АЛГОРИТМИ: системи, аналіз, оптимізація, моделі і моделювання, верифіувція, алгоритмічні мови, програмування, системи та прикладне програмне забезпечення

УДК 519.7
© Л. Мороз[1], А. Гринчишин[1], 2014

ШВИДКЕ ОБЧИСЛЕННЯ ОБЕРНЕНОГО КВАДРАТНОГО КОРЕНЯ З ВИКОРИСТАННЯМ МАГІЧНОЇ КОНСТАНТИ - АНАЛІТИЧНИЙ ПІДХІД

Подано математичний опис перетворень при швидкому обчисленні оберненого квадратного кореня з використанням магічної константи для чисел з плаваючою точкою та визначення оптимальних значень магічних констант для забезпечення мінімальних абсолютних і відносних похибок обчислень для початкового наближення першої та другої ітерацій за формулою Ньютона-Рафсона.

Mathematical description of the transformations is given at the fast calculation of inverse square root with the use of magic constants for floating point numbers and determination of optimum values of magic constants for providing of minimum absolute and relative errors for the initial values, first and second Newton-Raphson's iterations.

1. ВСТУП

У даній роботі вперше викладено достатньо простий та одночасно строгий математичний опис роботи алгоритму швидкого обчислення оберненого квадратного кореня з використанням магічної константи для чисел з плаваючою точкою (типу float) у форматі одинарної точності (single precision) стандарту IEEE-754. Подано

2. МЕТА РОБОТИ

Метою роботи є подання строгого математичного опису перетворень при обчисленні оберненого квадратного кореня з використанням магічної константи для чисел з плаваючою точкою

[1] **Н**аціональний університет "Львівська політехніка"

(типу float) та визначення оптимальних значень магічних констант для забезпечення мінімальних похибок обчислень (як абсолютних, так і відносних) для початкового наближення, для першої та другої ітерацій за формулою Ньютона-Рафсона.

3. ОПИС АЛГОРИТМУ

Алгоритм швидкого обчислення оберненого квадратного кореня подано нижче (інша назва - функція швидкого InvSqrt):

```
float InvSqrt(float x)
{
float halfnumber = 0.5*x;
int i = *(int)*&x;          { перевід числа x з floating-point y integer }
i = 0x5f3759df – (i»1);{початкове наближення для inverse square root, де 0x5f3759df – магічна константа R }
x = *(float*)&i ; { перевід числа i з integer y floating-point }
x = x*(1.5-halfnumber*x * x );{перша Ньютонівська ітерація}
x = x*(1.5-halfnumber*x * x );{друга Ньютонівська ітерація }
return x ;
}
```

Цей алгоритм знайшов широке застосування насамперед у комп'ютерній 3D графіці при нормалізації векторів. Інші використання алгоритму можна знайти також у [3,4,7,8,9], причому основна привабливість такого підходу полягає у високій швидкості обчислення оберненого квадратного кореня (приблизно у 3-4 рази вищій, ніж з використанням бібліотечних функцій). Огляд історії виникнення цієї функції можна знайти, наприклад, у [10]. Існує декілька робіт теоретичного плану [2,5,6,10,11] , де здійснені спроби аналітичного дослідження найкращих (у певному сенсі) магічних констант. Однак назвати вдалими ці спроби не можна, оскільки вони не дають строгого математичного опису процесів перетворень, що відбуваються на різних кроках алгоритму.

4. ОСНОВНІ РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

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

1.Задається додатнє число $x > 0$ типу float.

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

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

де S_x - знак (у даному випадку $S_x=0$, оскільки x - завжди додатнє число);

E_x - порядок, який визначається за формулою :

$$E_x=\lfloor \log_2(x) \rfloor = floor[\log_2(x)]; \quad (2)$$

M_x - мантиса, яка обраховується за формулою :

$$M_x=\frac{x}{2^{E_x}}, \quad (3)$$

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

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

$$x = (-1)^{S_x} 1.f \cdot 2^{e_x} = (-1)^{S_x} \cdot (1+m_x) \cdot 2^{e_x}, \quad (4)$$

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

Однак у двійковому представленні мантиса має так званий фантомний біт, який насправді не показується, тому вираз $1.f$ зображується лише у вигляді $0.f$. Тоді число, яке зберігається у 32-бітному регістрі, має вигляд :

біт 31	30 ← біти → 23	22 ← біти → 0
s_x	e_x	$0.f \cdot N_m = m_x \cdot N_m$

де $N_m = 2^{23}$. Звідси ціле число I_x, яке відповідає двійковому представленню числа x у стандарті IEEE-754, зображується як

$$I_x=e_x \cdot N_m+0.f \cdot N_m=(e_x+m_x)N_m, \quad (5)$$
$$m_x=0.f.$$

Наприклад, якщо $x=16$; $N_m=2^{23}$, то $E_{x=16}=4$; $S_{x=16}=0$; $e_{x=16}=131$; $M_{x=16}=1+m_{x=16}=1$; $m_{x=16}=0.f=0$; $I_{x=16}=(131+0)\cdot 2^{23}=1098907648$.

2. Тепер переходимо до зображення магічної константи R ($0x5f3759df$ - її шістнадцятковий запис)

0	Q	T

- двійкове представлення магічної константи у 32-бітному регістрі, де $S=0$, $Q=190$, $T=3627487$ [10]. Звідси магічну константу можна зобразити у вигляді цілого числа як

$$I_R=Q\cdot N_m+T=1597463007. \tag{6}$$

Для магічної константи Ломонта [2] $0x5f375a86$ значення такі: $S=0$, $Q=190$, $T=3627654$. Усі відомі з робіт [1,2,6,7,10] магічні константи відрізняються лише значенням T.

3. Наступна операція - це зсув цілого представлення I_x числа x на один розряд вправо (тобто, цілочисельне ділення на 2 з відсіканням дробової частини частки) - це шокуюча операція для програмістів, зокрема, тому що тут відбувається, наприклад, заміна старшого біту мантиси на молодший біт порядку:

$$I_{x/2}=\left\lfloor \frac{I_x}{2} \right\rfloor = floor\left[\frac{I_x}{2} \right] \tag{7}$$

4. Далі шукається цілочисельна різниця d

$$d=I_R-I_{x/2} \tag{8}$$

5. Після цього d переводиться у дійсне число типу float, яке і буде початковим наближенням y_0 функції оберненого квадратного кореня, тобто $y_0\approx\frac{1}{\sqrt{x}}$. Алгоритм переводу такий:

- знаходиться зсунутий порядок

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

- знаходиться справжній (незсунутий) порядок

$$E_p = e_p - 127; \tag{10}$$

- 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:

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

Далі можна оцінити абсолютну

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

та відносну похибки

$$\delta_0 = \frac{\Delta_0}{y_t} \tag{14}$$

початкового наближення.

6. Після знаходження y_0 проводяться ітерації за формулою Ньютона-Рафсона для $y_t = \frac{1}{\sqrt{x}}$

$$y_1 = \frac{y_0}{2}(3 - x y_0^2) \tag{15}$$

$$y_2 = \frac{y_1}{2}(3 - x y_1^2) \tag{16}$$

або у загальному випадку

$$y_{n+1} = \frac{y_n}{2}(3 - x y_n^2) \tag{17}$$

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 перевід x в I_x, та в $d = I_R - I_{x/2}$ і зворотній перевід d в y_0 з допомогою рівнянь (1)-(12), то в результаті отримаємо, що початкове наближення y_0 у загальному випадку описується рівнянням:

$$y_0=(1+\frac{Q\cdot N_m+T-I_{x/2}-e_p\cdot N_m}{N_m})\cdot 2^{E_p}, \quad (18)$$

$$e_p=floor(\frac{Q\cdot N_m+T-I_{x/2}}{N_m}) \quad (19)$$

$$I_{x/2}=floor[\frac{N_m}{2}(bias+E_x+x\cdot 2^{E_x}-1)], \quad (20)$$

причому E_x та E_p визначаються за формулами(2) та (10) відповідно.

Вирази (18)-(20) є імітаційною моделлю формування початкового наближення y_0 для форматів single precision та double precision стандарту IEEE-754. На рис.1 наведено графік абсолютної та відносної похибок для початкового наближення y_0 на проміжку $x\in[0.5,2)$

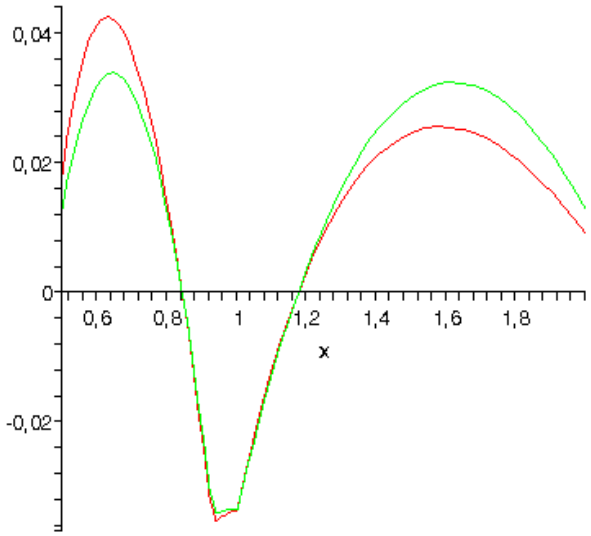

Рис.1 Графік абсолютної та відносної похибок початкового наближення y_0 (червона крива – Δ_0, зелена- δ_0)

Аналітичне дослідження поведінки y_0 з допомогою рівнянь (18)-(20) ускладнене через наявність тут декількох функцій типу $floor$.

Достатньо точне наближення y_0 можна отримати з допомогою виразу:

$$y_{00}=(\frac{3}{2}+Q+t-\frac{1}{2}bias-\frac{1}{2}E_x-x2^{-(E_x+1)}-e_p)\cdot 2^{E_p}, \quad (21)$$

причому $t=\frac{T}{N_m}$, якщо в (20) замінити

$$I_{x/2}=floor[\frac{N_m}{2}(bias+E_x+x\cdot 2^{E_x}-1)]$$

на відповідний йому вираз

$$I_{x/2}=\frac{N_m}{2}(bias+E_x+x\cdot 2^{E_x}-1),$$

тобто вилучити з (20) функцію $floor$, що стоїть перед квадратними дужками. Така заміна є прийнятною з точки зору обчислювальної точності, оскільки функцію $floor$ забрано лише у змінній $I_{x/2}$, що змінює її значення для будь-якого x у найгіршому випадку тільки на похибку відсікання. У форматі single precision ця відносна похибка складає лише 1 одиницю молодшого розряду, тобто $2^{-23}\approx 1.2\cdot 10^{-7}$. На рис.2 показано відносну похибку відхилення y_{00} від y_0 (тобто, $\frac{y_{00}-y_0}{y_0}$) на проміжку $[2^{-11},256)$.

Аналіз рівняння (21) показує, що наближення y_{00} можна подати у вигляді:

$$y_{00}=(\alpha x+\beta)\cdot 2^{E_p}, \quad (22)$$

де

$$\alpha=-2^{-(E_x+1)} \quad (23)$$

$$\beta=[\frac{1}{2}(3-bias)+Q+t-\frac{1}{2}E_x-e_p], \quad (24)$$

звідки випливає, що y_0 - це кусково-лінійне наближення функції $\frac{1}{\sqrt{x}}$.

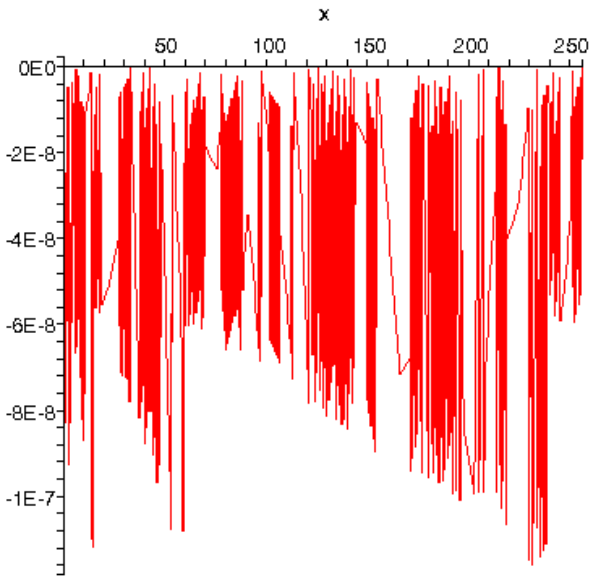

Рис. 2 Відносна похибка відхилення y_{00} *від* y_0 *на проміжку* $x \in [2^{-11}, 256)$

Тепер, маючи аналітичний вираз початкового наближення для будь-якого x, можна звузити діапазон значень аргумента, які підлягають дослідженню на максимуми похибок. Для цього застосуємо наступний прийом. При обчисленні оберненого квадратного кореня для чисел з плаваючою точкою використовується формат, подібний до IEEE-754:

$$x = A \cdot 2^{E_x},$$

де E_x - ціле парне число (обовязково парне!), у той час як число A - мантиса, значення якої лежать, як правило, у діапазонах $A \in [1,4)$, $A \in [0.25,1)$ або $A \in [0.5,2)$. Тоді

$$y_t = \frac{1}{\sqrt{x}} = \frac{2^{-E_x/2}}{\sqrt{A}} \qquad (25)$$

Тому достатньо проаналізувати поведінку похибки наближення лише на одному з вказаних проміжків значень x з межами, кратними 4, щоб описати закон поведінки похибки у всьому діапазоні зміни

аргумента x, заданого у вигляді чисел типу float. У даній роботі для аналізу обрано проміжок $[0.5,2)$.

Спочатку отримаємо прості аналітичні вирази для y_{00}, а потім перейдемо до аналізу рівняння абсолютної та відносної похибок на окремих ділянках проміжку $[0.5,2)$. Цей проміжок слід розбити на три сегменти:

$$x\in[0.5,x_t);$$
$$x\in[x_t,1);$$
$$x\in[1,2).$$

Тут $x_t=\frac{1}{2}+t$.

Спочатку розглянемо сегмент $x\in[0.5,x_t)$, Тут значення параметрів будуть такими: $E_x=-1$; $e_p=127$; $E_p=0$. Тоді

$$y_{01}=(-x+\frac{3}{2}+t)\cdot 2^0=-x+\frac{3}{2}+t \qquad (26)$$

Відповідно для сегмента $x\in[x_t,1)$ значення параметрів будуть такими: $E_x=-1$; $e_p=126$; $E_p=-1$. Тоді

$$y_{02}=(-x+\frac{5}{2}+t)\cdot 2^{-1}=-\frac{1}{2}x+\frac{5}{4}+\frac{t}{2} \qquad (27)$$

Для $x\in[1,2)$ $E_x=0$; $e_p=126$; $E_p=-1$; $Q=190$. Якщо підставити ці значення у рівняння (22-24), то вираз для y_{00} набере вигляду

$$y_{03}=(-\frac{1}{2}x+2+t)\cdot 2^{-1}=-\frac{1}{4}x+1+\frac{t}{2} \qquad (28)$$

Очевидно, що наближення y_{01}, y_{02}, y_{03}, записані у вигляді рівнянь (26-28), відрізнятимуться від y_0 на ті ж самі значення, що показані на рис.2.

Маючи аналітичні вирази початкового наближення y_{00}, перейдемо до вибору оптимального значення магічної константи R, яка забезпечить мінімальну похибку обчислення функції y_t (як

відносну, так і абсолютну) як для початкового наближення, так і після проведення першої та другої ітерацій за формулою Ньютона-Рафсона.

У ряді робіт є нерозуміння того, чому найкраще у сенсі мінімальної відносної похибки початкове наближення може давати гірший результат у сенсі мінімальної відносної похибки після проведення першої ітерації Ньютона-Рафсона [2,11]. Спробуємо розібратись у цьому питанні, використовуючи для цього знайдені лінійні наближення y_{00}.

Аналіз максимальних абсолютної $\Delta_{0\max}$ та відносної δ_0 похибки початкового наближення y_{00} показує, що вони виникають на першому сегменті $x \in [0.5, x_t)$. Знайдемо значення магічних констант $R_{0absjlute}$ та $R_{0relative}$, які забезпечать мінімальні значення Δ_0 та δ_0.

Запишемо абсолютну похибку через лінійне наближення y_{01} на цьому сегменті

$$\Delta_0 = -x + \frac{3}{2} + t - \frac{1}{\sqrt{x}}.$$

Максимальне значення похибки виникає при

$$\frac{d\Delta_0}{dx} = 0,$$

або

$$\frac{1}{2 \cdot x^{3/2}} - 1 = 0$$

Звідси знайдемо значення x, яке позначимо через $x_{\max\_a\_0}$, при якому виникає максимум абсолютної похибки початкового наближення: $x_{\max\_a\_0}$=0.629960524947.

Тепер сформуємо рівняння

$$E_1 - E_2 = 0,$$

де

$$E_1 = \Delta_0(x_{\max})$$
$$E_2 = \Delta_0(x_t),$$

або у розгорнутому вигляді

$$\frac{5}{2} - \frac{2}{\sqrt{2+4t}} - \frac{3}{2}2^{1/3} + t = 0$$

Звідси знайдемо відповідне значення t:

$$t_{a\_0} = -\frac{1}{2} + \frac{((108+12\sqrt{96-2162^{\frac{1}{3}}+1622^{\frac{2}{3}}})^{\frac{2}{3}} - 24 + 182^{\frac{1}{3}})^2}{36(108+12\sqrt{96-2162^{\frac{1}{3}}+1622^{\frac{2}{3}}})^{\frac{2}{3}}}$$

чи

$$t_{a\_0} = 0.427967981537;$$
$$T_{a\_0} = 3590055.633663,$$

або з врахуванням заокруглення до найближчого цілого

$$T_{a\_0} = 3590056.$$

Значення магічної константи $R_{0absolute}$ при цьому буде таким:

$$R_{0absolute} = 0x5F36C7A8. \quad (29)$$

Аналогічно можна знайти значення магічної константи $R_{0relative}$, яка забезпечить мінімально можливе значення максимальної відносної похибки δ_0:

$$\delta_0 = \frac{\Delta_0}{y_t} = -x^{3/2} + \frac{3}{2}\sqrt{x} + t\sqrt{x} - 1.$$

Значення x, при якому виникає максимум відносної похибки:

$$x_{\max\_r\_0} = \frac{1}{2}\cdot + \frac{1}{3}t$$

На підставі цього можна знайти відповідне значення констант:

$$t_{r\_0} = 0.432744889959,$$
$$T_{r\_0} = 3630127.2458729$$

або з врахуванням заокруглення до найближчого цілого

$$T_{r\_0} = 3630127.$$

Тоді

$$R_{0relative} = 0x5F37642F, \quad (30)$$

що збігається з результатами роботи Ломонта [2], отриманих шляхом повного перебору.

Однак, як уже зазначалось, значення $R_{0absolute}$ та $R_{0relative}$ не забезпечують мінімально можливі значення абсолютної та відносної похибок після проведення першої ітерації Ньютона-Рафсона.

Запишемо явні вирази поточної абсолютної Δ_1 та відносної δ_1 похибок на першому сегменті з врахуванням початкового наближення y_{01} після проведення першої ітерації:

$$\Delta_1=\frac{1}{16}\cdot(3+2t-2x)\cdot(-12+9x+12xt-12x^2+4xt^2-8x^2t+4x^3)-\frac{1}{\sqrt{x}};$$

$$\delta_1 = \Delta_1 \cdot \sqrt{x}\ .$$

Максимум абсолютної похибки тут виникає у двох точках, значення x, при якому виникає максимум абсолютної похибки Δ_1, складає:

$$x_{\max\_a\_11}=0.637104704644,$$

та у точці

$$x_{\max\_a\_12} = \frac{1}{2}\cdot + t_{a\_1}$$

Відповідно, $x_t = 0.930064414514$.

Максимум відносної похибки δ_1,теж виникає у двох точках :

$$x_{\max\_r\_11}=\frac{1}{2}\cdot+\frac{1}{3}t_{r\_1}=0.644150028263;$$

$$x_{\max\_r\_12} = \frac{1}{2}\cdot + t_{r\_1};$$

$$x_t = 0.932450056076.$$

Звідси можна знайти відповідні значення t_{a_1} та t_{r_1}

$$t_{a\_1}=0.430064414514,$$

якому відповідає значення

$$T_{a\_1} = 3607642,$$

та

$$t_{r\_1}=0.432450084790$$

з відповідним

$$T_{r\_1} = 3627654 .$$

Значення магічних констант при цьому буде таким:

$$R_{1absolute} = 0x5F370C5A \quad (31)$$

$$R_{1relative} = 0x5F375A86 \quad (32)$$

Останнє значення знову збігається з магічною константою Ломонта [2].

Очевидно, що абсолютні похибки, отримані з допомогою $R_{1absolute}$ на інших ділянках x будуть відрізнятися за значеннями, знайдених для проміжку $x \in [0.5,2)$, однак вони будуть мінімальними з поміж усіх інших абсолютних похибок для будь-яких інших значень магічних констант.

Аналогічним чином визначаються відповідні константи для другої ітерації Ньютона-Рафсона:

$$R_{2absolute} = 0x5F373366 \quad (33)$$

$$R_{2relative} = 0x5F375A86 \quad (34)$$

5.ВИСНОВКИ

У даній роботі наведено теоретичне обґрунтування оптимального вибору магічних констант для забезпечення мінімальних значень похибок обчислень (як абсолютних, так і відносних) для початкового наближення, для першої та другої ітерацій за формулою Ньютона-Рафсона.

1.id software, quake3-1. 32b/ code/ game/ q_ math. c , Quake III Arena, 1999. 2.C. Lomont, "Fast inverse square root," Purdue University, Tech. Rep.,2003. [Online].available: http://www.lomont.org/Math/Papers/2003/InvSqrt.pdf. 3.Zafar S., Adapa R. Hardware architecture design and mapping of 'Fast Inverse Square Root's algorithm' //Advances in Electrical Engineering (ICAEE), 2014 International Conference on. – IEEE, 2014. – pp. 1-4. 4.Jim Blinn, Floating-point tricks, IEEE Computer Graphics and Applications 17 (1997), no.4. 5.David Eberly, Fast inverse square root, Geometric Tools, LLC(2010), http://geometrictools.com/Documentation/FastInverseSqrt.pdf. 6.Charles McEniry: The Mathematics Behind the Fast Inverse Square Root Function Code. Tech. rep., 2007. 7.Q. Avril and V. Gouranton and B. Arnaldi Fast Collision Culling in Large-Scale Environments Using GPU Mapping Function "ACM Eurographics Parallel Graphics and Visualization, Cagliari : Italy (2012)".8.Edoardo Ardizzone, Roberto Gallea, Orazio Gambino and Roberto Pirrone Effective and Efficient Interpolation for Mutual Information based Multimodality Elastic Image Registration 2003. 9.Jerome L.V.M. Stanislaus and Tinoosh Mohsenin High Performance Compressive Sensing Reconstruction Hardware with QRD Process,2012. 10. Matthew Robertson A Brief History of InvSqrt 2012. 11.Ben Self. Efficiently Computing the Inverse Square Root Using Integer Operations. May 31, 2012.