

Nombres de Fermat et test de Pépin

Marc Lorenzi

20 décembre 2025

Définition 1. Pour tout $n \in \mathbb{N}$, le n^e nombre de Fermat est

$$F_n = 2^{2^n} + 1$$

Proposition 1. *Pour tout $n \geq 1$, $F_n \equiv 2 \pmod{3}$.*

Démonstration. Montrons par récurrence sur n que pour tout $n \geq 1$, $2^{2^n} \equiv 1 \pmod{3}$. On a

$$2^{2^1} = 4 \equiv 1 \pmod{3}$$

Soit $n \geq 1$. Supposons la propriété vérifiée par n . On a alors

$$2^{2^{n+1}} = (2^{2^n})^2 \equiv 1^2 = 1 \pmod{3}$$

De là, pour tout $n \geq 1$,

$$F_n = 2^{2^n} + 1 \equiv 1 + 1 = 2 \pmod{3}$$

□

Comme 3 est premier, ce résultat nous dit en particulier que 3 et F_n sont premiers entre-eux. Ainsi, 3 est inversible dans l'anneau $\mathbb{Z}/F_n\mathbb{Z}$.

Proposition 2. *Pour tout $n \geq 1$, $F_n \equiv 1 \pmod{4}$.*

Démonstration. En effet, $2^n \geq 2$ donc $4 = 2^2$ divise 2^{2^n} . □

Proposition 3. [TEST DE PÉPIN]

Soit $n \geq 1$. Le nombre de Fermat F_n est premier si et seulement si

$$3^{(F_n-1)/2} \equiv -1 \pmod{F_n}$$

Démonstration. ($\Rightarrow$) Supposons F_n premier. On a

$$3^{(F_n-1)/2} \equiv \left(\frac{3}{F_n}\right) \pmod{F_n}$$

où la fraction entre parenthèses désigne le symbole de Legendre.

Par la loi de réciprocité quadratique,

$$\left(\frac{3}{F_n}\right) = (-1)^{\frac{3-1}{2} \frac{F_n-1}{2}} \left(\frac{F_n}{3}\right) = (-1)^{\frac{F_n-1}{2}} \left(\frac{2}{3}\right) = -(-1)^{\frac{F_n-1}{2}}$$

Comme $F_n \equiv 1 \pmod{4}$, la puissance de -1 est égale à 1, d'où le résultat.

($\Leftarrow$) Supposons $3^{(F_n-1)/2} \equiv -1 \pmod{F_n}$. Notons ω l'ordre de 3 dans l'anneau $\mathbb{Z}/F_n\mathbb{Z}$. On a

$$3^{2^{n-1}} \equiv -1 \pmod{F_n}$$

et donc, $\omega = 2^{2^n}$. Par ailleurs, ω divise $\varphi(F_n)$ où φ est la fonction indicatrice d'Euler. Ainsi,

$$\varphi(F_n) \geq \omega = F_n - 1$$

Comme $\varphi(F_n) \leq F_n - 1$, on en déduit $\varphi(F_n) = F_n - 1$. Ainsi, F_n est premier. $\square$

Remarquons que $(F_n - 1)/2 = 2^{2^n - 1}$. La fonction ci-dessous renvoie **True** si F_n est premier et **False** sinon. Sur une machine récente, le résultat est renvoyé quasi instantanément pour $n \leq 12$. Pour $n = 15$, il faut attendre environ une minute avant d'obtenir **False**.

```
def pepin(n):
    F = fermat(n)
    x = 3
    for k in range(2 ** n - 1):
        x = (x * x) % F
    return x == F - 1
```