Détermination avec Maxima de la fraction continue associée à un nombre réel.

Introduction

On se propose d'obtenir à l'aide de Maxima, les N+1 premiers termes de la suite (a_n) d'entiers qui définit la fraction continue associée à un nombre réel x .

La suite (x_n) des successeurs de x est définie par $x_0 = x$ et la relation $x_{n+1} = 1/(x_n – a_n)$ où a_n est la partie entière de x_n .

Ce programme vient en complément du programme "cf" de Maxima qui s'applique aux nombres rationnels et aux racines carrées de nombres entiers.

Il utilisera la virgule flottante pour obtenir la partie entière de nombres irrationnels.

Il n'est pas possible de prévoir d'avance la précision nécessaire pour effectuer la totalité des calculs. La précision nécessaire pour déterminer le terme a_n , sera estimée à partir des calculs effectués pour déterminer le terme a_{n-1} .

Précision d'amorçage

On a besoin d'une valeur approchée valide de x pour effectuer les calculs. En fait x est l'expression d'un nombre réel à l'aide des fonctions élémentaires.
Pour une précision P , bfloat(x) donne une valeur approchée décimale s_P de x.

Dans le meilleur des cas, pour tout $P \geq 1$, on a $|x – s_P| < 10^{E–(P–1)}$ où E est l'entier qui vérifie $10^E \leq |x| < 10^{E+1}$ pour $x \# 0$. On pose $E = –1$ si x est une expression de 0 .
Si $x \# 0$, s_P est constitué des P premiers chiffres du développement décimal de $|x| \times 10^{-E}$ le dernier pouvant être augmenté d'une unité.

En général, nous aurons à estimer le plus petit entier L non nul telle que $|x – s_P| < 10^{E-(P-L)}$ pour tout $P \geq L$. L est appelé précision d'amorçage de bfloat(x).
Les chiffres valides de la valeur approchée de x seront les P+1–L premiers chiffres de bfloat(x).

Indice de régularité

Pour une estimation rapide de L nous utiliserons la notion de régularité de la virgule flottante sur un intervalle de précision.

Cette notion est basée sur l'idée intuitive que, si la précision augmente d'une unité, on doit voir apparaître un chiffre supplémentaire du développement décimal de x .

Pour chaque expression de x, il existe un plus petit entier T tel que, si la virgule flottante est régulière sur un intervalle de précision [A,B] contenant au moins T éléments, alors $A \geq L$.
T est appelé indice de régularité de bfloat(x) .

Contrôle des résultats obtenus à l'aide d'un "test de convergence"

On utilise un test qui repose sur la rapidité de convergence des réduites vers x.

Application

Une deuxième partie sera consacrée à la détermination de "la meilleure fraction égale à x à ε près".

PLAN DE L'ETUDE

I- Programmes élémentaires

Programme dans le cas d'un quotient de deux entiers

La réduite associée au dernier terme calculé est donnée par B/D.

Programme CFR(x,n)

```
CFR(x,n):=(u:P:num(x), v:Q:denom(x),  A:0, B:1, C:1, D:0, U:[],
            for i:0 while i <= n and v # 0 do
                  (a:entier(u/v), U:endcons(a,U), B:A+(A:B)*a, D:C+(C:D)*a, u:-v, v:B*Q-D*P),
            U) $
```

a_n est le terme de rang n de la fraction continue de x = P/Q.

Les relations de base issues des propositions A-2 et A-3 de l'annexe, sont:

$A_n = B_{n-1}$, $B_n = A_{n-1} + B_{n-1}\, a_n$, a_n est la partie entière de $x_n = \dfrac{(C_{n-1}\, x - A_{n-1})}{(B_{n-1} - D_{n-1}\, x)}$

$C_n = D_{n-1}$, $D_n = C_{n-1} + D_{n-1}\, a_n$, $C_n\, x - A_n = -(B_{n-1} - D_{n-1}\, x)$, B_n/D_n est la réduite de rang n.

Les relations B:A+(A:B)*a et D:C+(C:D)*a déterminent les paramètres A, B, C et D de rang i à l'aide de a_i = a et des paramètres A, B, C et D de rang i-1 . x_i est représenté par u/v .

On évite les divisions qui ralentissent le programme.
Si v = 0 , x = B/D. On a obtenu le dernier terme de la fraction continue de x. Les calculs s'arrêtent.

```
(%i2) (CFR(sum(1/i!,i,0,20),50),disp("q"=length(U),"B/D"=B/D),U);
```

q = 38

B/D = $\dfrac{66133133192480800 01}{24329020081766400 00}$

```
(%o2) [2,1,2,1,1,4,1,1,6,1,1,8,1,1,10,1,1,12,1,1,14,1,1,17,8,1,6,5,35,2,2,2,1,6,1,3,25,3]
```

q est le nombre de termes calculés. Il est inférieur au nombre de termes demandés.
37 est le rang du dernier terme de la fraction continue de x . x = B/D .

Programme qui utilise bfloat(x)

La précision affichée doit être suffisante pour effectuer la totalité des calculs.
Si rien n'est précisé, c'est la précision par défaut de Maxima qui est égale à 16.

Programme CFb(x,n)

```
CFb(x,n):=(Z:fpprec, t:bfloat(x), u:t, v:1, A:0, B:1, C:1, D:0, U:[],
            for i:0 while i <= n and v # 0.0b0 do (a:entier(u/v), B:A+(A:B)*a, D:C+(C:D)*a,
                                                   U:endcons(a,U), u:-v, v:B-D*t), fpprec:Z,
            U) $
```

```
(%i3) fpprec:16$ CFb(sqrt(2),30);
(%o3) [1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1]   (précision trop faible)

(%i5)  fpprec:100$ CFb(sqrt(2),30);
(%o5) [1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2]
```

II- Résultats utiles à l'élaboration des programmes autonomes

Pour un entier n donné, il est impossible de prévoir à l'avance la précision à utiliser pour obtenir la fraction continue jusqu'au rang n. Le programme autonome détermine la précision nécessaire au calcul du terme de rang i à partir des calculs effectués au rang i–1.

Par ailleurs, on doit évaluer la précision à partir de laquelle on obtient des valeurs approchées valides de x.

Pour déterminer la partie entière de x_i , il faut s'assurer que les décimales valides de la valeur approchée de x_i ne comporte pas que des 0 ou que des 9 auxquels cas la partie entière serait définie à une unité près.

Nous donnons les prérequis nécessaires. Ces prérequis sont développés en annexe.

A- Conditions d'obtention d'une partie entière à l'aide d'une valeur approchée

La partie entière de x est notée E(x) . Le développement décimal de x est noté DD de x.

Proposition C-4

Soient un nombre réel x , un entier $m \geq 0$ et $c_m(x) = E(10^m x) - 10^m E(x)$.

(1) L'équation : $x = E(x) + c\ 10^{-m} + b\ 10^{-m}$ admet une solution unique (c,b) telle que c soit un nombre entier et b un nombre réel vérifiant $0 \leq b < 1$: $c = c_m(x)$ et $b = 10^m x - E(10^m x)$.

(2) En base 10, $c_m(x)$ s'écrit : $a_1a_2\ldots a_m$ où a_1 , a_2 , … , a_m sont les m premières décimales du développement décimal de x .

(3) $0 \leq c_m(x) \leq 10^m - 1$.

(4) Si x est entier, $c_m(x) = 0$ pour tout entier $m \geq 0$.

(5) Si x n'est pas entier, il existe un entier p tel que, pour tout $m \geq p$, on ait $1 \leq c_m(x) \leq 10^m - 2$.

La relation $1 \leq c_m(x) \leq 10^m - 2$ exprime que les m premières décimales du DD de x ne sont ni toutes nulles ni toutes égales à 9 .

Exemple : $x = 5{,}947695234$, $c_6(x) = E(10^6 x) - 10^6 E(x) = 5947695 - 5000000 = 947695$

Nous serons amenés à utiliser la condition :

$3 \leq c_m(y) \leq 10^m - 3$ C(m)

Proposition C-5

Soit x un nombre entier.
Alors pour tout entier $m > 0$ et tout y vérifiant $|x - y| < 10^{-m}$ on a:

soit $c_m(y) = 0$ et $x = E(y)$

soit $c_m(y) = 10^m - 1$ et $x = E(y) + 1$

Proposition C-6

Soit x un nombre différent d'un entier.

(1) Soient un entier $m > 0$ et y un nombre réel tel que: $|x - y| < 10^{-m}$ et $1 \le c_m(y) \le 10^m - 2$.
Alors : $E(x) = E(y)$.
Si, de plus, $3 \le c_m(y) \le 10^m - 3$, pour tout z vérifiant $|x - z| < 10^{-m}$, on a $E(z) = E(x)$ et $c_m(z) \ge 1$.

(2) Il existe un entier $p > 0$ tel que :
pour tout entier $m \ge p$ et tout y vérifiant : $|x - y| < 10^{-m}$, on ait : $3 \le c_m(y) \le 10^m - 3$.

B- Précision d'amorçage et régularité de la virgule flottante

Précision d'amorçage

Soient x un nombre réel et E le nombre entier qui vérifie $10^E \le |x| < 10^{E+1}$ si $x \# 0$.
$E = -1$ si $x = 0$.
Pour une précision P, on pose t_P = bfloat(x) et $s_P = \sigma \times \mathbf{m} \times 10^{\mathbf{e}+1-\mathbf{P}}$ où σ est le signe, **m** la mantisse, **e** l'exposant de bfloat(x). s_P est un nombre décimal. s_P = round(t_P*10^(P–e–1))*10^(e+1–P) .

Sous certaines conditions, il existe un entier $L \ge 1$ tel que :
$|x - s_P| < 10^{E-(P-L)}$ pour tout $P \ge L$. Si $P \ge L$ on a $E - 1 \le e \le E + 1$.
Pour une valeur suffisante de P, e = E.

Définition

La précision d'amorçage de bfloat(x) est le plus petit entier L tel que $|x - s_P| < 10^{E-(P-L)}$
pour toute précision $P \ge L$.

Régularité de la virgule flottante

Si $A \ge L$, on vérifie : $|x - s_n| < 10^{E-(n-A)}$ pour tout $n \in [A,B]$.

Régularité de bfloat(x) sur un intervalle de précision [A,B]

On ne suppose plus $A \ge L$.
On dit que bfloat(x) est régulier sur l'intervalle [A,B] si: $|x - s_n| < 10^{E-(n-A)}$ pour tout $n \in [A,B]$.

Indice de régularité de bfloat(x)

On considère l'ensemble des intervalles réguliers [A,B] tel que $A < L$.
Le nombre de ces intervalles est fini (proposition F-2). Soit N le nombre d'éléments de celui qui a le plus grand nombre d'éléments. L'indice de régularité de bfloat(x) est T = N+1 .

Proposition F-3

Si la virgule flottante est régulière sur un intervalle [A,B] contenant au moins T éléments,
alors $A \ge L$.

P-régularité

Quand x est irrationnel, x n'est accessible que par les valeurs de bfloat(x) calculées à différentes précisions on va remplacer x par bfloat(x) calculé pour une précision P suffisante et supérieure à B.

On dit que bfloat(x) est P-régulier sur un intervalle [A,B] si $|s_P - s_n| < 10^{E-(n-A)}$ pour tout $n \in [A,B]$.

Avec ce critère et une estimation e de E à une unité près, on obtient une estimation L_0 de L qui vérifie $L_0 \le L \le L_0+2$ et $L_0+e+1 \ge L+E$ (proposition F-7).

Les programmes utiliseront $|t_P - t_n|$ au lieu de $|s_P - s_n|$ et la précision d'amorçage pourrait être sousestimée d'une unité (proposition F-8) . D'où $L_0+e+2 \ge L+E$

C- Recherche de la précision V_n nécessaire au calcul de a_n

Soient x un nombre réel, x_n est le successeur de rang n de x , b_n est la réduite de rang n de x , $H_n(t) = (C_{n-1}\, t - A_{n-1})/(B_{n-1} - D_{n-1}\, t)$, $x_n = H_n(x)$.

Le principe est d'estimer V_n à partir des éléments qui sont issus du calcul a_{n-1} .
La proposition D-4 donne les éléments nécessaires à l'élaboration du programme.

Proposition D-4

Soit H_n la fonction associée à la fraction continue de x.
On peut trouver deux suites (q_n) et (r_n) de nombres réels, des suites (m_n) , (K_n) , (V_n) d'entiers positifs et une suite (O_n) d'intervalles ouverts contenant x qui vérifient:

(1) $r_n = H_n(q_n)$.

(2) $(D_{n-1}\, 10^{m_{n-1}})^2 \le 10^{K_n}$ pour n > 0 .

(3) $| x_n - r_n | < 10^{-m_n}$ et $3 \le c_{m_n}(r_n) \le 10^{m_n} - 3$.

(4) $K_0 \ge 0$, $V_0 \ge K_0 + m_0$ et $V_n \ge \max(K_n + m_n , V_{n-1})$ pour n > 0.

(5) O_n est l'ensemble des nombres q tels que $| x - q | < 10^{-V_n}$. $O_n \subseteq O_{n-1}$ pour n > 0.

(6) $q_n \in O_n$ et pour tout $q \in O_n$, la fraction continue de q coïncident avec celle de x au moins jusqu'au rang n .

D- Choix de la précision de la virgule flottante nécessaire au calcul de r_n

On choisit $K_0 = \max(0,-E)$.

Soit n > 0 et g_{n-1} l'entier qui vérifie $10^{g_{n-1}-1} < D_{n-1} \le 10^{g_{n-1}}$. On choisit $K_n = 2\,(g_{n-1} + m_{n-1})$.

Pour une précision P, on a $|x - s_P| < 10^{E-(P-L)}$.

$|x_n - r_n| < |x - s_P|\, 10^{K_n}$ est la relation qui permet d'estimer la précision nécessaire au calcul de a_n pour n > 0 (proposition D-3).

La condition $|x_n - r_n| < 10^{-m_{n-1}}$ est satisfaite si $|x - s_P| < 10^{-V_n}$.

Pour obtenir $10^{E-(P-L)} \le 10^{-V_n}$, il suffit de choisir : $P_n \ge L+E+K_n+ m_n$ et $P_n \ge P_{n-1}$.

Ce choix de P_n est aussi valable pour n = 0 .

m_n est le premier entier ≥ 4 déterminé par le programme tel que $3 \le c_{m_n}(r_n) \le 10^{m_n} - 3$.

Le programme va utiliser les paramètres Q et W_n définis par $Q = L+ E$ et $W_n = Q+K_n+m_n$.

D'où $P_n \ge W_n$ et $P_n \ge P_{n-1}$.

On vérifie $P_n > L$, pour tout n : $P_n \ge P_0 \ge L+E+K_0 + m_0 > L$.

III- Programmes autonomes

Le domaine de validité des programmes qui suivent peut être élargi en augmentant la valeur de la précision initiale.

A- Programme dans le cas irrationnel

Le programme CFI(x,n), donne la fraction continue d'un nombre irrationnel . b(x) = bfloat(x) .
Les calculs sont effectués en virgule flottante. Pour une précision P, on utilise directement
t_P = bfloat(x) au lieu de s_P en évaluant les pertes éventuelles de précision par arrondi.

Estimation de la précsion d'amorçage à l'aide du test de régularité

Programme EI(z)

```
EI(z):=(d:1, if z<1 then (for k:1 while d*z<1 do (d:10*d, e:-k))
                     else (for k:0 while d<=z do (d:10*d, e:k)))$
```

Le programme EI(z) détermine l'entier e qui vérifie $10^e \le z < 10^{e+1}$ où z = |t| .

Si z < 1 , le programme cherche le premier entier k qui vérifie $10^k \times z \ge 1$. Alors e = -k .
Si z ≥ 1 , le programme cherche le premier entier k qui vérifie $10^{k+1} > z$. Alors e = k .

Programme L(x)

```
L(x):=(for i:0 while i<=1 and y=1 do  (F:10^(e-i*Z), fpprec:L+i*Z, f:t-b(x),
                                        if  -F<f  and  f<F then 1  else y:0)) $
```

L(x) est le test de régularité. $s_P - s_n$ est représenté par t – b(x) calculé pour la précision n .

La condition -F<f and f<F est utilisée à la place de abs(f) < F , ce qui permet d'invalider les valeurs complexes de b(x) .

Pour vérifier la régularité de bfloat(x) sur l'intervalle [L,L+Z] on se contente de vérifier l'inégalité
$|t_P - t_n| < 10^{e-(n-L)}$ pour les valeurs n = L et n = L+Z où Z = fpprec .

En augmentant la précision initiale, on augmente la longueur de l'intervalle sur lequel est vérifiée la régularité. Ainsi, la sensibilité du test est augmentée et le domaine de validité du programme élargi.

Programme ELI(x)

```
ELI(x):=(y:0,for i while y=0 do (y:1,L:L+Z,fpprec:P:L+Z+100,t:b(x),
                                  if t<0 or t>0 then (z:abs(t), EI(z) else y:0,if y=1 then L(x) else 1),
          fpprec:P) $
```

La condition t<0 or t>0 invalide les valeurs nulles ou complexes de t .
Le programme ELI(x) coordonne les deux programmes précédents. Si t est invalidé ou en cas d'échec du test de régularité, on remplace L par L + Z , on recalcule e et on refait le test.
L'opération se renouvelle aussi longtemps que le test échoue. La valeur initiale de L est Z .

Détermination des termes de la fraction continue

L+e+2 est une estimation par le programme de la valeur exacte de L+E.
On pose Q = L+e+2 . Au rang i, on pose W = Q + K + m où K est calculé au rang i-1 pour i > 0.
K=max(0,-e) pour i = 0.

Au lieu de calculer s_p pour chaque valeur de n, on utilise une précision forfaitaire P qui permet de garder la même valeur de s_P tant que W ≤ P . Si W > P , on remplace P par W+100 .

Programme ai(x)

```
ai(x):=(o:-o, c:0, for j while c<3 or c>s-3 do
              (m:m+m, s:s*s, W:Q+K+m,
               if W>P then (fpprec:P:W+100, t:b(x), u:C*t-A) else 1,
               if y=1 then (v:B-D*t, if o*v >0 then (r:u/v, a:entier(r), c:entier(s*r)-s*a) else (c:3,y:0))
                      else 1))$
```

Le terme a_i est calculé en effectuant le test de la condition C(m) pour satisfaire aux conditions de la proposition C-6 (1). s est égal à 10^m . Le nombre c qui représente $E(r\ 10^m) – 10^m\ E(r)$, doit être supérieur ou égal à 3 et inférieur ou égal à $10^m – 3$ (représenté par s-3) .

Initialement égale à 4 la valeur de m est doublée à chaque échec du test de la condition C(m).
Une valeur convenable de m sera obtenue après un nombre fini d'opérations (propositions C-6 (2)).

t est utilisé à la place de s_P . Le calcul est effectué en virgule flottante.

Pour L ≥ 16, la valeur de P est suffisante pour absorber les pertes de précision dues au calcul en virgule flottante (proposition E-2).

o représente $(-1)^i$. L'inégalité o*(v:B-D*t) > 0 traduit la condition $(-1)^{i-1}(t- b_{i-1}) > 0$ qui exprime que t appartient au domaine de définition de la fonction H_i (proposition A-3) .

Programme g(D)

```
g(D):=(for k:g+1 while D>d do (d:10*d, g:k)) $
```

g est la première valeur de k telle que $D \le 10^k$ (proposition D-4) .

Initialement, on pose g = 0 , K = max(0,-e) et d = 1.

La démarche du programme AI(x) est analogue à celle des programmes élémentaires. Après chaque application de ai(x) , g(D) détermine la nouvelle valeur de g nécessaire au calcul du terme suivant.
K = 2 (g+m) où g = g(D) .
La condition a>0 pour i > 0 traduit la propriété $a_n \ge 1$ pour n ≥ 1 des fractions continues.

Programme AI(x)

```
AI(x):=(o:-1, Q:L+e+2, g:0, K:max(0,-e), d:1, A:0, B:1, C:1, D:0, u:t, U:[],
        for i:0 while y=1 and i<=n do (m:2, s:100, ai(x), if i=0 or (y=1 and a>0)
                                                    then (B:A+(A:B)*a, D:C+(C:D)*a,  g(D),
                                                          K:2*(g+m), u:-v, U:endcons(a,U))
                                                    else y:0)) $
```

Test de convergence

Le test est basé sur la rapidité de convergence des réduites vers x quand n tend vers l'infini.
Il est nécessaire de calculer d'abord le terme de rang n+1 de la fraction continue en appliquant ai(x).

Soit t = bfloat(x) calculé pour une précision V = P+4 Z, où P est la dernière précision affichée.
Ce qui permet d'augmenter la sensibilité du test en augmentant la précision initiale.
D'après la proposition D-4 la fraction continue de s_V coïncide avec celle de x jusqu'au rang n+1.
Le test de convergence est obtenu en utilisant la proposition A-5 .

$|\sigma - b_n| = \dfrac{1}{(q_{n-1}+q_n\sigma_{n+1})q_n}$ où $\sigma = s_V$ et $b_n = p_n/q_n$. On a $|x_{n+1} - r_{n+1}| < 10^{Q+K-P} \le 10^{-m_{n+1}}$.

De plus, $|\sigma_{n+1} - x_{n+1}| < 10^K |\sigma - x| < 10^K\, 10^{e+1-(V-L-1)} = 10^{Q+K-P-4z}$.

Or $|\sigma_{n+1} - r_{n+1}| \le |\sigma_{n+1} - x_{n+1}| + |x_{n+1} - r_{n+1}|$. D'où $|\sigma_{n+1} - r_{n+1}| < 10^{Q+K-P}\,(10^{-4z} + 1) < 2\,10^{Q+K-P}$.

D'où : $1/(q_{n-1} + q_n (r_{n+1} + h)) < |q_n s_V - p_n| < 1/(q_{n+1} + q_n (r_{n+1} - h))$ où $h = 2\,10^{Q+K-P}$.

Quand n tend vers l'infini, les bornes de l'encadrement tendent vers 0 et la précision utilisée pour le calcul de b(x), tend vers l'infini.

Si l'un des termes de la fraction continue de x a été mal calculé, la suite (B_n/D_n) tendra vers x' # x (proposition A-10). La suite $|b(x) - B_n/D_n|$ tendra vers x' – x # 0 et il existera un rang n à partir duquel l'une au moins des inégalités ne sera pas vérifiée.

Les résultats sont d'autant plus fiables que le nombre de termes calculés est élevé .

Programme TI(x)

```
TI(x):=(m:10, s:10^10, ai(x), if y=1 and a>0 then (F:C+D*r, f:2*D*10^(Q+K-P), fpprec:V:P+4*Z,
                                                    d:o*(B-D*b(x)),
                                                    if  1 < (F+f)*d  and  (F-f)*d <1 then 1  else  y:0)
                                                else y:0) $
```

B/D représente b_n , b(x) représente s_V , a représente a_{n+1} ,
F représente $q_{n-1} + q_n r_{n+1}$ et f représsente 2 D 10^{Q+K-P} . La valeur minimale de m est 20.

La double inégalité est équivalente à: $1 < (F+f)|D\,b(x) - B|$ et $(F-f)|D\,b(x) - B| < 1$.
o*(B-D*b(x)) représente $|D\,b(x) - B|$.

Programme CFI(x,n)

```
CFI(x,n):= (Z:fpprec, b(x):=bfloat(x),
            L:0, y:0, for i while y=0 do (ELI(x), AI(x) , if y=1 then TI(x) else 1), fpprec:Z, U) $
```

```
(%i2) CFI(10^10*log(1+10^-10),17);
(%o2) [0,1,20000000000,3,10000000000,5,6666666666,1,4,4,555555555,2,1,8,2,1,444444443,1]

(%i3) CFI(%pi,30);
(%o3) [3,7,15,1,292,1,1,1,2,1,3,1,14,2,1,1,2,2,2,2,1,84,2,1,1,15,3,13,1,4,2]

(%i4)  CFI(sin(exp(-10))-exp(-10)+ exp(-30)/6,20);
(%o4) [0,62216466346098148020976 0,19,5,5,2,2,4,4,3,2,6,1,35,1,6,28,3,2,2,6]
```

Programme récapitulatif

```
CFI(x,n):=(Z:fpprec, b(x):=bfloat(x),

EI(z):=(d:1, if z<1 then (for k:1 while d*z<1 do (d:10*d, e:-k))
                    else (for k:0 while d<=z do (d:10*d, e:k))),

L(x):=(for i:0 while i<=1 and y=1 do  (F:10^(e-i*Z), fpprec:L+i*Z, f:t-b(x),
                                        if  -F<f  and  f<F then 1  else y:0)),

ELI(x):=(y:0,for i while y=0 do (y:1,L:L+Z,fpprec:P:L+Z+100,t:b(x),
                                  if t<0 or t>0 then (z:abs(t), EI(z)) else y:0,if y=1 then L(x) else 1),
          fpprec:P) ,

g(D):=(for k:g+1 while D>d do (d:10*d, g:k)),

ai(x):=(o:-o, c:0, for j while c<3 or c>s-3 do
            (m:m+m, s:s*s, W:Q+K+m,
             if W>P then (fpprec:P:W+100, t:b(x), u:C*t-A) else 1,
             if y=1 then (v:B-D*t, if o*v >0 then (r:u/v, a:entier(r), c:entier(s*r)-s*a) else (c:3,y:0))
                    else 1)),

AI(x):=(o:-1, Q:L+e+2, g:0, K:max(0,-e), d:1, A:0, B:1, C:1, D:0, u:t, U:[],
        for i:0 while y=1 and i<=n do (m:2, s:100, ai(x), if i=0 or (y=1 and a>0)
                                                    then (B:A+(A:B)*a, D:C+(C:D)*a,  g(D),
                                                          K:2*(g+m), u:-v, U:endcons(a,U))
                                                    else y:0))  ,

TI(x):=(m:10, s:10^10, ai(x), if y=1 and a>0 then (F:C+D*r, f:2*D*10^(Q+K-P), fpprec:V:P+4*Z,
                                                    d:o*(B-D*b(x)),
                                                     if  1 < (F+f)*d  and  (F-f)*d <1 then 1  else  y:0)
                                                else y:0) ,

L:0, y:0, for i while y=0 do (ELI(x), AI(x) , if y=1 then TI(x) else 1), fpprec:Z, U) $
```

B- Programme général

Le programme CF(x,n) donne la fraction continue de la partie réelle d'un nombre complexe.
Pour pouvoir travailler dans le domaine réel ou le domaine complexe, on choisit b(x) comme suit:

```
if domain=complex then b(x):=bfloat(realpart(x)) else  b(x):=realpart(bfloat(x)),
```

Programme E(z)

```
E(z):=( if z>0 then (d:1,if z<1 then (for k:1 while d*z<1 do (d:10*d, e:-k),
                                        if e<Y then (Y:e-1, h:h-1,y:1-h) else 1)
                                  else (for k:0 while d<=z do (d:10*d,e:k)))
                else h:0) $
```

Les modifications apportées au programme EI(z) permettent de prendre en compte le cas où x est une expression de 0 (par exemple x = log(27) – 3 log(3)) .

Si z est nul, on pose j = 0 .

Si t est une valeur approchée non nulle de 0 . La valeur estimée de e sera très inférieure à -1 et le test de régularité qui doit fonctionner avec e = -1 , échouera.
On impose une valeur minimale Y pour la valeur de e.
Initialement Y = -50 . Au départ des calculs on pose h = 2.

Si, on obtient e < Y, on remplace Y par e-1, on remplace h par h-1 et on effectue le test de régularité. Si le test échoue, on recalcule la valeur de e. Si à nouveau e < Y on pose e = -1 et on effectue le test de régularité.

On réinitialise la valeur de h chaque fois que le test de régularité échoue avec h = 0.

Le test de régularité reste inchangé.

Programme EL(x)

```
EL(x):=(y:0, h:2, for i while y<=0 do (y:1, L:L+Z, fpprec:P:L+Z+100, t:b(x), z:abs(t), E(z),
                                        if h=0 then e:-1 else 1, L(x),
                                        if y=1 then 1 elseif h=0 then h:2 else 1),
          fpprec:P) $
```

Calcul de a_i

Quand x est rationnel le nombre de termes calculés est fini.
Si n+1 est supérieur ou égal au rang p du dernier terme de la fraction continue de x, le programme ai(x) aura à déterminer la partie entière de x_p qui est entier.
La condition C(m) ne sera jamais vérifiée et le test de la condition C(m) va s'exécuter indéfiniment.

Pour échapper à cette situation, il faudra plafonner le nombre de boucles dans ce test.
Le nombre de boucles du test de la condition C(m) sera limitée à 2^J . A chaque échec du test de convergence, J sera doublé. La valeur initiale de J est fixée à 2.

Dans certains cas, une précision insuffisante du test de convergence ne pourra pas invalider un résultat erroné. Il faudra élargir le domaine de validité du programme en augmentant la précision initiale, ce qui aura pour conséquence d'augmenter la précision du test de convergence.

Programme a(x)

```
a(x):=(o:-o, c:0,
       for j while (c<3 or c>s-3) and j<=J do
           (m:m+m, s:s*s, W:Q+K+m ,
            if W>P then (fpprec:P:W+100, t:b(x), u:C*t-A) else 1,
            if y=1 then (v:B-D*t, if o*v >0 then (r:u/v, a:entier(r), c:entier(s*r)-s*a) else (c:3,y:0))
                   else 1),
       if y>0 then (if c=0 then y:2 elseif c=s-1 then (y:2, a:a+1) else 1) else 1) $
```

Cas où x_i est entier

D'après la proposition C-5 si la condition c= 0 est vérifiée avec j = J, alors a_i = a.
Si la condition c= s-1 est vérifiée avec j = J, alors a_i = a+1 .

Pour arrêter les calculs on pose y=2.

Programme A(x)

```
A(x):=(o:-1, Q:L+e+2, g:0, K:max(0,-e), d:1, A:0, B:1, C:1, D:0, u:t, U:[],
        for i:0 while y=1 and i<=n do (m:2, s:100, a(x), if i=0 or (y=1 and a>0) or (y=2 and a>1)
                                                    then (B:A+(A:B)*a, D:C+(C:D)*a, g(D),
                                                          K:2*(g+m), u:-v, U:endcons(a,U))
                                                 else y:0)) $
```

Test de convergence

Programme T(x)

```
T(x):=(if y=1 then (m:10, s:10^10, a(x), if (y=1 and a>0) or (y=2 and a>1)
                          then (fpprec:V:P+4*Z, F:C+D*r, f:2*D*10^(Q+K-P), fpprec:V:P+4*Z,
                                d:o*(B-D*b(x)),  if 1 < (F+f)*d  and  (F-f)*d <1 then 1  else  y:0)
                          else y:0)
        else (fpprec:V:P+4*Z, if 10^(V-Q)*abs(D*b(x)-B) < D then 1 else y:0)) $
```

Si y=1, on calcule le terme de terme de rang n+1 et on utilise les éléments de TI(x) .

Si y=2, soit p le rang du dernier terme calculé . x_p est entier. D'après la proposition A-5, $x = B_p / D_p$.
b(x) est calculé pour une précision V = P+4 Z. On pose $H=10^{V-Q}$.

L'égalité est traduite par l'inégalité $|s_V - x| < 1/H$.

Quand V tend vers l'infini, 1/H tend vers 0 et s_V tend vers x . Si $b_p \neq x$ il existera une valeur de V à partir de laquelle l'inégalité ne sera pas vérifiée.

x = B/D . La condition à vérifier est H*abs(D*b(x)-B) < D .

Programme CF(x,n)

```
CF(x,n):=
(Z:fpprec, if domain=complex then b(x):=bfloat(realpart(x)) else  b(x):=realpart(bfloat(x)),
J:1, L:0, Y:-50, y:0, for i while y=0 do (J:J+J, EL(x),  A(x), if y>0 then T(x) else 1), fpprec:Z, U) $
```

```
(%i2)  CF(log(3/2),20);
(%o2) [0,2,2,6,1,11,2,1,2,2,1,4,3,1,1,7,2,1,1,4,1]

(%i3) CF(sum((-1)^(i+1)*2^-i/i,i,1,50),20);
(%o3) [0,2,2,6,1,11,2,1,2,2,1,4,3,1,1,7,2,1,1,4,1]

(%i4) CF(48915654/985389+log(8)-3*log(2),20);
(%o4) [49,1,1,1,3,1,1,1,9,11,1,6,3,3]
```

Nécessité d'augmenter la précision initiale

```
(%i6) fpprec:16$CF(1+sin(exp(-1000)),10);
(%o6) [1]
```

```
(%i8) fpprec:70$ CF(1+sin(exp(-1000)),10);
(%o8)  [1,19700711140170469938887935224[375 digits]9597058441895090500470742175 68,4,
         2,2,3,1,1,1,1,11]
```

Résultat direct avec CFI(x,n).

Programme récapitulatif

```
CF(x,n):=(Z:fpprec,   if domain=complex then  b(x):=bfloat(realpart(x)) else  b(x):=realpart(bfloat(x)),
E(z):=( if z>0 then (d:1,if z<1 then (for k:1 while d*z<1 do (d:10*d, e:-k),
                                        if e<Y then (Y:e-1, h:h-1,y:1-h) else 1)
                                    else (for k:0 while d<=z do (d:10*d,e:k)))
                 else h:0),

L(x):=(for i:0 while i<=1 and y=1 do  (F:10^(e-i*Z), fpprec:L+i*Z, f:t-b(x),
                                                                  if  -F<f  and  f<F then 1  else y:0)),

EL(x):=(y:0, h:2, for i while y<=0 do (y:1, L:L+Z, fpprec:P:L+Z+100, t:b(x), z:abs(t), E(z),
                                        if h=0 then e:-1 else 1, L(x),
                                        if y=1 then 1 elseif h=0 then h:2 else 1),
          fpprec:P) ,

g(D):=(for k:g+1 while D>d do (d:10*d, g:k)),

a(x):=(o:-o, c:0,
       for j while (c<3 or c>s-3) and j<=J do
           (m:m+m, s:s*s, W:Q+K+m ,
            if W>P  then (fpprec:P:W+100, t:b(x), u:C*t-A) else 1,
            if y=1 then (v:B-D*t, if o*v >0 then (r:u/v, a:entier(r), c:entier(s*r)-s*a) else (c:3,y:0))
                   else 1),
       if y>0 then (if c=0 then y:2 elseif c=s-1 then (y:2, a:a+1) else 1) else 1),

A(x):=(o:-1, Q:L+e+2, g:0, K:max(0,-e), d:1, A:0, B:1, C:1, D:0, u:t, U:[],
        for i:0 while y=1 and i<=n do (m:2, s:100, a(x), if i=0 or (y=1 and a>0) or (y=2 and a>1)
                                                          then ( B:A+(A:B)*a, D:C+(C:D)*a, g(D),
                                                                 K:2*(g+m), u:-v, U:endcons(a,U))
                                                     else y:0)),

T(x):=(if y=1 then (m:10, s:10^10, a(x), if (y=1 and a>0) or (y=2 and a>1)
                                then (fpprec:V:P+4*Z, F:C+D*r, f:2*D*10^(Q+K-P), fpprec:V:P+4*Z,
                                      d:o*(B-D*b(x)),   if 1 < (F+f)*d  and  (F-f)*d <1 then 1  else  y:0)
                                else y:0)
             else (fpprec:V:P+4*Z, if 10^(V-Q)*abs(D*b(x)-B) < D then 1 else y:0)),

J:1, L:0, Y:-50,  y:0, for i while y=0 do (J:J+J, EL(x), A(x), if y>0 then T(x) else 1), fpprec:Z, U) $
```

Cas des nombres complexes

On obtient les fractions continues des parties réelle et imaginaire de x avec:

```
CFc(x,n):=(disp("CFr" = CF(x,n) , "CFi" = CF(-%i*x,n))) $
```

Quelque soit le domaine, on obtient:

```
(%i10) CFc(exp(2)*exp(%i*%pi/12),20);
CFr=[7,7,3,1,1,14,1,2,9,7,2,4,1,2,1,1,1,20,2,6,1]
CFi=[1,1,10,2,2,1,1,2,8,6,1,2,3,1,59,2,1,70,1,61,1]
(%o10) done
```

Cas où l'expression de x comporte plusieurs déterminations

Si x n'a pas de détermination réelle, CFc(x) choisit la détermination principale quel que soit le domaine. Le domaine réel est le domaine par défaut de Maxima.

Si x a une détermination réelle, CFc(x) choisit la détermination réelle dans le domaine réel , CFc(x) choisit la détermination principale dans le domaine complexe , car la commande realpart(x) appelle la détermination principale de x.

```
(%i11) x:tan(2)$
```

```
(%i12) CFc(log(x),20);
CFr=[0,1,3,1,1,2,1,1,1,4,1,2,8,1,14,1,12,1,5,1,2]
CFi=[3,7,15,1,292,1,1,1,2,1,3,1,14,2,1,1,2,2,2,2,1]      (fraction continue de π)
(%o12) done
```

```
(%i13) CFc(sqrt(x),20);
CFr=[0]
CFi=[1,2,10,1,25,24,1,1,2,2,12,1,3,18,1,33,1,3,13,1,5]   (fraction continue de √|tan(2)| )
(%o13) done
```

$\tan(2)^{1/3}$ posséde trois déterminations qui sont les solutions de l'équation $z^3 = \tan(2)$.
L'une est réelle, elle est égale à $-\sqrt[3]{|\tan(2)|}$ (car $\tan(2) < 0$) .

```
(%i14) CFc(x^(1/3),20);
CFr=[-2,1,2,2,1,3,1,1,7,2,7,41,3,2,1,3,1,13,5,1,6]
CFi=[0]
(%o14) done
```

Dans ce domaine les autres déterminations sont accessibles avec CFc($e^{2i\pi/3}\ x^{1/3}$,n) et CFc($e^{4i\pi/3}\ x^{1/3}$,n) (détermination principale de $x^{1/3}$).

```
(%i15) domain:complex$
```

```
(%i16) CFc(x^(1/3),20);
CFr=[0,1,1,1,5,1,1,3,1,3,4,3,1,1,20,6,1,2,1,1,27]
CFi=[1,8,12,1,2,1,103,1,3,1,2,6,42,1,1,2,1,42,1,7,2]
(%o16) done
```

Dans ce domaine les autres déterminations sont accessibles avec CFc($e^{2i\pi/3}\ x^{1/3}$,n) (détermination réelle de $x^{1/3}$) et CFc($e^{4i\pi/3}\ x^{1/3}$,n) .

C- Extension du domaine de validité de la virgule flottante

Programme BFLOAT(x)

```
BFLOAT(x):=(Z:fpprec,
                if domain=complex then  b(x):=bfloat(realpart(x)) else  b(x):=realpart(bfloat(x)),
Fv(x):=(

 E(z):=( if z>0 then (d:1,if z<1 then (for k:1 while d*z<1 do (d:10*d, e:-k),
                                        if e<Y then (Y:e-1, h:h-1,y:1-h) else 1)
                                  else (for k:0 while d<=z do (d:10*d,e:k)))
            else h:0),

 L(x):=(for i:0 while i<=1 and y=1 do  (F:10^(e-i*Z), fpprec:L+i*Z, f:t-b(x),
                                                    if  -F<f  and  f<F then 1  else y:0)),

EL(x):=(y:0, h:2, for i while y<=0 do (y:1, L:L+Z, fpprec:P:L+Z+100, t:b(x), z:abs(t), E(z),
                                        if h=0 then e:-1 else 1, L(x),
                                        if y=1 then 1 elseif h=0 then h:2 else 1),
        fpprec:P) ,

  L:0, Y:-50, EL(x), fpprec:Z, t:bfloat(t)),

Fv(x), t0:t , Fv(-%i*x), t0+t*%i) $
```

Soit u la partie réelle ou imaginaire de x.

Pour une précision Z , BFLOAT(u) donne une valeur approchée décimale de u, notée σ_Z , telle que : $|u - \sigma_Z| < k\ 10^{E-(Z-1)}$ avec $k < 1+10^{-101}$.

$(|u - \sigma_Z| \le |u - s_P| + |s_P - \sigma_Z| < 10^{E-(P-L)} + 10^{E-(Z-1)} \le 10^{E-(Z-1)}\ [1+10^{-101}])$

```
(%i2) (fpprec:30, BFLOAT(sin(exp(-10))-exp(-10)+exp(-30)/6));
(%o2) 1.607291539891053514899475341 61b-24

(%i3) ( domain:complex, x:tan(2), fpprec:30, BFLOAT(x^(1/3)));
(%o3) 1.123786351408711291861843038 53b0*%i+6.488183524974468393305963944 72b-1

(%i4)  (domain:real,x:tan(2), fpprec:30, BFLOAT(x^(1/3)));
(%o4) -1.297636704994893678661192788 94b0

(%i5)  (fpprec:30,BFLOAT(log(x)));
(%o5)  3.141592653589793238462643383 28b0*%i+7.816340724367478139927096307 37b-1

(%i6)  (fpprec:30,BFLOAT(sqrt(x)));
(%o6)  1.478188033797297045229029913 27b0*%i
```

D- Programme de détermination de la précision d'amorçage de la virgule flottante

Pour un nombre irrationnel, le programme CFL(x,S) détermine :
- l'entier E qui vérifie $10^E \le |x| < 10^{E+1}$.
- la précision d'amorçage L de bfloat(x).
- l'indice de régularité T .

S+1 est une estimation par excès de l'indice de régularité.
On peut augmenter la précision initiale pour se rapprocher des conditions d'application de la proposition F-4
On utilise les programmes EI(z), L(x) et les suivants :
ELI(x) est remplacé par ELL(x) (voir programme récapitulatif).

Programme EE(x)

```
EE(x):=(Q:L+S+max(e+1,0), m:2, s:100, c:0, for j while  c=0 or c=s-1 do
                (m:m+m, s:s*s, if (W:Q+m)>P then (fpprec:P:W+100, t:b(x), z:abs(t), EI(z)) else 1,
                  r:z*10^-e, a:entier(r), c:entier(s*r)-s*a), sp:round(t*10^(P-e-1))*10^(e+1-P), E:e)  $
```

Le programme EE(x) détermine le premier chiffre non nul "a" du DD de |x| , ce qui permet d'obtenir la valeur exacte de E.

Programme LL(A,B)

```
LL(A,B):=(for i:A while i<=B and  y=1 do
                                        (p:i-A+1, F:10^(E+A-i), fpprec:i, u:b(x), if u<0 or u>0
                                          then (z:abs(u),EI(z), si:round(u*10^(i-e-1))*10^(e+1-i),
                                                  f:sp-si, if -F<f and f<F then 1 else y:0)
                                          else y:0)) $
```

Le programme LL(A,B) est le test de régularité sur un intervalle [A,B]

Programme CL(x)

```
CL(x):=(L:0, y:0,  for i while y=0 do (y:1, L:L+1, LL(L,L+S))) $
```

Le programme CL(x) détermine la valeur exacte de la précision d'amorçage. Il détermine la première valeur de L pour laquelle bfloat(x) est régulier sur l'intervalle [L,L+T-1].

Programme CT(x)

```
CT(x):=(T:1, j:1, for j while j <= L-1 do (y:1,  LL(j,j+S), T:max(p,T))) $
```

Pour chaque valeur de j de l'intervalle [1,L-1] , le programme CT(x) cherche la première valeur de p pour laquelle l'intervalle [j,j+p-1] n'est pas régulier. La plus grande des valeurs de p est l'indice de régularité.

Programme CFL(x,S)

```
CFL(x,S):=(Z:fpprec, b(x):=bfloat(x), L:0, ELL(x), EE(x), CL(x), CT(x), fpprec:Z,
disp("E"=E, "L"=L, "T"=T)) $
```

Programme récapitulatif

```
CFL(x,S):=(Z:fpprec, b(x):=bfloat(x),
EI(z):=(d:1, if z<1 then (for k:1 while d*z<1 do (d:10*d, e:-k))
                    else (for k:0 while d<=z do (d:10*d, e:k))),

L(x):=(for i:0 while i<=1 and y=1 do  (F:10^(e-i*S), fpprec:L+i*S, f:t-b(x),
                                        if  -F<f  and  f<F then 1  else y:0)),

ELL(x):=(y:0,for i while y=0 do (y:1,L:L+Z,fpprec:P:L+S+100,t:b(x),
                                  if t<0 or t>0 then (z:abs(t), EI(z)) else y:0,if y=1 then L(x) else 1),
          fpprec:P),

EE(x):=(Q:L+S+max(e+1,0), m:2, s:100, c:0, for j while  c=0 or c=s-1 do
              (m:m+m, s:s*s, if (W:Q+m)>P then (fpprec:P:W+100, t:b(x), z:abs(t), EI(z)) else 1,
               r:z*10^-e, a:entier(r), c:entier(s*r)-s*a), sp:round(t*10^(P-e-1))*10^(e+1-P), E:e) ,

LL(A,B):=(for i:A while i<=B and  y=1 do
                                    (p:i-A+1, F:10^(E+A-i), fpprec:i, u:b(x), if u<0 or u>0
                                      then (z:abs(u),EI(z), si:round(u*10^(i-e-1))*10^(e+1-i),
                                              f:sp-si, if -F<f and f<F then 1 else y:0)
                                      else y:0)),

CL(x):=(L:0, y:0, for i while y=0 do (y:1, L:L+1, LL(L,L+S))),

CT(x):=(T:1, j:1, for j while j <= L-1 do (y:1, LL(j,j+S), T:max(p,T))),

L:0, ELL(x), EE(x),  CL(x), CT(x), fpprec:Z, disp("E"=E, "L"=L, "T"=T)) $
```

```
(%i3)  x:10^5*log(1+exp(-50))$ CFL(x,100,10);
 E = -17
 L = 22
 T = 1
(%o3) done

(%i5) fpprec:51$ bfloat(x); BFLOAT(x);
(%o4) 1.92874984796391778301715681273 025534361374155865392b-17
(%o5) 1.92874984796391778301715681272 821153295468469035706b-17
                                        21 chiffres corrects
(%i6)  CFL(exp(1000),100);
 E = 434
 L = 2
 T = 1
(%o6) done

(%i7)  CFL(sin(exp(1000)),100);
 E = -1
 L = 435
 T = 3
(%o7) done

(%i8)  CFL(sin(exp(-10))-exp(-10)+exp(-30)/6,100);
 E = -24
 L = 20
 T = 52
(%o8) done
```

E- Programme de partie entière d'un nombre différent d'un entier

Programme ENTIER(x)

```
ENTIER(x):=(Z:fpprec, b(x):=bfloat(x),

EI(z):=(d:1, if z<1 then (for k:1 while d*z<1 do (d:10*d, e:-k))
                    else (for k:0 while d<=z do (d:10*d, e:k))),

L(x):=(for i:0 while i<=1 and y=1 do  (F:10^(e-i*Z), fpprec:L+i*Z, f:t-b(x),
                                        if  -F<f  and  f<F then 1  else y:0)),

ELI(x):=(y:0,for i while y=0 do (y:1,L:L+Z,fpprec:P:L+Z+100,t:b(x),
                                  if t<0 or t>0 then (z:abs(t), EI(z)) else y:0,if y=1 then L(x) else 1),
        fpprec:P) ,

a0(x):=(if E<-1 then a:entier(t)
              else (Q:L+e+2, m:2, s:100, c:0, for j while c=0 or c=s-1 do
                    (m:m+m, s:s*s, if (W:Q+m) > P then (fpprec:W+100, t:b(x)) else 1,
                                                    a:entier(t), c:entier(s*t)-s*a))) ,

L:0,y:0, for i while y=0 do (ELI(x), if y=1 then a0(x) else 1),  fpprec:Z, a) $
```

```
(%i5)  x:10^5*sin(exp(450))$ entier(bfloat(x));
(%o5) 47113

(%i7)  ENTIER(x);"L"=L;
(%o6)  –63274
(%o7) L = 208

(%i8)  CFL(x,100,10);
E = 4
L = 196
T = 2
(%o8) done

(%i10)  fpprec:195+4$entier(bfloat(x));
(%o10) -63271

(%i12)  fpprec:196+4$entier(bfloat(x));
(%o12) -63274
```

VI- Obtention de la meilleure fraction égale à x à ε près

Problème

Soient x un nombre réel non nul et ε un nombre réel positif qui sera choisi rationnel dans les programmes qui suivent.

Si $x > 0$, on se propose de déterminer les plus petits entiers $p \geq 0$ et $q > 0$ tels que $|x - p/q| < \varepsilon$. Dans ce cas p/q est la meilleure fraction égale à x à ε près.

Si $x < 0$, la meilleure fraction égale à x à ε près est l'opposée de la meilleure fraction égale à |x| à ε près.

A- Première approche

On se place dans le cas où x est un nombre positif ou nul et ε un rationnel.
On peut élaborer un programme direct dans le cas où x est une fraction.

On pose x = P/Q . La meilleure fraction I/J égale à x à ε près doit vérifier: $|P/Q - I/J| < \varepsilon$.

En supposant $P \geq 0$, $Q > 0$, $I \geq 0$ et $J > 0$, cette inégalité est équivalente à:

$$(P-\varepsilon\, Q)\, J/Q \; < I \; < (P+\varepsilon\, Q)\, J/Q \quad 2-3$$

L'idée est de donner à J des valeurs croissantes à partir de 1.
Pour chaque valeur de J on détermine la plus petite valeur de $I \geq 0$ qui vérifie la première inégalité.
Comme I est entier il en résulte que I = max(0 , entier((P–ε Q) J/Q) +1) .

La double inégalité est donc équivalente à : max(0,entier((P–ε Q) J/Q) +1) < (P+ε Q) J/Q .

Si D est la première valeur de J pour laquelle cette condition est réalisée, alors la meilleure fraction est max(0,Entier((P–ε Q) D/Q) +1)/D .

ε est représenté par ec.

Programme BFA(x,ec)

```
BFA(x,ec):=(P:num(x),Q:denom(x), E:(P-ec*Q)/Q, F:(P+ec*Q)/Q, b:0, B:1,for J:1 while B>= b do
                (D:J, B: max(0,entier(E*J)+1) , b:F*J), B/D) $
```

On a placé en face de chaque résultat la fraction continue de la fraction obtenue.

(%i2) BFA(50149/23778,10^-9);
(%o2) $\frac{50149}{23778}$ [2,9,5,1,7,3,8,2]

(%i3) BFA(50149/23778,10^-8);
(%o3) $\frac{23653}{11215}$ [2,9,5,1,7,3,8]

(%i4) BFA(50149/23778,3*10^-8);
(%o4) $\frac{17967}{8519}$ [2,9,5,1,7,3,6]

(%i5) BFA(50149/23778,10^-6);
(%o5) $\frac{1934}{917}$ [2,9,5,1,7,2]

(%i6) BFA(50149/23778,10^-5);
(%o6) $\frac{793}{376}$ [2,9,5,1,6]

Dans les calculs précédents les termes de la fraction continue des meilleures fractions, sauf peut-être le dernier terme, coïncident avec les premiers termes de la fraction continue de $\frac{50149}{23778}$.

Pour $\varepsilon = 10^{-8}$ la meilleure fraction est une réduite.

De façon générale que x soit rationnel ou irrationnel, les meilleures fractions seront obtenues à partir des réduites associées à la fraction continue de x.

Nous donnons les éléments qui serons utilisés dans l'élaboration des programmes et qui ont été développés en annexe.

On devra utiliser la fonction G_n définie par $G_n(z) = \dfrac{A_{n-1} + z\,B_{n-1}}{C_{n-1} + z\,D_{n-1}}$ avec $B_{n-1}.C_{n-1} - A_{n-1}.D_{n-1} = (-1)^n$

ainsi que la fonction H_n déjà utilisée antérieurement.

Résultats (proposition B-2, B-3)

Soit x > 0 .

(1) Il existe un plus petit entier n tel que $|x - b_n| < \varepsilon$. Si $\varepsilon \geq 1$, n = 0 .

(2) b_n est la meilleure réduite égale à x à ε près.

(3) Il existe un plus petit entier d positif ou nul tel que $|x - G_n(d)| < \varepsilon$.

(4) $G_n(d)$ est la meilleure fraction égale à x à ε près.
$G_n(d)$ restitue la fraction définie par la suite $[a_0, a_1, a_2, \ldots, a_{n-1}, d]$

(5) Soit H_n l'application réciproque de G_n . Alors $d = \max(E(H_n(x-(-1)^n \varepsilon))+1,0)$.

B- Programme dans le cas rationnel

ε est le quotient de deux entiers positifs . ε sera représenté dans le programme par e = M/N.
Ce programme, noté BFR(x,ec), ne concerne que les expressions de x qui sont le quotient de deux entiers. Le mode de calcul est celui du programme CFR(x,n) .
Nous procéderons en trois étapes.

(1) Détermination du rang n de la meilleure réduite égale à |x| à ε près.

(2) Recherche de la première valeur de l'entier d tel que $||x| - G_n(d)| < \varepsilon$ à l'aide du résultat (5) .

(3) Calcul de $G_n(d)$ qui est la meilleure fraction égale à |x| à ε près.

Détermination du rang n de la meilleure réduite

```
AR(x):= (u:P:num(X), v:Q:denom(X), A:0, B:1, C:1, D:0, c:M*Q , o:1,
          for i:0 while o*N*v >= c*D and v # 0 do
                              (o:-o, n:i, a:entier(u/v), B:A+(A:B)*a, D:C+(C:D)*a, u:-v, v:B*Q-D*P),
          A:B-(B:A)*a, C:D-(D:C)*a) $
```

X représente |x| . X = P/Q . $o = (-1)^{i+1}$. On recherche le premier entier n tel que |P/Q – B/D| < M/N .
Cette inégalité est équivalente à N*o*(B*Q – D*P) < M*Q*D .
Les termes de la fraction continue sont calculé aussi longtemps que N*o*(B*Q–D*P) >= M*Q*D .

Quand l'entier n est atteint, les coefficients A, B , C , D au rang n , doivent être rétrogradés au rang n-1 pour pouvoir définir les fonction H_n et G_n : A:B-(B:A)*a, C:D-(D:C)*a .
$H_n(t)$ = (C*t-A)/(B-D*t) , $G_n(z)$ = (A+z*B)/(C+z*D) .

Recherche de d

d = max(E(H_n(X-(-1)^n ε))+1,0) . w = X-(-1)^n ε = X+o*ec.

```
DR(x):=( w:X+o*ec,  d:max(entier((C*w-A)/(B-D*w))+1,0)) $
```

Programme BFR(x,ec)

```
BFR(x,ec):=(M:num(ec), N:denom(ec),sg:if x<0 then -1 else 1, X:sg*x , AR(x), DR(x),
                                                                    sg*(A+d*B)/(C+d*D)) $
```

sg est le signe de x. |x|= sg*x. La meilleure fraction est: sg*(A+d*B)/(C+d*D) = sg G_n(d).

Programme récapitulatif

```
BFR(x,ec):=(
AR(x):= (u:P:num(X), v:Q:denom(X), A:0, B:1, C:1, D:0, c:M*Q , o:1,
          for i:0 while o*N*v >= c*D and v # 0 do
                              (o:-o, n:i, a:entier(u/v), B:A+(A:B)*a, D:C+(C:D)*a, u:-v, v:B*Q-D*P),
          A:B-(B:A)*a, C:D-(D:C)*a),

DR(x):=( w:X+o*ec,  d:max(entier((C*w-A)/(B-D*w))+1,0)) ,

M:num(ec), N:denom(ec),sg:if x<0 then -1 else 1, X:sg*x , AR(x), DR(x), sg*(A+d*B)/(C+d*D))$
```

(%i2) BFR(50149/23778,3*10^-8);

(%o2) $\frac{17967}{8519}$

(%i3) BFR(50149/23778,10^-10);

(%o3) $\frac{50149}{23778}$

C- Cas irrationnel

Proposition B-7

Soit x un nombre irrationnel positif. Soit p le plus petit entier tel que 1/ (q_{p-1} + q_p) q_p ≤ ε .
Alors le plus petit entier n tel que |x - b_n| < ε est p-1 ou p .

Pour déterminer le rang n de la meilleur fraction, on procède en deux étapes.

(1) On détermine d'abord l'entier p de la proposition B-7.

(2) Puis on procède par élimination en testant l'inégalité |x - b_{p-1}| < ε .

On utilise toujours la démarche du programmes CFI(x,n).
Les programmes EI(z), L(x), ELI(x), ai(x) , TI(x) sont conservés.

Détermination de p

On établit un programme auxiliaire Ae(x) analogue au programme AI(x).

Programme Ae(x)

```
Ae(x):=(o:-1,Q:L+e+2, g:0, K:max(0,-e), d:1, A:0,B:1,C:1,D:0, u:t,
        for i:0 while y=1 and  N>(C+D)*D*M  do
            (p:i, a0:a, m:2, s:100, ai(x), if i=0 or (y=1 and a>0) then (B:A+(A:B)*a, D:C+(C:D)*a,
                                                                        g(D), K:2*(g+m), u:-v)
                                                                  else y:0) $
```

La condition $1/((q_{i-1} + q_i)\, q_i) \le \varepsilon$ s'écrit $1/((C+D)\, D) \le e$ ou encore: $N \le (C+D)*D*M$.

On détermine la première valeur de p pour laquelle $N \le (C+D)*D*M$.

Si on obtient une valeur de p non nulle, $a0 = a_{p-1}$ et $a = a_p$. On pose $o1 = o = (-1)^p$.

Les paramètres au rang p-1, notés A1 , B1 , C1 , D1 sont obtenus en rétrogradant les paramètres A, B, C, D :

A1:B-(B1:A)*a, C1:D-(D1:C)*a

Détermination du rang n de la meilleure réduite

Cas p = 0

n = 0 . On pose o1=1 , A1=0 , B1=1 , C1=1 , D1=0 . Ce sont les paramètres utilisés par le programme pour définir les fonction H_0 et G_0 et déterminer le nombre d .

Cas p > 0

On pose X = |x| . La double inégalité $0 \le |X - b_{p-1}| < ec$ est équivalente à $0 \le |X - b_{p-1}|/ec < 1$. Ce qui revient à dire que la partie entière de $(-1)^{p-1}(X - b_{p-1})/ec$ est nulle.

Précision nécessaire pour calculer la partie entière

On pose $F(X) = (-1)^{p-1}(X - b_{p-1})/ec$. Soit K un entier vérifiant $1/ec \le 10^K$.

On a: $10^{Kp+1} \ge (C_p + D_p\, s)^2 > (C_p + D_p)\, D_p \ge N/M = 1/ec$. On peut choisir $K = K_{p+1}$.

Soit q une valeur approchée de X telle que $|X - q| < 10^{Q-P}$.

On vérifie $|F(X) - F(q)| = (-1)^p\, (X - q)/ec \le |X-q|\, 10^K$. D'où $|F(X) - F(q)| < 10^{Q+K-P}$.

Pour obtenir $|F(X) - F(q)| < 10^{-m}$, il suffit de choisir $P \ge Q+K_{p+1}+m = P_{p+1}+m$.

Programme SEL(x)

```
SEL(x):=(if p=0
            then (o1:1, A1:0, B1:1, C1:1, D1:0)
            else (m:1, s:10, c:0, for j while c=0 or c=s-1 do
                    (m:m+m, s:s*s, W:P+m , if V<W then (fpprec:V:W+100, t:b(x)) else 1,
                      r:(B1/D1-t)*N/M, if o1*r >0 then (a:entier(r), c:entier(s*r)-s*a)  else (c:1,y:0)),
                  if y=1 then (if a=0 then (o1:-o1, A1:B1-(B1:A1)*a0, C1:D1-(D1:C1)*a0) else  1)
                       else 1)) $
```

On utilise la valeur de t calculée pour la précision P. Initialement o1 = $(-1)^p$

On pose F(q) est représenté par r = o1*(B1/D1-t)/ec et on cherche la partie entière de r.

Puis on détermine les paramètres qui définissent H_n et G_n :

Si a = 0 , n = p-1 . On a $(-1)^n = (-1)^{p-1}$ et on remplace o1 par -o1.

Les paramètres A1, B1, C1, D1 sont rétrogradés au rang p-2 à l'aide de la séquence:

A1:B1-(B1:A1)*a0, C1:D1-(D1:C1)*a0 .

Si a # 0 , n = p .On a $(-1)^n = (-1)^p$. Les paramètres o1 , A1 , B1 , C1 , D1 sont inchangés .

Détermination de d

Estimation de la précision nécessaire au calcul de d

On rappelle que $d = \max(E(H_n(X-(-1)^n \varepsilon))+1,0)$.

Soit q une valeur approchée de X telle que $|X - q| < 10^{Q-P}$.

Supposons par exemple n impair.

Le calcul donne: $|H_n(X+\varepsilon) - H_n(q+\varepsilon)| = |X - q|/[D_{n-1}{}^2 |(X+\varepsilon) - b_{n-1}||(q+\varepsilon) - b_{n-1}|]$

On a $|b_{n-1} - (X+\varepsilon)| = X+\varepsilon - b_{n-1} > X - b_{n-1} > 0$.

De même $|b_{n-1} - (q+\varepsilon)| = q+\varepsilon - b_{n-1} > q - b_{n-1} \geq X - b_{n-1} > 0$.

D'où $|H_n(X+\varepsilon)-H_n(q+\varepsilon)| < |X-q|/[D_{n-1}^2 (X - b_{n-1})^2] = |H_n(X) - H_n(q)| < |X-q|10^{Kn}$.

D'où $|H_n(X+\varepsilon) - H_n(q+\varepsilon)| < 10^{Q+Kn-P}$.

Pour obtenir $|H_n(X+\varepsilon) - H_n(q+\varepsilon)| < 10^{-m}$ il suffit de choisir $P \geq Q+K_n+m = P_n+m$.

Ce choix est aussi valable si n est pair .

Programme D(x)

```
D(x):=(m:1 , s:10 , c:0 ,
        for j while c=0 or c=s-1 do
                        (m:m+m, s:s*s, W:P+m, if V<W then (fpprec:V:W+100, t:b(x))  else 1,
                          w:t-o1*ec , v:B1-D1*w , if o1*v > 0
                                                  then (a:entier(r:(C1*w-A1)/v) , c:entier(s*r)-s*a)
                                                  else (c:1,y:0)),
        if y=1  then d:max(a+1,0) else 1) $
```

On utilise la valeur de t calculée pour la précision P .

$q-(-1)^n \varepsilon$ est représenté par w = t-o1*ec .

$H_n(q-(-1)^n \varepsilon)$ est représenté par r = (C1*w-A1)/(B1-D1*w) .

Programme BF(x,ec)

```
BF(x,ec):=(Z:fpprec, b(x):=bfloat(x),
            M:num(ec), N:denom(ec), L:0, y:0 ,
                  for i while y=0 do (ELI(x), if t<0 then (t:-t,sg:-1) else sg:1, X:sg*x, Ae(X),
                                      if y=1 then TI(X) else 1, if y=1 then SEL(X) else 1,
                                      if y=1 then D(X) else 1),
            fpprec:Z, sg*(A1+d*B1)/(C1+d*D1)) $
```

La meilleure fraction égale à x à ε près est: sg*(A1+d*B1)/(C1+d*D1) (= sg $G_n(d)$) .

```
(%i3) (x:sin(exp(100)),"y"=y:BF(x,10^-20));
```

$$(\%o3) \qquad y = \frac{1214130659}{8538302952}$$

```
(%i4) BFLOAT(x-y);
(%o4)  1.519172621677087b-21

(%i5) (x:exp(20), disp(entier(x) , BF(x,10^5)));
485165195
485065196
(%o5) done

(%i6) BF(10^10*log(1+10^-10),10^-20);
```

$$(\%o6) \qquad \frac{19999999997}{19999999998}$$

Programme récapitulatif

```
BF(x,ec):=(Z:fpprec, b(x):=bfloat(x),

EI(z):=(d:1, if z<1 then (for k:1 while d*z<1 do (d:10*d, e:-k))
                    else (for k:0 while d<=z do (d:10*d, e:k))),

L(x):=(for i:0 while i<=1 and y=1 do  (F:10^(e-i*Z), fpprec:L+i*Z, f:t-b(x),
                                        if  -F<f  and  f<F then 1  else y:0)),

ELI(x):=(y:0,for i while y=0 do (y:1,L:L+Z,fpprec:P:L+Z+100,t:b(x),
                                  if t<0 or t>0 then (z:abs(t),EI(z)) else y:0,if y=1 then L(x) else 1),
         fpprec:P) ,

g(D):=(for k:g+1 while D>d do (d:10*d, g:k)),

ai(x):=(o:-o, c:0, for j while c<3 or c>s-3 do
           (m:m+m, s:s*s, W:Q+K+m,
            if W>P then (fpprec:P:W+100, t:b(x), u:C*t-A) else 1,
            if y=1 then (v:B-D*t, if o*v >0 then (r:u/v, a:entier(r), c:entier(s*r)-s*a) else (c:3,y:0))
                   else 1)),

TI(x):=(m:10, s:10^10, ai(x), if y=1 and a>0 then (F:C+D*r, f:2*D*10^(Q+K-P), fpprec:V:P+4*Z,
                                                    d:o*(B-D*b(x)),
                                                     if  1 < (F+f)*d  and  (F-f)*d <1 then 1  else  y:0)
                                                else y:0) ,

Ae(x):=(o:-1,Q:L+e+2, g:0, K:max(0,-e), d:1, A:0,B:1,C:1,D:0, u:t,
        for i:0 while y=1 and  N>(C+D)*D*M  do
           (p:i, a0:a, m:2, s:100, ai(x), if i=0 or (y=1 and a>0) then (B:A+(A:B)*a, D:C+(C:D)*a,
                                                                       g(D), K:2*(g+m), u:-v)
                                                                  else y:0),
        if y=1 then (o1:o, A1:B-(B1:A)*a, C1:D-(D1:C)*a) else 1) ,

SEL(x):=(if p=0
           then (o1:1, A1:0, B1:1, C1:1, D1:0)
           else (m:1, s:10, c:0, for j while c=0 or c=s-1 do
                    (m:m+m, s:s*s, W:P+m , if V<W then (fpprec:V:W+100, t:b(x)) else 1,
                     r:(B1/D1-t)*N/M, if o1*r >0 then (a:entier(r), c:entier(s*r)-s*a)  else (c:1,y:0)),
                  if y=1 then (if a=0 then (o1:-o1, A1:B1-(B1:A1)*a0, C1:D1-(D1:C1)*a0) else  1)
                         else 1)),

D(x):=(m:1 , s:10 , c:0 ,
       for j while c=0 or c=s-1 do
                          (m:m+m, s:s*s, W:P+m, if V<W then (fpprec:V:W+100, t:b(x))  else 1,
                           w:t-o1*ec , v:B1-D1*w , if o1*v > 0
                                                  then (a:entier(r:(C1*w-A1)/v) , c:entier(s*r)-s*a)
                                                  else (c:1,y:0)),
       if y=1  then d:max(a+1,0) else 1),

M:num(ec), N:denom(ec), L:0, y:0 , M:num(ec), N:denom(ec), L:0, y:0 ,
for i while y=0 do (ELI(x), if t<0 then (t:-t, sg:-1) else sg:1, X:sg*x,  Ae(X),
                    if y=1 then TI(X) else 1, if y=1 then SEL(X) else 1, if y=1 then D(X) else 1) ,
fpprec:Z, sg*(A1+d*B1)/(C1+d*D1)) $
```

D- Programme de détermination des meilleures fractions de rang n

Proposition B-5

Soit x>0. Si n = 0, les meilleures fractions sont les entiers de l'intervalle $[0,a_0]$.
Si n>0, $G_n(d)$ est une meilleure fraction de rang n si et seulement si d appartient à l'intervalle $[a,a_n]$ où $a = E(H_n(2\ x - b_{n-1})) +1$.

Cas rationnel:

Soit X = |x| .

Programme ARn(x)

```
ARn(x):= (u:P:num(X), v:Q:denom(X), A:0, B:1, C:1, D:0,  for i:0 while i <= n and v # 0 do
                                    (p:i, a:entier(u/v),  B:A+(A:B)*a,D:C+(C:D)*a, u:-v, v:B*Q-D*P),
                                     an:a, A1:B-(B1:A)*a,C1:D-(D1:C)*a) $
```

Le programme ARn(x) détermine les termes de la fraction continue jusqu'au rang n et les paramètres qui définissent les fonctions H_n et G_n en rétrogradant les paramètres de rang n :

A1:B-(B1:A)*a , C1:D-(D1:C)*a .

Détermination de a

```
DRa(x):=(w:2*X-B1/D1, a:entier((A1-C1*w)/(D1*w-B1))+1) $
```

$2\ x - b_{n-1}$ est représenté par w = 2*X-B1/D1 .

$H_n(2\ x - b_{n-1})$ est représenté par (A1-C1*w)/(D1*w-B1) .

Programme BFRn(x,n)

```
BFRn(x,n):=(sg: if x < 0 then -1 else 1, X:sg*x , ARn(x), M:[],
                if n=0 then M:makelist(i,i,0,a)
                        elseif n <= p then (DRa(x), M:[B/D],
                                            for i:a while i < an do (B:B-B1, D:D-D1, M:cons(B/D,M)))
                        else 1 ,
sg*M) $
```

Pour n = 0 , les meilleures fractions sont les entiers de l'intervalle $[0, a_0]$.

Pour déterminer les meilleures fractions pour n>0 , on utilise la séquence:

M:[B/D], for i:a while i<an do (B:B-B1, D:D-D1,M:cons(B/D,M),

Si n est supérieur au rang du dernier terme de la fraction continue de x , M = [] .

Programme récapitulatif

```
BFRn(x,n):=(
ARn(x):= (u:P:num(X), v:Q:denom(X), A:0, B:1, C:1, D:0,  for i:0 while i <= n and v # 0 do
                                    (p:i, a:entier(u/v),  B:A+(A:B)*a,D:C+(C:D)*a, u:-v, v:B*Q-D*P),
                                     A1:B-(B1:A)*a,C1:D-(D1:C)*a, an:a),

DRa(x):=(w:2*X-B1/D1, a:entier((A1-C1*w)/(D1*w-B1))+1),

sg: if x < 0 then -1 else 1, X:sg*x , ARn(x), M:[],
if n = 0  then M:makelist(i,i,0,a)
          elseif n <= p then (DRa(x), M:[B/D], for i:a while i < an do
                                                    (B:B-B1, D:D-D1, M:cons(B/D,M)))
                        else 1 ,
 sg*M) $
```

(%i2) BFRn(sum(1/i!,i,0,5),0);

(%o2) [0,1,2]

(%i3) BFRn(sum(1/i!,i,0,5),1);

(%o3) [3]

(%i4) BFRn(sum(1/i!,i,0,5),2);

(%o4) $[\frac{5}{2},\frac{8}{3}]$

(%i5) BFRn(sum(1/i!,i,0,5),3);

(%o5) $[\frac{11}{4}]$

(%i6) BFRn(sum(1/i!,i,0,5),4);

(%o6) $[\frac{19}{7}]$

(%i7) BFRn(sum(1/i!,i,0,5),5);

(%o7) $[\frac{87}{32},\frac{106}{39},\frac{125}{46},\frac{144}{53},\frac{163}{60}]$

(%i8) BFRn(sum(1/i!,i,0,5),6);

(%o8) [ ]

On obtient la liste des meilleures fractions associées à x. En gras, les réduites.

[0 , 1 , **2** , **3** , $\frac{5}{2}$, $\mathbf{\frac{8}{3}}$, $\mathbf{\frac{11}{4}}$, $\mathbf{\frac{19}{7}}$, $\frac{87}{32}$, $\frac{106}{39}$, $\frac{125}{46}$, $\frac{144}{53}$, $\mathbf{\frac{163}{60}}$]

x = $\frac{163}{60}$

Cas irrationnel

Le programme noté BFbn(x,n) donnera les meilleures fractions de rang n de |x|.

Elles seront affectées du signe "–" si x < 0 . On conserve les programmes EI(z) , L(x) et ELI(x).

Programme An(x)

```
An(x):=(o:-1,Q: L+e+2 , g:0, K:max(0,-e), d:1,  A:0, B:1, C:1, D:0, u:t ,
          for i:0 while y=1 and i <= n do
                  (m:2, s:100, ai(x), if i=0 or (y=1 and a>0) then (B:A+(A:B)*a, D:C+(C:D)*a, g(D),
                                                                    K:2*(g+m), u:-v)
                                                                 else y:0),
        if y=1 then (an:a, Bn:B, Dn:D, o1:o , A1:B-(B1:A)*a, C1:D-(D1:C)*a) else 1) $
```

On détermine les termes de la fraction continue jusqu'au rang n.

Au rang n on mémorise les termes : an , Bn , Dn et o1 = $(-1)^n$.

On détermine les paramètres des fonctions H_n et G_n en rétrogradant les paramètres de rang n:

A1:B-(B1:A)*a , C1:D-(D1:C)*a .

Détermination de a

Comme dans le cas rationnel pour n > 0 , on doit déterminer l'entier a défini par:

$$a = \text{Entier}(H_n(2\,x - b_{n-1})) + 1$$

Précision nécessaire au calcul de a pour n > 0

Soit q une valeur approchée de X telle que $|X - q| < 10^{Q-P}$.

Le calcul donne:

$|H_n(2\,X - b_{n-1}) - H_n(2\,q - b_{n-1})| = |X - q| / (4\,D_{n-1}^2\,|X - b_{n-1}|\,|q - b_{n-1}|) =$

$|H_n(X) - H_n(q)| / 4 < |X-q|\,10^{Kn} < 10^{Q+Kn-P}$.

Pour obtenir $|H_n(2\,X - b_{n-1}) - H_n(2\,q - b_{n-1})| < 10^{-m}$ il suffit de choisir $P \geq Q+K_n+m = P_n+m$.

Programme Da(x)

```
Da(x):=(if n=0 then 1
                  else (m:1, s:10, c:0,  for j while c=0 or c=s-1 do
                                  (m:m+m, s:s*s, W:P+m , if V<W then (fpprec:V:W+100, t:b(x)) else 1,
                                   w:2*t-B1/D1,  v:B1-D1*w,
                                   if  o1*v > 0 then (a:entier(r:(C1*w-A1)/v),c:entier(s*r)-s*a)
                                                else (c:1,y:0)),
                            if  y=1 and a>= 0  then a:a+1 else y:0)) $
```

On utilise la valeur de t calculée pour la précision P. On a $(-1)^n = o1$.

On pose w = 2*t – B1/D1 et on cherche la partie entière de r = (w*C1-A1)/(B1-D1*w).

On vérifie : o1*(B1-D1*w) = 2*o1*(B1-D1*t) > 0 . a = entier(r)+1.

Programme BFn(x,n)

```
BFn(x,n):=(Z:fpprec,  b(x):=bfloat(x),
            L:0  , y:0 , for i while y=0 do (ELI(x), if t<0 then (t:-t, sg:-1) else sg:1, X:sg*x, An(X),
                                                 if y=1 then TI(X) else 1 , if y=1 then Da(X) else 1),
            if n = 0 then M:makelist(i,i,0,an)
                     else (M:[Bn/Dn] ,
                              for i:a while i<an do (Bn:Bn-B1 , Dn:Dn-D1, M:cons(Bn/Dn,M))),
            fpprec:Z, sg*M ) $
```

Pour n = 0 les meilleures fractions sont données par: M:makelist(i,i,0,an) .

Pour n > 0 les meilleures fractions sont données par la séquence:

M:[Bn/Dn] , for b:a while b < an do (Bn:Bn-B1 , Dn:Dn-D1, M:cons(Bn/Dn,M),

Le résultat final est donné par sg*M .

(%i2) BFn(%pi,0);

(%o2) [0,1,2,3]

(%i3) BFn(exp(1),5);

(%o3) $\left[\ \frac{30}{11}\ ,\ \frac{49}{18}\ ,\ \frac{68}{25}\ ,\ \frac{87}{32}\ \right]$

Exemple illustrant le résultat 5

Soit $x = \pi$. Pour n = 2 , on a $a_2 = 15$, $b_1 = 22/7$, $b_2 = 333/106$.
On doit déterminer le plus petit entier a tel que: $G_2(a) + b_1 - 2\pi > 0$
On donne les 15 valeurs de $G_2(c)$ pour $c \in [1,15]$ dans l'ordre croissant.

$$\left[\frac{25}{8},\frac{47}{15},\frac{69}{22},\frac{91}{29},\frac{113}{36},\frac{135}{43},\frac{157}{50},\frac{179}{57},\frac{201}{64},\frac{223}{71},\frac{245}{78},\frac{267}{85},\frac{289}{92},\frac{311}{99},\frac{333}{106}\right]$$

Nous avons $157/50 + 22/7 < 2\pi$ et $179/57 + 22/7 > 2\pi$. Alors a = 8 .
Les seules meilleures fractions possibles de rang 2 sont :

$$\left[\frac{179}{57},\frac{201}{64},\frac{223}{71},\frac{245}{78},\frac{267}{85},\frac{289}{92},\frac{311}{99},\frac{333}{106}\right]$$

(%i4) BFn(%pi,2);

(%o4) $$\left[\frac{179}{57},\frac{201}{64},\frac{223}{71},\frac{245}{78},\frac{267}{85},\frac{289}{92},\frac{311}{99},\frac{333}{106}\right]$$

Programme récapitulatif

```
BFn(x,n):=(Z:fpprec, b(x):=bfloat(x),

EI(z):=(d:1, if z<1 then (for k:1 while d*z<1 do (d:10*d, e:-k))
                    else (for k:0 while d<=z do (d:10*d, e:k))),

L(x):=(for i:0 while i<=1 and y=1 do  (F:10^(e-i*Z), fpprec:L+i*Z, f:t-b(x),
                                        if  -F<f  and  f<F then 1  else y:0)),

ELI(x):=(y:0,for i while y=0 do (y:1,L:L+Z,fpprec:P:L+Z+100,t:b(x),
                                  if t<0 or t>0 then (z:abs(t), EI(z)) else y:0,if y=1 then L(x) else 1),
          fpprec:P) ,

g(D):=(for k:g+1 while D>d do (d:10*d, g:k)),

ai(x):=(o:-o, c:0, for j while c<3 or c>s-3 do
            (m:m+m, s:s*s, W:Q+K+m,
             if W>P then (fpprec:P:W+100, t:b(x), u:C*t-A) else 1,
             if y=1 then (v:B-D*t, if o*v >0 then (r:u/v, a:entier(r), c:entier(s*r)-s*a) else (c:3,y:0))
                    else 1)),

An(x):=(o:-1,Q: L+e+2 , g:0, K:max(0,-e), d:1,  A:0, B:1, C:1, D:0, u:t ,
          for i:0 while y=1 and i <= n do
                (m:2, s:100, ai(x), if i=0 or (y=1 and a>0) then (B:A+(A:B)*a, D:C+(C:D)*a, g(D),
                                                                  K:2*(g+m), u:-v)
                                                              else y:0),
         if y=1 then (an:a, Bn:B, Dn:D, o1:o , A1:B-(B1:A)*a, C1:D-(D1:C)*a) else 1) ,

TI(x):=(m:10, s:10^10, ai(x), if y=1 and a>0 then (F:C+D*r, f:2*D*10^(Q+K-P), fpprec:V:P+4*Z,
                                                    d:o*(B-D*b(x)),
                                                     if  1 < (F+f)*d  and  (F-f)*d <1 then 1  else  y:0)
                                               else y:0) ,

Da(x):=(if n=0 then 1
                 else (m:1, s:10, c:0,  for j while c=0 or c=s-1 do
                                (m:m+m, s:s*s, W:P+m , if V<W then (fpprec:V:W+100, t:b(x)) else 1,
                                 w:2*t-B1/D1,  v:B1-D1*w,
                                 if  o1*v > 0 then (a:entier(r:(C1*w-A1)/v),c:entier(s*r)-s*a)
                                              else (c:1,y:0)),
                           if  y=1 and a>= 0  then a:a+1 else y:0)) ,

 L:0, y:0,  for i while y=0 do (ELI(x), if t<0 then (t:-t, sg:-1) else sg:1, X:sg*x, An(X),
                                  if y=1 then TI(X) else 1 , if y=1 then Da(X) else 1),
 if n = 0 then M:makelist(i,i,0,an)
          else (M:[Bn/Dn],  for i:a while i<an do (Bn:Bn-B1 , Dn:Dn-D1, M:cons(Bn/Dn,M))),
 fpprec:Z, sg*M )  $
```

Les calculs qui suivent mettent en évidence une valeur de ε pour laquelle $\frac{179}{57}$ est la meilleure fraction. On montre aussi que $\frac{157}{50}$ ne peut être une meilleure fraction pour π .

```
(%i5) float(%pi-179/57);
(%o5) 0.00124177639681066 8
```

```
(%i6) BF(%pi,1242*10^-6);
```

(%o6) $\frac{179}{57}$

Pour ε = 1242 10^{-6} , $\frac{179}{57}$ est la meilleure fraction égale à π à ε près.

```
(%i7) float(%pi-157/50);
(%o7) 0.00159265358979299 2
```

```
(%i8) BF(%pi,1592*10^-6);
```

(%o8) $\frac{22}{7}$

On a π -157/50 > 1592 10^{-6} . On peut conclure qu'il n'existe pas de nombre ε tel que $\frac{157}{50}$ soit la meilleure fraction égale à π à ε près.

Un cas où la meilleure fraction et la meilleure réduite sont identiques

Proposition B-6

Soit x un nombre réel positif tel que b_{n+1} # x . Pour ε = 1/ (q_{n+1} q_n), b_n est la meilleure réduite égale à x à ε près, c'est aussi la meilleure fraction égale à x à ε près.

```
(%i10) x:sqrt(1+sin(50*exp(20))^2);
```

(%o10) $\sqrt{\sin(50\ \%e^{20})^2 + 1}$

On se propose de montrer que la meilleure fraction est la réduite de rang 3 pour ε = 1/(q_4 q_3).
En effectuant CF(x,4), la réduite de rang 3 est A/C, q_3= C et q_4= D . On effectue le calcul suivant :

```
(%i11) (CF(x,4) , disp(A/C) , BF(x,1/(D*C)));
```

$\frac{234}{167}$

(%o11) $\frac{234}{167}$

Cas où x est rationnel et où $b_{n+1} = x$

Résultat

Si $b_{n+1} = x$, x est rationnel et $| x - b_n| = 1/(q_{n+1}\, q_n)$. Pour $\varepsilon = 1/(q_{n+1}\, q_n)$, la meilleure réduite est b_{n+1} et la meilleure fraction est la première des meilleures fractions de rang n+1.

On choisit, par exemple, comme valeur de x la réduite de rang 3 de l'exemple précédent :

$x = \frac{234}{167}$.

```
(%i3)  (x:234/167,CFR(x,10));
(%o3)  [1,2,2,33]
```

3 est le rang du dernier terme de la fraction continue de x.
On se propose de calculer la meilleure fraction pour $\varepsilon = 1/(q_3\, q_2)$.
On détermine toutes les meilleures fraction de rang 3 .

```
(%i4) BFRn(x,3);
(%o4)
```

$$\left[\frac{122}{87},\frac{129}{92},\frac{136}{97},\frac{143}{102},\frac{150}{107},\frac{157}{112},\frac{164}{117},\frac{171}{122},\frac{178}{127},\frac{185}{132},\frac{192}{137},\frac{199}{142},\frac{206}{147},\frac{213}{152},\frac{220}{157},\frac{227}{162},\frac{234}{167}\right]$$

En appliquant CFR(x,3), on obtient $C = q_2$ et $D = q_3$. Alors $\varepsilon = 1/(D\, C)$.
On effectue BFR(x,1/(D*C)) .

```
(%i5) (CFR(x,3) , BFR(x,1/(D*C)));
```

(%o5) $\frac{122}{87}$

La meilleure fraction est la première des meilleures fractions de rang 3 .

VII- Annexe:

A- Résultats utiles concernant les fractions continues

Fraction continue associée à un nombre réel

Définitions

N désigne l'ensemble des entiers naturels, E(x) désigne la partie entière de x .

À un nombre réel x , on associe les suites (x_n) et (a_n) définies comme suit:
Pour n = 0 , on pose $x_0 = x$, $a_0 = E(x)$.
Pour n > 0, et $x_{n-1} \# a_{n-1}$, on pose $x_n = 1/(x_{n-1} - a_{n-1})$ et $a_n = E(x_n)$.
S'il existe un entier p tel que $x_p = a_p$, x_n et a_n ne sont pas définis pour n > p .

On désigne par **M** l'ensemble des entiers n pour lesquels x_n et a_n sont définis.
S'il existe p tel que $x_p = a_p$, **M** est l'intervalle [0,p] de **N** . Sinon **M** = **N** .

La suite (a_n) est appelée fraction continue associée à x .
La suite (x_n) est appelée suite des successeurs de x .

Pour tout $q \in \mathbf{M}$, la suite $(a_q , \ldots)$ obtenue en supprimant les q premiers termes de la suite (a_n) , est la fraction continue associée à x_q .

Proposition A-1

(1) Pour tout $n \in \mathbf{M}$ avec $n > 0$, $x_n > 1$ et $a_n \geq 1$.

(2) Si **M** = [0,p] et si $p > 0$, alors $a_p \geq 2$.

(3) Si $n+1 \in \mathbf{M}$, alors $x_n = a_n + 1/x_{n+1}$.

(4) x est rationnel, si et seulement si la suite (a_n) est finie .

Réduite d'ordre n associée à x

Pour $n \in \mathbf{M}$, la réduite d'ordre n associée à x est le nombre rationnel défini par:

$$b_n = a_0 + 1/(a_1 + 1/(a_2 + \ldots + 1/(a_{n-1} + 1/a_n) \ldots))$$

On dit aussi que b_n est la réduite définie par la suite finie $(a_0, \ldots, a_n)$

Par exemple, pour n = 4 , $b_4 = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4}}}}$

Proposition A-2

Soient x un nombre réel non nul et b_n la réduite d'ordre n associée à x .
On pose $b_n = p_n / q_n$ où p_n / q_n est une fraction irréductible avec $q_n > 0$.
On pose $p_{-2} = 0$, $q_{-2} = 1$, $p_{-1} = 1$ et $q_{-1} = 0$.

(1) Pour tout $n \in \mathbf{M}$, on a : $p_n = p_{n-2} + p_{n-1}\, a_n$
$q_n = q_{n-2} + q_{n-1}\, a_n$

(2) Pour tout $n \in \mathbf{M} \cup \{-1\}$, $p_n\, q_{n-1} - p_{n-1}\, q_n = (-1)^{n+1}$.

(3) Le suite (q_n) est croissante, avec $q_{n+1} > q_n$ pour $n \geq 1$.

(4) Si x est irrationnel, la suite (q_n) tend vers l'infini .

Suites (G_n) et (H_n) de fonctions associée à la suite (a_n)

Proposition A-3

Soit $n \in \mathbf{M}$. On pose $G_n(z) = \dfrac{p_{n-2} + z\, p_{n-1}}{q_{n-2} + z\, q_{n-1}}$ et $H_n(t) = \dfrac{t\, q_{n-2} - p_{n-2}}{p_{n-1} - t\, q_{n-1}}$

(1) Pour n = 0 , G_0 est défini sur $\mathbf{R}$ par $G_0(z) = z$.
Pour n = 1 , G_1 est défini sur $]0,+\infty[$ par $G_1(z) = a_0 + 1/z$.
Pour n > 1 , G_n est défini sur $]-q_{n-2}/q_{n-1}, +\infty[$.
G_n est continue strictement croissante si n est pair et decroissante si n est impair .

(2) H_n est l'application réciproque de G_n .
Pour n = 0 , H_0 est défini sur $G_0(\mathbf{R}) = \mathbf{R}$ par $H_0(t) = t$.
Pour n = 1 , H_1 est défini sur $G_1(]0,+\infty[) =]a_0, +\infty[$ par $H_1(t) = 1/(t - a_0)$.
Pour n > 1 , H_n est défini sur $G_n(]-q_{n-2}/q_{n-1}, +\infty[)$.
$G_n(]-q_{n-2}/q_{n-1}, +\infty[) =]-\infty, b_{n-1}[$ pour n pair et $]b_{n-1}, +\infty[$ pour n impair.
Pour n > 0, t appartient au domaine de définition de H_n si et seulement si $(-1)^{n-1}(t - b_{n-1}) > 0$.
H_n est continue strictement croissante si n est pair et decroissante si n est impair .

(3) On a $b_n = G_n(a_n)$, $x = G_n(x_n)$, $x_n = H_n(x)$ et $a_n = H_n(b_n)$

Proposition A-4

Soit $x > 0$.

Pour n = 0, on pose $F_0 = G_0([0, a_0])$. Alors : $F_0 = [0, a_0]$

Pour n > 0, on pose $F_n = G_n(]0, a_n])$. Alors: $F_n =]b_{n-2}, b_n]$ pour n pair et $[b_n, b_{n-2}[$ pour n impair.

Si n est pair $F_n \subset [0,x]$. Si n est impair, $F_n \subset [x,+\infty[$.

Les intervalles F_n sont deux à deux disjoints.

Si x est irrationnel, les intervalles F_n constituent une partition de $[0,x[\cup]x,+\infty[$.

Si x est rationnel avec $x = b_p$, les intervalles F_n constituent une partition de

$[0,x] \cup [b_{p-1},+\infty[$ si p est pair
$[0,b_{p-1}] \cup [x,+\infty[$ si p est impair

Encadrement de $|x - b_n|$

Proposition A-5

Soit $n+1 \in \mathbf{M}$.

(1) $$x - b_n = \frac{(-1)^n}{(q_{n-1}+q_n x_{n+1})q_n} \quad \text{et} \quad |x - b_n| = (-1)^n (x - b_n)$$

(2) Si $n+1 \in \mathbf{M}$, $$\frac{1}{(q_n+q_{n+1})q_n} < |x - b_n| \leq \frac{1}{q_{n+1}q_n} \leq \frac{1}{(q_{n-1}+q_n)q_n} .$$

On a $|x - b_n| = 1/(q_{n+1} q_n)$ seulement si x est rationnel et si $b_{n+1} = x$.

(3) Si $n+1 \in \mathbf{M}$, $|q_n x - p_n| > |q_{n+1} x - p_{n+1}|$ et $|x - b_n| > |x - b_{n+1}|$.

(4) La suite (b_{2k}) est strictement croissante. La suite (b_{2k+1}) est strictement décroissante.

(5) Si x est irrationnel, la suite (b_n) converge vers x , les suites (b_{2k}) et (b_{2k+1}) sont adjacentes.

(6) Si x est rationnel, la dernière réduite b_p est égale à x .

Quelques propriétés algébriques des fractions continues et des réduites

Proposition A-6

Soient $(x_0 , x_1 , x_2 , x_3 , \dots , x_n , \dots)$ la suite des successeurs de x ,
$(a_0 , a_1 , a_2 , a_3 , \dots , a_n , \dots)$ la fraction continue associée à x .
et $(b_0 , b_1 , b_2 , b_3 , \dots , b_n , \dots)$ la suite des réduites associée à x .

Soit $a \in Z$.
Suite des successeurs de $x+a$: $(x_0 +a, x_1 , x_2 , x_3 , \dots , x_n , \dots)$.
Fraction continue associée à $x + a$: $(a_0 +a , a_1 , a_2 , a_3 , \dots , a_n , \dots)$.
Suite des réduites associée à $x + a$: $(b_0 +a , b_1 +a , b_2 +a , b_3 +a , \dots , b_n +a , \dots)$.

Soit $0 < x < 1$.
Suite des successeurs de $1/x$: $(x_1 , x_2 , x_3 , \dots , x_n , \dots)$.
Fraction continue associée à $1/x$: $(a_1 , a_2 , a_3 , \dots , a_{n+1} , \dots)$.
Suite des réduites associée à $1/x$: $(1/b_1 , 1/b_2 , 1/b_3 , \dots , 1/b_{n+1} , \dots)$.

Soit $x > 1$.
Suite des successeurs de $1/x$: $(1/x_0 , x_0 , x_1 , x_2 , x_3 , \dots , x_n , \dots)$.
Fraction continue associée à $1/x$: $(0 , a_0 , a_1 , a_2 , a_3 , \dots , a_{n-1} , \dots)$.
Suite des réduites associée à $1/x$: $(0 , 1/b_0 , 1/b_1 , 1/b_2 , 1/b_3 , \dots , 1/b_{n-1} , \dots)$.

Soient $x > 0$ et $x - a_0 > 1/2$ $(a_1 = 1)$.
Suite des successeurs de $-x$: $(-x_0 , x_2 +1 , x_3 , \dots , x_n , \dots)$.
Fraction continue associée à $-x$: $(-a_0-1, a_2 +1, a_3 , \dots , a_n , \dots)$.
Suite des réduites associées à $-x$: $(-b_1 , -b_2 , -b_3 , \dots , -b_{n+1} , \dots)$.

Soient $x > 0$ et $x - a_0 < 1/2$ $(a_1 > 1)$.
Suite des successeurs de $-x$: $(-x_0 , 1+1/(x_1 -1) , x_1 -1 , x_2 , x_3 , \dots , x_n , \dots)$.
Fraction continue associée à $-x$: $(-a_0-1, 1 , a_1-1, a_2 , a_3 , \dots , a_n , \dots)$.
Suite des réduites associée à $-x$: $(-b_0-1, -b_0 , -b_1 , -b_2 , -b_3 , \dots , -b_{n-1} , \dots)$.

Théorème A-7 (Meilleure approximation.)

Soit x un nombre irrationnel.
Pour tout entier p et tout entier q tel que $0 < q < q_{n+1}$ on a : $| q\,x - p | \geq | q_n\,x - p_n |$.

L'inégalité est une égalité si et seulement si $p = p_n$ et $q = q_n$.

Proposition A-8

Soit x un nombre irrationnel. Soient p et $q > 0$ deux entiers.

L'une des réduites $b_n = p_n/q_n$ ou $b_{n+1} = p_{n+1}/q_{n+1}$ satisfait à : $| x - p/q | < 1/2q^2$.

Proposition A-9

Soit x un nombre irrationnel. Soient p et q des entiers tels que $q > 0$.

La relation $| x - p/q | < 1/2q^2$ implique que p/q est une réduite de x .

Détermination d'une réduite définie par sa suite d'entiers avec Maxima

Soit b_n une réduite définie par la liste $F = [a_0, a_1, a_2, \ldots, a_{n-1}, a_n]$.

$$b_n = a_0+1/(a_1 + 1/(a_2 + \ldots + 1/(a_{n-1} + 1/a_n)...)) .$$

Un premier programme utilise la relation de récurrence: $\beta_n = a_n$, $\beta_i = a_i + 1/\beta_{i+1}$ pour $n > i \geq 0$.
Alors $b_n = \beta_0$.

```
R0(F):=(b:last(F) , F:rest(F,-1) , for j while F # [] do (a:last(F) , b:a+1/b , F:rest(F,-1) ) , b) $
```

Un autre programme, plus rapide, reconstitue les coefficients A_i , B_i , C_i , D_i en commençant par i = 0.

Pour les dernières valeurs calculées de B et D, on a x = B/D .

```
R(F):=(A:0 , B:1 , C:1 ,  D:0 , for j while F # []  do
                                (a:first(F) ,  B:A + (A:B)* a , D :C + (C:D) * a , F:rest(F,1) ) , B/D) $
```

```
(%i3)  R([0,1,2,3,4,5,6,7,8,9,10]);
(%o3)  5225670/7489051
```

Proposition A-10

(1) Soit $(a_0, \ldots, a_m)$ une suite finie d'entiers telle que $a_n \geq 1$ pour tout $n > 0$ et $a_m \geq 2$.
Alors la suite est la fraction continue d'un nombre rationnel .

(2) Soit $(a_n)_{n\in N}$ une suite infinie d'entiers telle que $a_n \geq 1$ pour tout $n > 0$.
Alors la suite est la fraction continue d'un nombre irrationnel .

Preuve

(1) On raisonne par récurrence sur m .
Pour m = 0, $x = a_0$. $E(x) = a_0$. (a_0) est la fraction continue de l'entier x.
Supposons la propriété vraie au rang m.
Soient $(a_0, \ldots, a_{m+1})$ et $x = a_0+1/(a_1 + 1/(a_2 + \ldots + 1/(a_m + 1/a_{m+1})...))$.
On pose $x_1 = a_1 + 1/(a_2 + \ldots + 1/(a_m + 1/a_{m+1})...)$.
D'après l'hypothèse de récurrence, $(a_1, \ldots, a_{m+1})$ est la fraction continue de x_1 .
De plus $x = a_0+1/x_1$. On vérifie : $x_1 > 1$. D'où $E(x) = a_0$ et $x_1 = 1/(x-a_0)$.
Donc $(a_0, \ldots, a_{m+1})$ est la fraction continue de x .

(2) On pose $p_{-2} = 0$, $q_{-2} = 1$, $p_{-1} = 1$ et $q_{-1} = 0$.
Pour tout $n \in \mathbf{N}$, on pose :
$p_n = p_{n-2} + p_{n-1} a_n$, $q_n = q_{n-2} + q_{n-1} a_n$ et $b_n = a_0+1/(a_1 + 1/(a_2 + \ldots + 1/(a_{n-1} + 1/a_n)...))$.
On vérifie les propriétés suivantes qui dépendent uniquement des définitions de p_n , q_n , b_n et des propriétés de la suite (a_n) :

(a) $b_n = p_n /q_n$.

(b) Pour tout $n \in \mathbf{N} \cup \{-1\}$, $p_n q_{n-1} - p_{n-1} q_n = (-1)^{n+1}$.

(c) Le suite (q_n) est croissante, avec $q_{n+1} > q_n$ pour $n \geq 1$.

(d) La suite (q_n) tend vers l'infini .

(e) $b_{n+2} - b_n = (-1)^n a_{n+2} /q_{n+2} q_n$ et $|b_{n+2} - b_n| < 1/q_{n+1} q_n$ pour $n \geq 0$.

(f) $b_{n+1} - b_n = (-1)^n /q_{n+1} q_n$ pour $n \geq 0$.

Il en résulte que la suite (b_{2k}) est strictement croissante, la suite (b_{2k+1}) est strictement décroissante et leur différence tend vers 0. Ce sont deux suites adjacentes qui convergent vers un nombre réel x qui vérifie $b_{2k} < x < b_{2k+1}$. Alors la suite (b_n) converge vers x.

Soit un entier $s > n+1$. Si $a_s > 1$, $(a_0, \ldots, a_s)$ est la fraction continue de b_s .
Si $a_s = 1$, $(a_0, \ldots, a_{s-2}, a_{s-1}+1)$ est la fraction continue de b_s .
Il résulte de la proposition A-5 : $(-1)^n (q_n b_s - p_n) > (-1)^{n+1} (q_{n+1} b_s - p_{n+1}) > 0$ pour $n \geq -1$.
Quand s tend vers l'infini, on obtient $(-1)^n (q_n x - p_n) \geq (-1)^{n+1} (q_{n+1} x - p_{n+1}) > 0$.
(L'égalité $q_{n+1} x - p_{n+1} = 0$ contredit la relation $b_{2k} < x < b_{2k+1}$) .
On pose $x_n = H_n(x)$. $H_n(x) = (q_{n-2} x - p_{n-2})/(p_{n-1} - q_{n-1} x)$.

Il reste à montrer que $E(x_n) = a_n$ et $x_{n+1} = 1/(x_n - a_n)$ pour tout n.
Le calcul donne : $x_n - a_n = H_n(x) - a_n = (q_n x - p_n)/(p_{n-1} - q_{n-1} x) = 1/x_{n+1}$.
D'où $x_{n+1} = 1/(x_n - a_n)$.
Comme $q_n x - p_n$ et $q_{n-1} x - p_{n-1}$ sont de signes contraires, on a $x_n - a_n > 0$.
De plus $(q_n x - p_n)/(p_{n-1} - q_{n-1} x) \leq 1$. Si $x_n - a_n < 1$, $E(x_n) = a_n$.

Que se passe-t-il si $x_n - a_n = 1$? Supposons par exemple n pair.
On a $x_n = a_n + 1$, $x = G_n(a_n+1) = G_{n+1}(1) \geq G_{n+1}(a_{n+1}) = b_{n+1}$. Ce qui contredit $x < b_{n+1}$.

Donc la suite (a_n) est la fraction continue de x et x est un nombre irrationnel.

B- Meilleure fraction égale à x à ε près

Dans tout ce qui suit ε désigne un nombre réel positif .

Soit x un nombre réel non nul.

Si x > 0, la meilleure fraction égale à x à ε près est la fraction p/q constituée des plus petits entiers p ≥ 0 et q > 0 tels que |x-p/q| < ε .

Si x < 0, la meilleure fraction égale à x à ε près est l'opposée de la meilleure fraction égale à |x| à ε près.

Dans ce qui suit, on suppose x > 0 .

Proposition B-1

Soient A, B, C et D et des entiers positifs ou nuls, tels que |A D – B C| = 1.
On a les propriétés suivantes:

(1) B et D sont premiers entre eux. A et C sont premiers entre eux.

(2) Soit d un nombre entier. Alors A + B d et C + D d sont premiers entre eux.

(3) Soient r/s une fraction irréductible avec r > 0 et s > 0. Alors:

(a) A s + B r et C s + D r sont premiers entre eux.

(b) Si r/s > d-1 où d est un entier positif ou nul, on a:
A s + B r ≥ A + B d et C s + D r ≥ C + D d .
Si aucun des entiers A, B, C et D n'est nul et si r/s # d , ces inégalités sont strictes.

Preuve

(1) , (2) et (3) (a) s'obtiennent à l'aide du théorème de Bezout.
Montrons (3) (b). Il suffit de montrer r ≥ d .
C'est évident dans les cas suivants d = 0 , d = 1 , s = 1 .
Si d ≥ 2 et s ≥ 2 on a r > s(d-1) ≥ d + d – 2 ≥ d.

Existence de la meilleure fraction

Proposition B-2

Soit x > 0 .

(1) Il existe un plus petit entier n tel que $|x - b_n| < \varepsilon$.

(2) Il existe un plus petit entier d positif ou nul tel que $|x - G_n(d)| < \varepsilon$.

(a) Si n =0, d est élément de l'intervalle $[0, a_0]$.

(b) Si n>0, d est élément de l'intervalle $[1, a_n]$.

(c) $G_n(d)$ est la meilleure fraction égale à x à ε près.

Preuve:

(1) Soit **A** l'ensemble des entiers n tel que $|x - b_n| < \varepsilon$. Si x est irrationnel, **A** est non vide car la suite $(|x - b_n|)$ tend vers 0. Si x est rationnel, **A** est non vide car il contient le rang p du dernier terme de sa fraction continue, puisque $b_p = x$. Donc **A** admet un plus petit élément.

(2) Soit **B** l'ensemble des entiers $c \in [0,a_n]$, tels que $|x - G_n(c)| < \varepsilon$. **B** est minoré par 0 et n'est pas vide car il contient a_n ($G_n(a_n) = b_n$). Donc **B** admet un plus petit élément d appartenant à $[0,a_n]$.

(b) Soit $n > 0$. Si $n = 1$, $d \neq 0$ car $G_1(0)$ n'est pas défini .
Dans les autres cas, $G_n(0) = b_{n-2}$ et $|x - b_{n-2}| \geq \varepsilon$. Donc $d \neq 0$.
Donc d appartient à l'intervalle $[1, a_n]$.

(c) Si 0 appartient à $]x-\varepsilon,x+\varepsilon[$, $n = 0$ et $G_0(0) = 0 = 0/1$ est la meilleure fraction.
Supposons que 0 n'appartienne pas à $]x-\varepsilon,x+\varepsilon[$. On vérifie $]x-\varepsilon,x+\varepsilon[\subset G_n(]0, +\infty[)$:
Par exemple, pour n pair > 0 , on a $x - b_{n-2} \geq \varepsilon$ et $b_{n-1} - x \geq \varepsilon$.
D'où $]x-\varepsilon,x+\varepsilon[\subset]b_{n-2}, b_{n-1}[= G_n(]0, +\infty[)$.

Soit p/q une fraction appartenant à $]x-\varepsilon,x+\varepsilon[$.
Il existe une fraction irréductible $r/s > 0$ tel que $s>0$ et $p/q = G_n(r/s)$.
Comme G_n est monotone et que d est le plus petit entier tel que $G_n(d)$ appartienne à $]x-\varepsilon,x+\varepsilon[$, on a $r/s > d-1$.

$G_n(r/s) = (s\, p_{n-2} + r\, p_{n-1})/(s\, q_{n-2} + r\, q_{n-1})$ et $G_n(d) = (q_{n-2} + d\, q_{n-1})/(q_{n-2} + d\, q_{n-1})$.

D'après la proposition B-1 , la fraction $(s\, p_{n-2} + r\, p_{n-1})/(s\, q_{n-2} + r\, q_{n-1})$ est irréductible et on a :

$$s\, p_{n-2} + r\, p_{n-1} \geq p_{n-2} + d\, p_{n-1} \quad \text{et} \quad s\, q_{n-2} + r\, q_{n-1} \geq q_{n-2} + d\, q_{n-1} .$$

Par suite $p \geq p_{n-2} + d\, p_{n-1}$ et $q \geq q_{n-2} + d\, q_{n-1}$.

Donc $G_n(d)$ est la meilleure fraction.

Proposition B-3

Soit H_n l'application réciproque de G_n . Alors $d = \max(E(H_n(x-(-1)^n \varepsilon))+1,0)$.

Preuve:

d est le plus petit entier qui vérifie $|x - G_n(d)| < \varepsilon$.
Soit H_n l'application réciproque de G_n .

Soit $n = 0$. $G_0(d) = d$. Pour $d \in [0, a_0]$ on a $0 < x - d < \varepsilon$.
d est le plus petit entier tel que $d > x - \varepsilon$ et $d \geq 0$.
Alors $d = \max(E(x-\varepsilon) + 1,0)$.

Soient $n > 0$ et n pair. G_n est une fonction strictement croissante sur $[0, a_n]$.
d et le plus petit entier qui vérifie: $0 < x - b_n = x - G_n(a_n) \leq x - G_n(d) < \varepsilon$
d et le plus petit entier qui vérifie: $G_n(d) > x - \varepsilon$. Alors $d = E(H_n(x-\varepsilon)) + 1$.

Pour $n > 0$ et n impair , on obtient de la même manière: $d = E(H_n(x+\varepsilon)) + 1$.

Dans tout les cas, on a: $d = \max(E(H_n(x-(-1)^n \varepsilon)+1,0)$.

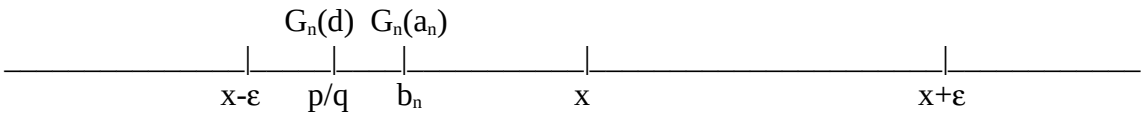
Remarques

(1) Soit n l'entier défini dans la proposition B-2.
Pour tout k > n, on a $|x - b_k| < \varepsilon$, $p_n \le p_k$ et $q_n \le q_k$ (ces inégalités sont strictes pour n > 0).
Nous dirons que b_n est la meilleure réduite égale à x à ε près.

(2) La meilleure réduite égale à x à ε près n'est pas toujours la meilleure fraction égale à x à ε près.

(3) La meilleure réduite est définie par la suite $(a_0 , a_1 , a_2 , a_3 , \dots , a_{n-1} , a_n)$.
La meilleure fraction est définie par la suite $(a_0 , a_1 , a_2 , a_3 , \dots , a_{n-1} , d)$.

Interprétation dans le cas où n est pair



G_n étant strictement croissante, il est parfois possible de trouver un entier $d \in [1,a_n[$ tel que $G_n(d) \in]x-\varepsilon, b_n[$.
Dans ce cas la meilleure fraction égale à x à ε près est différente de b_n .
L'objectif de ce qui suit est de caractériser, pour une valeur de n donnée, les entiers d tels que $G_n(d)$ soit la meilleure fraction égale à x à ε près, pour un choix convenable de ε .

Pour n = 0 , $G_0(d) = d$ et $a_0 = b_0$. Les valeurs convenables de d sont les entiers appartenant à l'intervalle $[0, a_0]$. Il suffit de choisir $\varepsilon = x-d+1/2$.

On déduit de la proposition A-4 le résultat suivant.

Proposition B-4

Soit x > 0 . Soit p/q la meilleure fraction égale à x à à ε près.
Le couple (n,d) qui vérifie $d \in [1, a_n]$ et $p/q = G_n(d)$ est unique.

Remarque

Soit n > 0 .
Si n est pair l'inégalité $b_{n-1} - x > x - G_n(d)$ est équivalente à $(-1)^n (G_n(d) + b_{n-1} -2 x) > 0$.
Si n est impair l'inégalité $x - b_{n-1} > G_n(d) – x$ est équivalente à $(-1)^n (G_n(d) + b_{n-1} -2 x) > 0$.
On vérifie que $(-1)^n (G_n(a_n) + b_{n-1} -2 x) > 0$.

Proposition B-5

Soit x > 0 et a_n le terme de rang n > 0 de la fraction continue associée à x .

(1) Le plus petit entier positif a qui vérifie: (C) $(-1)^n (G_n(a) + b_{n-1} – 2 x) > 0$
est défini par : $a = E(H_n(2 x – b_{n-1})) +1$

(2) Soit d un nombre entier. On peut trouver ε tel que $G_n(d)$ soit la meilleure fraction égale à x à ε près si et seulement si d élément de l'intervalle $[a,a_n]$

Preuve
Plaçons nous dans le cas où n est pair.
(1) La condition (C) s'écrit $G_n(a) > 2 x -b_{n-1}$.
Comme G_n est strictement croissante, $a = E(H_n(2 x - b_{n-1})) +1$

(2) Notons d'abord que $x - b_n < b_{n-1} - x$ et $G_n(a_n) = b_n$. Soit d un élément de $[a,a_n]$.
Si d = a, d'après la condition (C), la définition de a, et du fait que G_n est croissante, on a:
$0 \le x - b_n \le x - G_n(a) < b_{n-1} - x \le x - G_n(a-1)$.
Il suffit de choisir ε tel que $x - G_n(a) < \varepsilon < b_{n-1} - x$.
Si d > a , on a: $0 \le x - b_n \le x - G_n(d) < x - G_n(d-1) \le x - G_n(a) < b_{n-1} - x$.
Il suffit de choisir ε tel que $x - G_n(d) < \varepsilon < x - G_n(d-1)$.

Montrons que si d appartient à l'intervalle [1,a[, $G_n(d)$ ne peut pas être une meilleure fraction. Pour d élément de [1,a[, on a: $0 < b_{n-1} - x \le x - G_n(a-1) \le x - G_n(d)$.
S'il existait ε tel que $G_n(d)$ soit la meilleure fraction égale à x à ε près, on aurait $0 < b_{n-1} - x < \varepsilon$.
Il existerait un entier $p \le n-1$ et $d' \in [1,a_p]$ tel que $G_n(d)=G_p(d')$ ce qui contredit la proposition B-4.

Remarque

Soit x > 0 . Tout entier $d \in]0, a_0]$ est la meilleur fraction égale à x pour $\varepsilon \in]x-d,x-d+1]$,
0 est la meilleur fraction égale à x pour $\varepsilon > x$.

Un cas où la meilleure fraction et la meilleure réduite sont identiques

Proposition B-6

Soit x un nombre réel positif tel que $b_{n+1} \# x$. Soit $\varepsilon = 1/(q_{n+1} q_n)$. Alors:
(1) b_n est la meilleure réduite égale à x à ε près.
(2) b_n est la meilleure fraction égale à x à ε près.

Preuve

(1) D'après la proposition A-5, on a $|x - b_n| < \varepsilon$. Si n = 0, b_0 est la meilleure réduite.
Soit n > 0. $|x - b_{n-1}| > 1/(q_{n-1}(q_{n-1} + q_n)) \ge 1/(q_{n-1} q_{n+1}) \ge 1/(q_{n+1} q_n) = \varepsilon$.
Donc b_n est la meilleure réduite.
(2) Si $n \# 0$ et $a_n = 1$, la propriété est évidente. Dans les autres cas,
$|x - G_n(a_n - 1)| = (x_n - a_n + 1)/((q_n - q_{n-1})(x_n q_{n-1} + q_{n-2})) > 1/((q_n - q_{n-1})(q_n + q_{n-1})) \ge$
$1/((q_n - q_{n-1}) q_{n+1}) \ge 1/(q_{n+1} q_n) = \varepsilon$. Donc b_n est la meilleure fraction.

Remarque

On a $b_{n+1} = x$ quand x est rationnel. Dans ce cas, $| x - b_n| = 1/(q_{n+1} q_n) = \varepsilon$.
Alors la meilleure réduite est b_{n+1} et la meilleure fraction est la première des meilleures fractions de rang n+1.

Proposition B-7

Soit x un nombre irrationnel positif. Soit p le plus petit entier tel que $\dfrac{1}{(q_{p-1}+q_p)q_p} \le \varepsilon$.

Alors le plus petit entier n tel que $|x - b_n| < \varepsilon$ est p-1 ou p .

Preuve

On utilise le résultat suivant: $1/((q_n+q_{n+1}) q_n) < |x - b_n| < 1/((q_{n-1}+q_n) q_n)$ avec $n \ge 1$.
De cette relation on déduit $n \le p$.
Si p = 0 ou 1, la propriété est évidente.
Soit p >1 . On a $|x - b_{p-2}| > 1/(q_{p-2}(q_{p-2} + q_{p-1})) > 1/((q_{p-2} + q_{p-1}) q_{p-1}) > \varepsilon$. D'où n > p-2 .

C- Développement décimal d'un nombre réel

Proposition C-0

(1) $9\,(10^{m-1}+10^{m-2}+\ldots+10+1) = 10^m - 1$.

(2) $9\,(1/10+1/10^2+\ldots+1/10^m) = 1 - 1/10^m$.

(3) $9\,(1/10+1/10^2+\ldots+1/10^m)$ tend vers 1 quand m tend vers l'infini .

Soit $x = a_0 + a_1/10 + a_2/10^2 + \ldots + a_p/10^p$.
a_0 est un entier relatif, a_1 , a_2 , … , a_p sont des nombres entiers compris entre 0 et 9.
x est un nombre décimal car 10^p x est un nombre entier.

Soit m un entier tel que $0 \le m \le p$. On pose $S_m = a_0 + a_1/10 + a_2/10^2 + \ldots + a_m/10^m$.

Alors : $x = S_m + b/10^m$ où $b = a_{m+1}/10^1 + \ldots + a_p/10^{p-m}$.

On vérifie : $S_m = E(10^m x)/10^m$, $0 \le b \le 9\,(1/10+1/10^2+\ldots+1/10^{p-m}) = 1-1/10^{p-m} < 1$, $a_0 = E(x)$ et

$$0 \le x - (a_0 + a_1/10 + a_2/10^2 + \ldots + a_m/10^m) < 1/10^m$$

Ceci exprime simplement que pour $x = 5{,}947695234$, on a $0 \le x - 5{,}947695 < 1/10^6$

On peut compléter la suite $(a_0 , a_1 , a_2 , .., a_p)$ avec des termes nuls a_m pour $m > p$.
La relation précédente est encore valable pour tout entier $m \ge 0$.

L'objectif est de déterminer, pour un nombre réel x quelconque, une suite (a_m) infinie ayant la même propriété.

Cette suite sera le développement décimal de x (noté DD de x).

Proposition C-1

Soient un nombre réel x , un entier $m \ge 0$. On pose $d_m = E(10^m x)/10^m$.

(1) Il existe un seul décimal $d \in]\, x-1/10^m, x\,]$ tel que $10^m d$ soit entier. Il est égal à d_m .

(2) Il existe un seul décimal $d \in]\, x, x+1/10^m\,]$ tel que $d\,10^m$ soit entier. Il est égal à $d_m + 1/10^m$.

(3) Les suites (d_m) et $(d_m + 1/10^m)$ convergent vers x .

(4) Il existe une suite unique (a_m) qui vérifie :

$$0 \le a_m \le 9 \quad \text{et} \quad 0 \le x - \sum_{i=0}^{m} a_i/10^i < 1/10^m \text{ , pour tout entier } m \ge 0$$

Elle est définie par $a_0 = E(x)$ et $a_m = E(10^m x) - 10\,E(10^{m-1} x)$ pour $m > 0$.

(5) Si x est décimal $a_m = 0$ à partir d'un certain rang.

(6) Il n'existe pas de rang à partir duquel les termes a_m sont égaux à 9 .

Preuve :

(1) Soit d tel que 10^m d soit entier. La double inégalité $x - 10^{-m} < d \leq x$ est équivalente à $d \leq x < d + 10^{-m}$, puis à $10^m d \leq 10^m x < 10^m d + 1$, puis à $E(10^m x) = 10^m d$, puis à $d = d_m$. $10^m d_m$ est un nombre entier.

(2) La double inégalité $x < d \leq x+1/10^m$ est équivalente à $x - 10^{-m} < d - 10^{-m} \leq x$. Supposant $10^m d$ entier, $10^m (d - 10^{-m})$ est entier. D'après (1) , $d-10^{-m} = d_m$. Donc $d = d_m + 10^{-m}$. $10^m (d_m + 10^{-m})$ est un nombre entier.

(3) résulte de (1) et (2) et du fait que $1/10^m$ tend vers 0 quand m tend vers l'infini.

(4) Pour une telle suite, on pose $S_m = \sum_{i=0}^{m} a_i /10^i$. $10^m S_m$ est un entier et $S_m \in]\, x-1/10^m, x\,]$. Il en résulte que $S_m = d_m$.
$a_0 = S_0 = d_0 = E(x)$. Pour m > 0, le calcul donne $a_m = 10^m (d_m - d_{m-1}) = E(10^m x) - 10\, E(10^{m-1}x)$. Ce qui assure l'unicité de la suite (a_m) .

Soit $y = 10^{m-1} x$. $a_m = E(10\, y) - 10\, E(y)$. $E(y) \leq y < E(y) + 1$. $10\, E(y) \leq 10\, y < 10\, E(y) + 10$.
D'où : $10\, E(y) \leq E(10\, y) < 10\, E(y) + 10$, puis : $0 \leq E(10\, y) - 10\, E(y) < 10$.
Comme a_m est entier, on a $0 \leq a_m \leq 9$.
Dans ce qui suit, $a_0 = E(x)$ et $a_m = E(10^m x) - 10\, E(10^{m-1}x)$ pour m > 0 .
Montrons par récurrence : $S_m = d_m$.
Pour m = 0, $S_0 = a_0 = E(x) = d_0$.
Soit m > 0 . On suppose $S_{m-1} = d_{m-1}$. D'où $S_m = S_{m-1} + a_m /10^m = d_{m-1} + 10^m (d_m - d_{m-1})/10^m = d_m$.
Ce qui assure l'existence de la suite (a_m) .

La suite (a_m) est le développement décimal de x.

Pour exprimer que la suite (S_m) converge vers x, on écrit : $x = \sum_{i=0}^{\infty} a_i /10^i$.

(5) Si x est un nombre décimal, il existe un entier p tel que 10^p x soit un nombre entier.
Alors , pour tout entier m > p , on a $a_m = E(10^m x) - 10\, E(10^{m-1} x) = 10^m x - 10 \times 10^{m-1} x = 0$.

(6) Supposons qu'il existe un entier p tel que $a_m = 9$ pour tout m > p .
Dans ce cas x n'est pas décimal.

$x = \sum_{i=0}^{p} a_i /10^i + 1/10^p \sum_{i=1}^{\infty} 9/10^i = \sum_{i=0}^{p} a_i /10^i + 1/10^p$ (proposition C-0 (3)). x serait un nombre décimal.

Ce qui est contradictoire.

Écriture

Soit x > 0 . On pose $x = a_0,a_1\, a_2 \ldots a_m \ldots$ et $-x = -a_0\, ,a_1\, a_2 \ldots a_m \ldots$.

Exemple : $\pi = 3,141592653 \ldots$. $-\pi = -3,141592653 \ldots$.
Cependant $(-a_0\, , a_1\, , a_2\, , \ldots , a_m\, , \ldots)$ n'est pas le DD de -x .

Soit (a'_m) le DD de -x. Si x est entier : $E(-x) = -x$, sinon $E(-x) = -E(x) - 1$. Soit m > 0.

Si 10^{m-1} x et 10^m x ne sont pas entiers : $a'_m = 9 - a_m$.
Si 10^{m-1} x n'est pas entier et 10^m x entier : $a'_m = 10 - a_m$.
Si 10^{m-1} x et 10^m x sont entiers : $a'_m = 0$.

DD de $-\pi$: (- 4,8,5,8,4,0,7,3,4,6,…). $-\pi = -4 + 0,858407346 \ldots$.
Pour x = 51,289652395, DD de -x: (-52,7,1,0,3,4,7,6,0,5,0, … ,0, …). -x= -52+0,710347605 .

Arrondi de a_m à l'unité la plus proche

En utilisant les propositions C-0 et C-1 , on montre :

Proposition C-2

(1) Si $a_{m+1} \le 4$, on pose $D_m = S_m$. Alors $0 \le x - D_m < 5/10^{m+1}$.
(2) Si $a_{m+1} \ge 5$, on pose $D_m = S_m + 1/10^m$. Alors $0 < D_m - x \le 5/10^{m+1}$.
Dans tous les cas $|x - D_m| \le 5/10^{m+1}$.

Proposition C-3

On suppose que $d\,10^m$ est un entier.
Si $0 \le x - d < 5/10^{m+1}$, alors $d = S_m$ et $a_{m+1} \le 4$.
Si $0 < d - x < 5/10^{m+1}$, alors $d = S_m + 1/10^m$ et $a_{m+1} \ge 5$.
Si $|d - x| = 5/10^{m+1}$, alors $d = S_m$ ou $d = S_m + 1/10^m$ et $x = S_m + 5/10^{m+1}$.

Les m premières décimales du développement décimal de x

Proposition C-4

Soient un nombre réel x , un entier $m \ge 0$, $c_m(x) = E(10^m x) - 10^m E(x) = E(10^m (x - E(x))$.

(1) L'équation : $x = E(x) + c\,10^{-m} + b\,10^{-m}$ admet une solution unique (c,b) telle que c soit un nombre entier et b un nombre réel vérifiant $0 \le b < 1$: $c = c_m(x)$ et $b = 10^m x - E(10^m x)$.

(2) En base 10, $c_m(x)$ s'écrit : $a_1 a_2 \ldots a_m$ où $a_1 , a_2 , \ldots , a_m$ sont les m premières décimales du développement décimal de x .

(3) $0 \le c_m(x) \le 10^m - 1$.

(4) Si x est entier, $c_m(x) = 0$ pour tout entier $m \ge 0$.

(5) Si x n'est pas entier, il existe un entier p tel que, pour tout $m \ge p$, on ait $1 \le c_m(x) \le 10^m - 2$.

Preuve :

(1) On pose $d = E(x) + c\,10^{-m}$. L'équation est équivalente à $0 \le x - d < 1/10^m$ où $10^m d$ est un entier. D'où $E(x) + c\,10^{-m} = d_m = E(10^m x)/10^m$ (Proposition C-1 (1)).
Le calcul donne $c = c_m(x)$, puis $b = 10^m x - E(10^m x)$. c est entier et $0 \le b < 1$.

(2) $c_m(x) = 10^m (S_m - a_0) = \sum_{i=1}^{m} a_i\,10^{m-i}$.

(3) $0 = \sum_{i=1}^{m} 0\,10^{m-i} \le \sum_{i=1}^{m} a_i\,10^{m-i} \le \sum_{i=1}^{m} 9\,10^{m-i} = 10^m - 1$.

(4) Si x est entier, $10^m x$ est entier et $c_m(x) = E(10^m x) - 10^m E(x) = 10^m x - 10^m x = 0$.

(5) D'après la proposition C-1, il existe un rang $u > 0$ tel que $a_u \# 9$.
Alors pour tout $m \ge u$, $c_m(x) \# 10^m - 1$.

Si x n'est pas entier, il existe un rang $v > 0$ tel que $a_v \# 0$. Alors, pour tout $m \ge v$, $c_m(x) \# 0$.

Donc pour tout $m \ge \max(u,v) = p$, on a $1 \le c_m(x) \le 10^m - 2$.

Propriété C: $c_{m+p}(y) = 10^p\,c_m(y) + c_p(y)$ et $10^p\,c_m(y) \le c_{m+p}(y) \le 10^p\,c_m(y) + 10^p - 1$

Condition d'obtention d'une partie entière

Proposition C-5

Soit x un nombre entier. Alors, pour tout entier $m>0$ et tout y vérifiant $|x - y| < 10^{-m}$, on a:

soit $c_m(y) = 0$ et $x = E(y)$

soit $c_m(y) = 10^m - 1$ et $x = E(y) + 1$

Preuve :

On a $y = x + a\,10^{-m}$ avec $-1 < a < 1$.

Si $0 \le a < 1$, on a $c_m(y) = 0$ et $E(y) = x$.

Si $-1 < a < 0$, $y = x - 1 + (1+a\,10^{-m})$ où $0 < 1+a\,10^{-m} < 1$.
Alors $E(y) = x - 1$.
De plus, $y = x - 1 + (10^m - 1)\,10^{-m} + (1+a)\,10^{-m}$ avec $0 < 1+a < 1$. Alors $c_m(y) = 10^m - 1$.

Proposition C-6

Soit x un nombre différent d'un entier.

(1) Soient un entier $m > 0$ et y un nombre réel tel que: $|x - y| < 10^{-m}$ et $1 \le c_m(y) \le 10^m - 2$.
Alors : $E(x) = E(y)$.
Si, de plus, $3 \le c_m(y) \le 10^m - 3$, pour tout z vérifiant $|x - z| < 10^{-m}$, on a $E(z) = E(x)$ et $c_m(z) \ge 1$.

(2) Il existe un entier $p > 0$ tel que :
pour tout entier $m \ge p$ et tout y vérifiant : $|x - y| < 10^{-m}$, on ait : $3 \le c_m(y) \le 10^m - 3$.

Preuve

(1) $x = y + a\,10^{-m}$ où $-1 < a < 1$.

$y = E(y) + c_m(y)\,10^{-m} + b\,10^{-m}$ où $0 \le b < 1$. D'où $x = E(y) + c_m(y)\,10^{-m} + (a+b)\,10^{-m}$.

On pose : $d = c_m(y) + a + b$. On a : $0 < d < 10^m$. D'où $0 < d\,10^{-m} < 1$ et $E(x) = E(y)$.

On suppose : $3 \le c_m(y) \le 10^m - 3$. Soit z tel que $|x - z| < 10^{-m}$. $z = x + e\,10^{-m}$ avec $-1 < e < 1$.

$z = E(x) + (c_m(y) + a + b + e)\,10^{-m}$. On pose $f = c_m(y) + a + b + e$. On a $1 < f < 10^m$.

D'où $0 < f\,10^{-m} < 1$ et $E(z) = E(x)$. De plus $1 \le E(f) = c_m(z) \le 10^m - 1$.

(3) Comme x n'est pas entier, il existe $q > 0$ tel que $1 \le c_q(x) \le 10^q - 2$.

Soit un entier $r \ge 1$ et $m = q + r$.

On a : $10^r \le c_{q+r}(x) \le 10^{q+r} - 10^r - 1$ (propriété C).

$x = E(x) + c_{q+r}(x)\,10^{-q-r} + b\,10^{-q-r}$ avec $0 \le b < 1$.

Soit y tel que $|x - y| < 10^{-q-r}$.

$y = x + a\,10^{-q-r}$ avec $-1 < a < 1$. $y = E(x) + c_{q+r}(x)\,10^{-q-r} + (a+b)\,10^{-q-r}$.

$d = c_{q+r}(x) + a+b$. $10^r - 1 < d < 10^{q+r} - 10^r + 1$.

D'où $3 < 10^r - 1 \le E(d) = c_m(y) \le 10^m - 10^r < 10^m - 3$. Il suffit de choisir $p = q+1$.

D- Précision nécessaire au calcul de a_n

Dans ce qui suit, x désigne un nombre irrationnel. Quel que soit n , x_n n'est pas un nombre entier.

Les fonctions G_n et H_n réciproques l'une de l'autre sont définies par :

$$G_n(z)=\frac{(A_{n-1}+B_{n-1}z)}{(C_{n-1}+D_{n-1}z)} \qquad \text{et} \qquad H_n(t)=\frac{(C_{n-1}t-A_{n-1})}{(B_{n-1}-D_{n-1}t)}$$

On déduit de la proposition A-3 les résultats suivants :

Pour n > 0, q appartient au domaine de définition de H_n si et seulement si $(-1)^{n-1}(q - b_{n-1}) > 0$.

Pour tout entier p > 0 et tout entier n, G_n est défini sur l'intervalle $I =]x_n - 10^{-p} , x_n + 10^{-p} [$.
C'est vrai pour n = 0 . C'est vrai aussi pour n > 0 car $x_n > 1$.

Soit J l'ensemble des nombres q tels que $|x_n - H_n(q)| < 10^{-p}$. Comme G_n est continue strictement monotone, J est l'intervalle ouvert contenant x défini par $J = G_n(I)$.

On déduit de ce qui précède et de la proposition C-6 :

Proposition D-1

Soient (m_n) une suite d'entiers positifs et (r_n) une suite de nombres réels telles que
$|x_n - r_n| < 10^{-m_n}$ et $3 \le c_{m_n}(r_n) \le 10^{m_n} - 3$.

(1) Soit I_n l'ensemble des nombres z tels que $|x_n - z| < 10^{-m_n}$.
Soit J_n l'ensemble des nombres q tels que $|x_n - H_n(q)| < 10^{-m_n}$. On a $J_n = G_n(I_n)$.
J_n est un intervalle ouvert qui contient x . Pour tout $q \in J_n$, $E(H_n(q)) = a_n$.

(2) Soit N_n l'intersection des intervalles J_i pour $0 \le i \le n$. N_n est un intervalle ouvert qui contient x.
Pour tout $q \in N_n$, la fraction continue de q coïncide avec celle de x au moins jusqu'au rang n.

Le calcul de x_n et a_n est itératif. La valeur de m_n n'est pas connue à priori. Il faut déterminer la précision nécessaire avec laquelle il faut choisir la valeur approchée q_n de x pour obtenir une valeur de m_n satisfaisant à la propriété (1) avec $r_n = H_n(q_n)$. On utilisera les éléments issus du calcul de a_{n-1}.

Proposition D-2

Soient n > 0 , x un nombre irrationnel et q un nombre réel tel que $(-1)^{n-1}(q - b_{n-1}) > 0$.
On suppose que la fraction continue de q coïncide avec celle de x jusqu'au rang n-1. Alors :

$$| x_n - H_n(q) | = |x - q | \ (C_{n-1} + D_{n-1} x_n)(C_{n-1} + D_{n-1} H_n(q)) \qquad (1)$$

Preuve:
Comme $q \# b_{n-1}$, la fraction continue de q est définie au moins jusqu'au rang n.
La fonction H_n définie à l'aide des paramètres de rang n-1, est la même pour x et pour q .

On a $x_n = H_n(x)$ et on pose $r = H_n(q)$. Le calcul donne:

$$|x_n - r| = |H_n(x) - H_n(q)| = \frac{|x-q|}{|D_{n-1}x - B_{n-1}||D_{n-1}q - B_{n-1}|} = \frac{|x-q|}{D_{n-1}^2|x - b_{n-1}||q - b_{n-1}|}$$

Remarquant que $|A_{n-1} D_{n-1} - B_{n-1} C_{n-1}| = 1$ et utilisant la proposition A-5, on obtient :

$|x - b_{n-1}| = 1/[(C_{n-1} + D_{n-1} x_n) D_{n-1}]$ et $|q - b_{n-1}| = 1/[(C_{n-1} + D_{n-1} r) D_{n-1}]$. La relation (1) en résulte.

Proposition D-3

Soient $n > 0$ et q un nombre réel tel que $(-1)^{n-1}(q - b_{n-1}) > 0$.
On suppose que la fraction continue de q coïncide avec celle de x jusqu'au rang n-1.

Soient un entier $m > 0$ et r un nombre tels que : $| x_{n-1} - r | < 10^{-m}$ et $3 \le c_m(r) \le 10^m - 3$.

Soit K_n un entier vérifiant : $(D_{n-1}\, 10^m)^2 \le 10^{K_n}$.

On suppose : $| x_{n-1} - H_{n-1}(q)| < 10^{-m}$.
Alors:

$$(C_{n-1} + D_{n-1}\, x_n\ )\ (C_{n-1} + D_{n-1}\, H_n(q)) < (D_{n-1}\, 10^m)^2 \quad (2)$$

$$| x_n - H_n(q) | < |x - q|\, 10^{K_n} \quad (3)$$

Preuve
On pose $c_m = c_m(r)$.

D'après la proposition C-6 (1) , on a $E(r) = E(x_{n-1}) = a_{n-1}$.
$r = a_{n-1} + c_m\, 10^{-m} + b\, 10^{-m}$ avec $0 \le b < 1$.

Par définition, $x_n = 1/(x_{n-1} - a_{n-1})$.

On a $x_{n-1} = r + a\, 10^{-m}$ avec $-1 < a < 1$.
D'où $x_{n-1} - a_{n-1} = (c_m + a + b)10^{-m} > 2\, 10^{-m}$, puis $x_n < 10^m /2$.
D'où $C_{n-1} + D_{n-1}\, x_n < C_{n-1} + D_{n-1}\, 10^m /2 \le D_{n-1}\, (1+ 10^m/2)$.

Comme la fraction continue de q coïncide avec celle de x jusqu'au rang n-1, la fonction H_{n-1} est la même pour x et pour q .

On pose $H_{n-1}(q) = z$ et $H_n(q) = s$. D'où $s = 1/(z - a_{n-1})$.

Comme $| x_{n-1} - z| < 10^{-m}$, on a $c_m(z) \ge 1$ et $E(z) = E(x_{n-1}) = a_{n-1}$ (proposition C-6 (1)) .
$z = a_{n-1} + c_m(z)10^{-m} + d\, 10^{-m}$ avec $0 \le d < 1$. D'où $z - a_{n-1} \ge 10^{-m}$, puis $s \le 10^m$.

D'où $C_{n-1} + D_{n-1}\, s < C_{n-1} + D_{n-1}\, 10^m \le D_{n-1}\, (1+ 10^m)$.
Puis $(C_{n-1} + D_{n-1}\, x_n\ )\ (C_{n-1} + D_{n-1}\, s\ ) \le (D_{n-1})^2\, (10^{2m} +3\, 10^m+2)/2 < (D_{n-1})^2\, 10^{2m}$ (car $m \ge 1$).
(3) résulte des relations (1) et (2).

Proposition D-4

Soit H_n la fonction associée à la fraction continue de x.
On peut trouver deux suites (q_n) et (r_n) de nombres réels, des suites (m_n) , (K_n) , (V_n) d'entiers positifs et une suite (O_n) d'intervalles ouverts contenant x qui vérifient:

(1) $r_n = H_n(q_n)$.

(2) $(D_{n-1}\, 10^{m_{n-1}})^2 \le 10^{K_n}$ pour $n > 0$.

(3) $| x_n - r_n | < 10^{-m_n}$ et $3 \le c_{m_n}(r_n) \le 10^{m_n} - 3$.

(4) $K_0 \ge 0$, $V_0 \ge K_0 + m_0$ et $V_n \ge \max(K_n + m_n , V_{n-1})$ pour $n > 0$.

(5) O_n est l'ensemble des nombres q tels que $| x - q | < 10^{-V_n}$. $O_n \subseteq O_{n-1}$ pour $n > 0$.

(6) $q_n \in O_n$ et pour tout $q \in O_n$, la fraction continue de q coïncide avec celle de x au moins jusqu'au rang n .

Preuve :

Pour un entier n, J_n est l'ensemble des nombres q tels que $|x_n - H_n(q)| < 10^{-m_n}$ et $N_n = \bigcap_{i=0}^{i=n} J_i$.

On raisonne par récurrence sur n en montrant conjointement la propriété : $O_n \subseteq N_n$.

Soit n = 0 .

Comme x n'est pas entier, il existe un entier m_0 tel que pour tout y vérifiant $|x - y| < 10^{-m_0}$, on ait $3 \leq c_{m_0}(y) \leq 10^{m_0} - 3$ (proposition C-6 (2)) .

Pour un entier $K_0 \geq 0$, on choisit $V_0 \geq K_0 + m_0$.
On choisit q_0 (éventuellement décimal) tel que : $|x - q_0| < 10^{-V_0}$. On a $|x - q_0| < 10^{-m_0}$.
Alors $3 \leq c_{m_0}(q_0) \leq 10^{m_0} - 3$. D'après la proposition C-6 (1), $E(q_0) = a_0$ ($q_0 \in O_0$ et $r_0 = q_0$) .

Pour tout $q \in O_0$, $E(q) = a_0$ (proposition C-6 (1)) . On a $O_0 \subseteq J_0 = N_0$

Soit n > 0 .

Supposons la suite (q_n) définie jusqu'au rang n-1 .

Soit $q \in O_{n-1}$. On a $O_{n-1} \subseteq N_{n-1}$.

La fraction continue de q coïncide avec celle de x au moins jusqu'au rang n-1 (proposition D-1).

On a $|x_{n-1} - r_{n-1}| < 10^{-m_{n-1}}$, $3 \leq c_{m_{n-1}}(r_{n-1}) \leq 10^{m_{n-1}} - 3$ et $|x_{n-1} - H_{n-1}(q)| < 10^{-m_{n-1}}$.

D'après la proposition D-3 , si $(-1)^{n-1}(q - b_{n-1}) > 0$, on a $|x_n - H_n(q)| < |x - q| \, 10^{K_n}$ (7) .

Comme x_n n'est pas entier, on peut trouver un entier m_n tel que
pour tout y vérifiant $|x_n - y| < 10^{-m_n}$, on ait $3 \leq c_{m_n}(y) \leq 10^{m_n} - 3$ (proposition C-6 (2)).

On choisit un entier $V_n \geq \max(K_n + m_n , V_{n-1})$. On a $O_n \subseteq O_{n-1}$.

Montrons que tout $q \in O_n$ appartient au domaine de définition de H_n .

$(-1)^{n-1}(q - b_{n-1}) = (-1)^{n-1}(x - b_{n-1}) + (-1)^{n-1}(q - x) = |x - b_{n-1}| + (-1)^{n-1}(q - x) \geq |x - b_{n-1}| - |q - x|$.

Notons que $C_{n-1} + D_{n-1} x_n < D_{n-1}(1+10^{m_{n-1}}/2) < D_{n-1} 10^{m_{n-1}}$ (voir preuve de la proposition D-3).
Alors $|x - b_{n-1}| > 1/[10^{m_{n-1}} (D_{n-1})^2]$ (proposition A-5). De plus $|q - x| < 1/10^{K_n} \leq 1/(D_{n-1} 10^{m_{n-1}})^2$.

D'où $(-1)^{n-1}(q - b_{n-1}) > 1/[10^{m_{n-1}} (D_{n-1})^2] - 1/(D_{n-1} 10^{m_{n-1}})^2 > 0$.

On choisit q_n (éventuellement décimal) dans O_n et on pose $r_n = H_n(q_n)$.
Utilisant (7), on obtient $|x_n - r_n| < 10^{-m_n}$. Alors $3 \leq c_{m_n}(r_n) \leq 10^{m_n} - 3$.

De même pour tout $q \in O_n$, on a $|x_n - H_n(q)| < 10^{-m_n}$. D'où $O_n \subseteq O_{n-1} \cap J_n \subseteq N_{n-1} \cap J_n = N_n$.

D'après la proposition D-1 la fraction continue de tout $q \in O_n$ coïncident avec celle de x au moins jusqu'au rang n .

Remarques

Le programme de calcul propose des valeurs croissantes de m_n jusqu'à ce que la condition $3 \leq c_{m_n}(r_n) \leq 10^{m_n} - 3$ soit vérifiée.
La proposition C-6 (2) assure qu'une valeur convenable de m_n sera obtenue après un nombre fini d'opérations.

Si x est rationnel, le résultat précédent est valable pour les valeurs de n inférieures au rang p du dernier terme de la fraction continue de x. Pour n=p , si $c_{m_p}(r_p) = 0$, $E(r_p) = a_p$, sinon $E(r_p) = a_p - 1$.

E- Précision d'amorçage

Soient x un nombre réel et E le nombre entier qui vérifie $10^E \le |x| < 10^{E+1}$ si x # 0.
E = -1 si x = 0 .

Pour une précision P, on pose t_P = bfloat(x) et $s_P = \boldsymbol{\sigma} \times \mathbf{m} \times 10^{\mathbf{e}+1-\mathbf{P}}$ où $\boldsymbol{\sigma}$ est le signe, **m** est la mantisse qui comporte P chiffres, **e** est l'exposant de bfloat(x). s_P est un nombre décimal.
Pour une valeur suffisante de P, e = E .
Avec Maxima, s_P = round(t_P*10^(P–e–1))*10^(e+1–P) .

On pose X =$|x| \times 10^{-E}$. Soit S_{P-1} le développement décimal de X limité au rang P-1 .
Si x est un nombre décimal on fait les hypothèses suivantes:
(1) Soit $|s_P| \times 10^{-E} = S_{P-1}$, soit $|s_P| \times 10^{-E} = S_{P-1} + 10^{1-P}$.
(2) Si P est supérieur au rang du dernier chiffre non nul du DD de X alors $|s_P| \times 10^{-E} = S_{P-1} = X$.

On déduit de la proposition C-1 :
- Si $|s_P| \times 10^{-E} = S_{P-1}$, alors $0 \le |x| - |s_P| < 10^{E-(P-1)}$.
- Si $|s_P| \times 10^{-E-} = S_{P-1} + 10^{1-P}$, alors $0 < |s_P| - |x| < 10^{E-(P-1)}$. La condition (2) exclut $|s_P| - |x| = 10^{E-(P-1)}$.
Dans tous les cas on a $|x - s_P| < 10^{E-(P-1)}$.

Si x est quelconque on obtient, au mieux $|x - s_P| < 10^{E-(P-1)}$ pour tout $P \ge 1$.
Si de plus $|x - s_P| \le (5/10)10^{E-(P-1)}$ l'arrondi du dernier chiffre se fait à l'unité la plus proche.

De façon générale, sous certaines conditions, il existe un entier $L \ge 1$ tel que:
$|x - s_P| < 10^{E-(P-L)}$ pour tout $P \ge L$. Si $P \ge L$ on a $E - 1 \le e \le E + 1$.

Définition

La précision d'amorçage de bfloat(x) est le plus petit entier L tel que $|x - s_P| < 10^{E-(P-L)}$
pour toute précision $P \ge L$.

Si x est le quotient de deux nombres entiers, la précision d'amorçage est estimée à 1 .

Estimation de la précision d'amorçage à l'aide de l'inégalité des accroissements finis

Fonction des variables élémentaires

L'expression de x peut comporter des opérations entre divers nombres comme des nombres entiers, des valeurs de fonctions élémentaires appliquées à des rationnels.
Ces nombres sont notés t_1 , t_2 , ... , t_n .

Alors x apparaît comme la valeur d'une fonction $(y_1,y_2, \dots ,y_n) \rightarrow F(y_1,y_2, \dots ,y_n)$.
au point $(t_1,t_2, \dots ,t_n)$ et $y_1,y_2, \dots ,y_n$ sont les variables élémentaires.

Pour une précision P on pose y_i = bfloat(t_i) dont on connait la précision d'amorçage.

Le calcul de $F(y_1,y_2, \dots ,y_n)$ donne une évaluation de bfloat(x) .

Inégalité des accroissements finis dans le cas de deux variables

Soient (a,b) un point de $\mathbf{R}^2$ et U un ouvert de $\mathbf{R}^2$ contenant (a,b).
Soit (y,z) $\rightarrow$ F(y,z) une fonction numérique de classe C^1 sur U et dont les dérivées partielles de F sont bornées sur U .
Soient A et B des nombres positifs tels que, pour tout $M \in U$, $|\partial F/\partial y\ (M)| \le A$ et $|\partial F/\partial z(M)| \le B$.
Soient h et k des réels tels que (a+h,b+k) appartienne à U .

Alors $|F(a+h,b+k) - F(a,b)| \le A|h| + B|k|$

Remarque

L'inégalité des accroissements finis donne une estimation de L qui ne tient pas compte des arrondis de la virgule flottante à chaque étape du calcul de bfloat(x). Cependant, la majoration obtenue par l'inégalité des accroissement finis est souvent large. Si le nombre d'étapes du calcul de bfloat(x) n'est pas trop important, on obtient ainsi une bonne estimation de L.

Exemple

$x = \sqrt{\sqrt{2} - 1414/10^3}$
On considère la variable élémentaire $t = \sqrt{2}$. On pose $a = 1414/10^3$.
Pour une précision $P \geq 4$, a est une constante pour la virgule flottante .
On admet que la précision d'amorçage de bfloat(t) est égale à 1 .
On vérifie $14142\ 10^{-4} < t < 14143\ 10^{-4}$, $a+2\ 10^{-4} < t < a+3\ 10^{-4}$, $2\ 10^{-4} < t-a < 3\ 10^{-4}$. D'où E=-2.

On pose $F(y) = \sqrt{(y-a)}$. Alors $F'(y) = -1/(2\sqrt{(y-a)})$. F est de classe C^1 sur $]a,+\infty[$.

La fonction $y \rightarrow |F'(y)|$ est décroissante et bornée sur l'intervalle $U =]a+\alpha,+\infty[$ où $\alpha = 10^{-4}$.
On vérifie que t appartient à U.

Pour une précision P, on a $|t - bfloat(t)| < 10^{0-(P-1)}$.
Pour $P \geq 5$ on a $bfloat(t) > t - 10^{-4} > a + 10^{-4} = a+\alpha$.
Donc, pour tout $P \geq 5$, bfloat(t) appartient à U .

Comme $y \rightarrow |F'(y)|$ est décroissante sur U, on a $|F'(y)| \leq |F'(a+\alpha)| = 50$.

L'inégalité des accroissements finis s'écrit :

$$|F(t) - F(bfloat(t))| \leq A\,|t - bfloat(t)| < 50\,|t - bfloat(t)| < 10^{-(P-3)} = 10^{E-(P-3+E)}$$

Comme E = -2, L = 5 est une estimation convenable de la précision d'amorçage de bfloat(x).

```
(%i2) CFL(sqrt(sqrt(2)-1414/10^3),100,10);
E=-2
L=4
T=1
(%o2) done
```

Perte de précision par arrondi dans un calcul intermédiaire de la virgule flottante

La virgule flottante procède par étapes élémentaires au cours desquelles elle effectue un seul arrondi. Elle détermine ainsi les valeurs $bfloat(z_1)$, $bfloat(z_2)$, ... , $bfloat(z_q)$ associées à une suite de nombres $z_1, z_2, \ldots, z_q$ pour obtenir finalement $bfloat(x) = bfloat(z_q)$.
Chaque opération de la virgule flottante s'accompagne d'une perte de précision par arrondi.

Proposition E-1

Soit d la valeur approchée décimale de z_i calculée pour une précision $P > M > 0$ par la virgule flottante avant arrondi telle que $|z_i - d| < 10^{E_i-(P-M)}$ où E_i est l'entier qui vérifie $10^{E_i} \leq |z_i| < 10^{E_i+1}$.
Soit σ_i le nombre décimal défini par $bfloat(z_i)$.

Alors $|z_i - \sigma_i| < 10^{E_i-(P-M-\alpha)}$ où $\alpha = \log_{10}(1+10^{2-M})$.
α est la perte de précision par arrondi. Elle ne dépend pas de la précision appliquée.
Pour $M = 1$, $\alpha \approx 1$. Pour $M = 16$, $\alpha \approx 4{,}3\ 10^{-15}$.

Preuve

Soit e l'entier qui vérifie $10^e \le |d| < 10^{e+1}$. On a $E_i - 1 \le e \le E_i + 1$.

On a : $|d - \sigma_i| < 10^{e-(P-1)}$. D'où: $|z_i - \sigma_i| \le |z_i - d| + |d - \sigma_i| < 10^{Ei-(P-M)} + 10^{e-(P-1)} \le 10^{Ei-(P-M)}(1+10^{2-M})$.
On pose $\alpha = \log_{10}(1+10^{2-M})$. D'où : $|z_i - \sigma_i| < 10^{Ei-(P-M-\alpha)}$.

Comme dans l'exemple précédent, on peut montrer que la précision d'amorçage est définie pour les expressions de x dont la fonction des variables élémentaires est de classe C^1 au voisinage de $(t_1, t_2,..., t_n)$, sachant que la perte de précision par arrondi de la virgule flottante ne dépend pas de la précision appliquée mais seulement du nombre d'étapes intermédiaires dans le calcul de bfloat(x).

Soit x = log(27)-3*log(3). On peut observer un échec avec CFL(exp(1)+x^(1/3),100), alors que CFL(exp(1)+x^(4/3),100) donne L = 1.
Ceci est dû au fait que la dérivée de la fonction $t \to t^{1/3}$ tend vers l'infini quand t tend vers 0.

Perte de précision dans le calcul de u/v par le programme CFI(x,n) pour x > 0 et n > 1

Les étapes élémentaires du calcul de u/v par la virgule flottante peuvent être décrite comme suit : Calcul de u , calcul de v, calcul de u/v.

Dans le calcul de v on fait l'hypothèse que la virgule flottante effectue deux arrondis :
un dans le calcul de la valeur approchée de $z_1 = D\,x$ et un autre dans celui de la valeur approchée de de $z_2 = B - z_1$. (Il se peut qu'elle n'effectue qu'un seul arrondi).

Le calcul de u, valeur approchée de $z_0 = C\,x - A$, est analogue au cas de v.

Dans le calcul de la valeur approchée de $z_3 = z_0/z_2$, la virgule flottante effectue un seul arrondi.

Valeurs des paramètres

$L \ge 16$, $Q = L+E+2$, $C_{n-1} = C$, $D_{n-1} = D$, $K_n = 2\,g_{n-1}+2\,m_{n-1}$.

On rappelle que la suite (D_n) est croissante.

g_{n-2} et g_{n-1} sont les entiers qui vérifient $10^{g_{n-2}-1} < C \le 10^{g_{n-2}}$ et $10^{g_{n-1}-1} < D \le 10^{g_{n-1}}$. On a $g_{n-2} \le g_{n-1}$.

E , E_0 , E_1 , E_2 , E_3 sont les entiers qui vérifient :

$10^E \le |x| < 10^{E+1}$, $10^{E_0} \le |z_0| < 10^{E_0+1}$, $10^{E_1} \le |z_1| < 10^{E_1+1}$, $10^{E_2} \le |z_2| < 10^{E_2+1}$, $10^{E_3} \le |z_3| < 10^{E_3+1}$.

On a $P \ge Q + K_n + m_n = L+E+2+2\,g_{n-1}+2\,m_{n-1} + m_n$.

Si $E < 0$, $D \ge D_1 = a_1 = E(1/x)$. D'où $g_{n-1} > -E - 1$.

Alors $E+1+g_{n-1} > 0$ quel que soit le signe de E .

Perte de précision dans le calcul de z_1

On a $E+g_{n-1}-1 \le E_1 \le E+g_{n-1}$. D'où $0 \le E+g_{n-1}-E_1 \le 1$.

$|z_1 - D\,s_P| = D\,|x - s_P| < D\,10^{E-(P-L)} \le 10^{E+g_{n-1}-(P-L)} = 10^{E_1-(P-L-E-g_{n-1}+E_1)}$

On a $P \ge L+E+2+2\,g_{n-1}+2\,m_{n-1} + m_n > L+1 \ge L+E+g_{n-1}-E_1 = M_1 \ge L \ge 16$.

L'arrondi donne lieu à une perte de précision inférieure à $\alpha_1 = \log_{10}(1+10^{-14})$ (proposition E-1).
$|z_1 - \sigma_1| \le 10^{E+g_{n-1}-(P-L-\alpha_1)}$ où σ_1 est le nombre décimal défini par bfloat($D\,s_P$).

Perte de précision dans le calcul de z_2

$|z_2 – (B – \sigma_1)| = | z_1 - \sigma_1 | < 10^{E+g_{n-1}-(P-L-\alpha_1)}$. $|z_2| = |B – D\ x| = 1/(C+D\ x_n)$.

En utilisant les éléments de la démonstraion de la proposition D-3, on obtient $x_n < 10^{m_{n-1}}/2$,

puis $1/(D\,10^{m_{n-1}}) < |z_2| < 1/D$.

D'où $-g_{n-1}–m_{n-1} \le E_2 \le –g_{n-1}$ et $g_{n-1} \le –E_2 \le g_{n-1}+m_{n-1}$

$|z_2 – (B – \sigma_1)| < 10^{E_2-(P-L-E-g_{n-1}+E_2-\alpha_1)}$.

On a $P \ge L+E+2 + 2g_{n-1}+2m_{n-1} + m_n > L+E+g_{n-1} –E_2 + \alpha_1 = M_2 > L–1 \ge 15$.

L'arrondi donne lieu à une perte de précision inférieure à $\alpha_2 = \log_{10}(1+10^{-13})$

$|z_2 – \sigma_2| < 10^{E+g_{n-1}-(P-L-\alpha_1-\alpha_2)}$ où σ_2 est le nombre décimal défini par bfloat(B–σ_1).

$10^{E_2-1} \le |\sigma_2| < 10^{E_2+2}$.

Perte de précision dans le calcul de z_0

La perte de précision dans le calcul de u, valeur approchée de $z_0 = C\ x – A$, est analogue au cas de v.

$|z_0 – \sigma_0| < 10^{E+g_{n-2}-(P-L-\alpha_1-\alpha_2)}$ (pour n = 2, $|z_0 – \sigma_0| = |x – s_p| < 10^{E-(P-L)}$).

Perte de précision dans le calcul de z_3

Comme $10^{-E2-2} < 1/|\sigma_2| \le 10^{-E2+1}$ et $|z_0|/|z_2| < 10^{E3+1}$, il vient:

$|z_0/z_2 – \sigma_0/\sigma_2| \le (|z_0|/|z_2|)\ |z_2– \sigma_2|/|\sigma_2| + |z_0– \sigma_0|/|\sigma_2| \le 10^{E3–E2+2}\ |z_2– \sigma_2| + 10^{-E2+1}|z_0– \sigma_0|$.

$|z_2/z_0 – \sigma_2/\sigma_0| < 10^{E3-(P-L-E+E2-2-gn-1-\alpha1-\alpha2)} + 10^{E3-(P-L-E+E2-1+E3-gn-2-\alpha1-\alpha2)}$.

Or $E_3 \ge 0$ et $g_{n-2} \le g_{n-1}$. D'où $|z_2/z_0 – \sigma_2/\sigma_0| < (1+10^{-1})10^{E3-(P-L-E+E2-2-gn-1-\alpha1-\alpha2)}$.

D'où $|z_3 – \sigma_2/\sigma_0| \le 10^{E3-(P-L-E+E2-2-gn-1-\beta-\alpha1-\alpha2)}$ où $\beta = \log_{10}(1+10^{-1})$.

On a: $P \ge L+E+2 + 2g_{n-1}+2m_{n-1} + m_n > L+E–E_2+2+g_{n-1}+\beta+\alpha_1+\alpha_2 = M_3 > L+1 \ge 17$.

L'arrondi donne lieu à une perte de précision inférieure à $\alpha_3 = \log_{10}(1+10^{-15})$.

D'où $|z_3 – \sigma_3| < 10^{E_3-(P-L-E+E_2-2-g_{n-1}-\beta-\alpha_1-\alpha_2-\alpha_3)} = 10^{-(P-L-E-E_3+E_2-2-g_{n-1}-\beta-\alpha_1-\alpha_2-\alpha_3)}$ où σ_3 est le nombre décimal défini par bfloat(σ_2/σ_0) .

Il faut encore vérifier que la condition $|z_3 – \sigma_3| < 10^{-m_n}$ est bien satisfaite.

Comme $x_n < 10^{mn-1}/2$, on a $E_3 \le m_{n-1} – 1$. D'où $–E_3 \ge –m_{n-1} + 1$. De plus $E_2 \ge –g_{n-1}–m_{n-1}$.

$P–L–E–E_3+E_2–2–g_{n-1}–\beta–\alpha_1–\alpha_2–\alpha_3 \ge L+E+2 + 2g_{n-1}+2m_{n-1}+m_n–L–E–E_3+E_2–2–g_{n-1}–\beta–\alpha_1–\alpha_2–\alpha_3$

$\ge 2g_{n-1}+2m_{n-1}+m_n–m_{n-1}+1–g_{n-1}–m_{n-1}–g_{n-1}–\beta–\alpha_1–\alpha_2–\alpha_3 \ge m_n+1–\beta–\alpha_1–\alpha_2–\alpha_3 > m_n$.

(Dans les autres cas la perte de précision est encore inférieure à $\alpha_1+\alpha_2+\alpha_3$. En particulier, elle est nulle pour n = 0).

Conclusion

Proposition E-2

Le calcul de u/v en virgule flottante s'accompagne d'une perte de précision inférieure à :
$\alpha_1+\alpha_2+\alpha_3 < 5\ 10^{-14}$.

La valeur de P_n estimée dans la partie théorique est suffisante pour absorber les pertes de précision dues au calcul en virgule flottante.

F- Régularité de la virgule flottante sur un intervalle

On se propose de donner une méthode d'estimation de la précision d'amorçage de bfloat(x).
s_n est le nombre décimal déterminé par bfloat(x) calculé pour la précision n .
L est la précision d'amorçage de bfloat(x).

Pour $n \geq L$, on a $|x - s_n| < 10^{E-(n-L)}$. On vérifie facilement:

Proposition F-1

Soit [A,B] un intervalle de **N** tel que $A \geq L$.

Alors, pour tout $n \in [A,B]$, on a $|x - s_n| < 10^{E-(n-A)}$.

Dans ce qui suit on ne suppose plus $A \geq L$.

Définition

On dit que la virgule flottante est régulière sur [A,B] ,
si pour tout $n \in [A,B]$, on a $|x - s_n| < 10^{E-(n-A)}$.

Pour exprimer cette propriété, on dira simplement que l'intervalle [A,B] est régulier.

Si la virgule flottante est régulière sur [A,B] , elle l'est sur tout intervalle $[C,D] \subset [A,B]$.

Proposition F-2

(1) La virgule flottante est régulière sur $[A,+\infty[$ si et seulement si $A \geq L$.

(2) Le nombre d'intervalles réguliers, [A,B] vérifiant $A < L$ est fini .

Preuve

(1) Si $A \geq L$, $[A,+\infty[$ est régulier (proposition F-1).
Si $[A,+\infty[$ est régulier, pour tout entier $n \geq A$, on a $|x - s_n| < 10^{E-(n-A)}$.

D'après la définition de la précision d'amorçage, $A \geq L$.

(2) Soit **G** l'ensemble des entiers B pour lesquels l'intervalle [A,B] est régulier et vérifie $A < L$.
Si **G** est vide, la propriété est évidente. Supposons **G** non vide, et montrons que **G** est borné.

En effet, si cet ensemble n'était pas borné, l'intervalle $[L-1,+\infty[$ serait régulier. ce qui contredit la propriété (1).

Soit M le plus grand élément de **G** . Tout intervalle régulier [A,B] vérifiant $A \leq L$ est inclus dans [1,M]. Leur nombre est donc fini.

Indice de régularité

Comme le nombre d'intervalles réguliers [A,B] vérifiant A < L , est fini, il en existe un qui a le plus grand nombre d'éléments.

Soit N_0 le nombre de ses éléments. En l'absence d'intervalles réguliers [A,B] vérifiant A < L , en particulier si L = 1, on pose $N_0 = 0$.

On pose $T = N_0 + 1$. T est appelé indice de régularité de bfloat(x). Notons que M < L + T -2 .

Proposition F-3

Soit [A,B] un intervalle régulier contenant au moins T éléments. Alors :

(1) $A \geq L$.

(2) Si de plus $A - 1 = 0$ ou si $A - 1 > 0$ et [A-1,B-1] n'est pas régulier, alors L = A .

Exemple:

Pour x=sin(sqrt(501))*cos(sqrt(301)) , les valeurs de bfloat(x) quand fpprec varie de 1 à 15 , sont obtenues à l'aide du programme: b(x,m,n):=(for i:m while i<=n do (fpprec:i,disp([i,bfloat(x)]))) $
On effectue: b(x,1,15);

```
(%i2) x=sin(sqrt(501))*cos(sqrt(301))$ b(x,1,15);
[1,-1.0b-1]
[2,-3.6b-2]
[3,-2.78b-2]
[4,-2.702b-2]
[5,-2.6946b-2]
[6,-2.69378b-2]
[7,-2.693871b-2]
[8,-2.6938616b-2]
[9,-2.69386106b-2]
[10,-2.693861195b-2]
[11,-2.6938611982b-2]
[12,-2.69386119745b-2]
[13,-2.693861197392b-2]
[14,-2.6938611973976b-2]
[15,-2.69386119739711b-2]
(%o2) done
```

Au vu de cette liste on peut conjecturer que la précision d'amorçage est 2 ou 3.
Considérant que s_{15} est un représentant convenable de x, on a E = -2 . On voit qu'il n'y a pas d'intervalle régulier [A,B] pour A = 1 .
Cherchons le plus grand intervalle régulier [2,B] inclus dans [1,15] .

Le programme suivant permet de déterminer le plus grand intervalle régulier [A,B] contenu dans [A,P]. Le programme s'arrête dés que l'inéquation abs(t-u) < 10^(E-(n-A) n'est plus vérifiée.
Pour une précision "suffisante" P , on pose t = bfloat(x). Pour une précision n, on pose u = bfloat(x).
On utilise le programme suivant où t joue le rôle de x et où u celui de s_n :

```
c(x,E,A,P):=(fpprec:P , t:bfloat(x), f:0 , F:1 ,
                for n:A while n <= P  and  f < F do
                        (p:n, fpprec:n, u:bfloat(x), f:abs(t-u), F:10^(E-(n-A)), disp([n,E-(n-A),f]))) $
```

```
(%i4) c(x,-1,2,15);
[2,-2,9.1b-3]
[3,-3,8.32b-4]
[4,-4,8.488b-5]
[5,-5,7.0333b-6]
[6,-6,8.19564b-7]
[7,-7,9.778887b-8]
[8,-8,4.3655746b-9]
[9,-9,1.33150024b-9]
(%o4) done
```

Ce calcul montre que [2,8] est régulier mais pas [2,9] . Certainement L est supérieur à 2.
Par ailleurs on peut vérifier que l'intervalle [3,15] est régulier. On conjecture L = 3 et T = 8 .

On effectue : CFL(x,100,10); . On obtient : $E = -2$ $L = 3$ $T = 8$

Remarque

La valeur de x n'est pas toujours accessible (cas des nombres irrationnels).
Nous allons remplacer x par s_P pour une valeur suffisante de P .

P-régularité

Soient des entiers A , B et P vérifiant $A \le B < P$.
Nous dirons que la virgule flottante est P-régulière sur un intervalle [A,B] ou que l'intervalle [A,B] est P-régulier, si $|s_P - s_n| < 10^{E-(n-A)}$ pour tout entier $n \in [A,B]$.

Si la virgule flottante est P-régulière sur [A,B] , elle l'est sur tout intervalle $[C,D] \subset [A,B]$.

Proposition F-4

Soit x un nombre irrationnel et N un entier vérifiant $N \ge L + T$.
Soit **F** l'ensemble des intervalles réguliers [A,B] inclus dans [1,N] . Alors :

(1) Tout intervalle régulier [A,B] tel que $A < L$ est dans **F** .

(2) Il existe un entier P_0 tel que pour tout $P \ge P_0$, **F** soit l'ensemble des intervalles P-réguliers inclus dans [1,N] .

Preuve

(2) **F** est un ensemble fini.
Soit $[A,B] \in \mathbf{F}$. Soit $n \in [A,B]$ tel que $|x - s_n| < 10^{E-(n-A)}$.
Quand P tend vers l'infini, $|s_P - s_n|$ tend vers $|x - s_n|$.
Il existe un entier P_n tel que pour tout $P \ge P_n$, on ait $|s_P - s_n| < 10^{E-(n-A)}$.
Soit P_{AB} le maximum de P_n quand n parcourt [A,B] et P_F le maximum de P_{AB} quand [A,B] parcourt **F**. Alors , pour tout $P \ge P_F$, tout élément de **F** est P-régulier.

Soit **G** l'ensemble des intervalles inclus dans [1,N] qui ne sont pas réguliers. Cet ensemble est fini.
Soit $[A,B] \in \mathbf{G}$. Il existe $n \in [A,B]$ tel que $|x - s_n| > 10^{E-(n-A)}$ car x est non décimal.
Quand P tend vers l'infini, $|s_P - s_n|$ tend vers $|x - s_n|$.
Il existe un entier P'_{AB} tel que pour tout $P \ge P'_{AB}$ on ait $|s_P - s_n| \ge 10^{E-(n-A)}$.
Soit P_G le plus grand des entiers P'_{AB} quand [A,B] parcourt **G**.
Alors pour tout $P \ge P_G$, tout élément de **G** n'est pas P-régulier.

Soit $P_0 = \max(P_F,P_G)$. Pour tout $P \ge P_0$, **F** est l'ensemble des intervalles P-réguliers inclus dans [1,N].

Proposition F-5

Soient x un nombre irrationnel, N , L_0 et T_0 des entiers positifs tels que $T_0 \geq T$ et $N \geq L_0 + T_0$.
Soit P un entier tel que $P \geq P_0$, où P_0 est l'entier défini dans la proposition F-4.

(1) Si l'intervalle $[L_0, L_0+T_0-1]$ est P-régulier, alors $L_0 \geq L$.

(2) Si, de plus, $L_0 = 1$ ou si $[L_0-1, L_0+T_0-2]$ n'est pas P-régulier alors $L_0 = L$.

Remarque

Nous donnons des résultats complémentaires permettant une estimation plus large de L, plus facilement accessible que la valeur exacte de L sous les hypothèses de la proposition F-4.

Proposition F-6

Soit q un entier relatif tel que $e = E+q$. On suppose $P > B-A+L$.
On suppose que, pour tout $n \in [A,B]$ on a $|s_P - s_n| < 10^{e-(n-A)}$.

Si $q = -1$, alors [A,B] est régulier.

Si $q = 0$ et $B \geq A+1$, alors [A+1,B] est régulier.

Si $q = 1$ et $B \geq A+2$, alors [A+2,B] est régulier.

Preuve

Soit $n \in [A,B]$, on a $n-A-P+L \leq B-A-P+L \leq -1$.

Si $q = -1$ on a : $|x - s_n| \leq |x - s_P| + |s_P - s_n| < 10^{E-(P-L)} + 10^{e-(n-A)}$
$= 10^{E-(n-A)}[10^{n-A-P+L} + 1/10] \leq 10^{E-(n-A)}[2/10] < 10^{E-(n-A)}$.

Si $q \geq 0$ on a :
$|x - s_n| \leq |x - s_P| + |s_P - s_n| < 10^{E-(P-L)} + 10^{E0-(n-A)} = 10^{E-(n-(A+q))}[10^{n-A-q-P+L} + 1]$.

On a $n-A-q-P+L \leq -1-q \leq -1$.
Alors $|x - s_n| < 10^{E-(n-(A+q))}[10^{-1} + 1] < 10^{E-(n-(A+q+1))}$.

Proposition F-7

Soient e , L_0 , T_0 et P des entiers tels que $P > L_0+T_0 + 2$.
On suppose que pour tout $n \in [L_0, L_0+T_0-1]$ on a $|s_P - s_n| < 10^{e-(n-L0)}$.

Si $e = E-1$ et $T_0 \geq T$, alors $L_0 \geq L$.

Si $e = E$ et $T_0 \geq T+1$, alors $L_0 \geq L - 1$.

Si $e = E+1$ et $T_0 \geq T+2$, alors $L_0 \geq L - 2$.

Dans tous les cas, on a $L_0 + e + 1 \geq L + E$.

Preuve :

On applique les propositions F-3 et F-6 pour $A = L_0$ et $B = L_0+T_0-1$.

Remplacement de $|s_P - s_n|$ par $|t_P - t_n|$

Proposition F-8

Le remplacement de $|s_P - s_n|$ par $|t_P - t_n|$ calculé pour une précision $n < P$, peut engendrer une sous-évaluation de L d'une unité.

Preuve :

Soit F l'entier qui vérifie : $10^F \le |s_P - s_n| < 10^{F+1}$ $(s_P \neq s_n)$.

Calculé pour une précision $n < P$, $|t_P - t_n|$ est une estimation de bfloat($|s_P - s_n|$) calculé pour la précision n .

Soit σ_n le nombre décimal défini par $|t_P - t_n|$. $||s_P - s_n| - \sigma_n| < 10^{F-(n-1)} \le |s_P - s_n| 10^{1-n}$.

D'où $|s_P - s_n|(1 - 10^{1-n}) < \sigma_n < |s_P - s_n|(1+10^{1-n})$
et $\sigma_n /(1+ 10^{1-n}) < |s_P - s_n| < \sigma_n /(1-10^{1-n})$.
On pose $\alpha = -\log_{10}(1 - 10^{1-n})$. $\alpha > 0$.

$\sigma_n < 10^{E-(n-A)}$ entraîne $|s_P - s_n| < 10^{E-(n-(A+\alpha))}$.

Pour n = 16, $\alpha \approx 4{,}3\ 10^{-16}$.

La précision d'amorçage peut être sousestimée d'une unité.

Une unité suplémentaire a été ajoutée à la valeur de L estimée par le test de régularité.

G- Quelques curiosités

Fractions continus liées à l'exponentielle

```
(%i1) CFI(coth(1),20);
(%o1) [1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41]

(%i2) CFI(coth(1/2),20);
(%o3) [2,6,10,14,18,22,26,30,34,38,42,46,50,54,58,62,66,70,74,78,82]

(%i4) CFI(coth(1/10),20);
(%o5) [10,30,50,70,90,110,130,150,170,190,210,230,250,270,290,310,330,350,370,390,410]

(%i5) CFI(10^50*(exp(10^-50)-1),10);
(%o5) [1, 19999999999999999999999999999999999999999999999999, 2, 1,
          19999999999999999999999999999999999999999999999999, 4, 1,
          19999999999999999999999999999999999999999999999999, 6, 1]

(%i6) CFI(-10^50*(exp(-10^-50)-1),9);
(%o6) [0, 1, 19999999999999999999999999999999999999999999999999, 1, 2,
            19999999999999999999999999999999999999999999999999, 1, 4,
            19999999999999999999999999999999999999999999999999, 1, 6]
```

```
(%i7) CFI(exp(1),1000);
                Evaluation took 0.3120 seconds (0.3120 elapsed) using 101.307 MB.
(%o7)
 [2, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1, 10, 1, 1, 12, 1, 1, 14, 1, 1, 16, 1, 1, 18, 1, 1, 20, 1, 1,
 22, 1, 1, 24, 1, 1, 26, 1, 1, 28, 1, 1, 30, 1, 1, 32, 1, 1, 34, 1, 1, 36, 1, 1, 38, 1, 1, 40, 1, 1,
 42, 1, 1, 44, 1, 1, 46, 1, 1, 48, 1, 1, 50, 1, 1, 52, 1, 1, 54, 1, 1, 56, 1, 1, 58, 1, 1, 60, 1, 1,
 62, 1, 1, 64, 1, 1, 66, 1, 1, 68, 1, 1, 70, 1, 1, 72, 1, 1, 74, 1, 1, 76, 1, 1, 78, 1, 1, 80, 1, 1,
 82, 1, 1, 84, 1, 1, 86, 1, 1, 88, 1, 1, 90, 1, 1, 92, 1, 1, 94, 1, 1, 96, 1, 1, 98, 1, 1, 100, 1, 1, 102, 1, 1,
104, 1, 1, 106, 1, 1, 108, 1, 1, 110, 1, 1, 112, 1, 1, 114, 1, 1, 116, 1, 1, 118, 1, 1, 120, 1, 1, 122, 1, 1,
124, 1, 1, 126, 1, 1, 128, 1, 1, 130, 1, 1, 132, 1, 1, 134, 1, 1, 136, 1, 1, 138, 1, 1, 140, 1, 1, 142, 1, 1,
144, 1, 1, 146, 1, 1, 148, 1, 1, 150, 1, 1, 152, 1, 1, 154, 1, 1, 156, 1, 1, 158, 1, 1, 160, 1, 1, 162, 1, 1,
164, 1, 1, 166, 1, 1, 168, 1, 1, 170, 1, 1, 172, 1, 1, 174, 1, 1, 176, 1, 1, 178, 1, 1, 180, 1, 1, 182, 1, 1,
184, 1, 1, 186, 1, 1, 188, 1, 1, 190, 1, 1, 192, 1, 1, 194, 1, 1, 196, 1, 1, 198, 1, 1, 200, 1, 1, 202, 1, 1,
204, 1, 1, 206, 1, 1, 208, 1, 1, 210, 1, 1, 212, 1, 1, 214, 1, 1, 216, 1, 1, 218, 1, 1, 220, 1, 1, 222, 1, 1,
224, 1, 1, 226, 1, 1, 228, 1, 1, 230, 1, 1, 232, 1, 1, 234, 1, 1, 236, 1, 1, 238, 1, 1, 240, 1, 1, 242, 1, 1,
244, 1, 1, 246, 1, 1, 248, 1, 1, 250, 1, 1, 252, 1, 1, 254, 1, 1, 256, 1, 1, 258, 1, 1, 260, 1, 1, 262, 1, 1,
264, 1, 1, 266, 1, 1, 268, 1, 1, 270, 1, 1, 272, 1, 1, 274, 1, 1, 276, 1, 1, 278, 1, 1, 280, 1, 1, 282, 1, 1,
284, 1, 1, 286, 1, 1, 288, 1, 1, 290, 1, 1, 292, 1, 1, 294, 1, 1, 296, 1, 1, 298, 1, 1, 300, 1, 1, 302, 1, 1,
304, 1, 1, 306, 1, 1, 308, 1, 1, 310, 1, 1, 312, 1, 1, 314, 1, 1, 316, 1, 1, 318, 1, 1, 320, 1, 1, 322, 1, 1,
324, 1, 1, 326, 1, 1, 328, 1, 1, 330, 1, 1, 332, 1, 1, 334, 1, 1, 336, 1, 1, 338, 1, 1, 340, 1, 1, 342, 1, 1,
344, 1, 1, 346, 1, 1, 348, 1, 1, 350, 1, 1, 352, 1, 1, 354, 1, 1, 356, 1, 1, 358, 1, 1, 360, 1, 1, 362, 1, 1,
364, 1, 1, 366, 1, 1, 368, 1, 1, 370, 1, 1, 372, 1, 1, 374, 1, 1, 376, 1, 1, 378, 1, 1, 380, 1, 1, 382, 1, 1,
384, 1, 1, 386, 1, 1, 388, 1, 1, 390, 1, 1, 392, 1, 1, 394, 1, 1, 396, 1, 1, 398, 1, 1, 400, 1, 1, 402, 1, 1,
404, 1, 1, 406, 1, 1, 408, 1, 1, 410, 1, 1, 412, 1, 1, 414, 1, 1, 416, 1, 1, 418, 1, 1, 420, 1, 1, 422, 1, 1,
424, 1, 1, 426, 1, 1, 428, 1, 1, 430, 1, 1, 432, 1, 1, 434, 1, 1, 436, 1, 1, 438, 1, 1, 440, 1, 1, 442, 1, 1,
444, 1, 1, 446, 1, 1, 448, 1, 1, 450, 1, 1, 452, 1, 1, 454, 1, 1, 456, 1, 1, 458, 1, 1, 460, 1, 1, 462, 1, 1,
464, 1, 1, 466, 1, 1, 468, 1, 1, 470, 1, 1, 472, 1, 1, 474, 1, 1, 476, 1, 1, 478, 1, 1, 480, 1, 1, 482, 1, 1,
484, 1, 1, 486, 1, 1, 488, 1, 1, 490, 1, 1, 492, 1, 1, 494, 1, 1, 496, 1, 1, 498, 1, 1, 500, 1, 1, 502, 1, 1,
504, 1, 1, 506, 1, 1, 508, 1, 1, 510, 1, 1, 512, 1, 1, 514, 1, 1, 516, 1, 1, 518, 1, 1, 520, 1, 1, 522, 1, 1,
524, 1, 1, 526, 1, 1, 528, 1, 1, 530, 1, 1, 532, 1, 1, 534, 1, 1, 536, 1, 1, 538, 1, 1, 540, 1, 1, 542, 1, 1,
544, 1, 1, 546, 1, 1, 548, 1, 1, 550, 1, 1, 552, 1, 1, 554, 1, 1, 556, 1, 1, 558, 1, 1, 560, 1, 1, 562, 1, 1,
564, 1, 1, 566, 1, 1, 568, 1, 1, 570, 1, 1, 572, 1, 1, 574, 1, 1, 576, 1, 1, 578, 1, 1, 580, 1, 1, 582, 1, 1,
584, 1, 1, 586, 1, 1, 588, 1, 1, 590, 1, 1, 592, 1, 1, 594, 1, 1, 596, 1, 1, 598, 1, 1, 600, 1, 1, 602, 1, 1,
604, 1, 1, 606, 1, 1, 608, 1, 1, 610, 1, 1, 612, 1, 1, 614, 1, 1, 616, 1, 1, 618, 1, 1, 620, 1, 1, 622, 1, 1,
624, 1, 1, 626, 1, 1, 628, 1, 1, 630, 1, 1, 632, 1, 1, 634, 1, 1, 636, 1, 1, 638, 1, 1, 640, 1, 1, 642, 1, 1,
644, 1, 1, 646, 1, 1, 648, 1, 1, 650, 1, 1, 652, 1, 1, 654, 1, 1, 656, 1, 1, 658, 1, 1, 660, 1, 1, 662, 1, 1,
664, 1, 1, 666, 1, 1]
```

Fraction continue associée au nombre d'or

```
(%i8) CFI((1+sqrt(5))/2,77);
(%o8)
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

(%i9) CFI((1000000000000+sqrt(1000000000000^2+4))/2,9);
(%o9)      [1000000000000, 1000000000000, 1000000000000, 1000000000000, 1000000000000,
             1000000000000, 1000000000000, 1000000000000, 1000000000000, 1000000000000]
```

Bibliographie:

(1) Beeler, M., Gosper, R.W. Et Schroeppel, R. HAKMEM.
MIT(Massachusetts Institute of Technology) AI Note 239, February 29th, 1972

(2) Jean Vuillemin
Exact Real Computer Arithmetic with Continued Fractions. 14-27 (1988)
Electronic Edition (ACMDL) BiB Tex

(3) M. Couchouron
Développement d'un réel en fractions continues. Université de Rennes 1

(4) Université Paris 7 – Denis Diderot. Année 2007/2008. Licence 2. MA 3. Compléments sur les séries. 1 Le *développement décimal* d'un nombre réel.

(5) Jean-Michel Muller
Arithmétique virgule flottante – 3 septembre 2013– CNRS-INRIA-ENS Lyon-Univ. Claude Bernard

Dominique Drux St Zacharie janvier 2017