

Périmètre d'une ellipse

Asli Grimaud

Docteure et ingénieure en informatique

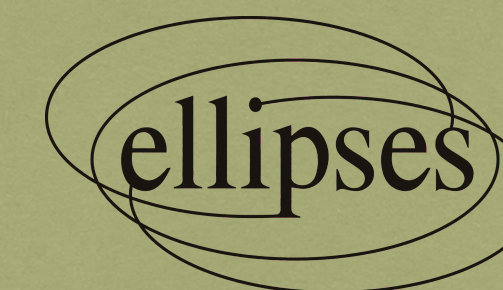
Professeure agrégée d'informatique (ex-mathématiques)

Professeure d'informatique en MP2I/MP1

Informatique

MP2I

*Cours complets, exercices corrigés et projets guidés
pour les classes préparatoires*



septembre 2025

Grimaud & Grimaud

Journées Académiques de l'IREM de Lille

28 Mars 2025

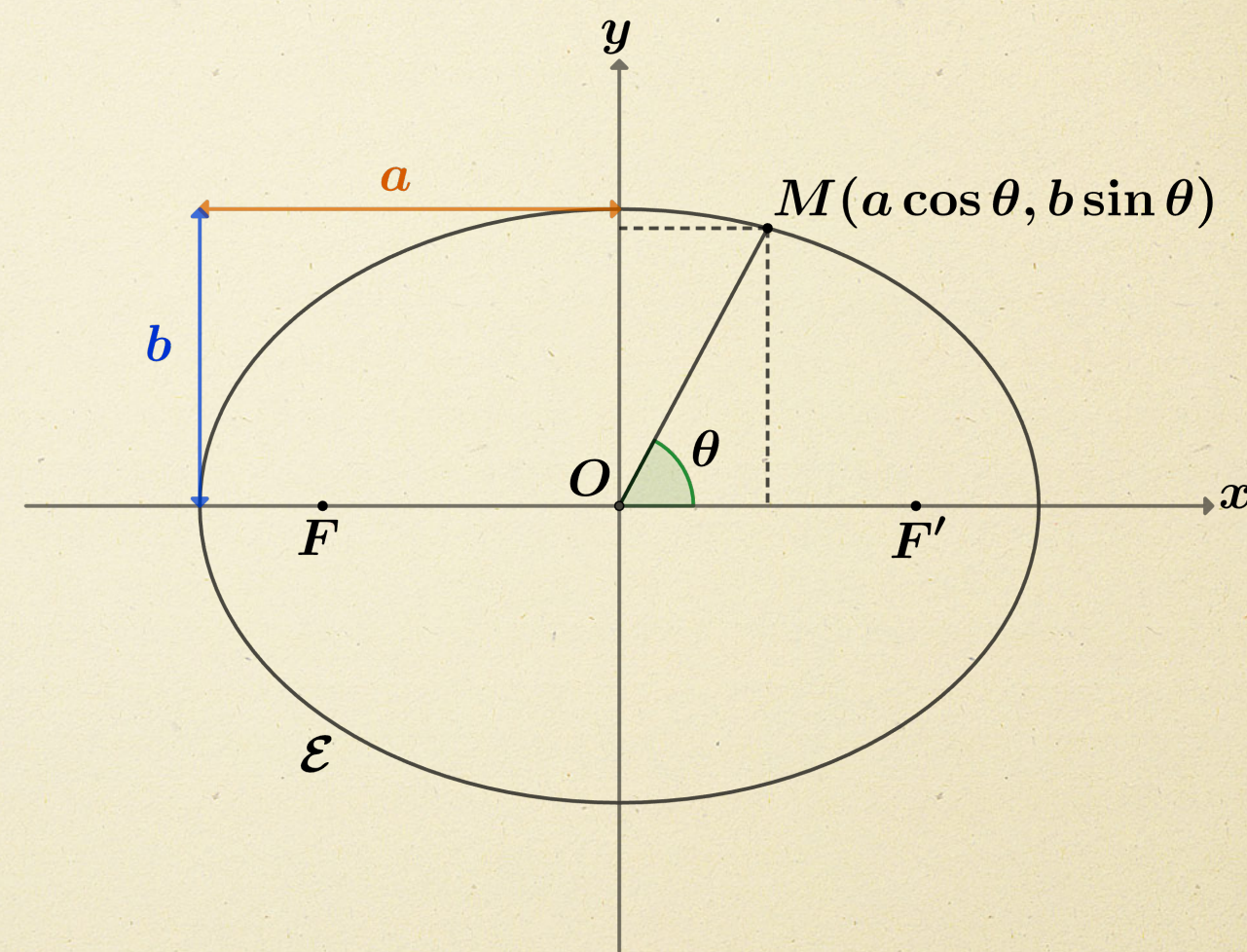
asli.grimaud@ac-lille.fr

Nombres flottants

$$\frac{1}{n} \times n \stackrel{?}{=} 1$$

multiplication binaire

Ellipse



addition binaire

- *programmation impérative*
- *programmation fonctionnelle*

Réels vs Nombres flottants

$$\frac{1}{n} \times n \stackrel{?}{=} 1$$

Mathématiques

Réels \mathbb{R}

notion abstraite

$$\forall n \in \mathbb{N}^*, \frac{1}{n} \times n = 1$$

Informatique

Nombres flottants

représentation des nombres réels

$$\forall n \in \mathbb{N}^*, \frac{1}{n} \times n \stackrel{?}{=} 1$$



Nombres flottants (8 octets)

$$\frac{1}{3} \times 3 = 1$$

$$\frac{1}{24} \times 24 = 1$$

$$\frac{1}{7} \times 7 = 1$$

$$\frac{1}{151} \times 151 = 1$$

$$\frac{1}{49} \times 49 \neq 1$$

$$\frac{1}{103} \times 103 \neq 1$$

$$\frac{1}{98} \times 98 \neq 1$$

$$\frac{1}{n} \times n \stackrel{?}{=} 1$$

Nombres flottants

Python : **float** (8 octets)

C : **float** (4 octets), **double** (8 octets)

OCaml : **float** (8 octets)

IEEE 754

précision simple « binary32 »

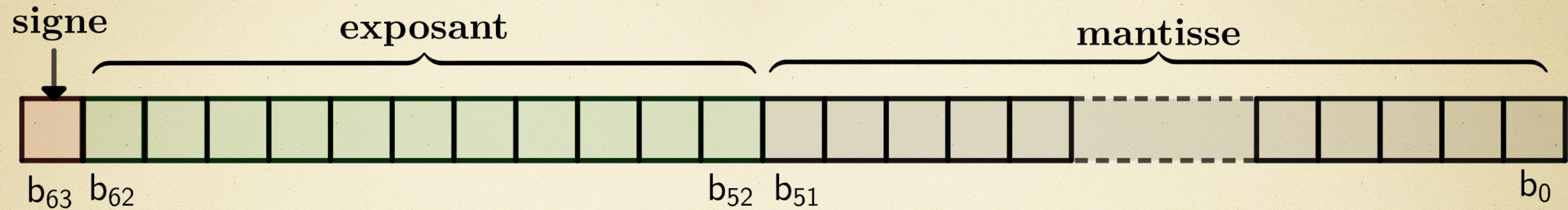
précision double « binary64 »

Notation scientifique d'un nombre réel en base 2

$$x = S \times 2^E \times M$$

- $S = \pm 1$ représente le signe ;
- $E \in \mathbb{Z}$ représente l'exposant ;
- $M \in [1; 2[$ représente la partie significative (en binaire) appelé la mantisse.

Nombres flottants (8 octets)



- Entier S ($s=1$ bit) : Le bit du poids plus fort est le signe du bit
- Entier E ($e=11$ bits) : L'exposant biaisé pour stocker sous forme d'entier non signé

$$1 \leq E' = E + 2^{e-1} - 1 \leq 2^e - 2$$

- Nombre M ($m=52$ bits) : La mantisse stockée par les premiers bits après la virgule
- Si $E' = M = 0$ alors $x = \pm 0$
- Si $E' = 2^e - 1$ et $M = 0$ alors $x = \pm \infty$
- Si $E' = 2^e - 1$ et $M \neq 0$ alors $x = NaN$

$$x = \frac{1}{3}$$

Nombres flottants (4 octets)

$$x = (-1)^0 \times 2^{-2} \times \frac{4}{3}$$

$$S = 1 \text{ puisque } x > 0$$

Le bit de signe vaut 0.

$$E = -2$$

$$E' = -2 + 127 = 125$$

$$M = \frac{4}{3} \in [1; 2[$$

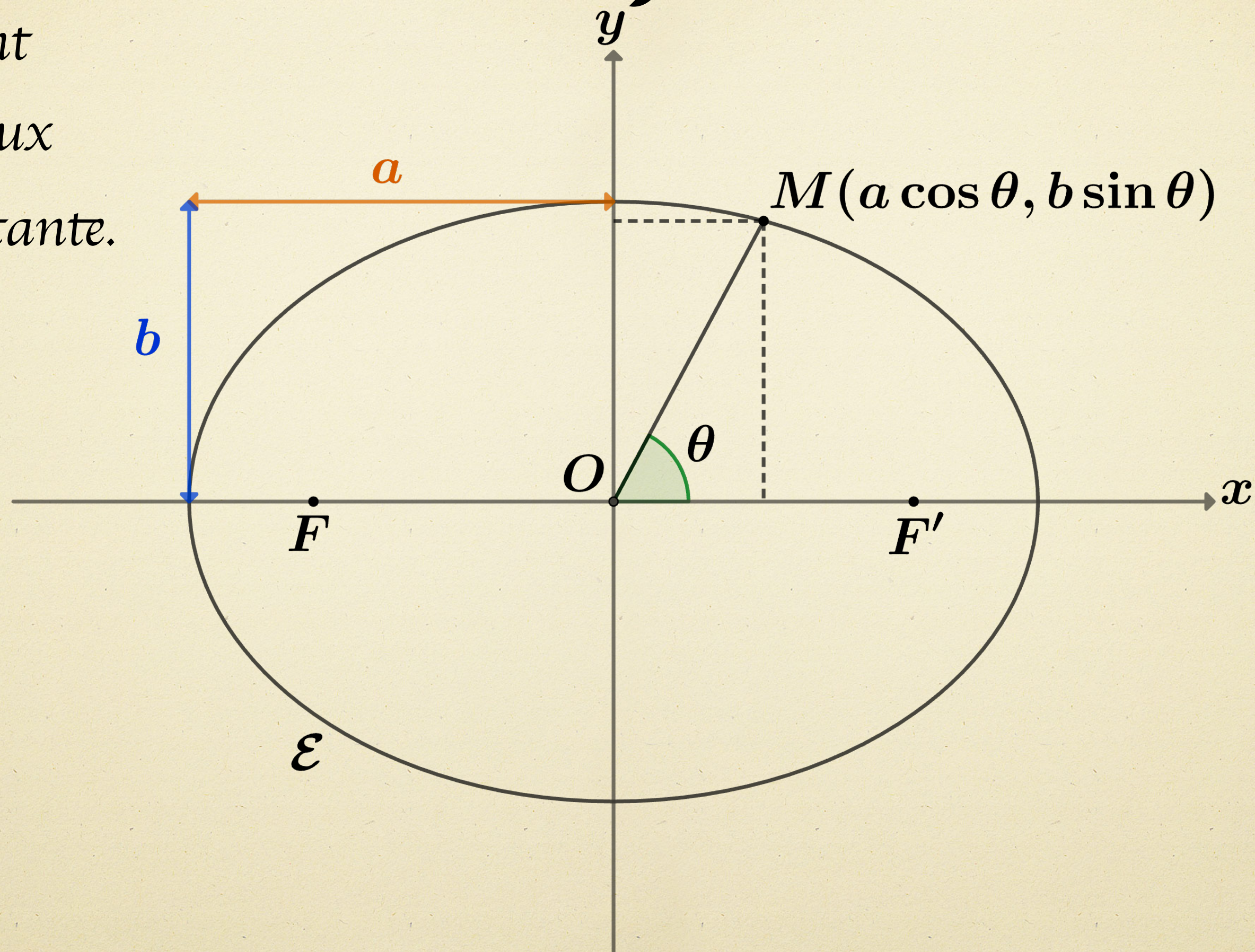
$$M = \frac{4}{3} = \frac{1}{1 - \frac{1}{4}} = \sum_{k=0}^{+\infty} \frac{1}{4^k} = \sum_{k=0}^{+\infty} \frac{1}{2^{2k}}$$

0 01111101 ~~01010101010101010101010101010101~~ 1010...
 0101010101010101010101010101011

0.3333333333333333333333331482962

Ellipse

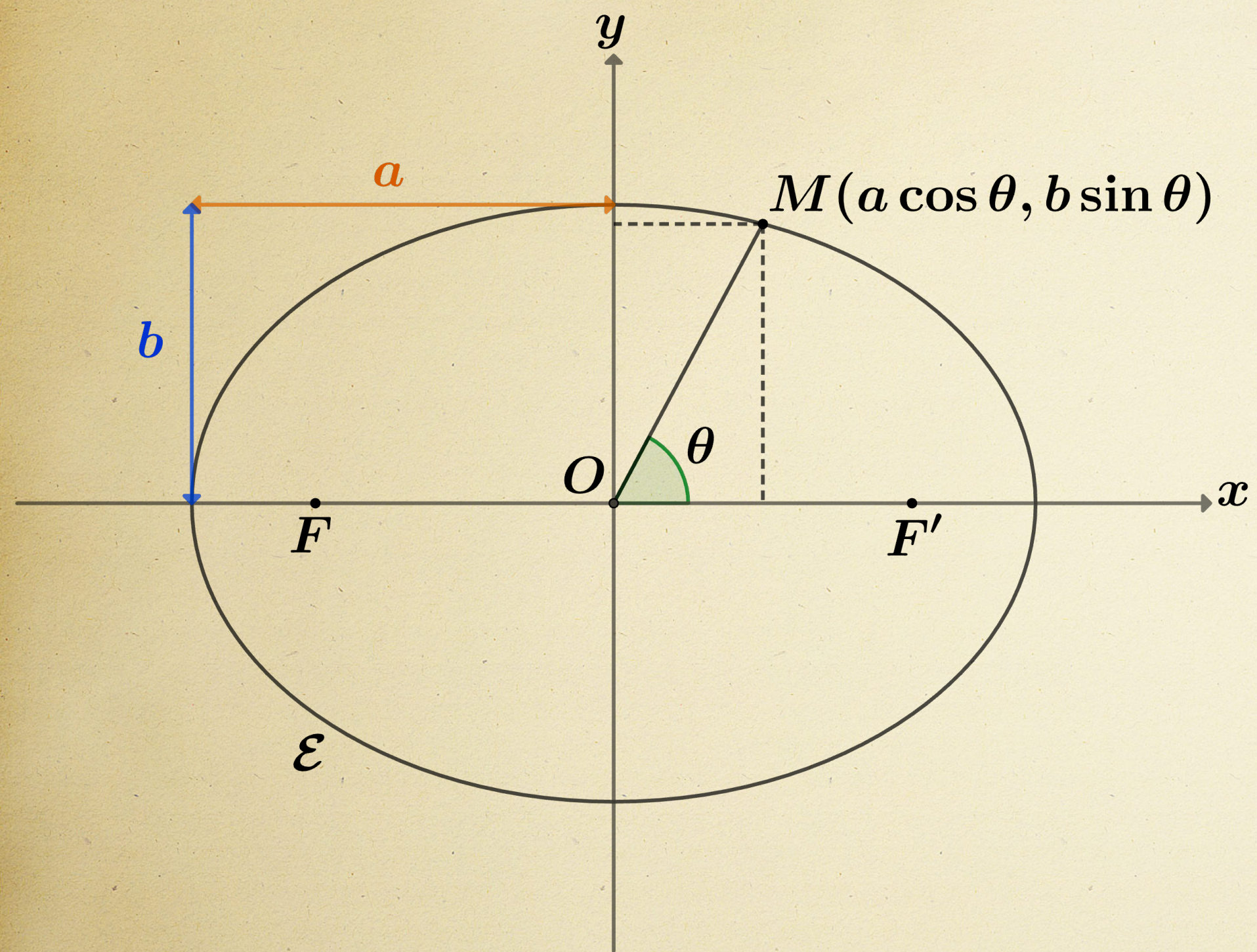
Ensemble des points dont
la somme des distances aux
deux foyers F et F' est constante.



$$a, b \in \mathbb{R}^{+*}$$

$$M(x, y) \in \mathcal{E} \Leftrightarrow \exists \theta \in [0, 2\pi[, x = a \cos \theta \text{ et } y = b \sin \theta$$

Ellipse



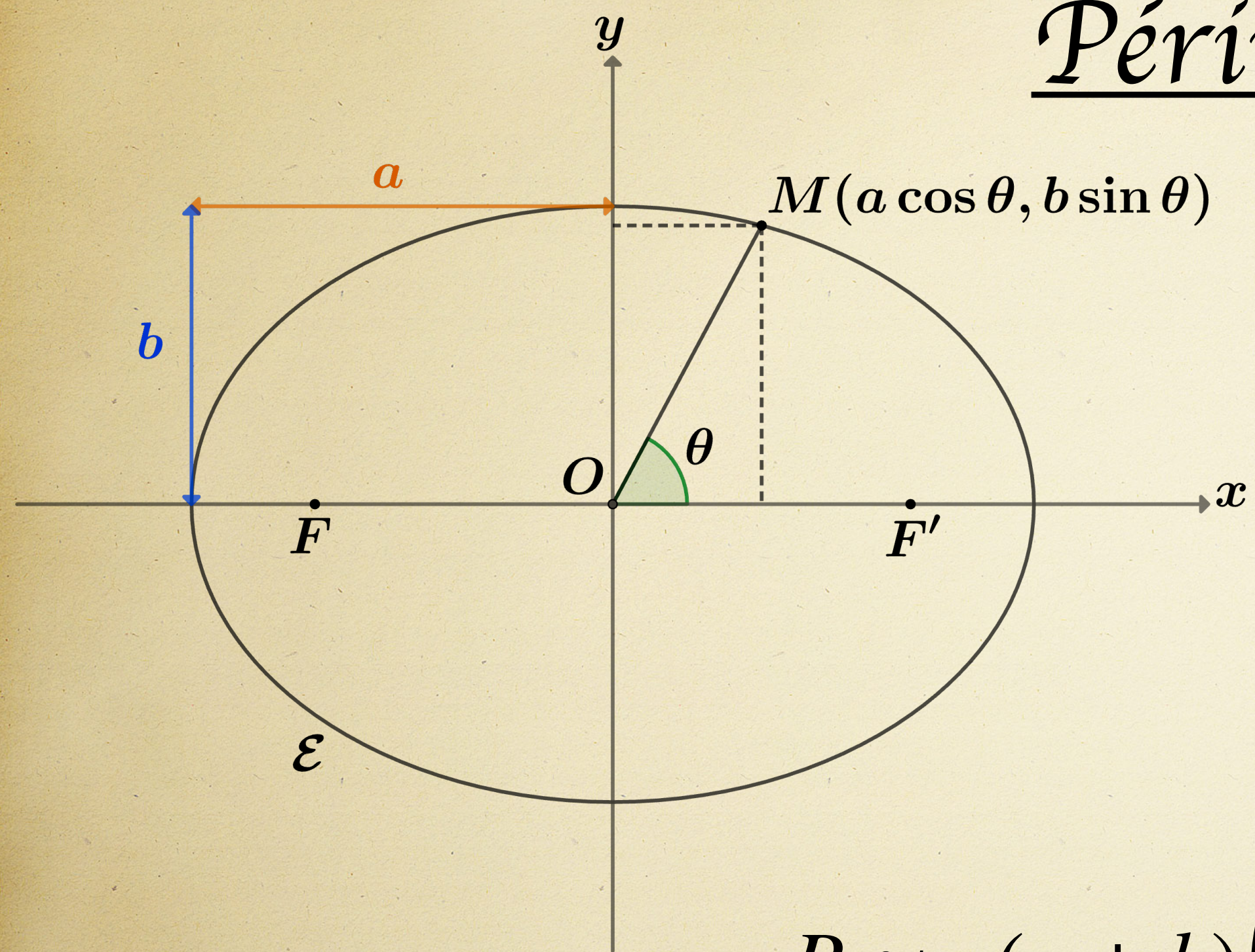
Aire

$$\pi \times a \times b$$

Périmètre

$$4 \int_0^{\frac{\pi}{2}} \sqrt{x'(\theta)^2 + y'(\theta)^2} d\theta = 4 \int_0^{\frac{\pi}{2}} \sqrt{1 - \left(1 - \frac{b^2}{a^2}\right) \sin^2 \theta} d\theta$$

Périmètre d'une ellipse

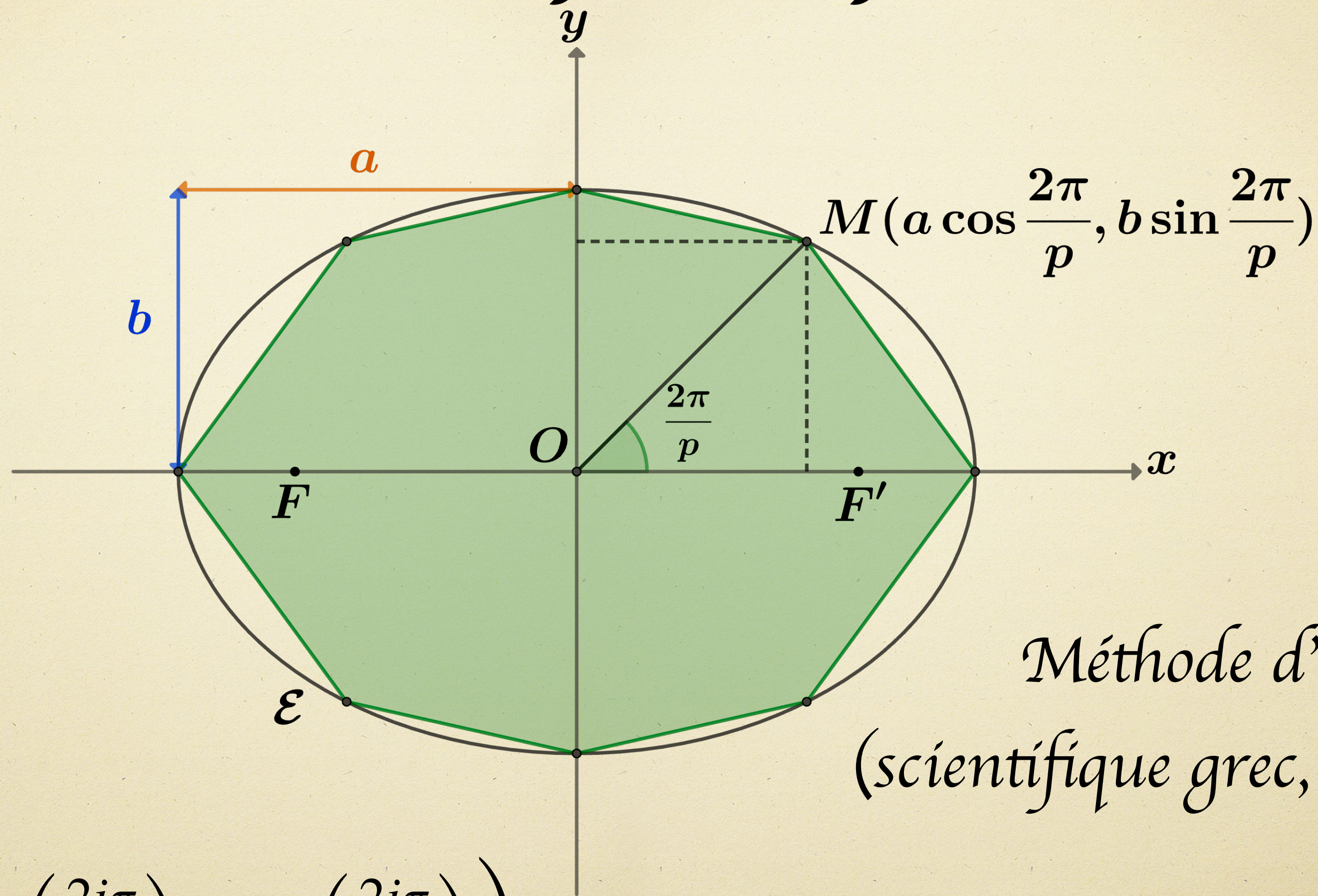


Formule de Ramanujan
(mathématicien indien, 1887-1920)

$$P \approx \pi(a + b) \left(1 + \frac{3h}{10 + \sqrt{4 - 3h}} \right) \quad \text{avec} \quad h = \frac{(a - b)^2}{(a + b)^2}$$

Périmètre d'une ellipse - implémentation

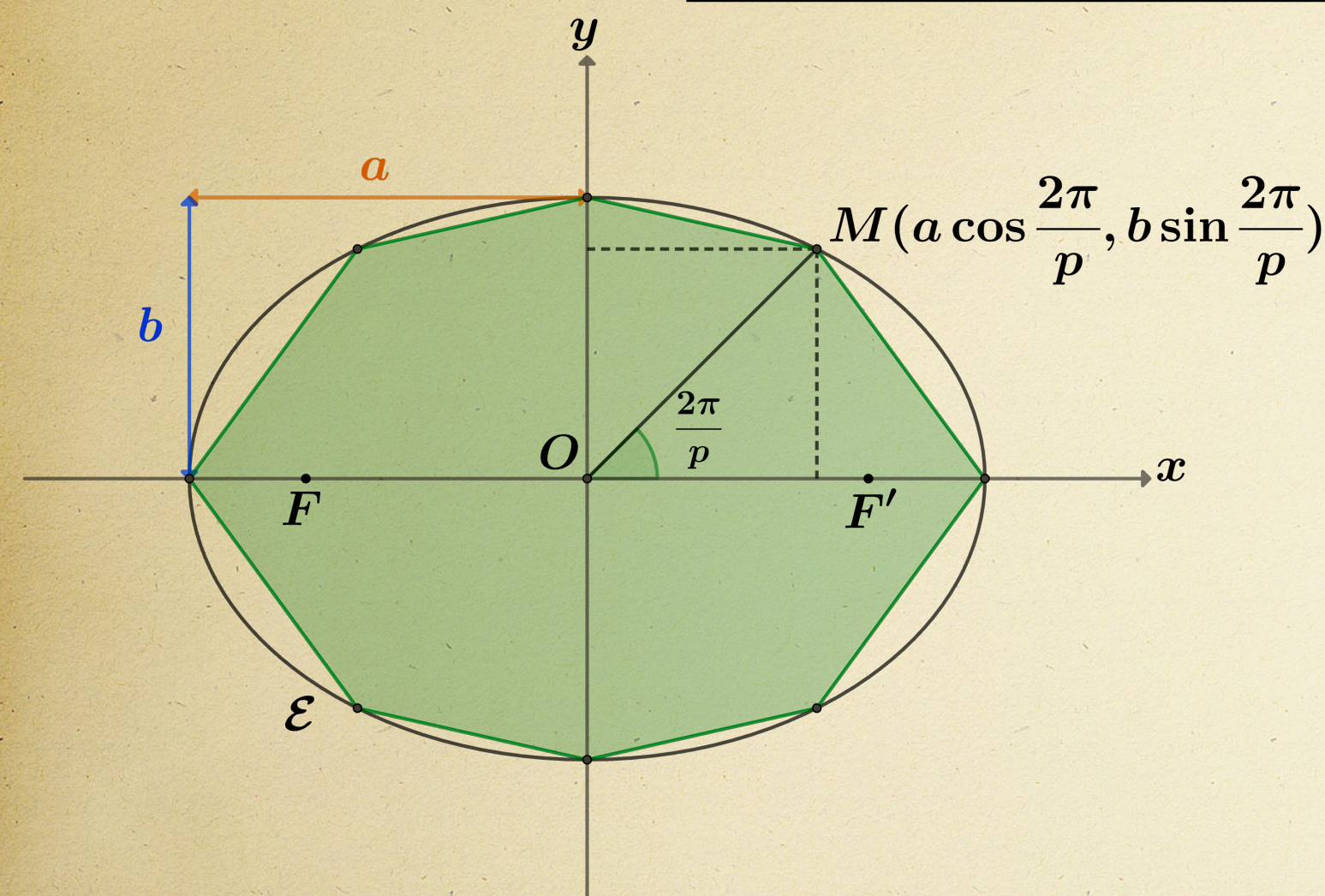
$p = 2^n$ côtés
 $n \in \mathbb{N}, n \geq 2$



Méthode d'Archimède
 (scientifique grec, 287-212 av.J.C.)

$$\forall i \in \llbracket 0, 2^n - 1 \rrbracket, M_i \left(a \cos \left(\frac{2i\pi}{p} \right), b \sin \left(\frac{2i\pi}{p} \right) \right) \text{ et } \forall i \in \llbracket 0, 2^n - 1 \rrbracket, l_i = M_i M_{i+1} \text{ avec } M_{2^n} = M_0$$

Périmètre d'une ellipse - version impérative



$$P_{a=b=1} = 2\pi \approx l_0 + l_1 + l_2 + l_3 \dots + l_{2^n-1}$$

- impérative : pour $n=23$, on s'approche π de
0.00000000000000000532907

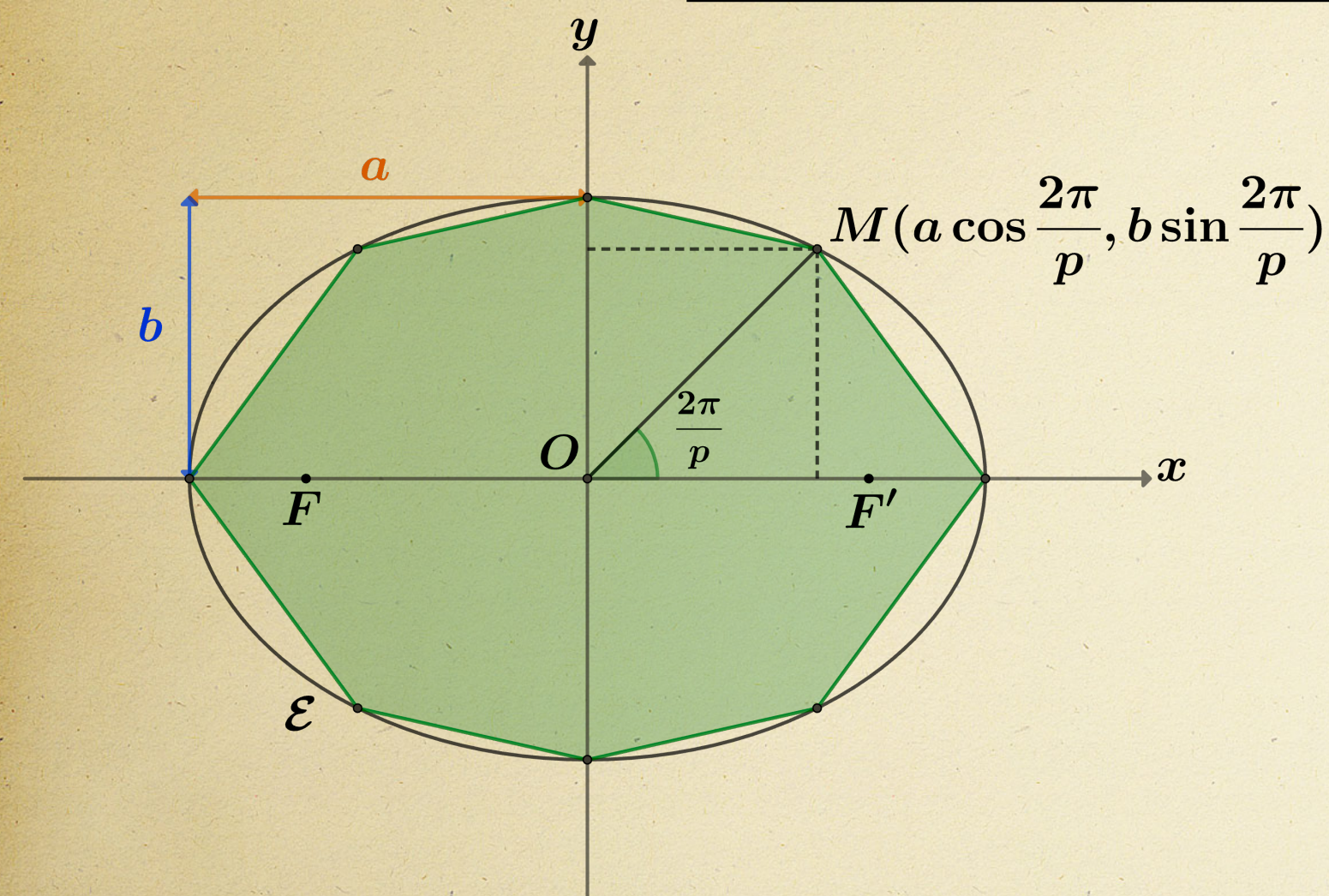
$$p = 2^n \text{ côtés}$$

$$n \in \mathbb{N}, n \geq 2$$

Erreurs dues aux nombres flottants :

- arrondi au supérieur
- perte d'information

Périmètre d'une ellipse - version impérative



```
let approx_perim_ellipse1 n a b =  
  if n < 2 then  
    failwith "approx_perim_ellipse1. Wrong parameter, need n >= 2."  
  else  
    let p = int_of_float(2. ** float_of_int(n)) in  
    let perim = ref 0. in  
    for i = 0 to p - 1 do  
      let a1 = float_of_int(i) *. 2. *. Float.pi /. float_of_int(p) in  
      let a2 = float_of_int(i + 1) *. 2. *. Float.pi /. float_of_int(p) in  
      let point1 = (a *. (cos a1), b *. (sin a1)) in  
      let point2 = (a *. (cos a2), b *. (sin a2)) in  
      perim := !perim +. (distance point1 point2);  
    done;  
    !perim
```


Arrondi au supérieur

Exemple : Addition binaire avec 4 chiffres significatifs après la virgule

$$x = (-1)^0 \times 2^0 \times 1,0000 = 1$$

$$y = (-1)^0 \times 2^{-3} \times 1,0110 = 0,171875$$

$$\begin{array}{r} 1,0000 \\ + 0,0010110 \\ \hline \cancel{1,0010}110 \\ 1,0011 \end{array}$$

Cela vaut : $1,1875 > 1,171875$

Perte d'information

$$P_{a=b=1} = 2\pi \approx \left(\dots \left(\left((l_0 + l_1) + l_2 \right) + l_3 \right) \dots + l_{2^n-1} \right)$$

Exemple : Addition binaire avec 4 chiffres significatifs après la virgule

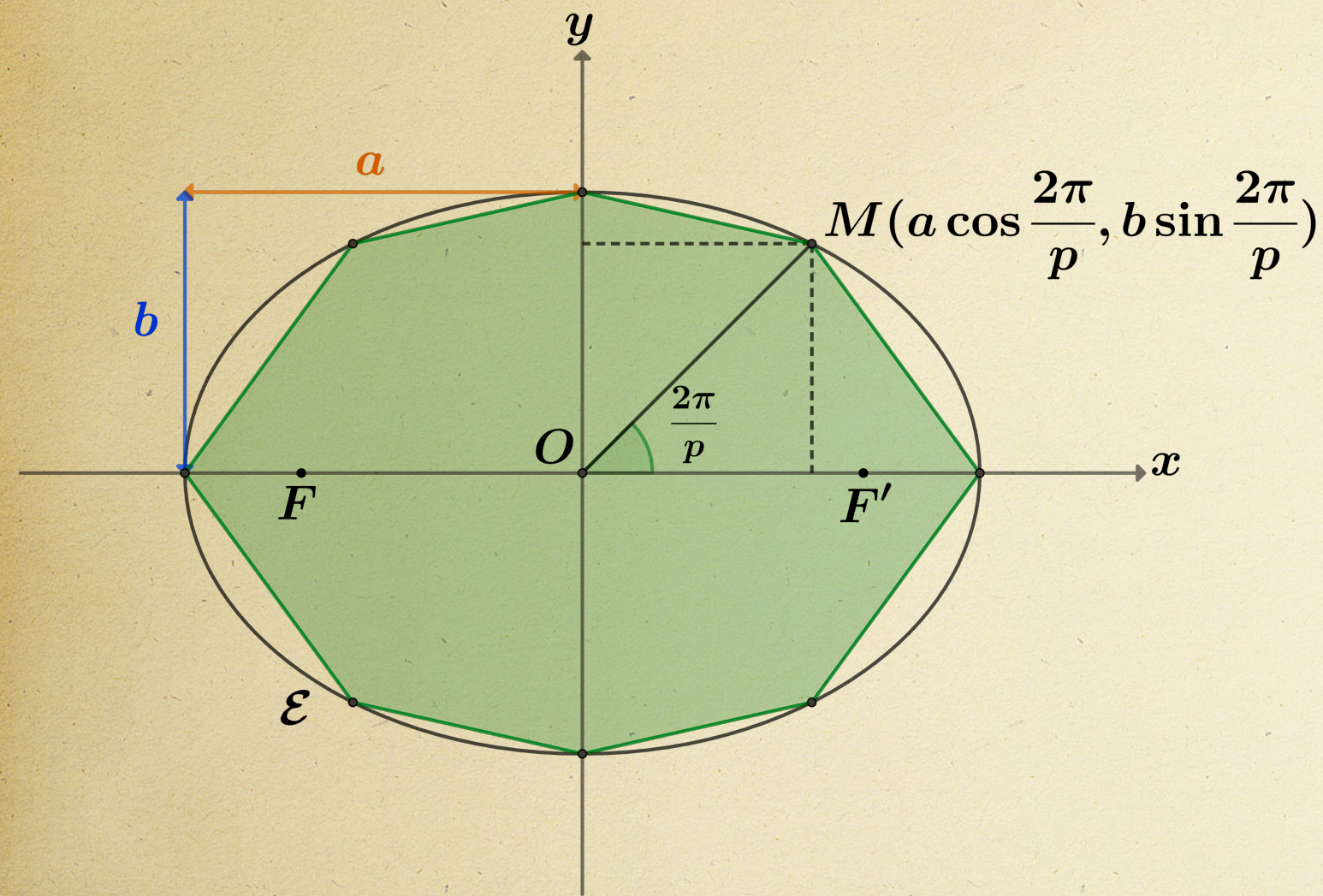
$$x = (-1)^0 \times 2^0 \times 1,0000 = 1$$

$$y = (-1)^0 \times 2^{-6} \times 1,0110 = 0,021484375$$

$$\begin{array}{r} 1,0000 \\ + 0,0000010110 \\ \hline 1,0000|010110 \end{array}$$

Cela vaut : 1, y est ignoré

Périmètre d'une ellipse - version fonctionnelle



$$P_{a=b=1} = 2\pi \approx l_0 + l_1 + l_2 + l_3 + \dots + l_{2^n-1}$$

$$P_{a=b=1} = 2\pi \approx \left(\dots \left((l_0 + l_1) + (l_2 + l_3) \right) + \dots + l_{2^n-1} \dots \right)$$

$p = 2^n$ côtés

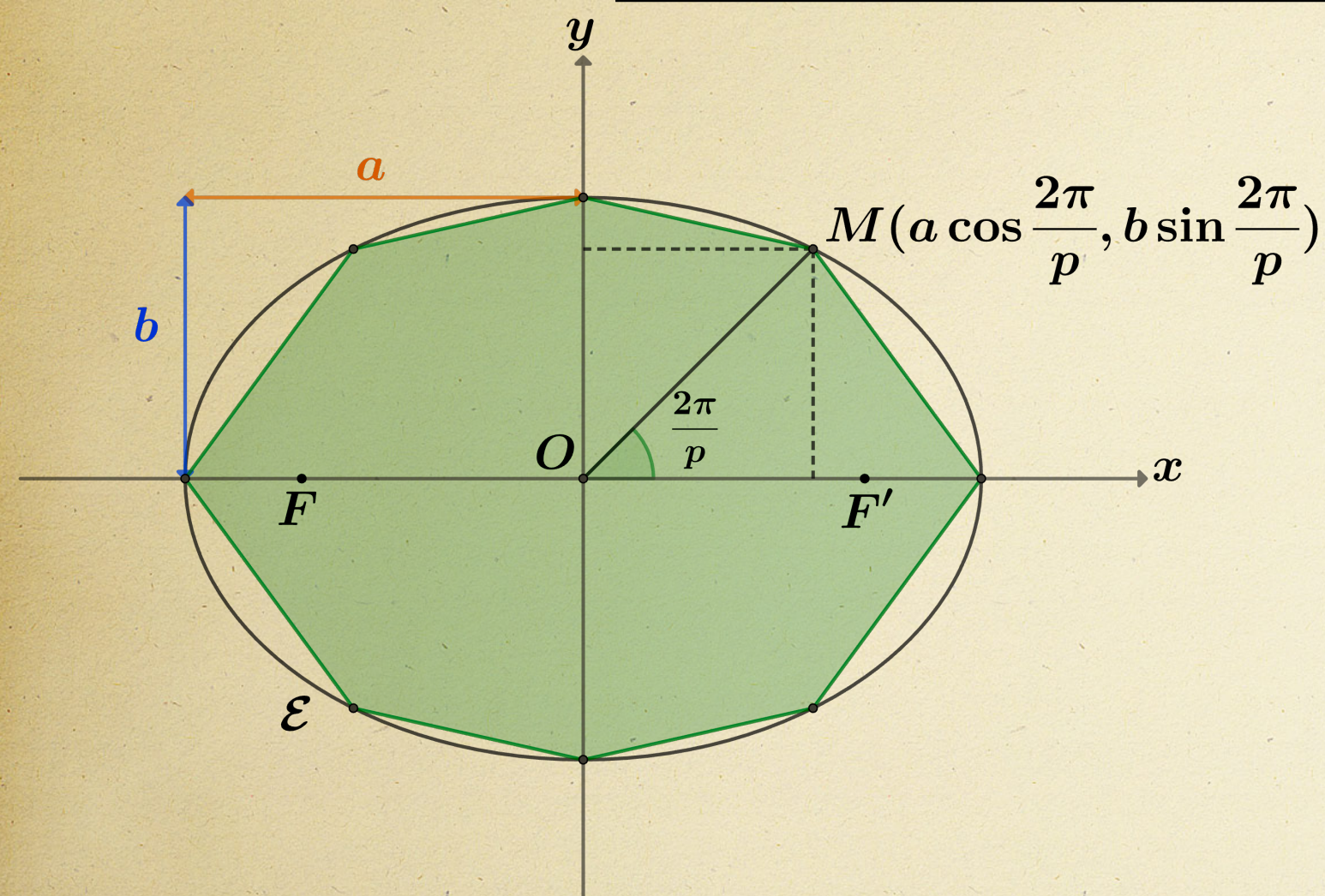
$n \in \mathbb{N}, n \geq 2$

- impérative : pour $n=23$, on s'approche π de

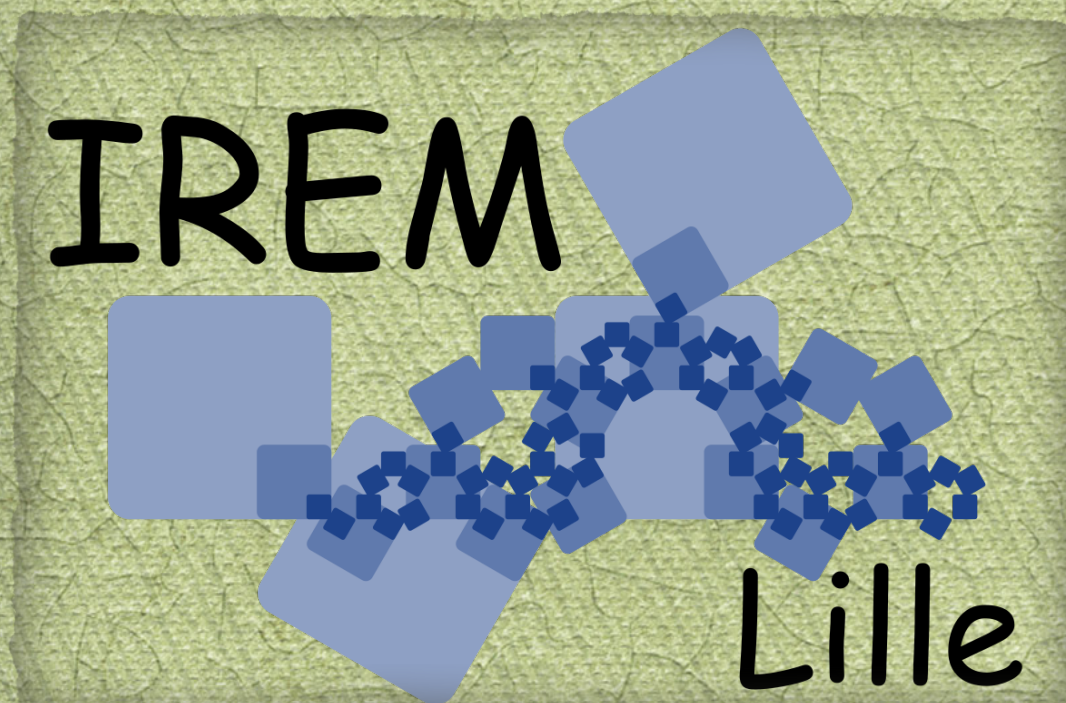
0.0000000000000000532907

- fonctionnelle : pour $n=27$, on a `Float.pi`

Périmètre d'une ellipse - version fonctionnelle



```
let approx_perim_ellipse2 n a b =  
  if n < 2 then  
    failwith "approx_perim_ellipse2. Wrong parameter, need n >= 2."  
  else  
    let rec aux n a b theta1 theta2 =  
      if n = 0 then  
        let point1 = (a *. (cos theta1), b *. (sin theta1)) in  
        let point2 = (a *. (cos theta2), b *. (sin theta2)) in  
        distance point1 point2  
      else  
        (aux (n-1) a b theta1 ((theta1+.theta2)/.2.)) +.  
        (aux (n-1) a b ((theta1+.theta2)/.2.) theta2)  
    in  
    aux n a b 0. (2. *. Float.pi)
```



Périmètre d'une ellipse

Asli Grimaud

Docteure et ingénieure en informatique

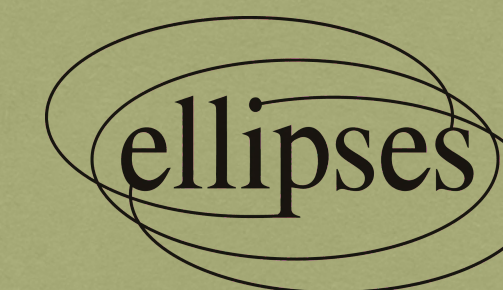
Professeure agrégée d'informatique (ex-mathématiques)

Professeure d'informatique en MP21/MP1

Informatique

MP21

*Cours complets, exercices corrigés et projets guidés
pour les classes préparatoires*



septembre 2025

Grimaud & Grimaud

Journées Académiques de l'IREM de Lille

MERCI

Mars 2025

asli.grimaud@ac-lille.fr