Nils Laurent

1 Calcul rapide de 1 divisé par racine de x

On souhaite réaliser une implémentation rapide $f : x \mapsto \frac{1}{\sqrt{x}}$, où x est un flottant codé sur 32 bits avec la norme IEEE 754.

On rappelle que $x = (-1)^S 2^{(E_x - 127)}(1 + M_x 2^{-23})$, où E_x est un entier exprimé sur 8 bits, et M_x un entier exprimé sur 23 bits. Pour la suite, on pose les quantités suivantes :

— $e_x = E_x - 127$

— $m_x = M_x 2^{-23}$

Et l'on suppose que $S = 0$.

Ainsi, $x = 2^{e_x}(1 + m_x)$.

1.1 Étude du passage au log

L'idée clé est de considérer le logarithme en base 2 de f :

$$\log_2(\frac{1}{\sqrt{x}}) = \frac{-1}{2}\log_2(x) \tag{1}$$

Puis, on vérifie aisément que $\log_2(x) = e_x + \log_2(1 + m_x)$. Le nouveau problème est maintenant de trouver une expression précise de $\log_2(1 + m_x)$.

Exploitons d'abord l'égalité suivante, valable pour $h \in \mathbb{R}_+^*$,

$$\log_2(1 + h) = \frac{\ln(1 + h)}{\ln(2)} \tag{2}$$

Puis remarquons que $m_x \in [0, 1]$ et restreignons-nous à l'étude $\log_2(1 + h)$ où h varie dans $[0, 1]$.

Remarque : En réalité, m_x appartient à un sous-ensemble de $[0, 1]$, qui permet d'être exprimé par la mantisse d'un flottant.

Maintenant, remarquons aussi l'égalité suivante,

$$\ln(1 + h) = h + \int_1^{1+h} \frac{1 + h - s}{s^2} ds \tag{3}$$

Ainsi, une approximation grossière de $\log_2(1 + h)$ est donnée par h, et l'erreur est donnée par la fonction

$$g(h) = \log_2(1 + h) - h \tag{4}$$

On souhaite maintenant trouver une constante σ^* qui minimise

$$J(\sigma) = \sup_{h \in [0,1]} |g(h) - \sigma| \tag{5}$$

Il faut donc faire l'étude de la fonction g sur $[0, 1]$.

L'erreur est nulle aux bords, c'est-à-dire que $g(0) = g(1) = 0$. Il faut trouver où g' s'annule, calculons donc g' :

$$g'(h) = \frac{d}{dh}\log_2(1+h) - 1 = \frac{d}{dh}\frac{\ln(1+h)}{\ln(2)} - 1 = \frac{1}{\ln(2)}\frac{1}{1+h} - 1 \tag{6}$$

Puis, cherchons une solution à l'équation

$$g'(h) = 0 \iff \frac{1}{\ln(2)}\frac{1}{1+h} - 1 = 0 \tag{7}$$

Il existe une unique solution, que l'on nomme h_0, exprimée par

$$h_0 = \frac{1}{\ln(2)} - 1 \tag{8}$$

qui est bien dans l'intervalle $[0, 1]$. La valeur σ^* qui minimise $J(\sigma)$ est donc donnée par :

$$\sigma^* = \frac{1}{2}g(h_0) = \frac{1}{2}\left(\log_2(1+h_0) - h_0\right) = \frac{1}{2}\left(\log_2\left(\frac{1}{\ln(2)}\right) - \frac{1}{\ln(2)} + 1\right) = \frac{1}{2} - \frac{\ln(\ln(2)) + 1}{2\ln(2)} \tag{9}$$

En conclusion, nous proposons donc l'approximation suivante,

$$\log_2\left(\frac{1}{\sqrt{x}}\right) = \frac{-1}{2}(e_x + \log_2(1+m_x)) \approx \frac{-1}{2}(e_x + m_x + \sigma^*) \tag{10}$$

1.2 Étude de la représentation entière

Posons $y = \frac{1}{\sqrt{x}}$. À partir de

$$\log_2\left(\frac{1}{\sqrt{x}}\right) = \frac{-1}{2}\log_2(x), \tag{11}$$

on propose, grâce à nos travaux précédents, l'approximation suivante,

$$e_y + m_y + \sigma^* \approx \frac{-1}{2}(e_x + m_x + \sigma^*) \tag{12}$$

Rappelons que notre problème est d'approcher y. Rappelons aussi que pour cela, il suffit de trouver E_y et M_y, en effet,

$$y = 2^{E_y - 127}(1 + M_y \times 2^{-23}) \tag{13}$$

Ici, il y a une observation importante à faire. Appelons I_y la représentation binaire de y. Celle-ci s'écrit

$$I_y = 2^{23}E_y + M_y = 2^{23}(E_y + m_y) \tag{14}$$

Pour résoudre notre problème, il suffit donc de trouver cette représentation. Supposons que

$$e_y + m_y + \sigma^* = \frac{-1}{2}(e_x + m_x + \sigma^*) \tag{15}$$

Nous pouvons alors faire les calculs suivants

$$E_y - 127 + m_y + \sigma^* = \frac{-1}{2}(E_x - 127 + m_x + \sigma^*) \tag{16}$$

$$\Leftrightarrow E_y + m_y = \frac{3}{2}(127 - \sigma^*) - \frac{1}{2}(E_x + m_x) \tag{17}$$

$$\Leftrightarrow 2^{23}(E_y + m_y) = 2^{23}\frac{3}{2}(127 - \sigma^*) - \frac{1}{2}2^{23}(E_x + m_x) \tag{18}$$

$$\Leftrightarrow I_y = 2^{22}3(127 - \sigma^*) - \frac{1}{2}I_x \tag{19}$$

Nous pouvons finalement conclure que $I_y = C^* - \frac{1}{2}I_x$ où $C^* = 2^{22}3(127 - \sigma^*)$.

1.3 Écriture du code C

Nous allons ici nous intéresser à l'écriture de $I_y = C^* - \frac{1}{2} I_x$ où $C^* = 2^{22}3(127 - \sigma^*)$ dans le langage C.

Tout d'abord, il faut expliciter la valeur de C^*. Sa valeur explicite est donnée par :

$$C^* = 2^{22}3\left(127 - \frac{1}{2} + \frac{\ln(\ln(2)) + 1}{2\ln(2)}\right) \tag{20}$$

Grâce à un calculateur, on obtient la valeur hexadécimale 0x5f37bcb6. Il en résulte ainsi la fonction suivante,

```
#include <stdint.h> // uint32_t

float fast_inverse_sqrt(float x)
{
    union
    {
        float f;
        uint32_t i;
    }
    u;

    u.f = x;
    u.i = 0x5f37bcb6 - (u.i >> 1);

    return u.f;
}
```

2 Sources

J. F. Blinn, "Floating-point tricks," in IEEE Computer Graphics and Applications, vol. 17, no. 4, pp. 80-84, July-Aug. 1997, doi : 10.1109/38.595279.