Problème du cercle de Gauss

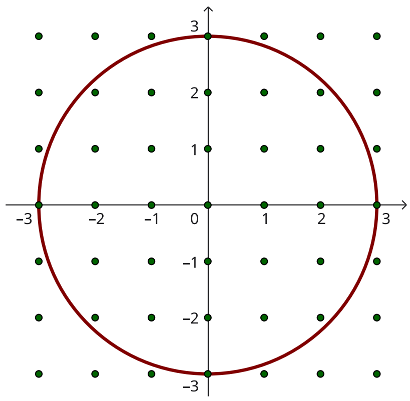
1. Énoncé

On se place dans le plan affine euclidien.

Soit $r \in \mathbb{N}^*$ et $\mathscr{C}_r$ le cercle centré en $(0;0)$, de rayon r.

Question : quel est le nombre p_r de points de $\mathbb{Z} \times \mathbb{Z}$ qui sont dans et sur le cercle ? Calculer p_5.

r	1	2	3	4	5	...
p_r	5	13	29	49	??	...



2. Stratégies de résolution

Nous allons aborder ce problème de deux manières différentes.

La méthode du mathématicien : trouver une formule explicite en fonction de r puis illustrer avec Geogebra.

La méthode de l'ingénieur : ne pas chercher de formule explicite mais réaliser un programme Python qui teste tous les points de $[\![-r, r]\!] \times [\![-r, r]\!]$

C'est parti !

3. Indices

La méthode du mathématicien

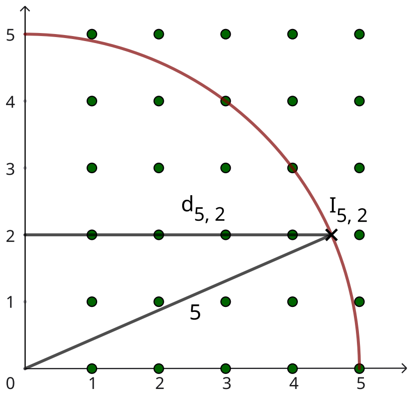
Dans un premier temps, on s'intéresse au nombre de points dans $\mathbb{N}^* \times \mathbb{N}$ (premier quart du plan). Pour un k donné, $1 \leqslant k \leqslant r$, le théorème de Pythagore nous donne la distance entre le point $(0, k)$ et le bord du quart de cercle d'ordonnée k.
À l'aide de la fonction « partie entière », on en déduit le nombre de points. La formule finale de la somme contient le symbole de la partie entière.

La méthode de l'ingénieur

On crée une double boucle : une première fait varier j de $-r$ à r et à l'intérieur de celle-ci, une deuxième fait varier k de $-r$ à r.
On utilise un compteur qui ajoute 1 si la condition `j**2 + k**2 <= r**2` est vérifiée. Attention à bien noter `for j in range(-r, r+1)`

4. Solutions

4.1. La méthode du mathématicien



Soit $r \in \mathbb{N}^*$. On note $\mathscr{D}$ l'ensemble des points de $\mathbb{Z} \times \mathbb{Z}$ qui sont dans ou sur $\mathscr{C}_r$ et on a $p_r := \mathrm{Card}(\mathscr{D})$

On note $\mathscr{C}_1$ le quart de cercle défini par $\mathscr{C}_1 := \mathscr{C} \cap \left(\mathbb{R}_+^* \times \mathbb{R}_+\right)$

On note $\mathscr{D}_1 = \mathscr{D} \cap (\mathbb{N}^* \times \mathbb{N})$ et on s'intéresse à son cardinal.

On a $\mathscr{D}_1 \subset (\llbracket 1, r \rrbracket \times \llbracket 0, r \rrbracket)$ puisqu'en-dehors de ce dernier ensemble, la distance de tout point de $\mathbb{N}^* \times \mathbb{N}$ sera strictement supérieure à r.

Soit $k \in \llbracket 0, r \rrbracket$

Pour $k = 0$, le nombre de points de $\mathscr{D}_1$ d'ordonnée $k = 0$ est égal à r (nombre d'entiers entre 1 et r).

Pour $k \neq 1$, on note $I_{r,k}$ le point d'intersection de $\mathscr{C}_1$ et de la droite d'équation $y = k$, et $d_{r,k}$ la distance entre le point $(0, k)$ et le point $I_{r,k}$. D'après le théorème de Pythagore, on a $d_{r,k} = \sqrt{r^2 - k^2}$

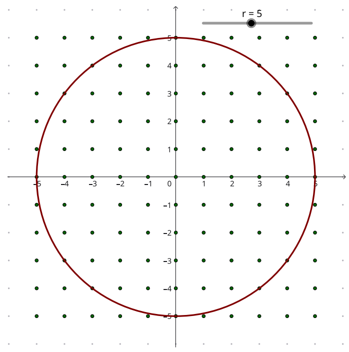
Le nombre de points de $\mathscr{D}_1$ d'ordonnée k est égal à $\lfloor d_{r,k} \rfloor$ (nombre d'entiers entre 1 et $d_{r,k}$ avec $\lfloor \cdot \rfloor$ désignant la partie entière).

On a alors : $\mathrm{Card}(\mathscr{D}_1) = \displaystyle\sum_{k=0}^{r} \left\lfloor \sqrt{r^2 - k^2} \right\rfloor$

Le nombre de points recherchés sur chacun des 3 autres « quarts de plan » est identique à $\mathrm{Card}(\mathscr{D}_1)$, par rotation. En ajoutant le point d'origine $(0;0)$, on a donc :

$$p_r = \mathrm{Card}(\mathscr{D}) = 1 + 4 \sum_{k=0}^{r} \left\lfloor \sqrt{r^2 - k^2} \right\rfloor \qquad \blacksquare$$

Illustration avec Geogebra :



- Créer un curseur `r = 5, min = 1, max = 10, Incrément = 1`
- Saisir `C = Cercle((0,0), r)`
- (Clic droit) > `Grille > (Points)`
- Entrer `p = 1 + Somme(4 * floor(sqrt(r^2-k^2)), k, 0, r)`
- Éventuellement, améliorer la présentation des points avec `G = Séquence(Séquence((j,k), j, -r, r), k, -r, r)`

La fenêtre « Algèbre » affiche `p = 81` (*i.e.* $p_5 = 81$)

4.2. La méthode de l'ingénieur

cercle-gauss.py

```python
def p(r):
   n = 0
   for j in range(-r, r+1):
      for k in range(-r, r+1):
         if(j**2 + k**2 <= r**2):
            n += 1
   return n

print(p(5))
```

Exécution du code

```
81
```

5. Pour aller plus loin...

On considère cette fois r, **réel** positif (les solutions seraient similaires : la formule pour Card($\mathscr{D}$) est identique au changement d'exposant près pour Σ et la partie informatique pourrait être ajustée).

Pour r grand[1], on a : $p_r \approx \pi r^2$

On note $E(r) = \pi r^2 - p_r$ le terme d'erreur.

On sait que $E(r) = \mathrm{O}(r^{\vartheta})$

Gauss a conjecturé[2] que $\vartheta = \frac{1}{2}$ et réussi à démontrer $\vartheta \leqslant 1$. En 1903, Voronoi a montré $\vartheta \leqslant \frac{2}{3}$ et en 2017, Bourgain et Watt ont légèrement amélioré le résultat avec $\vartheta \leqslant \frac{517}{824} \approx 0{,}627$

Cette conjecture est étroitement liée à l'hypothèse de Riemann : une preuve de l'une apporterait des éclairages sur l'autre.

En dimension supérieure à 3, le problème devient encore plus compliqué et les bornes sur le terme d'erreur sont encore moins précises.

Le problème du cercle de Gauss est sous-jacent à de nombreuses considérations en sciences appliquées : traitement d'images (flou gaussien, filtres), modélisations physiques (nombre de particules), cryptographie (réseaux euclidiens), recherche opérationnelle (optimisation logistique, couverture d'un service), etc.

1. En changeant la dernière ligne de code par `print(p(100)/100**2)`, on obtient une valeur approchée de π, ce qui peut d'ailleurs évoquer la méthode de Monte Carlo.

2. `https://petruconstantinescu.github.io/Gauss_circle_problem_presentation.pdf`