Traduction de l'anglais au français d'un article de Knuth sur la numération imaginaire[1]

Un système de numération imaginaire

Donald E. Knuth, Case Institute of Technology, Cleveland, Ohio

Pendant des siècles, le système des nombres décimaux a régné en maître, sauf peut-être chez les Indiens Mayas, jusqu'à ce que l'avènement des ordinateurs numériques mette les systèmes binaires et octaux sous le feu des projecteurs. Cet article présente un autre système numérique qui peut s'avérer utile pour manipuler des nombres complexes sur des machines.

Le système numérique "quart-imaginaire" utilise le nombre imaginaire $2i$ (ou $2j$ pour les ingénieurs électriciens) comme base. C'est-à-dire que le nombre

$$d_5d_4d_3d_4d_3d_2d_1d_0 \cdot d_{-1}d_{-2}d_{-3}$$

est la représentation quart-imaginaire de

$$d_5(32i) + d_4(16) + d_3(-8i) + d_2(-4) + d_1(2i) + d_0 + d_{-1}\left(-\frac{i}{2}\right) + d_{-2}\left(-\frac{1}{4}\right) + d_{-3}\left(\frac{i}{8}\right)$$

Chaque nombre complexe peut être représenté d'une manière unique dans le système quart-imaginaire par les chiffres 0, 1, 2 et 3. Les nombres quart-imaginaires sont sans signe par exemple, le nombre "plus unité" $(+1)$ est représenté par

$$1.$$

et le nombre "moins l'unité" (-1) est représenté par

$$103.$$

"Plus i" $(+i)$ est le nombre

$$10.2$$

et "moins i" $(-i)$ est

$$0.2.$$

En général, soit $z = x + iy$ n'importe quel nombre complexe, et soit

$$A = \pm a_p a_{p-1} \cdots a_0 \cdot a_{-1} \cdots a_{-q}$$

et

$$B = \pm b_r b_{r-1} \cdots b_0 \cdot b_{-1} \cdots b_{-s}$$

des représentations de x et de y, respectivement, dans le système numérique quaternaire ($base + 4$). Les entiers q et s peuvent être infinis par convention. Cette représentation est bien entendu facilement obtenue à partir de la représentation du système binaire. Ajustez p et r en ajoutant un zéro non significatif, si nécessaire, afin qu'ils soient tous deux pairs ; et ajustez de la même manière q

[1] Référence : https://dl.acm.org/doi/10.1145/367177.367233.
Traduction : Denise Vella-Chemla, assistée de Google translate, février 2024.

et s pour qu'ils soient soit infinis, soit impairs.

Si A est positif, former une nouvelle représentation A' où

$$a_k{}' = \begin{cases} 1 + a_k, & k \text{ pair} \\ 4 - a_k, & k \text{ impair} \end{cases}$$

pour $k = p, p-1, \cdots, -q$. Alors A' ne contient que les chiffres 1, 2, 3, et 4. Formez maintenant A'' en remplaçant les quatre par des zéros et en abaissant les chiffres immédiatement à leur gauche de un. Si le chiffre à gauche du 4 était zéro, remplacez-le par 3 et augmentez le chiffre précédent de 1, et ce processus se termine finalement.

Si A est négatif, former une nouvelle représentation A' où

$$u_k{}^r = \begin{cases} 4 - a_k, & k \ \text{ pair} \\ 1 + a_k, & k \ \text{ impair} \end{cases}$$

pour $k = p, p-1, \cdots, -q+1$, et soit $a'_{??} = 1, a'_{-q} = a_{-q}$. Puis passer en A'' en supprimant les 4 qui peuvent apparaître, comme ci-dessus.

Par exemple, supposons que A soit $+1, 23013$. Alors A' vaut 2.24421 et A'' vaut $2, 23021$. Et pour $A = -1, 23013$, nous aurions $A' = 13.31133 = A''$. Notez que A'' est maintenant le nombre x exprimé dans le système de base *moins quatre*, car

$$-1.23013 = 3.01030 - 10.30103$$

dans le système quaternaire ; une étude de cet exemple indiquera la motivation de la routine de conversion.

Le processus pour passer de A à A'' est illustré dans l'organigramme (fig. 1). Il convient maintenant d'appliquer le même processus[2] à B pour obtenir B''. Le nombre final

$$z = d_u d_{u-1} \cdots d_0 . d_{-1} \cdots d_{-v}$$

où $u = 2 \max(p+2, r+2), v = 2 \max(q, s)$, est alors défini par

$$d_k = \begin{cases} a''_n, & k = 2n \\ b''_n, & k = 2n+1 \end{cases}$$

pour $k = u, u-1, \ldots, -v$; et c'est la représentation quart-imaginaire requise de z.

Addition et multiplication

L'utilité possible du nouveau système réside dans sa facilité de multiplication. Les règles d'addition et de multiplication sont identiques à celles du système quaternaire ordinaire, avec la variation que

[2]Des méthodes plus rapides pour le processus de conversion peuvent être obtenues de manière évidente en considérant des groupes de 4 consécutifs.

le report se fait en soustrayant un de la deuxième colonne au-dessus, plutôt qu'en ajoutant un à la colonne adjacente. La retenue, si nécessaire, se fait en ajoutant un dans la deuxième colonne au-dessus, plutôt qu'en soustrayant un de la colonne adjacente. Avec ces règles de base, quelques multiplications simples sont effectuées comme suit :

```
1)                        10.2   [i]
                          10.2   [i]
                          ----
                          1.00
                       102
                       -------
                       103.00   [-1]

2)                      113311   [5+10i]
                         10210   [8+2i]
                      --------
                       1133110
                     112222
                   1133110
                   ----------
                   0000321310   [20+90 i]
```

Il est clair qu'une notation à virgule flottante pourrait être conçue pour travailler avec des nombres quart-imaginaires. Le plus gros problème est de trouver un bon algorithme de division : la plupart des processus de division reposent fortement sur les propriétés d'ordre des nombres réels, et la division dans ce nouveau système semble assez difficile. Les exemples de division longue, ou d'extraction de racines carrés, semblent nécessiter un coup de génie lorsque des essais de diviseurs sont effectués.

Long division: [(1+5i)/(5−i) = i]

```
                   10.20
          -----------
  10301.2|000031.200
          103012
         -------
            1123 20
            1123 20
           --------
             |     0
```

Square root: [$\sqrt{-3-4i} = 1-2i$]

```
              1  0  3  1
            √00 00 11 21
             01
            1 03 00 11
  203       1 03 00 11
                  10 21
  1021            10 21
```

[$\sqrt{-3-4i} = 2i - 1$]

```
               1  1  3
            √ 00 11 21
               01
             1 03 11
  21               21
             1 13 30 21
  223        1 13 30 21
```

Le système bi-imaginaire

Nous pouvons tout aussi bien considérer des systèmes de nombres n-imaginaires[3], où la base est considérée comme $\sqrt{-n}$. Ici, n est tout entier supérieur à 1 et les chiffres nécessaires à la représentation sont $0, 1, ..., n-1$. Le développement ci-dessus, à l'exception des exemples, peut être étendu simplement en remplaçant 4 par n et 3 par $n-1$ partout où ils apparaissent, et en laissant B être la représentation de $y/\sqrt{n}$.

En particulier, la saveur du système binaire peut être conservée dans un système numérique imaginaire en prenant $n = 2$. Dans ce système numérique bi-imaginaire, les seuls chiffres utilisés sont 0 et 1, et le système obtenu est donc bien adapté à une intégration dans des circuits électroniques. Le problème de la division pourrait être résolu plus simplement dans un tel système, car seuls deux choix (soustraire une fois ou pas du tout) doivent être faits lors de la production de chaque chiffre du quotient. Malheureusement, le nombre i lui-même doit être représenté de manière non terminale dans le système bi-imaginaire, et ainsi les erreurs de troncature et d'arrondi peuvent être plus courantes en pratique que si le système quart-imaginaire était utilisé.

[3] L'idée d'étendre cette théorie aux nombres n qui ne sont pas des carrés parfaits a été suggérée à l'auteur par Dr Arthur Wouk.

Une autre propriété du système bi-imaginaire est son caractère d'unicité de la représentation : chaque nombre complexe a une et une seule représentation dans le système. Ce fait peut être appliqué, par exemple, pour donner des preuves plus "élégantes" de théorèmes mathématiques tels que la non-dénombrabilité des nombres complexes.

L'apparente incapacité à se diviser facilement peut être un obstacle suffisant pour rendre les systèmes n-imaginaires impossibles à intégrer dans n'importe quel matériel : voici donc un défi lancé au lecteur de trouver un algorithme de division bi-imaginaire. Si quelqu'un pouvait imaginer un tel schéma, il accélérerait certainement les manipulations de nombres complexes à l'intérieur des machines.

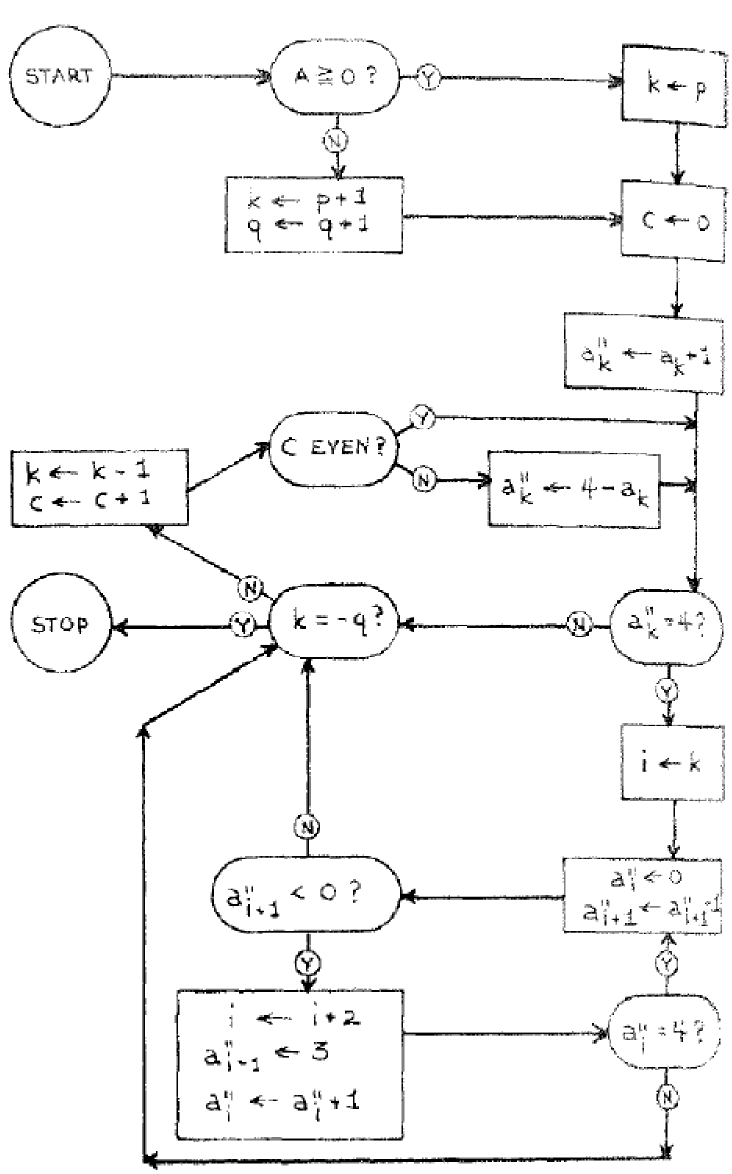

FIG. 1