

# Calculs en précision arbitraire

Très tôt dans l'histoire il semble que les mathématiciens aient été convaincus de l'existence d'un rapport constant entre le périmètre d'un cercle et son diamètre et aient cherché à approcher au plus près cette valeur. Au III<sup>e</sup> avant J.C ARCHIMÈDE, en considérant une succession de polygones inscrits et circonscrits au cercle unité obtient l'encadrement célèbre :  $3 + \frac{10}{71} < \pi < 3 + \frac{10}{70}$ . À l'aide de la même méthode, AL-KASHI obtient 16 décimales de  $\pi$  en 1424, et LUDOLPH 35 décimales en 1596. Dotés d'un ordinateur mais en suivant la même approche nous allons tenter de faire mieux.

## Calcul de $\pi$ par la méthode d'ARCHIMÈDE

Pour tout  $n \geq 2$  on note  $u_n$  le demi-périmètre d'un polygone à  $2^n$  côtés inscrit dans le cercle unité et  $v_n$  le demi-périmètre d'un polygone à  $2^n$  côtés circonscrit au cercle unité.

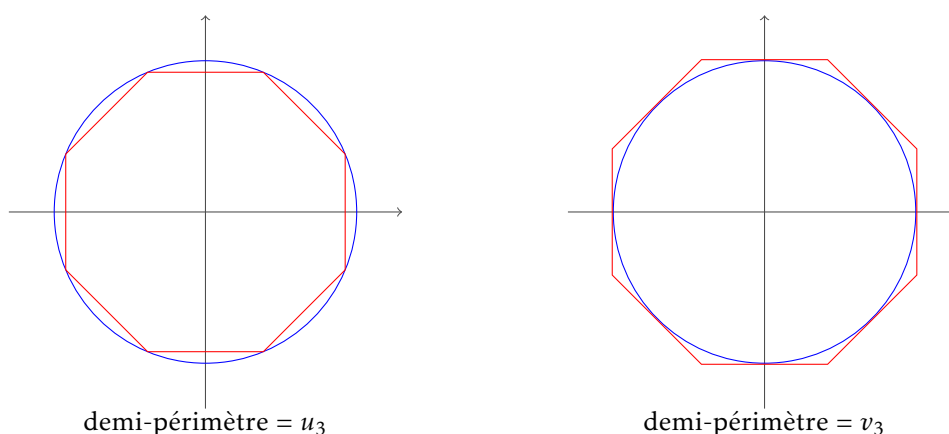


FIGURE 1 – Une représentation de l'octogone inscrit et de l'octogone circonscrit au cercle unité.

### Question 1.

- Montrer que  $u_n = 2^n \sin(\pi/2^n)$  et  $v_n = 2^n \tan(\pi/2^n)$  et déduire de ces formules que les suites  $u$  et  $v$  convergent vers  $\pi$  et vérifient :  $\forall n \in \mathbb{N}, u_n \leq \pi \leq v_n$  (on admettra la relation  $\sin x \leq x \leq \tan x$ , valable pour  $0 \leq x < \frac{\pi}{2}$ ).
- À l'aide des formules de duplication des arcs on prouve les relations suivantes (qu'on ne demande pas de démontrer) :

$$\forall n \geq 1, \quad u_{n+1} = \sqrt{u_n v_{n+1}} \quad \text{et} \quad v_{n+1} = \frac{2}{\frac{1}{u_n} + \frac{1}{v_n}} = \frac{2u_n v_n}{u_n + v_n}.$$

En utilisant ces relations ainsi que les valeurs initiales  $u_2 = 2\sqrt{2}$  et  $v_2 = 4$  rédiger une fonction PYTHON qui affiche les  $n$  premières valeurs des suites  $u$  et  $v$ .

- Utiliser cette fonction avec  $n = 40$ . À partir de quel rang devient-il inutile de poursuivre l'itération de ces suites ?

Si on veut obtenir plus de décimales de  $\pi$  on ne peut pas continuer à utiliser le type `float`. En revanche, PYTHON permet la manipulation d'entiers de taille arbitraire ; c'est pourquoi nous allons chercher à encadrer  $10^{200}\pi$  par deux entiers les plus proches possibles. Les chiffres que ces entiers auront en commun nous permettrons (en décalant la virgule de 200 chiffres vers la gauche) d'obtenir un certain nombre de décimales de  $\pi$ .

On définit quatre suites d'entiers  $a, b, c, d$  en posant :

$$a_2 = \lfloor \sqrt{8 \cdot 10^{400}} \rfloor, \quad b_2 = 4 \cdot 10^{200}, \quad c_2 = \lceil \sqrt{8 \cdot 10^{400}} \rceil, \quad d_2 = 4 \cdot 10^{200}$$

$$a_{n+1} = \lfloor \sqrt{a_n b_{n+1}} \rfloor, \quad b_{n+1} = \lfloor \frac{2a_n b_n}{a_n + b_n} \rfloor, \quad c_{n+1} = \lceil \sqrt{c_n d_{n+1}} \rceil, \quad d_{n+1} = \lceil \frac{2c_n d_n}{c_n + d_n} \rceil$$

**Question 2.** Montrer que pour tout  $n \geq 2$ ,  $a_n \leq 10^{200} u_n \leq c_n$  et  $b_n \leq 10^{200} v_n \leq d_n$ .

Pour pouvoir calculer les valeurs exactes des quatre suites entières  $a, b, c, d$  nous aurons besoin de quatre fonctions (définies sur  $\mathbb{N} \times \mathbb{N}$  ou sur  $\mathbb{N}$ ) :

$$\text{floor}(x, y) = \left\lfloor \frac{x}{y} \right\rfloor, \quad \text{ceil}(x, y) = \left\lceil \frac{x}{y} \right\rceil, \quad \text{fsqrt}(x) = \lfloor \sqrt{x} \rfloor, \quad \text{csqrt}(x) = \lceil \sqrt{x} \rceil.$$

**Question 3.** Définir les deux premières en PYTHON (sans recourir au type *float*).

### Calcul de la racine carrée de grands entiers

Pour calculer rapidement les deux autres fonctions, nous allons utiliser l'algorithme de HERON. Bien que datant du I<sup>er</sup> siècle après J.C cette méthode reste très efficace (vous apprendrez plus tard dans l'année qu'il s'agit d'un cas particulier d'une méthode plus générale de résolution d'une équation, la méthode de NEWTON-RAPHSON).

**Question 4.** Soit  $a \in \mathbb{N}^*$  et  $f : x \mapsto x^2 - a$ . On considère  $x_0 > \sqrt{a}$  et on note  $x_1$  l'abscisse du point d'intersection de la tangente à la courbe de  $f$  en  $x_0$  avec l'axe des abscisses.

- Montrer que  $\sqrt{a} < x_1 < x_0$  (faire un dessin, et étudier la position de la courbe par rapport à la tangente).
- Exprimer  $x_1$  en fonction de  $x_0$ , relation qu'on écrira :  $x_1 = g(x_0)$ , ce qui définit une fonction  $g$ .

On définit la suite  $r$  en posant  $r_0 = a$  et  $\forall n \in \mathbb{N}$ ,  $r_{n+1} = \lfloor g(r_n) \rfloor$ .

- Montrer qu'il existe un entier  $n$  tel que  $r_n \leq \sqrt{a}$ .
- Soit  $N$  le plus petit entier vérifiant cette inégalité. Montrer que  $r_N = \lfloor \sqrt{a} \rfloor$ .
- En déduire une fonction PYTHON nommée *fsqrt* calculant  $\lfloor \sqrt{a} \rfloor$  pour tout entier  $a$ .
- Comment modifier cette fonction pour calculer  $\lceil \sqrt{a} \rceil$ ? Rédiger la fonction *csqrt* correspondante.

### Calcul des décimales de $\pi$

**Question 5.** Lorsqu'on calcule les valeurs des suites  $a, b, c, d$  on observe que ces suites sont stationnaires. Si  $N$  désigne le rang à partir duquel elles stationnent, l'encadrement  $a_N \leq 10^{200} \pi \leq d_N$  est le meilleur que l'on puisse obtenir. Rédiger un script PYTHON qui détermine cette valeur de  $N$  ainsi que les valeurs correspondantes de  $a_N$  et de  $d_N$ . Combien de décimales exactes de  $\pi$  cet encadrement fournit-il ?

### Et pour ceux qui s'ennuient

Jadis on enseignait au collège une méthode d'extraction de racines carrées « à la main ». Pour calculer  $\lfloor \sqrt{n} \rfloor$  on utilisait le symbolisme de la division en disposant l'entier  $n$  à la place habituelle du nombre à diviser et on suivait l'algorithme ci-dessous :

- grouper les chiffres par tranches de 2, en partant de la droite ;
- chercher le plus grand entier  $a_1$  tel que  $a_1^2$  soit inférieur ou égal à la première tranche (en partant de la gauche). Ce chiffre est le premier de la racine carrée ; on l'écrit à la place usuelle du diviseur ;
- on calcule le premier reste, obtenu comme différence de la première tranche et de  $a_1^2$ . Il est écrit à la place usuelle des restes ;
- on abaisse la tranche suivante ;
- par essais successifs on cherche la plus grande valeur de  $i \in \llbracket 0, 9 \rrbracket$  telle que  $(20r + i) \times i$  soit inférieur au reste, où  $r$  est la racine provisoire, constituée des chiffres obtenus aux étapes précédentes. La valeur de  $i$  trouvée est le chiffre suivant de la racine, on l'écrit à la suite des autres ;
- on calcule le reste suivant, différence du reste précédent et de  $(20r + i) \times i$  pour la valeur de  $i$  trouvée à l'étape précédente ;
- tant qu'il reste des tranches on reprend à l'étape 4.

On trouvera figure 2 un exemple de calcul à la main de  $\lfloor \sqrt{71902} \rfloor$ .

### Question 6.

- Pour vous familiariser avec cette méthode, mettre en œuvre à la main l'algorithme proposé pour calculer  $\lfloor \sqrt{64015} \rfloor$ .
- Justifier la terminaison et la validité de cet algorithme, et en rédiger une version en PYTHON.
- Utiliser votre fonction pour calculer les 1000 premières décimales de  $\sqrt{2}$ .

$$\begin{array}{r|l}
 7.19.02 & 268 \\
 -4 & \textcircled{2} \times 2 = 4 \\
 \hline
 3.19 & 4 \textcircled{6} \times 6 = 276 \\
 -276 & 52 \textcircled{8} \times 8 = 4224 \\
 \hline
 43.02 & \\
 -4224 & \\
 \hline
 78 & 
 \end{array}$$

FIGURE 2 – Un exemple de calcul à la main d'une racine carrée :  $\lfloor \sqrt{71902} \rfloor = 268$ .