

- к созданию датчиков с расширенными возможностями архивации и обработки данных измерений, выполняющих функции обнаружения заданных событий и осуществления различных законов регулирования и логического управления [12];
- к созданию датчиков, определяющих и компенсирующих влияние электромагнитных наводок, а также плохих контактов в цепи путём применения проверки на достоверность;
- к созданию датчиков самостоятельно осуществляющих предварительную оценку состояния измеряемого параметра [13];
- к созданию датчиков, компенсирующих погрешность датирования.

С помощью математических и алгоритмических методов можно компенсировать:

- нелинейность, входящую в состав основной погрешности;
- увеличение нелинейности из-за изменения температуры;
- температурную погрешность из-за изменения свойств материала [7];
- погрешность датирования отсчета аналого-цифрового преобразователя или цифрового измерительного прибора.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Фрайден Дж.* Современные датчики. Справочник. – М.: Техносфера, 2005. 592 с.
2. <http://www.metran.ru/home/pr/pdf/1dd.pdf>
3. [http://kip.industry.su/honeywell/Field\\_Instruments/Transmitters/Pressure/ST3000\\_DIFFERENTIAL/Rus/34-ST-03-60\\_R.pdf](http://kip.industry.su/honeywell/Field_Instruments/Transmitters/Pressure/ST3000_DIFFERENTIAL/Rus/34-ST-03-60_R.pdf)
4. <http://www.tek-know.ru/1/102.html>
5. [www.emersonprocess.com/russia/files/4600\\_PDS\\_REV\\_AA\\_RUS.pdf](http://www.emersonprocess.com/russia/files/4600_PDS_REV_AA_RUS.pdf)
6. ГОСТ 8.009-84
7. <http://www.metran.ru/home/pr/pdf/leaflets/ddrosemount.pdf>
8. <http://chipinfo.ru/literature/chipnews/200105/3.html>
9. *Васильев В.А.* Методы уменьшения температурной погрешности датчиков давления.// Технология и конструирование в электронной аппаратуре. – 2002. – №4–5. – С. 50–54.
10. *Мартынов Д.Б.* Стучебников В.М. Температурная коррекция тензопреобразователей давления на основе КНС.// Датчики и Системы. – 2002. – №10. –С. 6–12.
11. [http://www.honeywell.ru/products/page3\\_2\\_7.shtml](http://www.honeywell.ru/products/page3_2_7.shtml).
12. *Винник Т.Л.* Современные технологии измерения температуры.// Мир измерений. – 2005. – № 3. – С. 4–11.
13. *Пьявченко О.Н.* Проектирование локальных микрокомпьютерных систем – Таганрог: Изд-во ТРТУ, 2005. 238 с.

**В.Е. Золотовский, Д.В.Золотовский**

#### **ВЫЧИСЛЕНИЕ ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ С АРГУМЕНТАМИ, ПРЕДСТАВЛЕННЫМИ ПОЛЯМИ**

Основные трудности численного определения элементарных функций с аргументом, представленных полями, является переменная точность вычислений, связанная с изменением формата данных в зависимости от задачи. Под полем данного понимается число, разрядность которого значительно превышает разрядность слова (32бита) или двойное слово (64 бита). Для представления поля принято формирование его в виде набора групп по  $m$  – бит, как правило, группа имеет 8 или 16 бит. Это позволяет упростить схемы ускоренного переноса в группах. Операции над группами ведутся независимо [1].

В вычислительных системах используются стандартные программы для фиксированных форматов данных (обычно  $n = 64$  бита). В системах структурного моделирования программы вычисления функций ведутся с аргументами, заданными полями.

В последнем случае использование полимиальных приближений затруднительно, так как для каждого случая требуется свой набор коэффициентов полинома. Использование же одного набора, ориентированного на максимальную точность, приводит к увеличению числа малопродуктивных операций.

В рассматриваемом случае наиболее удобным являются итерационные методы и разложения функции в ряды. Рассмотрим их реализацию для случая представления аргумента полем.

Итеративные процессы. Пусть задано выражение

$$y = f(x) \quad (1)$$

и требуется найти значение  $y$  в некоторой известной точке  $x = a$ , где  $a$  – фиксированная величина, заданная полем  $L(N)$  с числом групп  $N$ . Так как вычисление  $f(L(N))$  затруднительно, то будем искать значение  $y$  как корень уравнения  $F(y, L(N))$ .

В соответствии с [2] получаем следующий итерационный процесс

$$y_{n+1} = y_n - \frac{F(x, y_n)}{F'(x, y_n)}. \quad (2)$$

Последовательность  $y_{n+1}$  сходится к  $y$ , если выполняется условие Липшица с константой  $q$

$$|F(x, y_n) - F(x, y_{n+1})| \leq q \cdot |y_{n+1} - y_n|. \quad (3)$$

Тогда начальное приближение следует выбирать из окрестности, где

$$|F'(x, y_0)| \leq q < 1.$$

Так как входная переменная представлена полем, то итерационный процесс можно проводить, используя не все поля сразу, а последовательно достигая равенства, начиная со старшей группы.

Тогда вначале ищется приближение  $y$  для  $x = x_1$ , где  $x_1$  – старшая группа. Если  $|y_{n+1} - y_n| < \varepsilon$ , то условно можно считать, что  $y_{n+1} \approx y_n$ . В этом случае переходим на следующий этап операции, увеличивая разрядность  $x = x_1 + x_2$ , где  $x_2$  – следующая группа разрядов. Тогда имеем

$$y_{n+2} = y_n - \frac{F((x_1 + x_2), y_2)}{F'((x_1 + x_2), y_2)}. \quad (4)$$

Откуда

$$\Delta y_{n+2} = y_{n+1} - \frac{\partial}{\partial x_2} \left( \frac{F((x_1 + x_2), y_n)}{F'((x_1 + x_2), y_n)} \right) \Delta x_2$$

и

$$y_{n+2} = y_{n+1} + \Delta y_{n+2}.$$

Далее процесс продолжается по схеме (2) и если выполняется условие (3), то входной аргумент увеличивается на следующую группу и так до окончания поля.

Вторым методом, позволяющим проводить вычисление функций с аргументом, представленным полем, является разложение в степенной ряд (ряд Тейлора или Маклорена) [3]. Ряд Маклорена имеет вид

$$f(x) = f(0) + f'(0) \cdot x + \frac{f''(0)}{2!} \cdot x^2 + \dots + \frac{f^{(P)}(0)}{P!} \cdot x^P + \dots \quad (5)$$

Как известно [4], ошибка вычисления функции  $f(x)$  определяется как разность

$$E_0(x) = f(x) - \sum_{k=0}^P \frac{f^{(K)}(\theta \cdot x)}{k!} \cdot x^K, \quad (6)$$

которая может быть оценена по остаточному члену полинома, так как  $0 < \theta < 1$  и  $|x| < 1$ , то  $|R_n(x)| \leq \frac{1}{(p+1)!}$ , что позволяет получить предварительную оценку

числа членов полинома  $r^{-n} \leq \frac{1}{(p+1)!}$ , где  $r$  – основание системы счисления.

$$\text{Откуда } -n \ln r = \ln \left( \frac{1}{(p+1)!} \right) = - \sum_{K=1}^P \ln(k+1), \text{ и } \frac{1}{\ln r} \sum_{K=1}^P \ln(k+1) \geq n.$$

$$\text{Для двоичной системы счисления } n = 1,5 \sum_{K=1}^P \ln(k+1).$$

Например, если  $n = 128$ , то  $p \approx 43$ . Ясно, что данная оценка, как правило, является завышенной. Для организации вычислительного процесса используем метод Горнера. В этом случае имеем [5]

$$f(x) = \left( \left( \left( \frac{f_0^P}{p!} \cdot x + \frac{f_0^{p-1}}{(p-1)!} \right) \cdot x + \frac{f_0^{p-2}}{(p-2)!} \right) \cdot x + \dots + f_0' \right) \cdot x + f(0).$$

$$\text{Вычисление начинается со старшей скобки, тогда } \frac{1}{(p-1)!} \cdot \left( \frac{f_0^P}{p} \cdot x + f_0^{p-1} \right)$$

и

$$f(x) = \prod_{k=1}^N \frac{1}{(p-k)} \left( \frac{a_k}{(p-k+1)} x + f_0^{p-k} \right), \quad (7)$$

$$\text{где } a_1 = f_0^P, a_k = \left( \frac{a_{k-1}}{(p-k+1)} x + f_0^{p-k} \right)$$

Для исключения деления представим  $P$  в виде произведения простых чисел. В табл. 1 представлены простые числа и их обратная величина.

Таблица 1

К	r=10	r=2
2	0,5 (0)	0.1(0)
3	0.(3)	0.(01)
5	0.2(0)	0.(0011)
7	0.(142957)	0.(001)
11	0.(09)	0.(000111)
13	0.(076923)	0.(000100111011)
17	0.(05882352941176470)	0.(00001111)
19	0.(052631578473684210)	0.(000011100101)
23	0.04347826086956521738(9)	0.(0000101001)
29	0.(0310)	0.(000010001101000111101111)
31	0.0322(5806451612)	0.(00001001)
37	0.(027)	0.(01101110101100111110010001010011)
41	0.(024390)	0.(00000110000110100111)
43	0.(023255813953488372093)	0.(000001011111)
47	0.(021276595744680851.....)	0.(00000110001)
53	0.(0188679245283)	0.(000001001101)
57	0.(017543859649122807)	0.( )

Таким образом, для вычисления выражения (7) величина Р представляется как произведение простых чисел  $p = \prod_{K=1}^l p_K$ , где  $p_K$  – простое число. Тогда

$$x \cdot \frac{1}{p} = \left( \left( \left( x \cdot \frac{1}{p_1} \right) \cdot \frac{1}{p_2} \right) \dots \cdot \frac{1}{p_n} \right). \quad (8)$$

Для умножения на обратную величину простого числа достаточно получить произведение периода и исходного числа, а затем, сдвигая последовательно полученный результат на длину периода, складываем полученные значения друг с другом до достижения необходимой точности.

Пример. Пусть  $x = 01011011$  и необходимо найти  $x \cdot \frac{1}{p_1}$ , где  $p_1 = 3$ . Для

этого умножаем на период  $x \cdot \frac{1}{p} = 0.0001011011$ .

Затем сдвигаем результат на три разряда и складываем

$$\begin{array}{r} 00000001011011 \\ 00001011011000 \\ \hline 00001100110011 \end{array}$$

Еще раз осуществляем сдвиг

$$\begin{array}{r} 00000000001011011 \\ 00001100110011000 \\ \hline 00001100111110011 \end{array}$$

И так до получения необходимой точности. Затем полученный результат умножается на  $1/p_2$ ,  $1/p_3$  и т.д. до получения искомой величины.

В качестве примера рассмотрим вычисление значений показательной функции.

Пусть задана функция  $y = e^x$ , где  $x$  – аргумент, представленный полем, в полулогарифмической форме, т.е.

$$x = x \cdot 2^N \cdot 2^m, \tag{9}$$

где  $N$  – порядок, определяющий количество сдвигов на число разрядов групп, а  $m$  – порядок сдвига на число незначащих разрядов внутри старшей группы.

Следует отметить, что если  $N$  – положительное число, то число групп результата экспоненциально возрастает. Действительно, при  $N = 1$  число групп при  $n = 16$  (число разрядов в группе) возрастает до 16. При  $N = 2$  до 256 и т.д. В связи с этим введем ограничение. Если  $N$  достигает максимально-разрешённого значения  $P$ , то это считается переполнением, и вычисления прекращаются. Величина  $P$  устанавливается исходя из возможностей аппаратуры. Вторым фактором, устанавливающим ограничение на величину  $N$ , является необходимость представления величины  $e$  с требуемой точностью. Определим погрешность вычисления экспоненты. Для простоты изложения положим  $m = 0$ , тогда  $x$  представляется следующим образом:

$$x = \underbrace{\times \times \times \dots \times \times \times \times \times \times \times}_{N \text{ - группа}}, \underbrace{\times \times \times \dots \times \times \dots \times \times \times \times \times}_{(L - N) \text{ - группа}}$$

Разобьем  $x$  на две части – целую и дробную, т.е.  $x = x_y + x_{yp}$ . Тогда  $y = e^{xy} \cdot e^{x_{yp}}$ .

Максимальная погрешность образуется при максимальном значении целой части поля  $x_{max}$ , тогда  $x_{max} = 2^{m \cdot N} - 2^{-(L-N) \cdot m}$  и  $y = e^{2^{mN}} \cdot e^{-2^{-(L-N)m}}$ .

Величина  $y_T = e^{2^{mN}}$  может быть найдена как произведение  $y_T = \prod_{i=1}^{2^{mN}} e$ .

Таким образом, если  $e$  найдена с погрешностью  $\varepsilon$ , то

$$y^x = (e + \varepsilon)^{2^{mN}} \approx e^{2^{mN}} + 2^{mN} \cdot e^{(2^{mN} - 1)} \varepsilon,$$

и

$$\varepsilon_y = y^T - y^x = 2^{mN} \cdot e^{(2^{mN} - 1)} \varepsilon. \tag{10}$$

Полагая, что  $\varepsilon > 2^{-m(L-N)}$ , получим

$$\varepsilon_y = 2^{2^{mN} - 1 + mN - m(L-N)} = 2^{2^{mN} - mL - 1}.$$

Откуда следует, что общая длина дробной части при условии, что погрешность не превзойдет значения младшего разряда поля, должна быть равна  $L_{др} = (L-N)m + Rm$ , где  $R$  – число дополнительных групп дробной части ( $R = 2^{mN}$ ).

Длина целой части равна  $L_{цел} = Nm$ , а общая длина поля  $(L + R)$  - групп.

В настоящее время величина  $e$  определена с приблизительно 310 десятичных знаков ( $n \approx 1023$  двоичных разрядов или поле длиной  $L = 64$  группы при  $m=16$ ). Поле указанной длины можно считать предельным при построении систем моделирования. Если  $N = 0$ , то для вычисления используется непосредственно формула разложения

$$e^x = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^k}{k!}, \tag{11}$$

Для оценки количества членов разложения, обеспечивающих необходимую погрешность вычисления функции для заданного поля, воспользуемся формулой оценки погрешности [5]

$$R \leq \frac{q^{\kappa+1}}{k! \cdot k}.$$

Откуда

$$\left(2^{-16}\right)^L \approx \frac{q^{k+1}}{k! \cdot k},$$

где  $0 < q < 1$ .

Путем несложных преобразований  $q$  всегда можно привести к интервалу  $0 < |q| \leq 1/2$ . Действительно, если  $x$  лежит в интервале  $1/2 \leq 1$ , то, полагая  $x=1-(1-x)$ , получим  $e^x = e \cdot e^y$ , где  $y = -(1-x)$ .

Таким образом, имеем

$$2^{-16L} = \frac{2^{-(\kappa+1)}}{k! \cdot k}. \quad (12)$$

Прологарифмируем правую и левую части выражения (12), тогда

$$16N = (k+1) + \ln k! + \ln k. \quad (13)$$

Выражение (13) используется для итерационной оценки числа членов разложения. Для этого составляется таблица логарифмов факториала. Затем подбором находят значение  $K$ , обеспечивающее требуемое число разрядов. Например, для стандартной разрядности  $L = 4$  необходимо сохранять 15 членов ряда.

Таблица 2

Таблица факториалов и обратных им величин

n	n!	1/n!
2	2	0.5
3	6	0.1(6)
4	24	0.041(6)
5	120	$0.8(3) \cdot 10^{-2}$
6	720	$0.19889 \cdot 10^{-2}$
7	5040	$0.19841 \cdot 10^{-3}$
8	40320	$0.24802 \cdot 10^{-4}$
9	362880	$0.27557 \cdot 10^{-5}$
10	3628800	$0.27557 \cdot 10^{-6}$
11	39916800	$0.25052 \cdot 10^{-7}$
12	479001600	$0.20877 \cdot 10^{-8}$
13	6227020800	$0.16059 \cdot 10^{-9}$
14	87178291200	$0.11471 \cdot 10^{-10}$
15	1307674368000	$0.76472 \cdot 10^{-12}$
16	20922789888000	$0.47795 \cdot 10^{-13}$
17	355687428096000	$0.28115 \cdot 10^{-14}$
18	6402373705728000	$0.15619 \cdot 10^{-15}$
19	121645100408832000	$0.82206 \cdot 10^{-17}$
20	2432902008176640000	$0.41103 \cdot 10^{-18}$
21	51090942171709440000	$0.19572941 \cdot 10^{-19}$
22	112400072777607680000	$0.8896791392 \cdot 10^{-21}$
23	25852016738884976640000	$0.3868170170630 \cdot 10^{-22}$

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Золотовский В.Е.* Арифметические и алгоритмические основы проблемно-ориентированных вычислительных систем. – Таганрог: Изд. ТРТУ. 2003. – 185с.
2. *Люстерник Л.А., Абрамов А.А., Шестаков В.И., Шура-Бура М.Р.* Решение математических задач на автоматических цифровых машинах Изд. АН СССР, 1952.
3. *Смирнов В.И.* Курс высшей математики. Т.1. – М. Изд-во Физматлит. 1961.
4. *Благовещенский Ю. В., Теслер Г. С.* Вычисления элементарных функций на ЭВМ – М.: Изд-во «Техника», 1977. – 288с.
5. *Демидович Б. П., Марон И. А.* Основы вычислительной математики. – М., 1966. – 664с.

**В.Ф.Гузик, В.Е.Золотовский, П.В.Савельев**

**ВЫПОЛНЕНИЕ МАТЕМАТИЧЕСКИХ ОПЕРАЦИЙ В СИСТЕМЕ СИМВОЛЬНЫХ ВЫЧИСЛЕНИЙ**

Введение в комплекс структурного моделирования модулей работы с математическими данными, представленными в общем виде, дает возможность выйти на качественно новый этап развития средств моделирования. Так появится возможность решить задачи автоматического построения компьютерной модели из математических моделей компонент, оптимизировать данную модель, определить параллельные участки вычислений для случая моделирования сложных технических систем распределёнными средствами.

Благодаря выбранному формату представления символьных выражений – в виде графического дерева, выполнение операций сложения и вычитания требует минимум усилий от вычислительной системы.

Для вычисления суммы двух операндов, которые представлены в виде деревьев, необходимо объединить точки входа. Тем самым получится результирующий граф-дерево, расширенный в ширину на размер второго операнда. Дополнительные ребра как раз и сыграют роль выполняемой операции.

Для вычитания алгоритм несколько усложняется. Как известно, вычитание – это сложение с обратным знаком. Поэтому помимо объединения точек входа потребуется произвести действие над знаками первого ряда слагаемых вычитаемого. А именно, нужно изменить знаки на противоположные у тех слагаемых, которые соединены непосредственно с точкой входа вычитаемого.

Покажем на примере, как происходит склейка деревьев представления символьных выражений. Возьмём исходные выражения:

$$\begin{aligned}7A(2B + C) + 5E \\ 4E - 7A.\end{aligned}$$

Внутри вычислительной системы они будут выглядеть следующим образом (рис.1).