Информационные процессы, Том 12, № 3, 2012, стр. 248–252.

Рекурсивный алгоритм вычисления логарифма

М.С.Сальников

НИУ ВШЭ, факультет математики, Москва, Россия
Поступила в редколлегию 15.09.2012

Аннотация—В работе обосновывается рекурсивный алгоритм вычисления натуральных логарифмов, выводится оценка погрешности вычисления и описываются реализации алгоритма на языках ЛИСП и С++.

КЛЮЧЕВЫЕ СЛОВА: алгоритм, логарифм, рекурсия, погрешность.

1. ВВЕДЕНИЕ

В 1981 году В.Л.Стефанюк предложил подход [1] к рекурсивному[1] вычислению элементарных функций. В частности, для приближенного вычисления $sin(x)$ была предложена формула

$$Rsin(x;\delta) \;=\; \begin{cases} 3\cdot Rsin(x/3;\delta) - 4\cdot Rsin^3(x/3;\delta), & \text{если } |x|>\delta \\ x & \text{иначе,} \end{cases}$$

где число δ – параметр. Фактически, формула для вычисления $Rsin(x;\delta)$ сконструирована *из* тождества $sin(x) = 3sin(x/3) - 4sin^3(x/3)$ и
из первого замечательного предела: $sin(x) \sim x$ при $x \to 0$.

Для вычисления $ln(x)$ в [1] предложена формула

$$Rlog(x;\delta) \;=\; \begin{cases} 2\cdot Rlog(\sqrt{x};\delta), & \text{если } |x-1|>\delta \\ 2\cdot(x-1)/(x+1) & \text{иначе.} \end{cases}$$

В настоящей работе излагается подход к рекурсивному вычислению $ln(x)$, при котором отпадает необходимость считать элементарной операцию извлечения квадратного корня.

2. ПОДХОД К РЕКУРСИВНОМУ ОЦЕНИВАНИЮ ЛОГАРИФМА

Следуя [2], будем использовать для вычисления $ln(x)$ функцию $log1p(x) \overset{def}{=} ln(1+x)$, где $x > -1$. Если алгоритм вычисления $log1p(x)$ известен, то вычисление $ln(x)$ выполняется за одно действие:

$$ln(x) = log1p(x-1). \tag{1}$$

Легко проверить три примечательных свойства функции $log1p(x)$.

Свойство 1. *Для* $x > -1$ *справедливо тождество* $log1p(x) = log1p(\frac{x}{x+2}) - log1p(-\frac{x}{x+2})$

Свойство 2. $log1p(x) \sim x$ *при* $x \to 0$.

Свойство 3. *Для* $x > -1$ *справедливо неравенство* $|\frac{x}{x+2}| < |x|$.

[1] В современной науке оценивания элементарных функций методы такого типа именуются редукционными [2].

Свойства 1-3 позволяют определить рекурсивную формулу для приближенного вычисления $log1p(x)$, причем свойство 3 гарантирует алгоритмическую корректность формулы.

$$Rlog1p(x;\delta) \ = \ \begin{cases} Rlog1p(\frac{x}{x+2};\delta) - Rlog1p(-\frac{x}{x+2};\delta), & \text{если } |x| > \delta \\ x & \text{иначе.} \end{cases}$$

Здесь δ - параметр, ограничивающий дробление аргумента и влияющий на точность вычисления. Идея рекурсивного вычисления логарифма вытекает из (1) и выражается так:

$$ln(x) \approx Rlog1p(x-1;\delta) \tag{2}$$

На языке ЛИСП [3] метод (2) может быть реализован с помощью трех функций:

(**RLOG** (LAMBDA(X) (RLOG1P (SUB1 X))))

(**RLOG1P** (LAMBDA(X)
    (COND ((GREATERP (ABS X) DELTA) (MINX (FQUOTIENT X (PLUS X 2.))))
        (T X)   )))

(**MINX** (LAMBDA(U) (DIFFERENCE (RLOG1P U) (RLOG1P (MINUS U)))))

Функция RLOG вычисляет искомое приближение $ln(x)$, но фактически лишь вызывает основную функцию RLOG1P с аргументом $X-1$. При обращении к вспомогательной функции MINX один раз вычисляется $U = X/(X+2)$, а сама функция MINX:
– организует пару вызовов RLOG1P с аргументами U и $-U$, и
– вырабатывает в качестве результата разность полученных чисел.

Остановимся на вычислительных особенностях метода (2), позволяющим, в конечном итоге, оценить его погрешность. Прежде всего рассмотрим конкретный пример.

Пример 1. Вычислим $E = Rlog1p(-1/13; 0.02)$.

$$\begin{aligned} E \ = \ & Rlog1p(-1/25;0.02) - Rlog1p(+1/25;0.02) \quad = \\ & \Big(Rlog1p(-1/49;0.02) - Rlog1p(+1/49;0.02)\Big) \quad - \\ & \Big(Rlog1p(+1/51;0.02) - Rlog1p(-1/51;0.02)\Big) \quad = \\ & \Big(\Big(Rlog1p(-1/97;0.02) - Rlog1p(+1/97;0.02)\Big) \quad - \\ & \Big(Rlog1p(+1/99;0.02) - Rlog1p(-1/99;0.02)\Big)\Big) - \Big(+1/51 - (-1/51)\Big) = \\ & \Big(\Big(-1/97 - (+1/97)\Big) - \Big(+1/99 - (-1/99)\Big)\Big) - \Big(+1/51 - (-1/51)\Big) \end{aligned}$$

Полученное скобочное выражение однозначно представляется деревом, изображенным на рисунке 1. Окончательный результат вычисления выглядит так:

$$E \ = \ -\frac{2}{97} - \frac{2}{99} - \frac{2}{51} \ \approx \ -0.08$$

Конец примера.

Древовидная структура вычисления позволяет говорить о внутренних и терминальных узлах вычислительной схемы, о ее ярусном строении и глубине. Внутренние узлы имеют непосредственных потомков, этим узлам соответствует вычисление

$$Rlog1p(x;\delta) = Rlog1p(\tfrac{x}{x+2};\delta) - Rlog1p(-\tfrac{x}{x+2};\delta).$$

Терминальные узлы потомков не имеют, им соответствует вычисление $Rlog1p(x;\delta) = x$. Корень дерева составляет нулевой ярус вычислительной схемы; $(i+1)$-й ярус составляют все непосредственные потомки узлов i-го яруса. Глубина вычислительной схемы есть максимальный номер представленного в ней яруса.

Для примера 1 вычислительная схема имеет:
▷ 5 внутренних и 6 терминальных узлов;

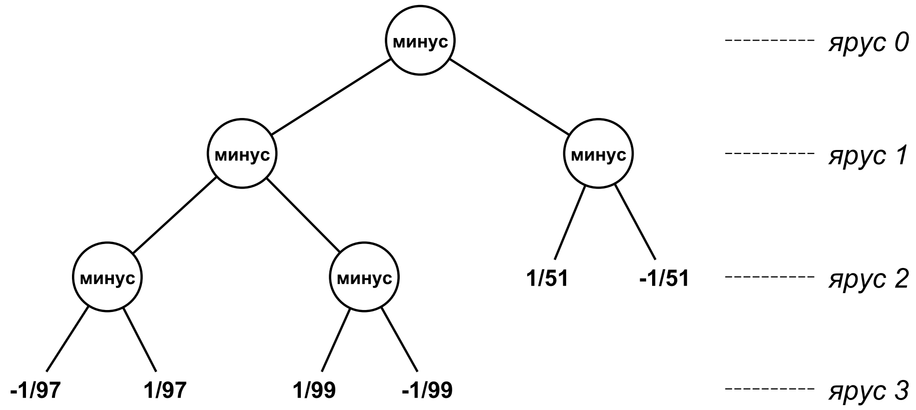

Рис.1 Вычисление $Rlog1p(-1/13; 0.02)$

▷ 2 внутренних узла первого яруса
для подзадач $Rlog1p(-1/25; 0.02)$ и $Rlog1p(+1/25; 0.02)$;
▷ 4 узла второго яруса:
2 внутренних для подзадач $Rlog1p(-1/49; 0.02)$ и $Rlog1p(+1/49; 0.02)$;
2 терминальных для подзадач $Rlog1p(+1/51; 0.02)$ и $Rlog1p(-1/51; 0.02)$;
▷ 4 терминальных узла третьего яруса
для подзадач $Rlog1p(-1/97; 0.02)$, $Rlog1p(+1/97; 0.02)$,
$Rlog1p(+1/99; 0.02)$, $Rlog1p(-1/99; 0.02)$.
Глубина вычислительной схемы из примера 1 равна 3.

Рекурсивный метод вычисления $Rlog1p$ применим ко всем $x > -1$. Однако в окрестности -1, а также при больших x объем вычислений сильно возрастает.

Пример 2.
Схема для $Rlog1p(-0.9999; 0.001)$ имеет глубину 24 и содержит 13348 терминальных узлов.
Схема для $Rlog1p(-0.999; 0.001)$ имеет глубину 20 и содержит 9976 терминальных узлов.
Схема для $Rlog1p(-0.5; 0.001)$ имеет глубину 10 и содержит 1000 терминальных узлов.
Схема для $Rlog1p(+0.5; 0.001)$ имеет глубину 9 и содержит 512 терминальных узлов.
Конец примера.

Указанный недостаток можно устранить, полагая, что метод (2) рассчитан на применение вычислительной техники. Как известно [2], в компьютере каждое положительное вещественное x хранится в виде целого числа *порядок*(x) и вещественного числа *мантисса*(x) таких, что:

$$x = 2^{\textit{порядок}(x)} \cdot \textit{мантисса}(x) \quad \text{и} \quad 0.5 \le \textit{мантисса}(x) < 1.$$

Отсюда следуют равенство (3) и выражение (4)

$$ln(x) = \textit{порядок}(x) \cdot ln(2) + log1p(\textit{мантисса}(x) - 1) \tag{3}$$
$$ln(x) \approx \textit{порядок}(x) \cdot ln(2) + Rlog1p(\textit{мантисса}(x) - 1; \delta). \tag{4}$$

Величина $ln(2)$ – известна заранее, поэтому в (4) основная тяжесть вычисления приходится на $Rlog1p(u; \delta)$. На языке C++ (с использованием библиотеки Math.h) вычисление выражения (4) выглядит так:

```
const double Delta = 0.001;

double Rlog1p(double X) { if (abs(X) <= Delta) return X;
                          double Z = X/(X+2);
                          return Rlog1p(Z) - Rlog1p(-Z);     }
```

```
double Rlog(double X)  { int N;
                         double U = frexp(X,&N);          // U=мантисса(X), N=порядок(X)
                         return N * 0.693147 + Rlog1p(U-1); }    // ln(2)=0.693147
```

Строго говоря, в выражении (4) аргумент u находится в пределах от -0.5 до 0. Однако в ходе рекурсивного вычисления в подзадачах могут возникнуть и положительные аргументы, впрочем из свойства 3 следует, что во всех подзадачах $|u| \leq 0.5$

3. ОЦЕНКА ПОГРЕШНОСТИ

Принципиальный вопрос, связанный с использованием выражения (4), состоит в возможности вычисления логарифма с заранее заданной точностью. В настоящем разделе будем игнорировать погрешности выполнения арифметических операций, полагая, что погрешность возникает только в терминальных узлах вычислительной схемы при замене $Rlog1p(x)$ на x. При таком подходе точность вычисления логарифма целиком определяется точностью вычисления $Rlog1p(x)$ для $|x| \leq 0.5$.

Для определения общей погрешности необходимо оценить:
–1– погрешность вычисления в одном терминальном узле; ▷▷ свойство 4
–2– количество терминальных узлов. ▷▷ свойство 5

Свойство 4. *Для* $|x| \leq \delta \leq 0.5$ *справедливо неравенство* $0 \leq x - log1p(x) \leq \frac{1}{2(1-\delta)} x^2$.

Свойство доказывается стандартным исследованием функций $f(x) = x - log1p(x)$ и $g(x) = \frac{1}{2(1-\delta)} x^2 - f(x)$ на монотонность и экстремум. Свойство иллюстрирует рисунок 2.

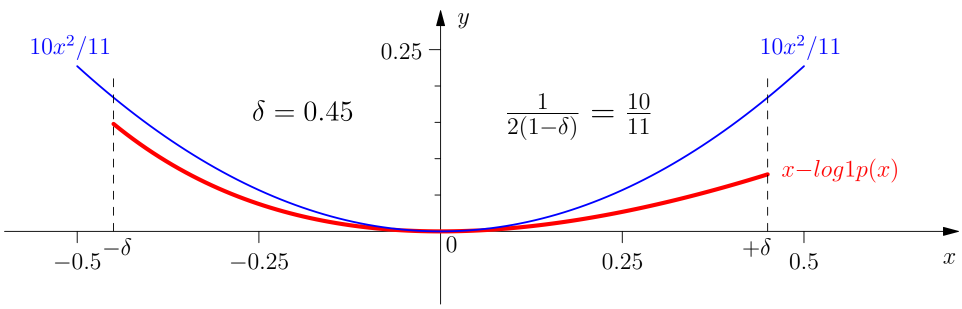

Рис.2 Графики $x - log1p(x)$ и $\frac{1}{2(1-\delta)} x^2$ для $\delta = 0.45$

В терминальном узле:
– погрешность вычисления есть величина $x - log1p(x)$;
– соблюдается ограничение $|x| \leq \delta$; и из свойства 4 вытекает
– оценка погрешности: $0 \leq x - log1p(x) \leq \frac{1}{2(1-\delta)} \delta^2$.

Свойство 5. *Если для некоторого* k, $k \geq 0$, *имеет место* $-\frac{1}{2^k+1} \leq x \leq \frac{1}{2^k+1}$, *то* $-\frac{1}{2^{k+1}+1} \overset{(a)}{\leq} \frac{x}{x+2} \overset{(b)}{\leq} \frac{1}{2^{k+1}+1}$, *и, следовательно,* $\left|\frac{x}{x+2}\right| \leq \frac{1}{2^{k+1}+1}$.

Для доказательства свойства обозначим: $A \overset{def}{=} \frac{1}{(2^{k+1}+1)(x+2)}$. При всех $|x| \leq 0.5$ и $k \geq 0$ справедливо: $A > 0$. Доказательство свойства состоит в исследовании двух случаев.

В первом случае, когда $0 \leq x \leq \frac{1}{2^k+1}$, и, следовательно, $1 - x2^k \geq x \geq 0$, имеем:
неравенство (a) очевидно,
неравенство (b) следует из $\frac{1}{2^{k+1}+1} - \frac{x}{x+2} = A(x + 2 - x2^{k+1} - x) = 2A(1 - x2^k) \geq 0$.

Во втором случае, когда $-\frac{1}{2^k+1} \leq x \leq 0$, и, следовательно, $(2^k + 1)x + 1 \geq 0$, имеем:

неравенство (b) очевидно,
неравенство (a) следует из $\frac{x}{x+2} + \frac{1}{2^{k+1}+1} \; = \; A(x2^{k+1}{+}x{+}x{+}2) \; = \; 2A\left((2^k+1)x+1\right) \; \geq \; 0.$

Утверждение. *Для* $|x| \leq 0.5$ *и* $n > 0$ *справедливо неравенство*

$$|log1p(x) - Rlog1p(x; 2^{-n})| \leq \frac{1}{2(1{-}2^{-n})}2^{-n}.$$

Доказательство. Обозначим x_i произвольный аргумент i-го яруса вычислительной схемы $Rlog1p(x; 2^{-n})$.

Поскольку $x_0 = x$, то	$|x_0| \; \leq \; 1/(2^0+1)$.
Согласно свойству 5	$|x_1| \; \leq \; 1/(2^1+1)$.
Аналогично	$|x_2| \; \leq \; 1/(2^2+1)$.
и т.д.	$|x_n| \; \leq \; 1/(2^n+1) \; \leq \; 2^{-n}$ – условие терминальности узла.

То есть на n-ом ярусе вычисления могут располагаться только терминальные узлы, а значит n – максимально возможная глубина вычислительной схемы. Согласно свойству 4 погрешность в каждой терминальной вершине не превышает $\frac{1}{2(1-2^{-n})}(2^{-n})^2$. В бинарном дереве, каковым и является схема вычисления $Rlog1p$, количество терминальных узлов T и глубина n связаны соотношением $T \; \leq \; 2^n$. Таким образом, суммарная ошибка вычисления $log1p(x)$ – величина $T\frac{1}{2(1-2^{-n})}(2^{-n})^2$ не превосходит $2^n\frac{1}{2(1-2^{-n})}(2^{-n})^2 = \frac{1}{2(1-2^{-n})}2^{-n}$.
Утверждение доказано.

Возвращаясь к исходной задаче приближенного вычисления логарифма, можно утверждать, что для любого $x > 0$ выполняется неравенство

$$|ln(x) - h| \; \leq \; \frac{1}{2(1{-}2^{-n})}2^{-n}, \quad \text{где} \quad h = \textit{порядок}(x) \cdot ln(2) + Rlog1p(\textit{мантисса}(x){-}1; 2^{-n}).$$

4. ЗАКЛЮЧЕНИЕ

Компактность и простота реализации описанного алгоритма позволяют надеяться, что он найдет применение для целей обучения, а также в качестве "спарринг-партнера" при тестировании традиционных алгоритмов вычисления логарифмов.

СПИСОК ЛИТЕРАТУРЫ

1. Стефанюк В.Л. Рекурсивное оценивание арифметических функций в системах ЛИСП. *Программирование*, 1981, No.5, стр. 92-94.
2. Brent R., Zimmermann P. *Modern Computer Arithmetic.* Cambridge University Press, 2011.
3. McCarthy J., Abrahams P., Edwards D., Hart T., Levin M. *LISP 1.5 Programmer's Manual.* M.I.T. Press, Cambridge, 1985.

A recursive algorithm for the logarithm evaluation

M.S.Salnikov

We propose recursive algorithm for evaluating the natural logarithms. This algorithm is based on formula $log(1{+}x){=}log(1{+}x/(x{+}2)){-}log(1{-}x/(x{+}2))$.

KEYWORDS: algorithm, logarithm, recursion, precision.