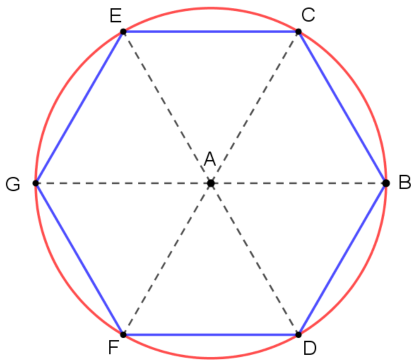
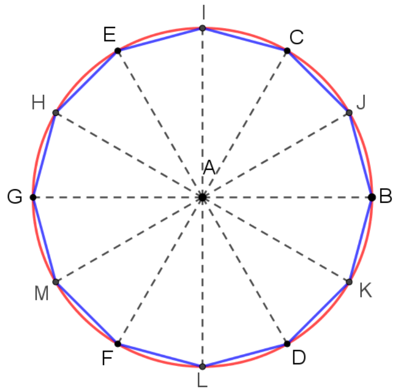
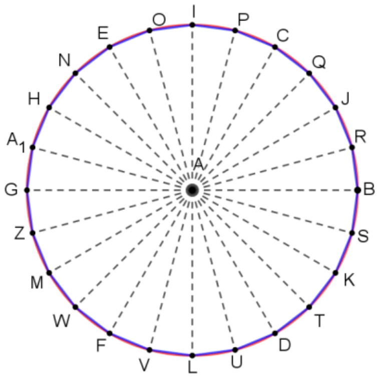
Calcul de Pi par Héron d'Alexandrie

Exercice 7D.1 Archimède

Sur la première figure est représenté un hexagone régulier dans un cercle.
A l'aide des bissectrices de chaque segment extérieur, on peut générer un dodécagone régulier dans ce cercle.
Ainsi de suite, sur la troisième figure, dans ce cercle est inscrit un polygone régulier possédant 24 côtés.
L'objectif de cet exercice est de déterminer l'évolution du périmètre en fonction du nombre de côtés.







1) Déterminer le périmètre de l'hexagone inscrit dans ce cercle de rayon égal à R.
2) Déterminer le périmètre du dodécagone inscrit dans ce cercle de rayon égal à R.
3) Déterminer le périmètre du polygone ayant 24 côtés inscrit dans ce cercle de rayon égal à R.
4) Ecrire un programme python qui donne le périmètre du polygone en fonction du nombre de cotés. (Les angles doivent être définis en radians)

Exercice 7D.2

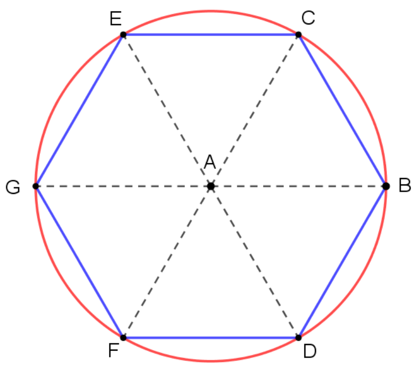
La méthode de Héron d'Alexandrie (Ier siècle) permet de calculer l'approximation d'une racine carrée d'un nombre strictement positif, à l'aide d'une suite convergeant vers cette valeur.

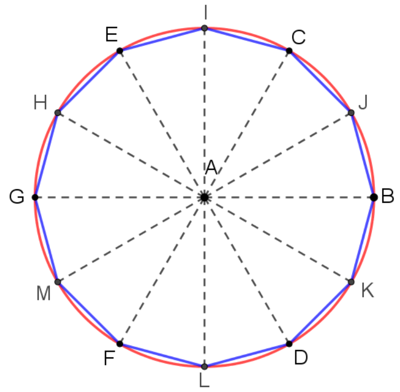
On commence par donner une valeur initiale strictement positive *a*, de préférence voisine de $\sqrt{A}$ et l'on cherche à encadrer la valeur $\sqrt{A}$ à l'aide de *a* et A.

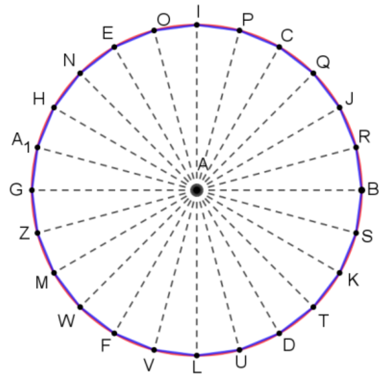
CORRIGE – Notre Dame de La Merci – Montpellier – M. Quet

Exercice 7D.1 **Archimède**

Sur la première figure est représenté un hexagone régulier dans un cercle.
A l'aide des bissectrices de chaque segment extérieur, on peut générer un dodécagone régulier dans ce cercle.
Ainsi de suite, sur la troisième figure, dans ce cercle est inscrit un polygone régulier possédant 24 côtés.
L'objectif de cet exercice est de déterminer l'évolution du périmètre en fonction du nombre de côtés.



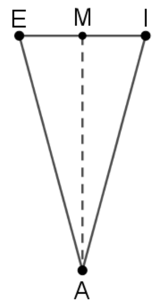




1) *Déterminer le périmètre de l'hexagone inscrit dans ce cercle de rayon égal à R.*
Tous les triangles sont équilatéraux, donc le périmètre mesure 6R.
Pour $R=1$, on obtient un périmètre égal à 6 (le périmètre du cercle étant égal à $2\pi \simeq 6{,}283$).

2) *Déterminer le périmètre du dodécagone inscrit dans ce cercle de rayon égal à R.*
L'angle au centre mesure $\frac{360}{12}=30°$ et tous les triangles intérieurs sont isocèles en A.

Soit M le milieu du segment $[EI]$: le triangle AME est rectangle en M :



$$\sin\left(\widehat{MAE}\right)=\frac{EM}{EA} \Leftrightarrow \sin 15=\frac{EM}{R} \Leftrightarrow EM=R\times\sin 15$$

Soit : $EM=R\times\sin\frac{30}{2}$

D'où : $EI=2R\times\sin\frac{30}{2}$

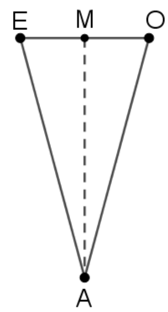
Le périmètre mesure :

$$12\times 2R\times\sin\frac{30}{2}=24R\times\sin\frac{30}{2}.$$

Pour $R=1$, on obtient un périmètre environ égal à 6,212.

3) Déterminer le périmètre du polygone ayant 24 côtés inscrit dans ce cercle de rayon égal à R.
L'angle au centre mesure $\frac{360}{24}=15°$ et tous les triangles intérieurs sont isocèles en A.

Soit M le milieu du segment $[EO]$: le triangle AME est rectangle en M :



$$\sin\left(\widehat{MAE}\right)=\frac{EM}{EA} \Leftrightarrow \sin 7{,}5=\frac{EM}{R} \Leftrightarrow EM=R\times\sin 7{,}5$$

Soit : $EM=R\times\sin\frac{30}{4}$

D'où : $EI=2R\times\sin\frac{30}{4}$

Le périmètre mesure :

$$24\times 2R\times\sin\frac{30}{4}=48R\times\sin\frac{30}{4}.$$

Pour $R=1$, on obtient un périmètre environ égal à 6,265.

4) *Ecrire un programme python qui donne le périmètre du polygone en fonction du nombre de cotés.*
A chaque division, le nombre de côtés double.
On peut demander à l'utilisateur le nombre de subdivisions de l'hexagone initial.
On peut définir une suite décrivant cette situation $\left(30° = \frac{\pi}{6} rad\right)$:

$$\forall n \in \mathbb{N} : u_n = 6 \times 2^n \times 2\text{R} \times \sin\left(\frac{\pi}{6} \times \frac{1}{2^n}\right) = 2^n \times 12\text{R} \times \sin\left(\frac{\pi}{6} \times \frac{1}{2^n}\right)$$

$$u_0 = 2^0 \times 12\text{R} \times \sin\left(\frac{\pi}{6} \times \frac{1}{2^0}\right) = 12\text{R} \times \sin\frac{\pi}{6} = 6\text{R}$$

$$u_1 = 2^1 \times 12\text{R} \times \sin\left(\frac{\pi}{6} \times \frac{1}{2^1}\right) = 24\text{R} \times \sin\frac{\pi}{12}$$

$$u_2 = 2^2 \times 12\text{R} \times \sin\left(\frac{\pi}{6} \times \frac{1}{2^2}\right) = 48\text{R} \times \sin\frac{\pi}{24}$$

On obtient :

```
from math import *
print("Le cercle étudié est de rayon égal à 1.")
n = int(input("Veuillez saisir le nombre de divisions:"))
print("Pour un polygone ayant",6*2**n,"côtés,")
print("on obtient un périmètre égal à",2**n*12*sin(pi/6*1/2**n))
```

```
Le cercle étudié est de rayon égal à 1.
Pour un polygone ayant 6 côtés,
on obtient un périmètre égal à 5.999999999999999

Le cercle étudié est de rayon égal à 1.
Pour un polygone ayant 12 côtés,
on obtient un périmètre égal à 6.211657082460498

Le cercle étudié est de rayon égal à 1.
Pour un polygone ayant 24 côtés,
on obtient un périmètre égal à 6.2652572265624755

Le cercle étudié est de rayon égal à 1.
Pour un polygone ayant 6291456 côtés,            # n = 20
on obtient un périmètre égal à 6.283185307179325
```

Exercice 7D.2

La méthode de Héron d'Alexandrie (Ier siècle) permet de calculer l'approximation d'une racine carrée d'un nombre strictement positif A, à l'aide d'une suite convergeant vers cette valeur.
On commence par donner une valeur initiale strictement positive *a*, de préférence voisine de $\sqrt{\text{A}}$ et l'on cherche à encadrer la valeur $\sqrt{\text{A}}$ à l'aide de *a* et A.
Deux cas sont à étudier :

- Soit $a < \sqrt{\text{A}} \Leftrightarrow \frac{1}{a} > \frac{1}{\sqrt{\text{A}}} \Leftrightarrow \frac{1}{a} \times \text{A} > \frac{1}{\sqrt{\text{A}}} \times \text{A} \Leftrightarrow \frac{\text{A}}{a} > \frac{\text{A}}{\sqrt{\text{A}}} \Leftrightarrow \frac{\text{A}}{a} > \sqrt{\text{A}}$

  →ainsi : $a < \sqrt{\text{A}} < \frac{\text{A}}{a}$

- Soit $a > \sqrt{\mathrm{A}} \Leftrightarrow \frac{1}{a} < \frac{1}{\sqrt{\mathrm{A}}} \Leftrightarrow \frac{1}{a} \times \mathrm{A} < \frac{1}{\sqrt{\mathrm{A}}} \times \mathrm{A} \Leftrightarrow \frac{\mathrm{A}}{a} < \frac{\mathrm{A}}{\sqrt{\mathrm{A}}} \Leftrightarrow \frac{\mathrm{A}}{a} < \sqrt{\mathrm{A}}$

  →ainsi : $\frac{\mathrm{A}}{a} < \sqrt{\mathrm{A}} < a$.

On en conclut que $\sqrt{\mathrm{A}}$ est toujours compris entre a et $\frac{\mathrm{A}}{a}$.

On réduit alors cet intervalle en prenant la moyenne arithmétique m des deux valeurs a et $\frac{\mathrm{A}}{a}$. On a alors :

$$m = \frac{1}{2}\left(a + \frac{\mathrm{A}}{a}\right).$$

On remplace alors la valeur de a par la valeur de m et on réitère le processus.
On construit une suite (u_n), suite des valeurs de a, définie sur $\mathbb{N}$ par :

$$\begin{cases} u_0 = a \\ u_{n+1} = \frac{1}{2}\left(u_n + \frac{\mathrm{A}}{u_n}\right) \end{cases}.$$

A et a étant strictement positif, il est immédiat que $\forall n \in \mathbb{N},\ u_n > 0$.

Montrons ensuite que $u_{n+1} > \sqrt{\mathrm{A}}$ pour $n > 1$: on compare u_{n+1} et $\sqrt{\mathrm{A}}$, ou mieux : $(u_{n+1})^2$ et A :

$$(u_{n+1})^2 - \mathrm{A} = \left[\frac{1}{2}\left(u_n + \frac{\mathrm{A}}{u_n}\right)\right]^2 - \mathrm{A} = \frac{u_n^{\ 4} + 2\mathrm{A}u_n^{\ 2} + \mathrm{A}^2}{4u_n^{\ 2}} - \mathrm{A} = \frac{u_n^{\ 4} + 2\mathrm{A}u_n^{\ 2} + \mathrm{A}^2 - 4\mathrm{A}u_n^{\ 2}}{4u_n^{\ 2}}$$

$$= \frac{u_n^{\ 4} - 2\mathrm{A}u_n^{\ 2} + \mathrm{A}^2}{4u_n^{\ 2}} = \left(\frac{u_n^{\ 2} - \mathrm{A}}{2u_n}\right)^2$$

Donc $\forall n \in \mathbb{N},\ (u_{n+1})^2 - \mathrm{A} > 0 \Leftrightarrow (u_{n+1})^2 > \mathrm{A} \Leftrightarrow u_{n+1} > \sqrt{\mathrm{A}}$

Ainsi : $\forall n \in \mathbb{N},\ u_n > \sqrt{\mathrm{A}}$

Montrons que (u_n) est décroissante à partir de $n=1$. Pour $n \in \mathbb{N}^*$, on a :

$$u_{n+1} - u_n = \frac{1}{2}\left(u_n + \frac{\mathrm{A}}{u_n}\right) - u_n = \frac{\mathrm{A}}{2u_n} - \frac{u_n}{2} = \frac{\mathrm{A} - u_n^{\ 2}}{2u_n}$$

Or $u_n > \sqrt{\mathrm{A}} \Leftrightarrow u_n^{\ 2} > \mathrm{A} \Leftrightarrow 0 > \mathrm{A} - u_n^{\ 2}$

Donc $u_{n+1} - u_n < 0$: (u_n) est décroissante

La suite (u_n) est décroissante et minorée par 0 à partir de $n=1$, d'après le théorème des suites monotones, la suite (u_n) converge.

La convergence est très rapide : à chaque itération, le nombre de décimales exactes est multiplié par deux!

```
def racine (A,n):
    i = 0
    while A>i**2:
        i+=1
    u=i
    for i in range (1,n+1):
        u=0.5*(u+A/u)
    return u
resultat=racine(5,10)
print(resultat)
```

→2.23606797749979 *PS tout est à reprendre, j'ai vu en Grand Oral une approche bien plus intuitive*