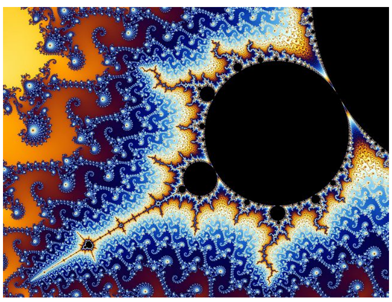
PT

Cours

Chapitre 1
Programmation récursive

Savoirs et compétences :

❒ *Alg – C15 : Récursivité : avantages et inconvénients.*



Courbe fractale de Mandelbrot [2]

1 Suites numériques 2

1.1 Suites définies explicitement 2

1.2 Suites définies par une relation de récurrence 2

2 Diviser pour régner 3

3 Courbes fractales 3

3.1 La courbe de Peano 3

Patrick Beynet

Xavier Pessoles

PT

Présentation

En mathématiques, en informatique, en biologie, mais aussi dans notre quotidien, nous faisons souvent face à des situations où un problème doit être résolu en utilisant une méthode de résolution qui est répétée plusieurs fois. Dans l'itération, cette méthode est appliquée par paliers de façon séquentielle. Dans la récursion, la méthode s'appelle elle-même. La récursion est si fondamentale qu'il n'est pas possible de l'éviter : l'auto-reproduction, qui constitue le fondement de toute vie, est un processus récursif.

1 Suites numériques

1.1 Suites définies explicitement

Une suite numérique peut dans certains cas être définie de manière explicite : $u_n = f(n)$.
La détermination du nième terme est alors aisée. Il suffit d'évaluer $f(n)$.

■ **Exemple** Puissances de 2 : *Comment évaluer le nombre 2 à la puissance n de manière explicite, $n \in \mathbb{N}$?* On définit de manière explicite la suite : $u_n = 2^n$.

■ **Pseudo Code**

P2_explicite : Calcul de la nieme puissance de 2 – Méthode explicite

Entrée :
- n, int : un nombre entier

Sortie :
- un nombre entier.

```
P2_explicite(n) :
    Retourner 2 ∧ n
```

■

Malheureusement, toutes les suites numériques ne peuvent pas être définies de manière explicite.

1.2 Suites définies par une relation de récurrence

Dans une suite définie de manière récurrente, il est possible de calculer le terme u_n de la suite en connaissant les termes précédents. Les égalités de la forme $u_n = f(u_{n-1})$, $u_n = f(u_{n-1}, u_{n-2})$, *etc.* s'appellent des relations de récurrence. Les égalités qui définissent les premiers termes d'une suite sont appelées des conditions de départ. Une suite recursive est donc définie par une relation de récurrence et une(des) condition(s) de départ. On reprend l'exemple 1 : puissances de 2. On définit de manière récursive la suite : $u_0 = 1$, $u_n = 2 \cdot u_{n-1}$. On propose de manière proche de la définition, un algorithme récursif.

On propose de manière proche de la définition, un algorithme récursif.

■ **Pseudo Code**

P2_explicite : Calcul de la nieme puissance de 2 – Méthode récursive

Entrée :
- n, int : un nombre entier

Sortie :
- un nombre entier.

```
P2_recursive(n) :
    Si n == 0 alors :
        Retourner 1
    Sinon :
        Retourner 2*P2_recursive(n − 1)
```

Il est aussi possible de proposer un algorithme itératif permettant d'aboutir au même résultat.

■ **Pseudo Code**

P2_iterative : Calcul de la nieme puissance de 2 – Méthode récursive

Entrée :
- n, int : un nombre entier

Sortie :
- un nombre entier.

P2_iterative(n) :
    $x \leftarrow 1$
    tant que $n > 0$ **faire :**
        $x \leftarrow 2 * x$
        $n \leftarrow n - 1$
    fin tant que
    Retourner x

2 Diviser pour régner

3 Courbes fractales

3.1 La courbe de Peano

Références

[1] Patrick Beynet, *Supports de cours de TSI 2*, Lycée Rouvière, Toulon.

[2] « Mandel zool 08 satellite antenna ». Sous licence CC BY-SA via Wikimedia Commons - `https://fr.wikipedia.org/wiki/Ensemble_de_Mandelbrot#/media/File:Mandel_zoom_08_satellite_antenna.jpg`