

Глава 15. Адаптивные экономические асинхронные итерационные методы «цифра за цифрой»

15.1. Состояние проблемы

При аппаратной реализации алгоритмов вычисления функций важную роль играют методы “цифра за цифрой”. Методы “цифра за цифрой” позволяют последовательно получить очередную цифру результата в заданной системе счисления. Эти методы восходят к алгоритму Г. Бригса (1624 г.) вычисления логарифмической функции. Бурное развитие данных методов связано с появлением работ Д. Волдера [1] и Д. Меджита [2]. Большое число исследователей внесло свой вклад в развитие и модернизацию этих методов, обширная библиография по данному вопросу в работах [3, 4]. В основном, исследования и модернизация методов “цифра за цифрой” производятся в части их обобщения на произвольную систему счисления, оценки сходимости методов, исследования погрешности, оценки времени вычисления, определения аппаратных затрат, асинхронной организации процесса вычислений, использования избыточных систем счисления для сокращения времени счета и получения результата старшими разрядами вперед, сокращения числа итераций за счет использования завершающих приближений, расширения областей применения этих методов и т.д. В последнее время проведена модернизация методов на случай реализации на параллельных ВС [5, 6]. На данную тему защищены кандидатские и докторские диссертации как у нас в стране, так и за рубежом. В настоящее время осуществляется реализация этих методов в БИС и СБИС.

Одной из причин широкого распространения методов Волдера и Меджита является то обстоятельство, что для их реализации требуется простой набор операций: сдвиги, сложение (вычитание), определение знака числа и выборки из памяти. Помимо этого, разработчиков данных методов привлекает прямая зависимость числа итераций от числа разрядов в представлении искомой функции. Для эффективной реализации методов “цифра за цифрой” необходимо использование сдвигателей на произвольное число разрядов, специальной структуры АЛУ, а также для вычисления большинства функций – наличие ЗУ с малым временем выборки. В последнем случае эти методы являются разновидностью таблично-алгоритмических методов. В силу последнего обстоятельства они могут быть использованы для ЭВМ, работающих с фиксированной и переменной разрядностью, например, для одинарной и двойной точности, но их нельзя использовать для счета с произвольной разрядностью. Появление умножителей и делителей, выполненных на комбинационных схемах, позволило значительно сократить время выполнения этих операций, что в ряде случаев сделало более предпочтительным для вычисления функций использование многочленной и дробно-рациональной аппроксимации. Помимо этого, увеличение

объемов ЗУ и сокращение времени выборки из них сделало целесообразным использование методов сегментной аппроксимации. Как уже отмечалось выше, для эффективной реализации алгоритмов, реализующих методы “цифра за цифрой”, целесообразно использовать АЛУ со специфической структурой. Все это вынудило искать дополнительные средства для ускорения методов типа “цифра за цифрой”. Эти средства ускорения используют следующие факторы: организацию асинхронного режима счета, использование избыточных систем счисления, использование начальных и завершающих приближений, замену целых этапов вычислений на выборку из таблиц их значений, а также параллельное выполнение определенных участков алгоритма.

Такой подход, состоящий из сочетания метода “цифра за цифрой” с другими методами вычисления функций, назовем комбинированным. Комбинированный подход использует несколько методов и является для определенных областей применения более эффективным, чем каждый из составляющих его методов. Обычно на практике используются комбинированные методы, сочетающие на первом этапе вычислений прямые методы счета для нахождения начального приближения, а на втором этапе – итерационные методы для уточнения значения искомой функции. В рассматриваемом случае в большинстве исследуемых ниже методов на первом этапе используется итерационный метод “цифра за цифрой”, а на последующих этапах – используются прямые или другие итерационные методы вычисления искомой функции. Благодаря такому подходу, появилась возможность адаптировать эти алгоритмы к различным условиям применения.

Методы Д. Волдера и Д. Меджита в основном используются для синхронного режима счета. При вычислении ряда функций для обеспечения сходимости этих методов даже в двоичной системе счисления возникает необходимость в повторных итерациях. Это приводит к ухудшению показателя: отношение производительность/стоимость. Характерной особенностью аппаратной реализации методов «цифра за цифрой» является реализация операций псевдоделения и псевдоумножения над базисом соответствующих констант типа $\ln(1+N^m)$, $\arctg N^m$ и т.д., являющихся функциональными весами разрядов. Для реализации методов могут быть использованы регистровая, таблично–регистровая и матричные структуры устройств псевдоделения и псевдоумножения [3]. Существуют и другие подходы для реализации этих методов [5, 6]. Временные, аппаратные и другие затраты при реализации методов могут заменяться путем использования канонических систем счисления с различными основаниями типа (2,4,8,10,16,...), избыточных цифровых наборов типа $(\bar{1},0,1)$, $(\bar{9}, \bar{8}, \dots, 0,1, \dots 9)$, $(\bar{1}\bar{5}, \bar{1}\bar{4}, \dots, 0,1,2, \dots 15)$, включая иррациональные основания системы счисления типа $(1 + \sqrt{5})/2$, $\sqrt[4]{2}$, $1 + 2^{1-a}$, использование синхронных и асинхронных режимов вычисления, а также использование начальных и завершающих приближений и т.д. Использование асинхронной организации позво-

ляет увеличить производительность вычислительных средств за счет того, что обработка информации проводится по мере готовности информации и устройства обработки. Применение асинхронных алгоритмов эквивалентно логическим методам ускорения вычислительных операций в арифметико-логических устройствах, так как предполагает исключение расчетов, когда очередная цифра анализируемого числа равна нулю (для некоторых функций – единице).

С точки зрения поступления исходной информации будем различать статический (полноциклический) и динамический (темп накачки), т.е. конвейерный режим работы вычислительных устройств, реализующих эти методы. При этом способы реализации алгоритмов могут быть аппаратными, микропрограммно-аппаратными и программно-аппаратными [7].

Остановимся более подробно на проблеме ускорения вычислений в методах «цифра за цифрой». Как отмечалось в работе [7], одним из наиболее быстрых методов класса «цифра за цифрой» является асинхронный метод Т.Х. Чена [8, 9] – наряду с асинхронностью, которая обеспечивает для двоичной системы счисления сокращение числа итераций примерно в два раза. В методе Т.Х. Чена используются линейные завершающие приближения, которые для большинства приведенных алгоритмов содержат операцию умножения. Но благодаря этому можно добиться уменьшения общего числа итераций еще в два раза.

Для дальнейшего сокращения числа итераций автором было предложено использовать нелинейные завершающие приближения, представленные в виде модифицированного разложения функций в ряд невязок [7]. В настоящей работе показано, что для этих целей можно также использовать широкий спектр модифицированных разложений функций по невязкам. Наряду с целью сокращения числа умножений (делений), а в некоторых случаях и полного отказа от операций умножения, но с сохранением квадратичной сходимости алгоритма, автором предложено введение экономичного начального приближения перед вторым этапом работы алгоритма [7]. Такие подходы позволяют сократить число итераций и расширить спектр СВТ для применения методов «цифра за цифрой». Использование методов «цифра за цифрой» представляет собой метод сужения исходного интервала изменения аргумента и слабо поддается распараллеливанию. Однако вычисление начальных и завершающих приближений допускает процесс эффективного распараллеливания многочленных либо дробно-рациональных выражений. Это позволяет использовать приведенные методы на многопроцессорных ВС. Дополнительный эффект в сокращении времени достигается за счет использования инкрементной (сокращенной) информации. Соотношение между итерационным алгоритмом сужения информации и количеством членов в начальном либо завершающем приближении соответствует случаю, рассмотренному при исследовании z_m – аппроксимации функций [11]. Это связано с тем, что использование z_m – аппроксимации функций и метода «цифра за цифрой» – приводит к одинаковому эффекту уменьшения исходного

интервала изменения аргумента, но разными средствами. В основу разработанных базовых адаптивных алгоритмов положен метод Т.Х. Чена, хотя для этих целей могут быть использованы и алгоритмы типа Д. Волдера или Д. Меджита. Термин «адаптация» в данном контексте будем понимать в смысле функционально-аппаратного соответствия.

Использование адаптивных алгоритмов позволяет, наряду со специальными структурами, характерными для методов «цифра за цифрой», использовать и современные структуры АЛУ, содержащие параллельные сумматоры, умножители, быстрые сдвигатели, а также устройства определения количества левых нулей, используемых обычно в аппаратуре для реализации команд сложения в режиме с плавающей запятой. Помимо этого, имеется возможность уменьшить время счета элементарных функций за счет более полного использования «быстрых» постоянных запоминающих устройств и таблично– алгоритмических методов. При этом рассматривается введение начальных приближений перед началом первого или второго этапа счета по методу «цифра за цифрой», а также использование завершающих приближений. Наряду с обычными аппроксимациями, рассматриваются и сегментные аппроксимации. Таким образом, использование адаптивных алгоритмов позволяет в широких пределах варьировать номенклатуру используемых аппаратурных средств за счет различных модификаций алгоритмов метода “цифра за цифрой”.

Благодаря применению экономичных асинхронных алгоритмов с использованием на втором этапе итерации экономичных линейных приближений, не содержащих операций умножений, удается получить для двоичной системы счисления такое же ускорение вычислений, как и в методе Т.Х. Чена, но при этом отсутствует операция умножения. Среднее число итераций уменьшается примерно в 4 раза (в 2 раза за счет асинхронности и в 2 раза за счет начального приближения на втором этапе вычислений). Если же в этих алгоритмах использовать экономичные квадратичные приближения [11], содержащие одну операцию умножения, то количество итераций сократится примерно в 6 раз (в 2 раза за счет асинхронности и в 3 раза за счет квадратичного приближения). За счет увеличения порядка начального приближения можно достичь дальнейшего уменьшения количества итераций, но при этом возрастает количество умножений и увеличиваются возможности для распараллеливания алгоритма. При использовании избыточных цифровых наборов $(\bar{1}, 0, 1)$ среднее количество нулей возрастает с половины до двух третей от общего количества цифр в представлении числа. В этом случае при экономичном линейном приближении число итераций уменьшается в 6 раз, а при экономичном квадратичном приближении число итераций в данном же случае уменьшается в 9 раз (в 3 раза за счет асинхронности и в 3 раза за счет использования начального приближения).

Помимо этого, при вычислении функции $\ell_n(x)$ по алгоритмам Д. Волдера и Д. Меджита требуются двойные итерации, а в рассматриваемом алго-

ритме этого не требуется, так как процесс приближения к результату односторонний. Это также дает дополнительный эффект ускорения вычислений.

15.2. Функциональные соотношения для получения адаптивных методов «цифра за цифрой»

Основу адаптивных аппроксимаций составляют соответствующие функциональные преобразования.

Перейдем к непосредственному описанию методов “цифра за цифрой” для вычисления функции $f(x)$ на основе модификации метода Т.Х. Чена с целью адаптации к различным видам структур вычислительных устройств и структурам данных. Для вычисления функции $f(x)$ так же, как и в случае получения ИФ формул на основе разложения функции по невязкам, будем рассматривать уравнение $z=F(x,y)$.

В общем случае $F(x,y)=y \otimes f(x)$ отлично от нуля. Знак \otimes означает некоторую арифметическую операцию (в основном $+$, $-$ или умножение). В рассматриваемом методе одновременно итерируется как параметр x_k , так и параметр y_k . При этом при стремлении величины x_k к некоторому пределу (обычно к нулю или к единице) величина y_k стремится к искомой функции $f(x)$, т.е. $\lim_{x_k \rightarrow c} y_k = z$. В базовом алгоритме имеется возможность получить более сложную функцию $z = a \otimes f(x)$, где a – заданное значение величины y_0 . В модифицированном алгоритме такой возможности не имеется, т.к. вместо величины a вводится начальное приближение. В рассматриваемом алгоритме проводятся такие итерационные преобразования пары чисел x_k, y_k , что функция $z=f(x_k, y_k)$ остается инвариантной относительно этих преобразований, т.е. $F(x_{k+1}, y_{k+1}) = F(x_k, y_k)$. При $x_0 = x$ и $y_0 = A$ $z=F(x_0, y_0)=F(x_1, y_1)=\dots=F(x_p, y_p)=A \otimes f(x)$. В результате получим $x_p \approx c$, $y_p \approx A \otimes f(x)$. Так, для функции $f(x) = 1/x$ имеем $z = y(1/x)$. Если использовать преобразования $x_{k+1} = p_k x_k$, $y_{k+1} = p_k y_k$, которые не изменяют величину z , т.к. $F(x_k, y_k) = F(x_{k+1}, y_{k+1}) = z$, то, использовав n раз такое преобразование, обеспечив $x_k \rightarrow c$, получим

$$z = \frac{y_0 \prod_{k=1}^n P_k}{x_0 \prod_{k=1}^n P_k} \approx C^{-1} y_0 \prod_{k=1}^n P_k .$$

При $C=1$ не требуется умножения на константу C^{-1} .

В некоторых случаях используется $C = \pm 2^\ell$ и даже $C=2/3$.

Если взять $P_k = 1+2^{-k}$, $\lim_{n \rightarrow \infty} x_0 \prod_{k=1}^n P_k = 1$, то можно записать следующие итерационные формулы:

$$x_{k+1} = x_k + 2^{-m} x_k ;$$

$$y_{k+1} = y_k + 2^{-m} y_k ,$$

где $x_0 = x$, $m = 1 + p$, p – число левых нулей в выражении $1 - x_k$. Процесс можно не доводить до конца, а завершить по итерационной формуле нахождения обратной величины $1 / x_p$.

Приведем функциональные преобразования для получения основных итерационных формул, работающих по методу «цифра за цифрой».

Для вычислений функций e^x , y/x , $\ln x$, $y/\sqrt[n]{x}$ воспользуемся преобразованиями [1–4].

Для функции $e^x = p e^{x - \ln p}$, откуда $e^x \approx \prod_{k=0}^n P_k e^{x - \sum_{k=0}^n P_k}$ либо

$$y_0 e^x \approx y_0 \prod_{k=0}^n P_k e^{x - \sum_{k=0}^n P_k}, \quad \lim_{n \rightarrow \infty} e^{x - \sum_{k=0}^n P_k} = 1 .$$

Для функции $\ln x = -\ln p + \ln px$, откуда $\ln x = \sum_{k=0}^n (-\ln P_k) + \ln(x \prod_{k=0}^n P_k)$

$$\text{либо } y_0 + \ln x = (y_0 - \sum_{k=0}^n \ln P_k) + \ln(x \prod_{k=0}^n P_k) .$$

Параметр P_k может быть выбран специальным способом, чтобы избежать операции умножения и обеспечить стремление в предел соответствующей константы. В работе [8] предлагается брать $P_k = 1 + 2^{m_k}$, где m_k подбирается так, чтобы обеспечить выполнение предельных переходов.

В работе [10] используется $P_k = 1 + 16^{m_k}$.

В общем случае можно брать $P_k = 1 + N^{mk}$, $N=2,4,8,10,16,\dots$, симметричные цифровые наборы или иррациональные избыточные основания системы счисления.

Для функций $\sin x$, $\cos x$, $\text{Arctg}x$, $\text{sh}x$ и $\text{Arth}x$ можно использовать функциональные преобразования, которые основаны на формулах сложения тригонометрических и гиперболических функций.

Для функций $\sin x$ и $\cos x$ справедливы следующие функциональные преобразования [4]:

$$\sin(x + p) = \cos p (\sin x + \cos x \operatorname{tg} p);$$

$$\cos(x + p) = \cos p (\cos x - \sin x \operatorname{tg} p),$$

где $p = \operatorname{arctg} N^{-1}$, N – основание системы счисления.

Для функции $\operatorname{arctg}(y/x) = \operatorname{arctg} p + \operatorname{arctg}((y - xp)/(x + yp))$.

Для функций $\text{sh}x$, $\text{ch}x$ используем функциональные преобразования

$$\operatorname{sh}(x + p) = \operatorname{ch}p (\operatorname{sh}x + \operatorname{ch}x \operatorname{th}p);$$

$$\operatorname{ch}(x + p) = \operatorname{ch}p (\operatorname{ch}x + \operatorname{sh}x \operatorname{th}p),$$

где $p = \operatorname{Arth} N^{-1}$.

Для функции $\operatorname{Arth}(y/x) = \operatorname{Arth} p + \operatorname{Arth}((y - xp)/(x + yp))$.

15.3. Вычисление функций на основе адаптивных методов «цифра за цифрой»

Как уже отмечалось выше, асинхронный алгоритм Т.Х. Чена [8] базируется так же, как методы Д. Волдера и Д. Меджита, на двух этапах вычислений, соответствующих псевдоумножению и псевдоделению. Но для уменьшения числа итераций используются линейное завершающее приближение и асинхронный режим счета на основе анализа числа левых единиц (нулей). Этот алгоритм допускает параллельное выполнение двух этапов итераций по методу «цифра за цифрой». В экономичном адаптивном алгоритме, предложенном автором [7], имеется возможность избавиться от операции умножения, используемой в завершающем приближении, путем введения между первым и вторым этапом промежуточного этапа, на котором используется экономичное приближение (не содержащее операции умножения). При необходимости распараллелить процесс вычислений либо сократить число итераций автором предложено использовать следующие подходы. Использование более сложных зависимостей (увеличение числа членов) для начального и/или завершающего приближений.

На основе приведенных функциональных соотношений и введенных начальных (завершающих) приближений получаем ряд адаптивных методов. В табл. 15.1 приведены адаптивные экономичные асинхронные алгоритмы, основанные на методе «цифра за цифрой».

В табл. 15.1 приняты следующие обозначения:

q – основание системы счисления; m – целое число; r_1 – число левых единиц в числе или число левых нулей в числе, записанном в дополнительном коде; r_2 – число левых нулей в числе; $(p - 1)$ – индекс последней итерации. Количество итераций и число членов в завершающей формуле определяются исходя из используемых аппаратных средств, погрешностей величин x_p , y_p , φ_p и вида аппроксимирующего приближения, используемого в качестве завершающей формулы.

Для функций $1/x$ и $1/\sqrt{x}$ в качестве завершающих приближений имеется возможность использовать итерационные формулы заданного порядка сходимости. Особый интерес использования этих формул представляется при вычислении с большой разрядностью. При этом может быть широко использована инкрементная информация.

Таблица 15.1

Адаптивные экономичные асинхронные алгоритмы

Функция	Итерационные формулы		Завершающие приближения	
	I этап	II этап	Алгоритм деления	
			I этап	II этап
$\frac{1}{x}$	$x_{k+1} = x_k + q^{-m} x_k$, $x_0 = x$, $m = 1 + r_1$, $k = 0, 1, \dots, p - 1$	$y_{k+1} = y_k + q^{-m} y_k$, $y_0 = 2 - x_p$, $y_p = 1/x$	–	–
$\frac{1}{\sqrt{x}}$	$t_{k+1} = x_k + q^{-m} x_k$, $x_{k+1} = t_k + q^{-m} t_k$, $x_0 = x$, $m = 2 + r_1$, $k = 0, 1, \dots, p - 1$	$y_{k+1} = y_k + q^{-m} y_k$; для $q = 2$, $y_0 = \frac{3}{2} - \frac{1}{2} x_p$; $y_p = 1/\sqrt{x}$	–	–
$\ln x$	$x_{k+1} = x_k - q^{-m} x_k$, $x_0 = x$, $m = 1 + r_2$, $k = 0, 1, \dots, p - 1$	$y_{k+1} = y_k - \ln(1 + q^{-m})$, $y_0 = -1 + x_p$ либо $y_0 = A + x_p$ $y_p = \ln x$.	–	–
e^x	$x_{k+1} = x_k - \ln(1 + q^{-m})$, $x_0 = x$, $m = 1 + r_2$, $k = 0, 1, \dots, p - 1$	$y_{k+1} = y_k + q^{-m} y_k$, $y_0 = 1 + x_p$ либо $y_0 = A + x_p$, $y_p = e^x$	–	–

$\operatorname{tg}\varphi$	$\varphi_{k+1} = \varphi_k - \operatorname{Arctg}q^{-m}$, $\varphi_0 = \varphi$, $m = 1 + r_2$, $k = 0, 1, \dots, p-1$	$y_{k+1} = y_k + q^{-m}x_k$, $x_{k+1} = x_k - q^{-m}y_k$, $x_0 = 1$, $y_0 = \varphi_p$	$x_{i+1} = x_i + q^{-m}x_i$, $x_0 = x_p$	$y_{i+1} = y_i + q^{-m}y_i$, $y_0 = y_p$, $y_e = \operatorname{tg}\varphi$
$\operatorname{arctg} \frac{y}{x}$	$y_{k+1} = y_k - q^{-m}x_k$, $x_{k+1} = x_k + q^{-m}y_k$, $x_0 = x$, $y_0 = y$, $m = r_2$, $k = 0, 1, \dots, p-1$	$\varphi_{k+1} = \varphi_k + \operatorname{Arctg}q^{-m}$, $\varphi_0 = y_p / x_p$	$x_{i+1} = x_i + q^{-m}x_i$, $x_0 = x_p$	$y_{i+1} = y_i + q^{-m}y_i$, $y_0 = y_p$
$\operatorname{th} \varphi$	$\varphi_{k+1} = \varphi_k - \operatorname{Arth}q^{-m}$, $\varphi_0 = \varphi$, $m = 1 + r_2$, $k = 0, 1, \dots, p-1$	$x_{k+1} = x_k + q^{-m}y_k$, $y_0 = \varphi_p$	$x_{i+1} = x_i + q^{-m}x_i$, $x_0 = x_p$	$y_{i+1} = y_i + q^{-m}y_i$, $y_0 = y_p$, $y_e = \operatorname{th}\varphi$
$\operatorname{Arth} \frac{y}{x}$	$y_{k+1} = y_k - q^{-m}x_k$, $x_{k+1} = x_k + q^{-m}y_k$, $x_0 = x$, $y_0 = y$, $m = r_2$, $k = 0, 1, \dots, p-1$	$\varphi_{k+1} = \varphi_k + \operatorname{Arth}q^{-m}$, $\varphi_0 = y_p / x_p$	$x_{i+1} = x_i + q^{-m}x_i$, $x_0 = x_p$	$y_{i+1} = y_i + q^{-m}y_i$, $y_0 = y_p$

Начальные и завершающие приближения (табл. 15.2) могут быть получены в виде разложений функций по невязкам (с модификацией невязки) на основе разложений в ряд Тейлора, аппроксимаций Паде, разложений в цепные дроби, наилучших (минимаксных) и близких к ним приближений, бесконечных произведений и т.д.

Таблица 15.2

Начальные и завершающие приближения

Функция	Вид завершающего приближения	Вид начального приближения
$\frac{1}{x}$	$\frac{1}{x} = y_p \sum_{k=0}^{\infty} z_p^k$; $z_p = 1 - y \cdot x_p \approx 1 - x_p$ $y_{i+1} = y_i \sum_{k=0}^l z_p^k$, <i>либо</i> $y_{i+1} = y_i \prod_{k=0}^l (1 + z_p^{2k})$, $y_0 = y_p$, $z_i = 1 - xy_i$	$y_0 = \sum_{k=0}^l z_p^k$, <i>либо</i> $y_0 = \prod_{k=0}^l (1 + z_p^{2k})$, $z_p = 1 - x_p$

$\frac{1}{\sqrt{x}}$	$\frac{1}{\sqrt{x}} = y_p \left(1 + \frac{1}{2} z_p + \frac{1.3}{2.4} z_p^2 + \frac{1.3.5}{2.4.6} z_p^3 + \dots \right),$ $z_p = 1 - y^2 x_p \approx 1 - x_p;$ $y_{i+1} = y_i \left(1 + \frac{1}{2} z_i + \frac{1.3}{2.4} z_i^2 + \dots \right),$ $z_i = 1 - x y_i^2, \quad y_0 = y_p$	$y_0 = 1 + \sum_{k=0}^l \frac{z_p^k (2+1) \dots (1+2(k-1))}{k! 2^k}$
$\ln x$	$\ln x = y_p - \sum_{k=1}^{\infty} z_p / k, \quad z_p = 1 - x_p;$ $\ln x = y_p - \frac{z_p}{1} - \frac{z_p}{2} - \frac{z_p}{3} - \frac{z_p}{2} - \frac{z_p}{2l+1}$	$y_0 = - \sum_{k=1}^{\infty} z_p / k$
e^x	$e^x = y_p \sum_{k=0}^{\infty} z_p / k!, \quad z_p = x_p;$ $e^x = y_p \left(\frac{1}{1} - \frac{2z_p}{2} + \frac{z_p^2}{6} - \dots + \frac{z_p^2}{2(2l+1)} \right)$ $e^x = y_p \frac{1 + \frac{1}{2} z_p + \frac{3}{28} z_p^2 + \frac{1}{84} z_p^3 + \dots}{1 - \frac{1}{2} z_p + \frac{3}{28} z_p^2 - \frac{1}{84} z_p^3},$ $e^x = y_p (I_0(\beta) + 2 \sum_{n=1}^{\infty} I_n(\beta) T_n(u));$ $u = (2z - \beta) / \beta, \quad z_p \in [0, \beta], \quad u \in [-1, 1]$	$y_0 = \sum_{k=0}^n z_p^k / k!$ $y_p = I_0(\beta) + 2 \sum_{n=1}^m I_n(\beta) T_n(u)$
$\operatorname{tg} \varphi$	$\operatorname{tg} \varphi = \frac{\sin \varphi}{\cos \varphi} = \frac{\sin \left(\sum_m \operatorname{arctg} q^{-m} + \varphi_p \right)}{\cos \left(\sum_m \operatorname{arctg} q^{-m} + \varphi_p \right)} =$ $= \frac{y_p \cos \varphi_p + x_p \sin \varphi_p}{x_p \cos \varphi_p + y_p \sin \varphi_p},$ $\cos \varphi_p = \sum_{k=0}^{\infty} (-1)^k \frac{\varphi_p^{2k}}{(2k)!},$ $\sin \varphi_p = \sum_{k=1}^{\infty} (-1)^k \frac{\varphi_p^{2k+1}}{(2k+1)!},$	<p style="text-align: center;">—</p>

$th\varphi$	$th\varphi = \frac{sh\varphi}{ch\varphi} = \frac{y_p ch\varphi_p + x_p sh\varphi_p}{x_p ch\varphi_p + y_p sh\varphi_p},$ $sh\varphi_p = \sum_{k=0}^{\infty} \frac{\varphi_p^{2k+1}}{(2k+1)!}, \quad ch\varphi_p = \sum_{k=0}^{\infty} \frac{\varphi_p^{2k}}{(2k)!},$ <p>либо</p> $sh\varphi_p = \frac{\varphi_p(1+F)}{(1+F)^2 - \varphi_p^2/4},$ $ch\varphi_p = \frac{\varphi_p^2/2}{(1+F)^2 - \varphi_p^2/4},$ $F = \frac{3u}{1} + \frac{15u}{1} + \dots + \frac{(4(n-1)^2 - 1)u}{1} + \dots,$ $u = \varphi_p / 4$	<p>—</p>
$arctg \frac{y}{x}$	$arctg \frac{y}{x} \approx \varphi_p + arctg \frac{y}{x} \approx$ $\approx \varphi_p + \sum_{k=0}^{\infty} (-1)^k u^{2k+1} / (2k+1),$ <p>либо</p> $u = y_p / x_p$ $\approx \varphi_p + \frac{u}{1} + \frac{u^2}{3} + \frac{4u^2}{5} + \dots + \frac{n^2 u^2}{2n+1} + \dots,$ $u = y_p / x_p$	<p>—</p>
$Arth \frac{y}{x}$	$Arth \frac{y}{x} \approx \varphi_p + Arth \frac{y_p}{x_p} \approx$ $\approx \varphi_p + \sum_{k=0}^{\infty} u^{2k+1} / (2k+1),$ $u = y_p / x_p$	<p>—</p>

В табл. 15.3 приведены результаты расчета ряда элементарных функций на основе адаптивного экономичного асинхронного итерационного метода «цифра за цифрой». Из примеров расчета видно, что для получения точности $3 \cdot 10^{-7} - 3 \cdot 10^{-8}$ требуется от двух до четырех итераций в зависимости от вида начального приближения. В обычном методе Волдера для достижения такой точности требуется 22 – 25 итераций.

**Результаты вычислений по итерационным формулам адаптивного метода
«цифра за цифрой»**

Функция	Значение аргумента	Результат вычислений I этапа	Начальное приближение	Результат вычислений II этапа, погрешность
$\frac{1}{x}$	0,555 5555 0,555 5555	$\bar{m} = \{1, 3, 4, 8\}$ $x_4 = 9,9998463 \cdot 10^{-1}$ $\bar{m} = \{1, 3, 4\}$ $x_4 = 9,9609364 \cdot 10^{-1}$	$y_0 = 2 - x_4 = 1,0000154$ $y_0 = \frac{3}{4} + (z_p + \frac{1}{2})^2 = 1,003917$	$1/x \approx y_4 = 1,800003$, $ \Delta = 3 \cdot 10^{-7}$ $1/x \approx y_3 = 1,800002$, $ \Delta = 2 \cdot 10^{-7}$
$\frac{1}{\sqrt{x}}$	0,555 5555 0,555 5555	$\bar{m} = \{2, 4, 7, 9\}$ $\bar{m} = \{2, 4, 7, 9\}$	$y_0 = \frac{3}{2} - \frac{1}{2} x^4 = 1,0003918$ $y_0 = 1,0000001 + \frac{1}{2} z = 1,0003919$	$1/\sqrt{x} \approx y_4 = 1,3416405$, $ \Delta = 3 \cdot 10^{-7}$ $1/\sqrt{x} \approx y_4 = 1,3416406$, $ \Delta = 2 \cdot 10^{-7}$
$\ln x$	0,555 5555 0,555 5555 0,555 5555	$\bar{m} = \{1, 3, 4, 8\}$ $\bar{m} = \{1, 3\}$ $x_2 = 0,93749991$ $\bar{m} = \{1, 3\}$ $x_3 = 0,93749991$	$y_0 = x_4 - 1 = -1,54 \cdot 10^{-5}$ $y_0 = 0,258165338 x_2 + 0,842795596 - \frac{1,633938093}{x_2 + 0,484101794} = -6,45385 \cdot 10^{-2}$ $y_0 = -(1 - x_3) - \frac{1}{2}(1 - x_3) = -391403 \cdot 10^{-3}$	$\ln x \approx y_4 = -5,8778673 \cdot 10^{-1}$, $ \Delta = 7 \cdot 10^{-8}$ $\ln x \approx y_2 = -5,8778662 \cdot 10^{-1}$, $ \Delta = 1,4 \cdot 10^{-8}$ $\ln x \approx x_3 = -5,8778677 \cdot 10^{-1}$, $ \Delta = 3 \cdot 10^{-8}$
e^x	0,555 5555 0,555 5555	$\bar{m} = \{1, 3, 5\}$ $x_3 = 0,1535722 \cdot 10^{-2}$ $\bar{m} = \{1, 3, 5\}$ $x_3 = 0,1535722 \cdot 10^{-2}$	$y_0 = 1 + x_3 = 1,00115357$ $y_0 = 1,0000009 + x_3 = 1,00115366$	$e^x \approx y_3 = 1,7429068$, $ \Delta = 2 \cdot 10^{-6}$ $e^x \approx y_3 = 1,7429068$, $ \Delta = 5 \cdot 10^{-7}$
$\operatorname{tg} \varphi$	0,555 5555 0,555 5555	$\bar{m} = \{1, 4, 6\}$ $\varphi_3 = 1,3865350 \cdot 10^{-2}$ $\bar{m} = \{1, 4, 6\}$ $\varphi_3 = 1,3865350 \cdot 10^{-2}$	$y_0 = \varphi_3$ $x_0 = 1$ $y_0 = \varphi_3$ $x_0 = 0,99993915$	$\operatorname{tg} \varphi \approx y_3 / x_3 = 6,2077400 \cdot 10^{-1}$, $ \Delta = 1,2 \cdot 10^{-6}$ $\operatorname{tg} \varphi \approx y_3 / x_3 = 6,2077515 \cdot 10^{-1}$, $ \Delta = 8 \cdot 10^{-8}$
$\operatorname{arctg} \frac{y}{x}$	$y_0 = y = 0,555555$ $x_0 = x = 1$	$\bar{m} = \{1, 5, 7\}$ $y_3 = 5,628742 \cdot 10^{-3}$ $x_3 = 1,2796360$	—	$\operatorname{arctg} \frac{y}{x} = \theta = 0,5709848$, $ \Delta = 1 \cdot 10^{-8}$

th φ	0,555 5555	$\bar{m} = \{1, 8\}$ $\varphi_3 = 2,3432487 \cdot 10^{-3}$ $\bar{m} = \{1\}$ $\varphi_1 = 6,2494 \cdot 10^{-3}$	$\varphi_3 = 2,3431487 \cdot 10^{-3}$ $y_0 = \varphi_3 \quad x_0 = 1$ $y_0 = \varphi_1$ $x_0 = 0,99998475$	th $\varphi \approx y_2 / x_2 = 5,0467238 \cdot 10^{-1}$, $ \Delta = 2 \cdot 10^{-8}$ th $\varphi \approx y_1 / x_1 = 5,0467237 \cdot 10^{-1}$, $ \Delta = 1 \cdot 10^{-7}$
Arth $\frac{y}{x}$	$y_0 = y = 0,555555$ $x_0 = x = 1$	$\bar{m} = \{1, 4, 7\}$ $y_3 = 4,801377 \cdot 10^{-3}$ $x_3 = 7,1866863 \cdot 10^{-3}$	—	Arth $\frac{y}{x} = 6,2638123 \cdot 10^{-1}$, $ \Delta = 1,7 \cdot 10^{-7}$

Дальнейшее ускорение вычислений с использованием этих методов может быть достигнуто за счет использования сегментной аппроксимации для начальных (завершающих) приближений [11].

Таким образом, в работе приведен новый класс адаптивных алгоритмов, позволяющий ускорить вычисления за счет более широкого использования имеющегося в ЭВМ оборудования. Помимо этого, в отличие от обычных методов «цифра за цифрой», которые, в основном, предполагают аппаратную реализацию, описанные в статье адаптивные методы могут быть эффективными в ряде случаев и при программной реализации.

Список литературы

1. Volder J.E. The CORDIC trigonometric computing technique // IRE Trans. Electr. Comput. – 1959. – C8, N 3. – P. 330 – 334.
2. Meggit J.E. Psevdo division and psevdo multiplication processes // IBM J. – 1962. – № 6. – P. 210 – 226.
3. Байков В.Д., Смолов В.Б. Специализированные процессоры: Итерационные алгоритмы и структуры. – М.: Радио и связь, 1985. – 288 с.
4. Байков В.Д., Семотин С.А. Вычисление элементарных функций в ЭВМ. – М.: Радио и связь, 1982. – 64 с.
5. Евдокимов В.Ф., Стасюк А.Н. Параллельные вычислительные структуры на основе разрядных методов вычислений. – Киев: Наукова думка, 1987. – 312 с.
6. Самофалов К.Г., Луцкий Г.М. Основы теории многоуровневых конвейерных систем. – М.: Радио и связь, 1980. – 272 с.
7. Теслер Г.С. Организация параллельных вычислений элементарных функций с использованием экономичных алгоритмов // Тезисы докладов Всесоюзной школы-семинара «Распараллеливание обработки информации». – Львов. – 1985. – Кн.1. – С. 95 – 97.
8. Chen T.C. Automatic computation of exponentials, logarithms, rations and square roots // IBM Journal of Res. And Devel. – 1971. – N 4. – P. 380 – 388.
9. Wrathall C., Chen T.C. Convergence guarante, improvments for a fast hardware exp, log evaluation scheme // Proc. of 4-th Symp. on Comp. Arithmetic. – 1978. – P. 175 – 182.
10. Ercegovoc M.D. Radix-16 evaluation of certain elementary functions // IEEE Trans. of Comp. – 1973. – 22, N 6. – P. 561 – 566.
11. Попов Б.А., Теслер Г.С. Вычисление функций на ЭВМ: Справочник. – Киев: Наукова думка, 1984. – 600 с.