$

Нам потребуется очевидное соотношение для тензорного произведения матриц:

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD). \quad (16)$$

Подставив (15) в (14) и используя (16) получим

$$\tilde{Y} = (A_1^{n_1} \otimes A_1^{n_2}) (C_2^{n_1} \otimes C_2^{n_2}) \dots (C_m^{n_1} \otimes C_m^{n_2}) (B^{n_1} \otimes B^{n_2}) (A_n^{n_1} \otimes A_n^{n_2}) (A_{n-1}^{n_1} \otimes A_{n-1}^{n_2}) \dots (A_1^{n_1} \otimes A_1^{n_2}) \tilde{y}, \quad (17)$$

т. е. разложение вида (3). Заметим, что, как и в случае алгоритмов БПФ малых порядков $C_1 = A_1$.

Отображение входных и выходных данных при переходе к двумерному ДПФ основано на китайской теореме об остатках для целых чисел. Пусть $\text{НОД}(n_1, n_2) = 1$. Тогда правило отображения входных и выходных индексов вычисляются по правилу [1, с. 141]

$$\begin{aligned} y[j_1, j_2] &= x(j_1 n_2 + j_2 n_1 \pmod{N}) & j_1 &= 0, \dots, n_1 - 1 \\ Y[k \pmod{n_1}, k \pmod{n_2}] &= X(k) & j_2 &= 0, \dots, n_2 - 1 \\ & & k &= 0, \dots, N - 1, \end{aligned}$$

или в матричном виде

$$\tilde{y} = P x, \quad \tilde{Y} = R X,$$

где P и R – матрицы перестановок, определяемые следующим образом

$$\begin{aligned} P[l, p] &= \begin{cases} 1, & \text{при } l = j_1 n_2 + j_2 u \text{ и } p = j_1 n_2 + j_2 n_1 \pmod{N} \\ 0, & \text{при других } l \text{ и } p \end{cases} \\ R[l, p] &= \begin{cases} 1, & \text{при } l = (k \pmod{n_1}) \cdot n_2 + (k \pmod{n_2}) \text{ и } p = k \\ 0, & \text{при других } l \text{ и } p. \end{cases} \end{aligned} \quad (18)$$

Таким образом, формулу (17) можно записать

$$X = R^T A_1 C_2 \dots C_m B A_n A_{n-1} \dots A_1 P x, \quad (19)$$

где $A_l = A_l^{n_1} \otimes A_l^{n_2}$, $C_k = C_k^{n_1} \otimes C_k^{n_2}$ и $B = B^{n_1} \otimes B^{n_2}$.

Оценим вычислительную сложность полученного алгоритма. Пусть $M(N)$ – число умножений, $A(N)$ – число сложений, $a(N)$ – число дополнительных сложений, а $AS(N)$ – число операций типа “сложить/вычесть” в алгоритме БПФ. Тогда число арифметических операций в алгоритме БПФ задается формулами

$$\begin{aligned} M(N) &= m(n_1) m(n_2) - (m(n_1) - M(n_1))(m(n_2) - M(n_2)), \\ a(N) &= n_1 a(n_2) + m(n_2) a(n_1), \\ A(N) &= n_1 A(n_2) + n_2 A(n_1) + a(N), \\ AS(N) &= n_1 AS(n_2) + n_2 AS(n_1), \end{aligned}$$

где $m(n_1)$, $m(n_2)$ – длины диагоналей, $M(n_1)$, $M(n_2)$ – число умножений, $A(n_1)$, $A(n_2)$ – число сложений, $a(n_1)$, $a(n_2)$ – число сложений, зависящих от размера диагонали, и $AS(n_1)$, $AS(n_2)$ – число операций типа “сложить/вычесть” в соответствующих алгоритмах БПФ малых порядков n_1 и n_2 . Число умножений новых алгоритмов БПФ совпадает с числом умножений больших алгоритмов БПФ Винограда [1, с. 270], но число сложений значительно уменьшилось за счет факторизации матриц A и C в (2).

В заключение заметим, что если поменять местами схемы индексации входа и выхода (18), то (19) примет вид

$$X = P^T A_1 C_2 \dots C_m B A_n A_{n-1} \dots A_1 R x.$$

Рассмотрим пример. Пусть $n_1 = 5$ и $n_2 = 16$, тогда $N = 80$. Алгоритм БПФ порядка 16 имеет вид

$$F_{16} = A_1^{16} A_2^{16} A_3^{16} A_4^{16T} M^{16} A_4^{16} P_1^{16} A_3^{16} A_2^{16} A_1^{16}.$$

Алгоритм БПФ порядка 5 имеет вид

$$F_5 = A_1^5 A_2^5 A_3^{5T} M^5 A_3^5 A_2^5 A_1^5.$$

Для того чтобы воспользоваться формулой (19) перепишем последнее равенство следующим образом:

$$F_5 = A_1^5 A_2^5 I^5 A_3^{5T} M^5 A_3^5 I^5 A_2^5 A_1^5,$$

где I^5 – единичная (5×5)-матрица. Таким образом, приходим к окончательной формуле

$$X = R^T A_1 A_2 A_3 A_4^T M A_4 P_1 A_3 A_2 A_1 P x,$$

где $A_1 = A_1^5 \otimes A_1^{16}$, $A_2 = A_2^5 \otimes A_2^{16}$, $A_3 = I^5 \otimes A_3^{16}$, $A_4 = A_4^5 \otimes A_4^{16}$, $P_1 = I^5 \otimes P_1^{16}$, $M = M^5 \otimes M^{16}$ и матрицы P и R определяются по формуле (18).

Число умножений в полученном алгоритме БПФ порядка 80 равно 100, число сложений равно 648, из них: 266 операций типа “сложить/вычесть”, 32 сложения плюс 84 дополнительных сложений. При комплексном входе общее количество арифметических операций в записанном алгоритме составляет 1496, а в аналогичном алгоритме БПФ Винограда число арифметических операций равно 1568. При вычислении ДПФ длины 80 алгоритмом БПФ по основанию два длины 128 потребуется 4480 арифметических операций.

Заключение

Полученный алгоритм БПФ имеет более простую структуру, чем алгоритм БПФ Винограда, и в тоже время эффективнее в вычислительном отношении. Программная реализация алгоритма в значительной степени упрощается за счет того, что матрицы в (3) содержат операции типа “сложить/вычесть”. Большинство этих матриц одинаковы, что позволяет сократить количество программных модулей.

На Рис. 2 приведены графики вычислительной эффективности алгоритма БПФ по основанию два, алгоритма БПФ Винограда и представленного в данной работе нового алгоритма БПФ. Алгоритм Винограда эффективнее алгоритма БПФ по основанию два на 15-60% (в среднем на 37%). Новый алгоритм БПФ эффективней алгоритма БПФ Винограда на 0-17% (в среднем на 5%).

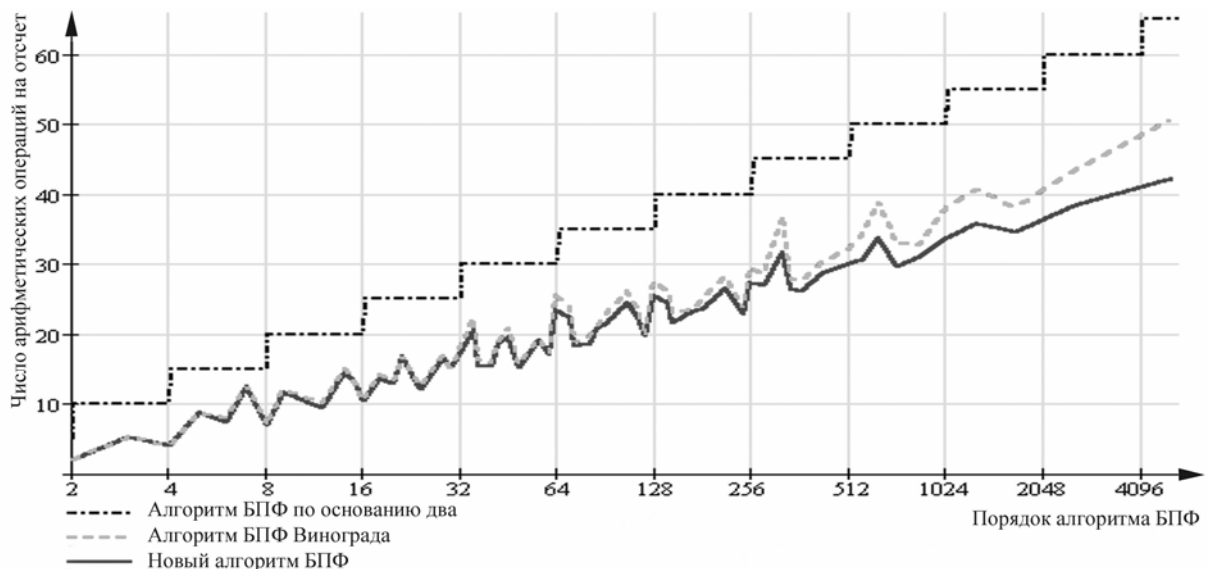


Рис. 2. Вычислительная эффективность алгоритмов БПФ

СПИСОК ЛИТЕРАТУРЫ

1. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. / Пер. с англ. М., 1989. 448 с. (Richard E. Blahut, Fast Algorithms for Digital Signal Processing. Addison-Wesley, 1985)
2. Малоземов В. Н., Просеков О. В. О быстром преобразовании Фурье малых порядков. / Вестн. С.-Петербург. ун-та. Сер. 1. 2003. Вып. 1. С. 36-45.