УДК 519.6:681.3

КУСОЧНО-ПОЛИНОМИАЛЬНАЯ АППРОКСИМАЦИЯ ФУНКЦИЙ НА ОСНОВЕ ИНТЕРПОЛЯЦИИ ПО НЬЮТОНУ В ПРИЛОЖЕНИИ К ПАРАЛЛЕЛЬНОЙ ЦИФРОВОЙ ОБРАБОТКЕ СИГНАЛОВ

Я.Е. Ромм, Л.Н. Аксайская

ГОУ ВПО «Таганрогский государственный педагогический институт», г. Таганрог

Рецензент Н.П. Пучков

Ключевые слова: вычисление функций; параллельные схемы; цифровая обработка сигналов.

Аннотация: Изложена компьютерная схема оптимизированной кусочно-полиномиальной аппроксимации функций, производных и определенных интегралов на основе интерполяционного полинома Ньютона для произвольно заданной границы погрешности. Параллельно вычисляется базис дискретных преобразований Фурье с логарифмической оценкой временной сложности при произвольном количестве отсчетов, при этом из асимптотики вычислительного алгоритма исключаются операции умножения.

Постановка вопроса. В работе представлены схемы оптимизации временной сложности компьютерного вычисления функций, производных и определенных интегралов на основе интерполяционного полинома Ньютона для произвольно задаваемой границы абсолютной погрешности. Обсуждаются аспекты построения последовательных и параллельных вычислительных алгоритмов. Параллельные алгоритмы прилагаются к вычислению элементов базиса, коэффициентов, а также непосредственно к выполнению дискретных прямых и обратных преобразований Фурье. Помимо того, рассматривается их приложение к другим схемам цифровой обработки сигналов. Временная сложность $T(R)$, где R – число процессоров, предлагаемых алгоритмов оценивается на модели неветвящихся парал-

Ромм Я.Е. – доктор технических наук, профессор, заведующий кафедрой «Информатика» Таганрогского государственного педагогического института, г. Таганрог; Аксайская Л.Н. – ассистент кафедры «Математика» Таганрогского государственного педагогического института, г. Таганрог.

лельных программ [7, 8]. При этом она измеряется количеством последовательных шагов алгоритма без учета обмена.

Общая цель работы – показать минимальную временную сложность и эффективность компьютерной реализации кусочно-полиномиальных аппроксимаций функций на основе интерполяции по Ньютону. В частности, в приложении к параллельной цифровой обработке сигналов будет показана логарифмическая оценка временной сложности дискретных преобразований с исключением из асимптотики операций умножения для произвольного количества отсчетов независимой переменной.

Кусочно-полиномиальная аппроксимация функций на основе интерполяции по Ньютону. Ниже излагаются схемы аппроксимации функций на основе интерполяционного полинома Ньютона с оптимизацией временной сложности. Минимизация степени интерполяционного полинома Ньютона [4, 5] достигается за счет сужения подынтервалов аппроксимации. Для функции одной действительной переменной вида

$$y = f(x), \quad x \in [a, b], \tag{1}$$

выбирается система непересекающихся подынтервалов равной длины:

$$[a,b] = \bigcup_{i=0}^{P-1} \left[x_i, x_{i+1} \right), \tag{2}$$

$$x_{i+1} - x_i = (b-a)/P, \quad i = 0, 1, \ldots, P-1, \quad P = 2^k, \quad k \in \{0, 1, \ldots\}. \tag{3}$$

При априори заданной границе ε погрешности аппроксимации функции (1) для каждого отдельного подынтервала из (2), (3) строится интерполяционный полином Ньютона степени n, где n выбирается минимальным для заданной точности приближения одновременно на всех подынтервалах. Полином преобразуется по дистрибутивности так, что принимает канонический вид для каждого подынтервала:

$$P_n(x) = a_{0if} + a_{1if}x + a_{2if}x^2 + \ldots + a_{nif}x^n, \ x \in [x_i, x_{i+1}),$$
$$n = \text{const}, \ i = \overline{0, P-1}. \tag{4}$$

Построение (4) выполняется для всех подынтервалов при условии:

$$\left| f(x) - P_n(x) \right| \le \varepsilon, \quad x \in [x_i, x_{i+1}), \ i = 0, 1, \ldots, P-1. \tag{5}$$

Если в (4), (5) минимально возможное n, одинаковое для всех подынтервалов, найдено, то для функции (1) и каждого подынтервала из (2) набор коэффициентов из (4) делается хранимым в памяти компьютера. Для вычисления функции дешифруется значение номера i подынтервала, которое служит математическим адресом выборки коэффициентов (4). Если $x \in [x_i, x_{i+1})$, то, в обозначении $h = x_{i+1} - x_i$, дешифрация индекса осуществляется по формуле $i = \operatorname{int}\left(\dfrac{x-a}{h}\right)$, где int – целая часть числа, a – из (1). Если для рассматриваемой функции вычислить и хранить коэффициенты для всех 2^k подынтервалов, то время вычисления функции зависит только от степени полинома (4). По схеме Горнера значение этого полинома вычисляется с временной сложностью $t(1) = n(t_у + t_с)$, где $t_у$, $t_с$ –

время бинарного умножения и сложения. За счет сужения подынтервала степень n можно сделать «сколь угодно» малой при соответственном возрастании P в (3). Схема минимизации степени полинома Ньютона конкретно осуществляется следующим образом. Если границы i-го подынтервала из (2), (3) обозначить как $a_{i\,0}$, $b_{i\,0}$, шаг интерполяции – $w_i = \dfrac{b_{i0} - a_{i0}}{n}$, то равноотстоящие узлы интерполяции на текущем шаге примут вид

$$x_{i\,j} = a_{i\,0} + j\,w_i\ , \quad j = 0, 1, ..., n-1\,. \tag{6}$$

Полином Ньютона на i-м подынтервале записывается в виде $\Psi_{n\,i}(x) = f(x_{i\,0}) + \sum_{j=1}^{n} \dfrac{\Delta^j y_{i\,0}}{j!\,w_i^j} \prod_{k=0}^{j-1} (x - x_{i\,k})$, где $\Delta^j y_{i\,0}$ – конечная разность j-го порядка в точке $x_{i\,0}$. Пусть

$$t = \frac{x - x_{i\,0}}{w_i}\,, \tag{7}$$

тогда [2]

$$\Psi_{n\,i}(t) = f(x_{i\,0}) + \sum_{j=1}^{n} \frac{\Delta^j y_{i\,0}}{j!} \prod_{k=0}^{j-1} (t - k)\ . \tag{8}$$

Процесс приведения (8) к виду (4) влечет значения коэффициентов

$$a_{0\,i\,f} = f(x_{i0})\,,\ a_{l\,i\,f} = \sum_{j=l}^{n} b_{i\,j} d_{l\,j}\ , \tag{9}$$

где $b_{i\,j} = \dfrac{\Delta^j y_{i0}}{j!}$, $d_{l\,j}$ – коэффициенты полиномов вида $P_{n\,j}(t) = d_{0\,j} + d_{1j}\,t + d_{2j}\,t^2 + ... + d_{n\,j}\,t^n$ с натуральными корнями, входящими в состав полинома (8).

Таблица 1

Значения степеней полинома Ньютона, интерполирующего функцию $y = 1/x$, $x \in [1/2, 1]$

ε \ k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
10^{-4}	7	5	4	3	2	2	2	1	1	1	1	1	1	1	1	1	1	1
10^{-5}	8	6	4	4	3	2	2	2	1	1	1	1	1	1	1	1	1	1
10^{-6}	10	7	5	4	3	3	2	2	2	2	1	1	1	1	1	1	1	1
10^{-7}	12	8	6	5	4	3	3	2	2	2	2	2	1	1	1	1	1	1
10^{-8}		10	7	6	4	4	3	3	3	2	2	2	2	2	1	1	1	1
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
10^{-17}				12	10	8	7	6	5	5	4	4	4	4	3	3	3	3
10^{-18}				12		9		8	6	5	5	4	4	4	4	3	3	3

Во входном столбце таблицы $\varepsilon = 10^{-4}, 10^{-5}$, ... обозначает априори задаваемую границу погрешности. Во входной строке k задает показатель степени $P = 2^k$ количества подынтервалов из (2), (3). На пересечении строки, содержащей ε, и столбца, содержащего k, указывается минимальное значение степени n, при которой функция аппроксимируется полиномом данной степени с точностью до ε на каждом из 2^k подынтервалов. Пустующая клетка означает, что в используемой версии языка программирования ε оказалось недостижимым для данного числа подынтервалов ни при одном значении n.

Вычисление определенного интеграла на основе кусочно-полиномиальной аппроксимации подынтегральной функции. Схема минимизации степени полинома Ньютона, интерполирующего функцию (1), позволяет построить схему аппроксимации определенного интеграла и производной [5].

Пусть имеют место соотношения (1) – (3). Вычисление интеграла от функции (1) по промежутку $[a,b]$ по аддитивности сводится к сумме интегралов по подынтервалам $\left[x_i, x_{i+1}\right)$ из (2), (3):

$$\int_a^b f\left(x\right)dx = \sum_{i=1}^{P-1} \int_{x_i}^{x_{i+1}} f(x)dx. \tag{10}$$

Для *i*-го подынтервала справедливо соотношение

$$f(x) \approx P_{ni}(x), \quad \forall\, x \in \left[x_i, x_{i+1}\right), \tag{11}$$

где $P_{n\,i}(x)$ из (5). Отсюда

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx \int_{x_i}^{x_{i+1}} P_{ni}(x)dx. \tag{12}$$

С учетом (10)

$$\int_a^b f\left(x\right)dx \approx \sum_{i=1}^{P-1} \int_{x_i}^{x_{i+1}} P_{ni}(x)dx. \tag{13}$$

Для интерполяции по Ньютону используется замена переменной (7). При ее выполнении значениям $x = x_i$ и $x = x_{i+1}$ соответствуют $t = 0$ и $t = n$. С учетом $P_{n\,i}\left(x\right) = \Psi_{n\,i}\left(t\right)$ получится $P_{n\,i}(x)dx = w_i \Psi_{ni}(t)dt$. В результате

$$\int_{x_i}^{x_{i+1}} P_{ni}(x)dx = w_i \int_0^n \Psi_{ni}(t)\,dt. \tag{14}$$

Интеграл в правой части вычисляется непосредственно:

$$w_i \int_0^n \Psi_{n\,i}(t)dt = w_i\ \tilde{\Psi}_{(n+1)\,i}(t)\Big|_0^n, \tag{15}$$

где

$$\tilde{\Psi}_{(n+1)\,i}\left(t\right) = c + \frac{a_{0if}}{1}t + \frac{a_{1if}}{2}t^2 + \frac{a_{2if}}{3}t^3 + \ldots + \frac{a_{nif}}{n+1}t^{n+1}. \qquad (16)$$

Соотношения (13) – (16) используются для построения формул приближенного вычисления интеграла. Для минимальной степени n интерполирующего функцию полинома и соответствующего значения k строится приближение определенного интеграла функции с помощью вычисления определенного интеграла от интерполирующего эту функцию полинома.

Подставляя верхние и нижние пределы интегрирования в правую часть (15), получаем

$$w_i \tilde{\Psi}_{(n+1)\,i}(t)\Big|_0^n = w_i \tilde{\Psi}_{(n+1)\,i}(n). \qquad (17)$$

Учитывая (17), равенство (15) можно записать в виде

$$w_i \int_0^n \Psi_{n\,i}(t)\,dt = w_i \tilde{\Psi}_{(n+1)\,i}(n). \qquad (18)$$

Значения $\tilde{\Psi}_{(n+1)\,i}(n)$ вычисляются по схеме Горнера

$$\tilde{\Psi}_{(n+1)i}\left(n\right) = \left(\ldots\left(\left(\frac{a_{nif}}{n+1}n + \frac{a_{(n-1)if}}{n}\right)n + \frac{a_{(n-2)if}}{n-1}\right)n + \cdots + \frac{a_{0if}}{1}\right)n. \qquad (19)$$

Замечание 1. Если функция наперед известна, то коэффициенты $\frac{a_{0if}}{1}, \frac{a_{1if}}{2}, \frac{a_{2if}}{3}, \ldots, \frac{a_{nif}}{n+1}$ в (16) можно рассчитать и занести в память компьютера для постоянного хранения, при этом вычисление интеграла в дальнейшем будет осуществляться путем выборки коэффициентов, соответствующих заданной функции и заданному подынтервалу, по аналогии с таблично-алгоритмической схемой аппроксимации функций. Более того, в данном частном случае можно хранить заранее вычисленные значения (19) для каждого номера подынтервала.

В общем случае, используя свойство аддитивности интеграла по длине промежутка, с учетом (13) – (19), получаем значение интеграла на всем промежутке $[a,b]$:

$$\int_a^b f\,(\,x\,)\,d\,x \approx \sum_{i=1}^{P-1} w_i \tilde{\Psi}_{(n+1)\,i}(n). \qquad (20)$$

Существенно, что формула (20) строится для минимизирующего значение показателя n алгоритма кусочно-полиномиальной аппроксимации. Этого значения, как показывает эксперимент, оказывается достаточным для достижения наперед заданной, сравнительно высокой, точности вычисления определенного интеграла. Экспериментальная проверка достигнутой таким путем точности вычисления определенных интегралов осуществлялась сравнением с точными значениями интегралов, вычисляемыми с помощью первообразных.

В случае существования первообразной $F(x)$ точное значение интеграла (20) вычислялось по формуле

$$\int_a^b f(x)\,dx = F(b) - F(a),$$

погрешность приближения – путем вычисления разности $F(b) - F(a) - \sum_{i=0}^{P-1} w_i \tilde{\Psi}_{(n+1)\,i}(n)$.

Аналогичные разности составлялись для формулы трапеций и для метода Симпсона с целью сопоставления погрешности приближения (20) с погрешностью известных методов.

В обозначениях $\delta = \left| F(b) - F(a) - \sum_{i=0}^{n} w_i \tilde{\Psi}_{(n+1)\,i}(n) \right|$, ε – априори заданная граница погрешности кусочно-полиномиального приближения подынтегральной функции, результаты эксперимента занесены в приводимую ниже табл. 2. Вычисление по формуле (20) выполнялось на подынтервалах длины $h = \frac{b-a}{2^k}$, которая принималась за шаг интегрирования сравниваемых методов.

Таблица 2

Сравнение погрешностей интегрирования функции

$f(x) = \sin(x)$, $[a,\ b] = \left[1,\ \frac{3}{2}\right]$, $\varepsilon = 10^{-10}$

n	k	δ для метода на основе полинома Ньютона степени n	δ для метода трапеции	δ для метода Симпсона
7	0	$7{,}24 \cdot 10^{-13}$	$9{,}82 \cdot 10^{-3}$	$1{,}03 \cdot 10^{-5}$
6	1	$4{,}58 \cdot 10^{-15}$	$2{,}45 \cdot 10^{-3}$	$6{,}38 \cdot 10^{-7}$
5	2	$5{,}22 \cdot 10^{-13}$	$6{,}12 \cdot 10^{-4}$	$3{,}98 \cdot 10^{-8}$
4	3	$1{,}45 \cdot 10^{-14}$	$1{,}53 \cdot 10^{-4}$	$2{,}49 \cdot 10^{-9}$
3	4	$2{,}26 \cdot 10^{-16}$	$3{,}82 \cdot 10^{-5}$	$1{,}55 \cdot 10^{-10}$
3	5	$4{,}32 \cdot 10^{-12}$	$9{,}55 \cdot 10^{-6}$	$9{,}72 \cdot 10^{-12}$
3	6	$2{,}7 \cdot 10^{-13}$	$2{,}39 \cdot 10^{-6}$	$6{,}07 \cdot 10^{-13}$
3	7	$1{,}69 \cdot 10^{-14}$	$5{,}97 \cdot 10^{-7}$	$3{,}8 \cdot 10^{-14}$
2	8	$2{,}37 \cdot 10^{-15}$	$1{,}49 \cdot 10^{-7}$	$2{,}37 \cdot 10^{-15}$
2	9	$1{,}48 \cdot 10^{-16}$	$3{,}73 \cdot 10^{-8}$	$1{,}47 \cdot 10^{-16}$
2	10	$9{,}57 \cdot 10^{-18}$	$9{,}33 \cdot 10^{-9}$	$9{,}57 \cdot 10^{-18}$
2	11	$4{,}88 \cdot 10^{-19}$	$2{,}33 \cdot 10^{-9}$	$4{,}88 \cdot 10^{-19}$
2	12	$3{,}25 \cdot 10^{-19}$	$5{,}83 \cdot 10^{-10}$	$2{,}98 \cdot 10^{-19}$
2	13	$4{,}07 \cdot 10^{-19}$	$1{,}46 \cdot 10^{-10}$	$6{,}23 \cdot 10^{-19}$
2	14	$6{,}51 \cdot 10^{-19}$	$3{,}64 \cdot 10^{-11}$	$6{,}23 \cdot 10^{-19}$
1	15	$9{,}11 \cdot 10^{-12}$	$9{,}11 \cdot 10^{-12}$	$6{,}78 \cdot 10^{-19}$
1	16	$2{,}28 \cdot 10^{-12}$	$2{,}28 \cdot 10^{-12}$	$1{,}52 \cdot 10^{-18}$
1	17	$5{,}69 \cdot 10^{-13}$	$5{,}69 \cdot 10^{-13}$	$1{,}17 \cdot 10^{-18}$

Результаты эксперимента показывают, что на основе полинома Ньютона может достигаться более высокая точность приближенного вычисления интеграла, чем по методу Симпсона или по формуле трапеций. Следует заметить, что вычисление интеграла по предложенной схеме с точностью 10^{-13} осуществляется для полинома седьмой степени на всем промежутке. В этом случае не требуется перехода к формулам численного интегрирования с мелким шагом. Для сравнения, та же точность вычисления интеграла по формуле Симпсона достигается только на 2^6 подынтервалах, это означает в 64 раза больший объем вычисления; формула трапеций такой точности достигает при использовании 2^{17} подынтервалов.

Помимо достижения высокой точности интегрирования без использования шага или без интегрального разбиения на промежутки достигается вторая отличительная особенность – быстродействие вычисления определенного интеграла, если наперед известен вид подынтегральной функции и промежуток интегрирования, поскольку в этом случае коэффициенты $\frac{a_{0if}}{1}, \frac{a_{1if}}{2}, \frac{a_{2if}}{3}, \dots, \frac{a_{nif}}{n+1}$ из (16), (19) являются хранимыми.

Отметим, что полученные формулы интегрирования отличаются стабильной точностью вычисления, как правило, более высокой по сравнению с известными методами. В частности, они не уступают по точности формулам трапеции и методу Симпсона при равенстве значения шага, а в пределе всегда достигают более высокой точности вычисления.

Изложенный метод (с целью экспериментальной оценки погрешности) проиллюстрирован для случая приближенного вычисления интеграла, когда первообразная известна. Очевидно, в общем случае это никак не исключает применения метода, когда первообразная неизвестна.

На основе данного метода можно вычислять интегралы вида $E_i(x) = \int\limits_{-\infty}^{x} \frac{e^x}{x} dx, \ (x<0)$ – интегральная показательная функция, $l_i(x) = \int\limits_0^x \frac{dx}{\ln x}, \ (x>0)$ – интегральный логарифм, $C(x) = \int\limits_0^x \cos\frac{\pi}{2}x^2 dx$, $S(x) = \int\limits_0^x \sin\frac{\pi}{2}x^2 dx$ – интегралы Френеля и др [3].

Вычисление производной на основе кусочно-полиномиальной интерполяции по Ньютону. Схема кусочно-полиномиальной аппроксимации функций следующим образом применима для вычисления производных. Как отмечалось, для i-го подынтервала из (2), (3) выполнено соотношение

$$f(x) \approx P_{ni}(x), \quad \forall\, x \in \left[x_i, x_{i+1}\right),$$

где $P_{n\,i}(x)$ из (5). При интерполяции по Ньютону $P_{ni}(x) = \Psi_{ni}(t)$ в этом случае

$$f(x) \approx \Psi_{ni}(t). \tag{21}$$

Взятие производной по x от обеих частей (21) влечет приближение

$$(f(x))'_x \approx (\Psi_{ni}(t))'_x .$$

С учетом (7) получится

$$f'(x) \approx (\Psi_{ni}(t))'_t \, t'_x ,$$

где

$$(\Psi_{ni}(t))'_t = a_{1if} + 2a_{2if}t + 3a_{3if}t^2 + ... + n\, a_{nif}t^{(n-1)}, \qquad t'_x = \frac{1}{w_i}.$$

Значение $\Psi'_{n\,i}(t)$ вычисляется по схеме Горнера:

$$\Psi'_{n\,i}(t) = \left(\left(\cdots\left(n\,a_{nif}t + (n-1)\,a_{(n-1)if}\right)\, t + \cdots + 2\,a_{(n-2)if}\right)t + a_{1\,i\,f}\right)\frac{1}{w_i}. \quad (22)$$

Соотношение (22) дает формулу кусочно-полиномиального приближения производных. С целью экспериментальной оценки погрешности данного приближения на каждом подынтервале из (2), (3) в проверочных точках (7) выполнялось сравнение (22) с аналитическим значением производной. Результаты сравнения сведены в табл. 3.

Здесь n – степень полинома Ньютона, 2^k – число подынтервалов, при этом сама функция $f(x) = \sin(x)$ вычислялась с точностью $\varepsilon = 10^{-19}$. Из табл. 3 видно, что с помощью предложенной схемы производная функции вычисляется практически с точностью аппроксимации функции. Существенно, что функция и производная могут вычисляться одновременно. При

Таблица 3

Погрешность кусочно-полиномиальной аппроксимации производной функции $f(x) = \sin(x)$, $[\,a,\ b\,] = \left[\,1,\ \frac{3}{2}\,\right]$, $\varepsilon = 10^{-19}$

n	k	Погрешность вычисления производной
11	0	$9{,}048 \cdot 10^{-17}$
10	1	$-5{,}421 \cdot 10^{-20}$
10	2	0
8	3	$9{,}926 \cdot 10^{-17}$
7	4	$9{,}758 \cdot 10^{-19}$
7	5	0
10	6	$-5{,}963 \cdot 10^{-19}$
5	7	$7{,}589 \cdot 10^{-19}$
5	8	$-2{,}602 \cdot 10^{-18}$
4	9	$-2{,}711 \cdot 10^{-19}$
4	10	$5{,}421 \cdot 10^{-19}$
4	11	$-2{,}873 \cdot 10^{-18}$
4	12	$-7{,}589 \cdot 10^{-19}$
4	13	$-3{,}524 \cdot 10^{-19}$
4	14	$-9{,}487 \cdot 10^{-19}$
4	15	$-2{,}927 \cdot 10^{-18}$

построении библиотеки стандартных подпрограмм, для случая наперед известной функции, коэффициенты аппроксимирующего полинома могут быть хранимыми для каждого подынтервала. В этом случае производная, как и функция, будет вычисляться за время n сложений и умножений (в табл. 3 достигается $n = 4$ для $\varepsilon = 10^{-19}$).

Предложенный способ кусочно-полиномиальной аппроксимации можно оценить временной сложностью $T = O(1)$ для вычисления стандартного вида функции и производной от нее одновременно, в эту оценку можно включить приближенное вычисление интеграла от той же функции на заданном промежутке.

Замечание 2. Для нескольких произвольно заданных значений аргумента функции, аппроксимируемой по предложенной схеме, вычисление коэффициентов (9) сохраняет синхронность и взаимную независимость для всех номеров подынтервалов (2), (3). Это влечет возможность параллельного вычисления функции одновременно по всем подынтервалам. Помимо того, параллелизм вычисления распространяется одновременно на все степени аппроксимирующих полиномов, на все значения границ погрешности и на наборы одновременно вычисляемых функций с готовыми значениями аргумента. В данную схему параллельного вычисления с очевидностью включаются изложенные схемы вычисления производных и интегралов.

Применимость метода для параллельного преобразования Фурье при цифровой обработке сигналов. При выполнении прямого дискретного преобразования Фурье $X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(nT)\,\mathrm{e}^{-i2nk/N}$ (**ДПФ**) и обратного ДПФ $x(nT) = \sum_{k=0}^{N-1} X(k)\,\mathrm{e}^{i2nk/N}$ (**ОДПФ**) изложенный метод можно использовать для динамического вычисления действительной и мнимой части элементов тригонометрического базиса $\mathrm{e}^{i2nk/N} = \cos(2nk/N) + i\sin(2nk/N)$. По рассмотренной схеме кусочно-полиномиальной аппроксимации элементы базиса ДПФ и ОДПФ можно представить в виде алгебраического полинома с заданными коэффициентами:

$$\cos(x) = a_{0i} + a_{1i}x + a_{2i}x^2 + \ldots + a_{ni}x^n, \tag{23}$$

$$\sin(x) = b_{0i} + b_{1i}x + b_{2i}x^2 + \ldots + b_{ni}x^n, \tag{24}$$

где i – номер подынтервала, $x = \frac{2\pi k}{N}$. Для всех значений аргументов при произвольном N, одновременно для всех номеров подынтервалов i, все элементы базиса в форме (23), (24) могут быть вычислены параллельно и синхронно. Для наперед заданной точности аппроксимации степень каждого интерполирующего полинома (23) и (24) может быть выбрана сколь угодно малым натуральным числом, поэтому временная сложность параллельного вычисления всех элементов базиса оценивается из соотношения

$$T(R) = O(1),$$

где число процессоров R совпадает с количеством одновременно вычисляемых элементов базиса, $R = N$.

Далее, вычисленные значения (23), (24) на тех же N процессорах одновременно умножаются на соответствующие коэффициенты $x(nT)$ ДПФ.

После умножения элементов базиса на коэффициенты ДПФ принимает вид суммы N слагаемых. По схеме сдваивания сумма N слагаемых параллельно вычисляется с логарифмической оценкой временной сложности [8].

$$T\left(\left\lceil \frac{N}{2} \right\rceil\right) = \lceil \log_2 N \rceil \, t_c .$$

Отсюда время параллельного выполнения по рассматриваемой схеме ДПФ с произвольным числом отсчетов составит

$$T(N) = O\left(\log_2 N\right). \tag{25}$$

Аналогичная оценка имеет место для вычисления ОДПФ.

Близкая к (25) оценка параллельного выполнения ДПФ и ОДПФ может быть получена на основе схемы Стоуна [6] применительно к полиномам Чебышева, с помощью которых выражаются элементы базиса. Однако схема на основе кусочно-интерполяционной аппроксимации базиса отличается исключением из асимптотики операций умножения. Поэтому на данной основе схема из [6] ускоряется во столько, во сколько машинная операция сложения быстрее умножения. Таким образом, достигается улучшение данной в [6] оценки временной сложности параллельного выполнения ДПФ и ОДПФ.

Предложенная схема сохранит порядок временной сложности при включении в нее алгоритмов корреляции, цифровой фильтрации и свертки [1].

Пусть $x(t)$ – опорный сигнал, $y(t)$ – сумма $x(t)$ и сигналов других форм. В случае действительных $x(t)$ и $y(t)$ корреляция определяется из соотношения

$$C_{xy}(k) = \frac{1}{\sigma_x \sigma_y N} \sum_{n=0}^{N-1} x_n \, y_{n+k}, \tag{26}$$

где x_n – выборка $x(t)$, y_n – выборка $y(t)$, y_{n+k} – последовательность, полученная сдвигом последовательности y_n относительно x_n на k шагов, σ_x, σ_y – среднеквадратичные значения x_n и y_n

$$\sigma_x^2 = \frac{1}{N} \sum_{n=0}^{N-1} x_n^2, \quad \sigma_y^2 = \frac{1}{N} \sum_{n=0}^{N-1} y_n^2 . \tag{27}$$

В случае комплексных $x(t)$ и $y(t)$ корреляция равна

$$C_{xy}(k) = \frac{1}{\sigma_x \sigma_y N} \sum_{n=0}^{N-1} x_n^* \, y_{n+k} . \tag{28}$$

Если комплексные амплитудные коэффициенты величин x_n^* и y_n обозначить X_m^* и Y_m, то конечные фурье-преобразования $C_{xy}(k)$ и $X_m^* Y_m$ можно представить в виде:

$$\sigma_x \sigma_y \sum_{k=0}^{N-1} C_{xy}(k) \exp\left(\frac{-i 2\pi m k}{N}\right) = N X_m^* Y_m, \qquad (29)$$

$$\sigma_x \sigma_y C_{xy}(k) = \sum_{m=0}^{N-1} X_m^* Y_m \exp\left(\frac{i 2\pi m k}{N}\right). \qquad (30)$$

Функция ковариации представляет собой корреляционную функцию, умноженную на среднеквадратичные значения x_n и y_n:

$$\mathrm{cov}[x, y(k)] \equiv \sigma_x \sigma_y C_{xy}(k). \qquad (31)$$

Для двух конечных причинных последовательностей $h(n)$ и $y(k)$ длиной соответственно N и K свертка определяется выражением

$$s(k) = h(n) \otimes y(k) \equiv h(n) * y(k) = \sum_{n=0}^{N} h(n) y(k-n), \qquad (32)$$

где $\otimes$ или $*$ символьные обозначения операции свертки; $y(k)$ представляет собой сигнал на входе системы; $h(n)$ – импульсный отклик системы; $s(k)$ – выходной сигнал системы.

Для вычисления корреляции (26) целесообразно предварительно по схемам сдваивания параллельно вычислить значения левых частей (27). При этом квадратный корень аппроксимируется по кусочно-полиномиальной схеме применительно к функции $y = x^{1/2}$. После этого с сохранением оценки (25) по схеме сдваивания выполняется параллельное вычисление (26). Порядок оценки (25) сохраняется также в случае комплексных значений $x(t)$ и $y(t)$ при параллельном вычислении корреляции (28) с использованием схемы сдваивания. В выражениях (29), (30) элементы базиса можно определить по предложенной параллельной схеме кусочно-полиномиальной аппроксимации. Очевидно, порядок рассматриваемой оценки временной сложности не изменится при вычислении ковариации (31). С распространением схемы сдваивания на операцию свертки (32) весь представленный набор вычислительных операций цифровой обработки оказывается выполнимым параллельно с временной сложностью порядка

$$T(N) = O(\log_2 N),$$

где N – произвольно задаваемое значение переменной (количество отсчетов).

Таким образом, фактически полный набор операций одномерной цифровой обработки сигналов с использованием предложенных схем можно выполнить параллельно с логарифмической оценкой временной сложности. При этом из асимптотики схемы сдваивания остаются исключенными операции умножения.

Список литературы

1. Айфичер, Э. Цифровая обработка сигналов: практический подход / Э. Айфичер, Б. Джервис. – 2-е изд. – М. : «Вильямс», 2004. – 989 с.
2. Березин, И.С. , Методы вычислений. В 2 т. Т. 1 / И.С. Березин, Н.Г. Жидков. – М. : Наука, 1970. – 464 с.
3. Люк, Ю. Специальные математические функции и их аппроксимации / Ю. Люк. – М. : Мир, 1980. – 608 с.
4. Ромм, Я.Е. Бесконфликтные и устойчивые методы детерминированной параллельной обработки : дис. … д-ра техн. наук. – Таганрог: ТРТУ, 1998. – 546 с.; ВНТИ Центр. – 05.990.001006.
5. Ромм, Я.Е. Кусочно-полиномиальное вычисление функций, производных и определенных интегралов на основе интерполяции по Ньютону / Я.Е. Ромм, Л.Н. Аксайская ; Таганрог. госуд. педагогич. ин-т. – Таганрог, 2007. – 36 с. – Деп. в ВИНИТИ 02.05.07, № 487.
6. Ромм, Я.Е. Параллельная схема дискретного и быстрого преобразований Фурье на основе полиномиального представления базиса / Я.Е. Ромм, С.А. Фирсова // Известия РАН. Математическое моделирование, 2006. – Т. 18, № 11. – С. 3 –12.
7. Солодовников, В.И. Верхние оценки сложности решения систем линейных уравнений / В.И. Солодовников ; В кн. : Теория сложности вычислений. I: Записки научных семинаров; ЛОМИ АН СССР. – Л., 1982. – Т. 118. – С. 159 – 187.
8. Фаддеева, В.Н. Параллельные вычисления в линейной алгебре / В.Н. Фаддеева, Д.К. Фаддеев // Кибернетика, 1977. – № 6, – С. 28 – 40.

Piecewise Polynomial Approximation of Functions on the Basis of Newton Interpolation with Application to Parallel Digital Signal Processing

Ya.E. Romm, L.N. Aksaiskaya

Taganrog State Teachers' Training Institute, Taganrog

Key words and phrases: functions calculation; parallel schemes; digital signal processing.

Abstract: Computer scheme of optimized piecewise polynomial approximation of functions, derivative and definite integrals on the basis of interpolation Newton polynomial for optional error boundary is presented. The basis of discrete Fourier transformations with logarithmic evaluation of temporal complexity under optional number of indications is calculated in parallel, thus multiplication operations are excluded from calculation algorithm.
