Algorithme de CORDIC

1 Introduction

L'algorithme CORDIC (*COordinate Rotation DIgital Computing*), inventé en 1959 pour calculer les valeurs des fonctions trigonométriques ou logarithme, a permis aux premiers calculateurs de poche (HP 35 en 1972) d'être rapides malgré la taille réduite de leur mémoire. C'est en effet un algorithme économe en calculs. Nous allons le décrire pour le calcul du logarithme népérien d'un nombre décimal x compris entre 1 et 10.

2 L'algorithme

- On prend habituellement pour une calculatrice $N = 10$. On constate que le programme utilise la fonction logarithme de certains nombres $\ln(10)$ et $\ln(1 + 10^{-i})$ pour i de 0 à N soit : $\ln 2$, $\ln(1,1)$, $\ln(1,01)$, $\ln(1,001)$, ...
- Les $N + 1$ valeurs de la fonction logarithme sont calculé d'une autre façon et sont stockées dans la base de données de la calculette.
- Lorsque l'on prend $N = 10$ la précision de la valeur approchée par excès Y de $\ln X$ est de 10^{-9}. Les valeurs stockées sont des approximation à 10^{-12}.

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:CORDIC
:Prompt N,X
:ln(10)→Y
:For(I,0,N)
:1+10^-I→Z
:While XZ≤10
:XZ→X
:Y-ln(Z)→Y
:End
:End
:Disp Y
```

Variables : N, I entiers, Y, Z réels
$X \in [1;10]$ réel
Entrées et initialisation
Lire N, X
$\ln 10 \to Y$
Traitement
pour I variant de 0 à N **faire**
$1 + 10^{-I} \to Z$
tant que $XZ \leqslant 10$ **faire**
$XZ \to X$
$Y - \ln(Z) \to Y$
fin
fin
Sorties : Afficher Y

On peut remarquer que les seuls produits à effectuer sont des produits par $(1 + 10^{-i})$ ce qui revient à faire un décalage de virgule puis une addition, qui sont deux opérations très simples. Les seules autres opérations à effectuer sont des soustractions pour trouver Y. Ceci explique que cet algorithme est particulièrement efficace et rapide.

Avec $N = 10$, les 11 valeurs stockées à 10^{-12} sont données dans le tableau suivant.

Z	ln Z
10	$a = 2{,}302\,585\,092\,994$
2	$a_0 = 0{,}693\,147\,180\,560$
1,1	$a_1 = 0{,}095\,310\,179\,804$
1,01	$a_2 = 0{,}009\,950\,330\,853$
1,001	$a_3 = 0{,}000\,999\,500\,333$
1,000 1	$a_4 = 0{,}000\,099\,995\,000$
1,000 01	$a_5 = 0{,}000\,009\,999\,950$
1,000 001	$a_6 = 0{,}000\,000\,999\,999$
1,000 000 1	$a_7 = 0{,}000\,000\,100\,000$
1,000 000 01	$a_8 = 0{,}000\,000\,010\,000$
1,000 000 001	$a_9 = 0{,}000\,000\,001\,000$
1,000 000 000 1	$a_{10} = 0{,}000\,000\,000\,100$

On teste le programme :

X	4,5	5,6
Y	1,504 077 396	1,722 766 597

Remarque : On peut remarquer que si l'on tape directement sur la calculatrice $\ln 4,5$ et $\ln 5,6$, on obtient exactement les mêmes résultats au dernier chiffre près.

3 Explication

Soit x un réel vérifiant $1 \leqslant x \leqslant 10 \quad \Leftrightarrow \quad \dfrac{1}{10} \leqslant \dfrac{1}{x} \leqslant 1 \quad \Leftrightarrow \quad 1 \leqslant \dfrac{10}{x} \leqslant 10$

1) Comme $2^0 = 1$ et $2^4 = 16 \geqslant 10$,

   il existe un entier α_0 avec $0 \leqslant \alpha_0 \leqslant 3$ tel que : $2^{\alpha_0} \leqslant \dfrac{10}{x} < 2^{\alpha_0+1}$

2) On divise l'encadrement par 2^{α_0} : $1 \leqslant \dfrac{10}{2^{\alpha_0}x} \leqslant 2$

   Comme $1,1^0 = 1$ et $1,1^8 \approx 2,14 \geqslant 2$,

   il existe un entier α_1 avec $0 \leqslant \alpha_1 \leqslant 8$ tel que : $1,1^{\alpha_1} \leqslant \dfrac{10}{2^{\alpha_0}x} < 1,1^{\alpha_1+1}$

   On en déduit alors : $2^{\alpha_0} \times 1,1^{\alpha_1} \leqslant \dfrac{10}{x} < 2^{\alpha_0} \times 1,1^{\alpha_1+1}$

3) On peut montrer par récurrence que, pour $\forall n \in \mathbb{N}$, il existe des entiers α_n avec $0 \leqslant \alpha_n < 10$ tels que :

$$2^{\alpha_0} \times 1,1^{\alpha_1} \times \cdots \times (1+10^{-n})^{\alpha_n} \leqslant \frac{10}{x} < 2^{\alpha_0} \times 1,1^{\alpha_1} \times \cdots \times (1+10^{-n})^{\alpha_n+1}$$

   Remarque :
   - Cette récurrence ne pose pas de problème particulier, cependant l'écriture étant un peu fastidieuse, nous ne la reproduisons pas ici.
   - On montre avec le binôme de Newton que : $(1+10^{-n})^{10} > 1 + 10^{-(n-1)}$

4) On pose : $y = 2^{\alpha_0} \times 1,1^{\alpha_1} \times 1,01^{\alpha_2} \times \cdots \times (1+10^{-n})^{\alpha_n}$

   En utilisant les propriétés fondamentales de la fonction logarithme :

   $\ln y = \alpha_0 \ln(2) + \alpha_1 \ln(1,1) + \cdots + \alpha_n \ln(1+10^{-n})$

   De la relation de récurrence, on a : $y \leqslant \dfrac{10}{x} < y(1+10^{-n})$

   Comme la fonction ln est croissante sur $\mathbb{R}_+^*$, on a :

$$\ln y \leqslant \ln 10 - \ln x < \ln y + \ln(1+10^{-n})$$

   De la relation $\ln(1+x) < x$, on a :

$$\ln y \leqslant \ln 10 - \ln x < \ln y + 10^{-n}$$
$$\ln y - \ln 10 \leqslant -\ln x < \ln y - \ln 10 + 10^{-n}$$
$$\ln 10 - \ln y - 10^{-n} < \ln x \leqslant \ln 10 - \ln y$$

   $\ln 10 - \ln y = \ln(10) - [\alpha_0 \ln 2 + \alpha_1 \ln 1,1 + \cdots + \alpha_n \ln(1+10^{-n})]$ est donc une valeur approchée de $\ln x$ à 10^{-n}

5) On prend maintenant $n = 10$. Les valeurs approchées a et a_i du tableau étant données avec une précision de 10^{-12}.

   On a : $\ln 10 - \ln y \approx a - [\alpha_0 a_0 + \alpha_1 a_1 + \cdots + \alpha_{10} a_{10}]$

   Comme les coefficients a_i sont donnés à une précision de 10^{-12} et que les coefficients α_i sont inférieurs à 10, alors les termes $\alpha_i a_i$ sont donnés avec une précision inférieure à $10 \times 10^{-12} = 10^{-11}$. Comme il y a 11 termes $\alpha_i a_i$ et le terme a, soit 12 termes, en additionnant les précisions, on obtient $12 \times 10^{-11} \approx 10^{-10}$.

   Comme $\ln 10 - \ln y$ donne une valeur approchée de $\ln x$ à $10^{-n} = 10^{-10}$,
   $a - [\alpha_0 a_0 + \alpha_1 a_1 + \cdots + \alpha_{10} a_{10}]$ donne la valeur de $\ln x$ à 2×10^{-10}

6) Dans l'algorithme, on prend les coefficient a_i les uns après les autres dans la boucle « **Pour** I variant de 0 à N **faire** ».
   On soustrait autant que possible a_i à Y dans la boucle « **Tant que** $XZ \leqslant 10$ **faire** ». Soit α_i fois.

4 Généralisation pour un nombre quelconque de la calculatrice

1) Soit x un nombre réel de la calculatrice, on a alors $10^{-100} < x < 10^{100}$.

   On encadre x entre deux puissance de 10 consécutives, négatives si $x < 1$ et positives si $x > 10$. Pour n compris entre -100 et 100, on a alors :

$$10^n \leqslant x < 10^{n+1}, \text{ en multipliant par } 10^{-n}, \text{ on a :}$$
$$1 \leqslant 10^{-n}x < 10$$

   Comme $1 \leqslant 10^{-n}x < 10$, on est revenu au calcul précédent, c'est à dire :

$$\begin{aligned}
\ln(10^{-n}x) &\approx a - [\alpha_0 a_0 + \alpha_1 a_1 + \cdots + \alpha_{10} a_{10}] \\
-n \ln 10 + \ln x &\approx a - [\alpha_0 a_0 + \alpha_1 a_1 + \cdots + \alpha_{10} a_{10}] \\
\ln x &\approx n \ln 10 + a - [\alpha_0 a_0 + \alpha_1 a_1 + \cdots + \alpha_{10} a_{10}] \\
\ln x &\approx na + a - [\alpha_0 a_0 + \alpha_1 a_1 + \cdots + \alpha_{10} a_{10}]
\end{aligned}$$

   - Comme a est donné à 10^{-12} près, $|na| < 100a$ est donné à 10^{-10} près.
   - Comme le calcul précédent était une approximation à 2×10^{-10} près
   - $\ln x$ est donc donné à 3×10^{-10} près.

2) Pour écrire un nouveau programme qui donne une approximation de $\ln x$ d'un réel quelconque de la calculatrice, on distinguera deux cas : $x < 1$ et $x \geqslant 10$.

   - Si $X > 10$, on divise par 10 tant que $X > 10$ en augmentant M de 1 à chaque boucle afin d'avoir $10^M \leqslant X < 10^{M+1}$.
   - Si $X < 1$, on multiplie par 10 tant que $X < 1$ en soustrayant M de 1 à chaque boucle afin d'avoir $10^M \leqslant X < 10^{M+1}$.

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:CORDIC2
:Prompt N,X
:ln(10)→Y
:0→M
:While X>10
:X/10→X
:M+1→M
:End
:While X<1
:10X→X
:M-1→M
:End
:For(I,0,N)
:1+10^-I→Z
:While XZ≤10
:XZ→X
:Y-ln(Z)→Y
:End
:End
```

Variables : N, I entiers, Y, Z réels
$X \in [10^{-100}\ ;\ 10^{100}]$ réel
Entrées et initialisation
- Lire N, X
- $\ln 10 \to Y$
- $0 \to M$
- **tant que** $X > 10$ **faire**
  - $\dfrac{X}{10} \to X$
  - $M + 1 \to M$
- **fin**
- **tant que** $X < 1$ **faire**
  - $10X \to X$
  - $M - 1 \to M$
- **fin**

Traitement
- **pour** I variant de 0 à N **faire**
  - $1 + 10^{-I} \to Z$
  - **tant que** $XZ \leqslant 10$ **faire**
    - $XZ \to X$
    - $Y - \ln(Z) \to Y$
  - **fin**
- **fin**
- **Sorties :** Afficher $M \ln 10 + Y$

On teste le programme :

X	4 567	0,001 467
Y	8,426 611 813	-6,524 535 78

Remarque : On peut remarquer que si l'on tape directement sur la calculatrice $\ln 4\,567$ et $\ln 0{,}001\,467$, on obtient exactement les mêmes résultats.