Комп'ютерна симуляція процесу керування мотором постійного струму з незалежним збудженням // Автоматика, вимірювання та керування: Вісник Нац. ун-ту "Львівська політехніка". – 2010. – № 665. – С. 12–18. 6.. Самотий В, Дзелендзяк У. Комп'ютерна симуляція системи керування мотором постійного струму з паралельним збудженням // Міжвідомчий науково-технічний збірник "Вимірювальна техніка та метрологія" – 2010. – № 71. – С. 51–58. 7. Самотий В., Дзелендзяк У. Математична модель каскаду "однофазний двопівперіодний випрямляч – мотор постійного струму з паралельним збудженням" // Автоматика, вимірювання та керування: Вісник Нац. ун-ту "Львівська політехніка". – 2013. – № 753. – С. 3–8. 8. Samotyy V., Dzelendzyak U.Mathematical model of thyristor's system control of DC motor with independent excitation // Czasopismo Techniczne. Automatyka. 1-AC/2013, p. 79–91.

УДК 519.7

Л. В. Мороз, А. Гринчишин
Національний університет "Львівська політехніка",
кафедра безпеки інформаційних технологій

ШВИДКЕ ОБЧИСЛЕННЯ ФУНКЦІЇ Y=1/X З ВИКОРИСТАННЯМ МАГІЧНОЇ КОНСТАНТИ

Подано математичний опис перетворень при швидкому обчисленні обернено-пропорційної залежності з використанням магічної константи для чисел типу float та визначення оптимальних значень зміщень для адитивної корекції з метою зменшення відносних похибок обчислень .

Ключові слова: магічна константа, числа типу float, адитивна корекція, відносна похибка обчислень.

Mathematical description of transformations is given at a fast computation reciprocal with the use of magic constant for the numbers of type of float and determination of optimum values of biases for a additive correction with the purpose of decreasing of relative errors of computation .

Key words: magic constant, the numbers of type of float, additive correction, fast reciprocal.

Вступ

У роботах [1,2] швидкий зворотний (або обернений) квадратний корінь (fast reciprocal square root – frsqrt), що базується на застосуванні магічної константи [4,5], використано для реалізації обернено-пропорційної залежності (англ. – reciprocal) таким способом:

$$y_t = \frac{1}{x} = \frac{1}{\sqrt{x}} \cdot \frac{1}{\sqrt{x}}. \qquad (1)$$

На наш погляд, таке рішення не є оптимальним та ефективним ні за швидкодією, ні за простотою програмної або апаратної реалізації. Ми пропонуємо свій, покращений алгоритм. Перше наближення такого типу алгоритму було описано у відомій роботі Д. Блінна (J. Blinn) [3], однак точність його невисока – відносна похибка після двох ньютонівських ітерацій становить приблизно 0.00244 (8.67 коректних бітів результату) .

Мета роботи

Метою роботи є створення автономного алгоритму для реалізації обернено-пропорційної залежності з використанням магічної константи для чисел з плаваючою точкою (типу float) у форматі одинарної точності (single precision) стандарту IEEE-754 та визначення оптимальних значень зміщень для адитивної корекції формул Ньютона–Рафсона з метою зменшення відносних похибок обчислень.

Опис алгоритму

Для реалізації обернено-пропорційної залежності $y_t = 1/x$ пропонується такий алгоритм:

float reciprocal (float x)

{

int i = *(int*)&x; { *перевід числа x з floating-point у integer* }

i = 0x7ef311c3 -i;);{*ціле початкове наближення для reciprocal, де 0x7ef311c3 – магічна константа* }

float y = *(float*)&i; { *перевід i з integer у floating-point для отримання початкового наближення y_0* }

y= y*(2.0f -x*y); {*перша ньютонівська ітерація - наближення y_1* }

y = y*(2.0f - x*y); {*друга ньютонівська ітерація - наближення y_2* }

return x;

}

Цей алгоритм забезпечує мінімальну відносну похибку обчислень y_t для усього діапазону значень чисел типу float (режим single precision для стандарту IEEE-754).

Основні результати досліджень

Опишемо дію алгоритму з одночасним теоретичним обґрунтуванням процесів, що відбуваються при цьому.

1. Задається число x типу float.

Подамо число x у форматі IEEE-754, тобто запишемо x у форматі з плаваючою точкою у вигляді нормалізованого числа

$$x = (-1)^{S_x} M_x \cdot 2^{E_x}, \tag{2}$$

де S_x – знак (у даному випадку $S_x = 0$); E_x – порядок, який визначають за формулою :

$$E_x = \lfloor \log_2 x \rfloor = floor[\log_2 x]; \tag{3}$$

M_x – мантиса, яку розраховують за формулою:

$$M_x = \frac{x}{2^{E_x}}, \tag{4}$$

причому M_x подано у вигляді $M_x = 1 + m_x = 1.f$, де f – дробова частина мантиси. Звідси $m_x = M_x - 1 = 0.f$.

Переведемо число x у формат single precision для стандарту IEEE-754, в якому для зберігання двійкового представлення числа використовується 32-бітний регістр:

– один біт для S_x;

– 8 біт для E_x;

– 23 біти для m_x.

У десятковій системі числення це число запишеться як

$$x = (-1)^{S_x} 1.f \cdot 2^{e_x} = (-1)^{S_x} \cdot (1 + m_x) \cdot 2^{e_x}, \tag{5}$$

де $e_x = E_x + bias$ – зміщений порядок (зміщення $bias = 127$ для формату single precision стандарту IEEE-754).

Однак у двійковому представленні мантиса має фантомний біт, який не показується, тому вираз $1.f$ зображується лише у вигляді $0.f$. Тому ціле число I_x, яке відповідає двійковому представленню числа x у стандарті IEEE-754, зображується як

$$I_x = e_x \cdot N_m + 0.f \cdot N_m = (e_x + m_x) N_m = (bias + E_x + x \cdot 2^{-E_x} - 1) \cdot N_m . \tag{6}$$

2. Тепер переходимо до зображення магічної константи R

S_R	Q	T

– таке двійкове представлення магічної константи у 32-бітному регістрі, де $S_R = 0$, $Q = 253$. Звідси магічну константу можна зобразити у вигляді цілого числа як

$$I_R = Q \cdot N_m + T \,. \tag{7}$$

3. Далі шукається цілочислова різниця d :

$$d = I_R - I_x \,. \tag{8}$$

4. Після цього d переводиться у дійсне число типу float, яке і буде початковим наближенням y_0 обернено-пропорційної залежності, тобто $y_0 \approx \dfrac{1}{x}$. Алгоритм переведення такий:

– знаходиться зсунутий порядок

$$e_p = floor\left[\frac{d}{N_m}\right]; \tag{9}$$

– знаходиться справжній (незсунутий) порядок

$$E_p = e_p - bias \,; \tag{10}$$

– знаходиться дробова частина мантиси

$$m_p = \frac{d - E_p \cdot N_m}{N_m} = \frac{d}{N_m} - E_p \,; \tag{11}$$

– знаходиться початкове наближення y_0 у формі:

$$y_0 = (1 + m_p) \cdot 2^{E_p} \,. \tag{12}$$

Тоді можна оцінити абсолютну

$$\Delta_0 = y_0 - y_t \tag{13}$$

та відносну похибки

$$d_0 = \frac{\Delta_0}{y_t} \tag{14}$$

початкового наближення.

6. Після знаходження y_0 проводяться ітерації за формулою Ньютона–Рафсона для $y_t = 1/x$:

$$y_1 = y_0(2 - xy_0)\,; \tag{15}$$

$$y_2 = y_1(2 - xy_1)\,, \tag{16}$$

або у загальному випадку

$$y_{n+1} = y_n(2 - xy_n)\,. \tag{17}$$

З поданого опису можна сформувати строгу математичну модель формування початкового наближення y_0. Для цього спочатку запишемо вираз для y_0 в аналітичному вигляді. Якщо послідовно описати переведення x в I_x та в $d = I_R - I_x$ і зворотне переведення d в y_0 за допомогою рівнянь (2)–(12), то в результаті отримаємо, що початкове наближення y_0 у загальному випадку описується рівнянням:

$$y_0 = (-x2^{-E_x} + 2 - 2 \cdot bias + Q + t - E_x - E_p) \cdot 2^{E_p} \,, \tag{18}$$

причому E_x та E_p визначаються за формулами (3) та (9), (10) відповідно, а $t = \dfrac{T}{N_m}$.

Вираз (18) є імітаційною моделлю формування початкового наближення y_0 для форматів single precision та double precision стандарту IEEE-754.

Аналіз рівняння (18) показує , що наближення y_0 можна подати у вигляді:

$$y_0 = (a\,x + b\,) \cdot 2^{E_p} \,, \tag{19}$$

де

$$a = -2^{-E_x} \tag{20}$$

$$b = 2\cdot(1-bias)+Q+t-E_x-E_p\,, \tag{21}$$

звідки випливає, що y_0 – це кусково-лінійне наближення функції $y_t = 1/x$.

Тепер, маючи аналітичний вираз початкового наближення для будь-якого x, можна звузити діапазон значень аргумента, які підлягають дослідженню на максимуми похибок. Для цього застосуємо такий прийом. При обчисленні функції $1/x$ для чисел з плаваючою точкою використовується формат, подібний до IEEE-754: $x = M_x \cdot 2^{E_x}$, де E_x – ціле число, тоді як число M_x – мантиса, значення якої лежать у діапазоні $M_x \in [1,2)$. Тоді

$$y_t = \frac{1}{x} = \frac{1}{M_x}\cdot 2^{-E_x}\,. \tag{22}$$

Тому достатньо проаналізувати поведінку похибки наближення лише на одному проміжку значень $x \in [1,2)$, щоб описати закон поведінки похибки у всьому діапазоні зміни аргумента x, заданого у вигляді чисел типу float.

Аналітичне дослідження поведінки y_0 за допомогою рівнянь (18)–(21) ускладнене через наявність тут двох функцій типу *floor* у виразах для E_x та E_p. Тому спочатку отримаємо прості аналітичні вирази для y_0, задаючи відповідні цілі значення E_x та E_p з проміжку $[1,2)$, а потім перейдемо до аналізу рівняння абсолютної та відносної похибок на окремих ділянках цього проміжку. Розіб'ємо проміжок на дві ділянки: $x \in [1, x_t)$; $x \in [x_t, 1)$; де $x_t = 1+t$. Розглянемо ділянку $x \in [1, x_t)$, де значення параметрів є такими: $E_x = 0$; $E_p = -1$. Лінійне початкове наближення буде таким:

$$y_{01} = -\frac{x}{2} + 1 + \frac{t}{2}\,. \tag{23}$$

Відповідно для ділянки $x \in [x_t, 2)$ значення параметрів є: $E_x = 0$; $E_p = -2$. Тоді

$$y_{02} = -\frac{x}{4} + \frac{3}{4} + \frac{t}{4}\,. \tag{24}$$

Маючи аналітичні вирази початкового наближення y_0, перейдемо до вибору оптимального значення магічної константи R, яка забезпечить мінімальну відносну похибку обчислення функції y_t як для початкового наближення, так і після проведення першої та другої ітерацій за формулою Ньютона–Рафсона.

Наші дослідження, подібні до тих, які детально описано у [5], показують, що для функції $1/x$ мінімальну відносну похибку як для початкового наближення, так і для першої та другої ітерацій за формулою Ньютона–Рафсона забезпечить одна й та сама магічна константа, а саме $0x7ef311c3$. Для цієї константи, поданої у вигляді (6), значення параметрів будуть такими:

$T = 7541187$; $t = 0.89897954463958740234375$; $x_t = 1 + t = 1.89897954463958740234375$.

Графіки абсолютної Δ_0 та відносної d_0 похибок будуть такими:

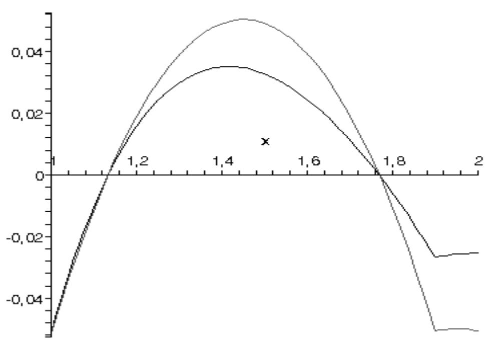
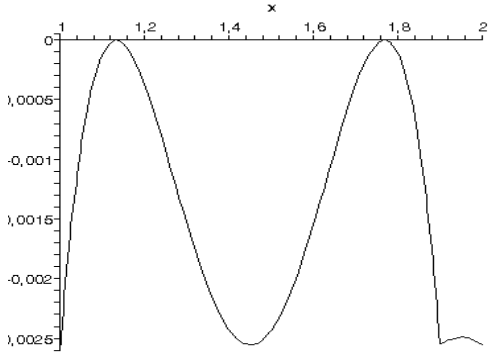


Рис. 1. Графік абсолютної та відносної похибок початкового наближення y_0 (червона крива – Δ_0, зелена- d_0) для $x \in [1,2)$

Після проведення першої ітерації за класичною формулою Ньютона–Рафсона (15) отримаємо відносну похибку $d_1 = (y_1 - y_t)/y_t$, графік якої подано на рис.2.

Рис. 2. Графік відносної похибки d_1 *для* $x \in [1,2)$



При цьому максимуми відносної похибки d_1 розміщено в точках:

$$x_{11\_\max} = 1;$$
$$x_{12\_\max} = 1 + t/2; \qquad (25)$$
$$x_{13\_\max} = 1 + t,$$

а нулі цієї ж похибки – у точках :

$$x_{11\_zero} = 1 + 1/2t - 1/2 \cdot (-4 + 4t + t^2)^{1/2};$$
$$x_{12\_zero} = 1 + 1/2t + 1/2 \cdot (-4 + 4t + t^2)^{1/2}. \qquad (26)$$

Другу ітерацію проводять за формулою (15). За характером зміни графік відносної похибки d_2 при цьому нагадує графік похибки d_1, а її максимальне значення становить $d_{2\max} \approx -6.51 \cdot 10^{-6}$ (17.2 коректних бітів) у всьому діапазоні значень чисел типу float. Отже, у запропонованому алгоритмі досягнуто зменшення відносної похибки більше ніж у 370 разів порівняно з алгоритмом Блінна. Слід також зазначити, що запропонований алгоритм має вищу швидкодію порівняно з [1,2], оскільки містить меншу кількість операцій.

Для подальшого підвищення точності можна застосувати спосіб адитивної корекції результатів обчислень на кожній ітерації, запропонований у роботі [6].

Тоді першу ітерацію слід провести за модифікованими формулами Ньютона–Рафсона

$$y_{1b} = y_0(2 + k_1 - xy_0) \qquad (27)$$

або у розгорнутому вигляді :

$$y_0 = y_{01};$$
$$y_{1b} = -2x - 1/2xk_1 - 1/4x^3 + x^2 + 1/2x^2t + 2 + k_1 - xt + t + 1/2tk_1 - 1/4xt^2, \; x \in [1, x_t) \;,$$

де k_1 – стале зміщення, яке потрібно визначити.

Для такого y_{1b} отримаємо відносну похибку

$$d_{1b} = (y_{1b} - y_t)/y_t,$$

явний вираз якої буде наступним:

$$d_{1b} = (-2x - 1/2xk_1 - 1/4x^3 + x^2 + 1/2x^2t + 2 + k_1 - xt + t + 1/2tk_1 - 1/4xt^2 - 1/x)x; \qquad (28)$$

Знайдемо точки максимуму відносної похибки з такого рівняння:

$$\frac{dd_{1b}}{dx} = -4x - xk_1 - x^3 + 3x^2 + 3/2x^2t - 2xt - 1/2xt^2 + 2 + k_1 + t + 1/2tk_1 = 0.$$

Його розв'язок дає три точки максимуму відносної похибки:

$$x_{11b\_\max} = 1/2t + 1;$$
$$x_{12b\_\max} = 1 + 1/2t - 1/2 \cdot (-4 + 4t + t^2 - 4k_1)^{1/2};$$
$$x_{13b\_\max} = 1 + 1/2t + 1/2 \cdot (-4 + 4t + t^2 - 4k_1)^{1/2}.$$

Створимо систему з двох рівнянь:

1) $x = x_{11b\_\max}$; $d_{11b} = (-2x - 1/2xk_1 - 1/4x^3 + x^2 + 1/2x^2t + 2 + k_1 - xt + t + 1/2tk_1 - 1/4xt^2 - 1/x)x$;

2) $x = x_{12\_zero}$; $d_{13b} = (-2x - 1/2xk_1 - 1/4x^3 + x^2 + 1/2x^2t + 2 + k_1 - xt + t + 1/2tk_1 - 1/4xt^2 - 1/x)x$.

Похибки при цих значеннях мають бути рівними за модулем, але протилежними за знаками:

$$d_{11b} - d_{13b} = 0 .$$

Розв'язком цього рівняння є значення

$$k_1 = 0.00130869 . \qquad (29)$$

Графік відносної похибки d_1 при цьому буде таким:

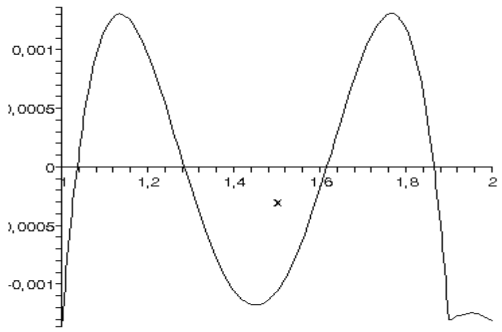


Рис. 3. Графік відносної похибки d_{1b} для $x \in [1,2)$

Однак слід підкреслити, що це теоретичне значення k_1 не враховує похибок заокруглень та відсікання, тому на практиці може спостерігатись відхилення у наймолодшому двійковому розряді мантиси при поданні уточненого коефіцієнта $(2 + k_1)$ у вигляді числа типу float.

Аналогічно проведемо другу ітерацію за модифікованою формулою:

$$y_{2b} = y_{1b}(2 + k_2 - xy_{1b}) , \qquad (30)$$

де k_2=0.00000085.

Остаточний вигляд програми буде таким:

```
float reciprocal (float x)
{
int i = *(int*)&x;
i = 0x7ef311c3 -i;
float y = *(float*)&i;
y = y*( 2.00130856 f -x*y);
y = y*( 2.00000084 f - x*y);
return y;
}
```

Графіки похибок d_2 та d_{2b} для теоретичних викладок та цієї програми наведено на рис. 4, 5.

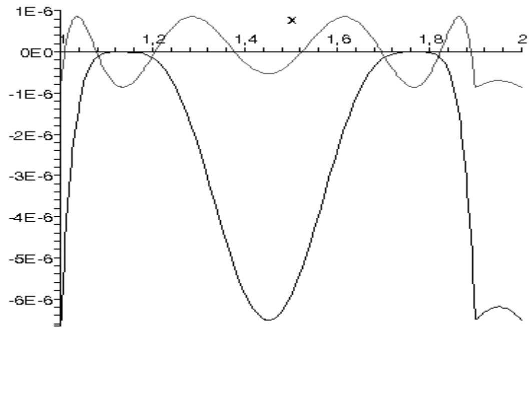


Рис.4. Графіки теоретично розрахованих відносних похибок d_2 та d_{2b} (червона крива $-d_2$, зелена – d_{2b}) для $x \in [1,2)$

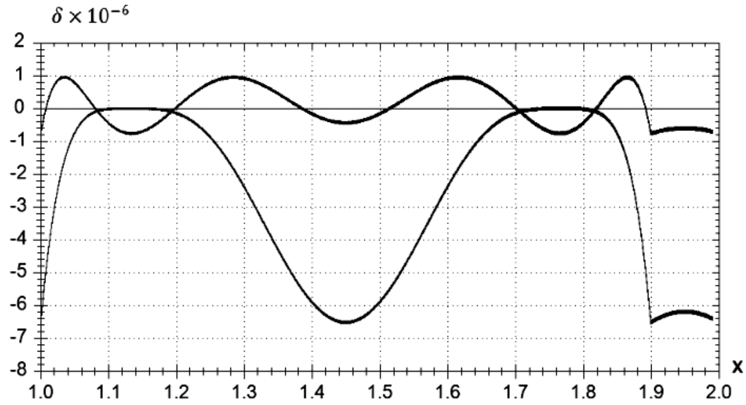


Рис.5. Графіки реальних відносних похибок d_2 та d_{2b} (червона крива $-d_2$, синя $-$ d_{2b}) для $x \in [1, 2)$

Модифікація формул Ньютона–Рафсона дає змогу підвищити точність обчислень ще приблизно у 6.5 разу (19.9 коректних бітів результатів обчислень): $d_{2\max} = -6.51 \cdot 10^{-6}$; $d_{2b\max} = 1.01 \cdot 10^{-6}$; $v \approx 6.5$.

Висновки

Наведено теоретичне обґрунтування оптимального вибору магічної константи для забезпечення мінімальних значень відносних похибок для початкового наближення, першої та другої ітерацій за формулою Ньютона–Рафсона при обчисленні функції $1/x$ для чисел типу float. Також визначено оптимальні значення зміщень для адитивної корекції формул Ньютона–Рафсона з метою зменшення відносних похибок обчислень приблизно у 6,5 разу.

1. C. Minos Niu, Sirish K. Nandyala, Won Joon Sohn, Terence D. Sanger. "Multi-scale hyper-time hardware emulation of human motor nervous system based on spiking neurons using FPGA." Advances in Neural Information Processing Systems. 2012. 2. Eric Papenhausen and Klaus Mueller. "Rapid Rabbit: Highly Optimized GPU Accelerated Cone-Beam CT Reconstruction". IEEE Medical Imaging Conference, Seoul, Korea, November 2013. 3. Jim Blinn. "Floating-point tricks". IEEE Computer Graphics and Applications,17 (1997), no. 4.Page(s): 80 – 84. 4.Chris Lomont, Fast Inverse Square Root, 2003. http://www.lomont.org/Math/Papers/2003/InvSqrt.pdf. 5. Мороз Л., Гринчишин А. Швидке обчислення оберненого квадратного кореня з використанням магічної константи – аналітичний підхід. // "Комп'ютерні технології друкарства". – 2014. – № 32. – С. 38–51. 6. Мороз Л. В. Теорія та швидкодіючі апаратно-програмні засоби ітераційних методів обчислення функцій. – Автореф. дис. … д-ра техн. наук за спеціальністю 05.13.05 – комп'ютерні системи і компоненти. – Львів: Національний університет "Львівська політехніка", 2013.