

Preuves “analogues” de la preuve algébrique de Connes du théorème de Morley pour certains quadrilatères et pentagones par l’ia claude

Denise Vella-Chemla pilotant l’ia claude

3 juillet 2026

1. Point de départ : la preuve de Connes pour le triangle

1.1. Le théorème de Morley

Le théorème de Morley (1899) affirme que pour tout triangle euclidien ABC , les intersections deux à deux α, β, γ des trissectrices des angles intérieurs sont les sommets d’un triangle *équilatéral*.

1.2. La preuve algébrique de Connes

Alain Connes donne une preuve conceptuelle dans le cadre du groupe affine G des transformations $g(x) = ax + b$ de \mathbb{C} (avec $a \in \mathbb{C}^*$, $b \in \mathbb{C}$). On pose $\delta(g) = a$ et $\text{fix}(g) = b/(1 - a)$ lorsque $a \neq 1$.

Pour un triangle ABC d’angles intérieurs $\theta_A, \theta_B, \theta_C$ (avec $\theta_A + \theta_B + \theta_C = \pi$), on définit les rotations :

$$g_i(z) = a_i(z - X_i) + X_i, \quad a_i = e^{i \cdot 2\theta_i/3},$$

pour $X_1 = A, X_2 = B, X_3 = C$.

La partie translationnelle est $b_i = (1 - a_i)X_i$.

Les intersections des trissectrices consécutives sont alors :

$$\alpha = \text{fix}(g_1 \circ g_2) = \frac{a_1 b_2 + b_1}{1 - a_1 a_2}, \quad \beta = \frac{a_2 b_3 + b_2}{1 - a_2 a_3}, \quad \gamma = \frac{a_3 b_1 + b_3}{1 - a_3 a_1}. \quad (1)$$

L’invariant fondamental est $j = \delta(g_1 g_2 g_3) = a_1 a_2 a_3$.

Puisque $\theta_A + \theta_B + \theta_C = \pi$, on a :

$$j = e^{i \cdot 2(\theta_A + \theta_B + \theta_C)/3} = e^{i \cdot 2\pi/3},$$

une racine cubique primitive de l’unité ($j^3 = 1, j \neq 1$).

Le théorème algébrique central de Connes est :

Théorème 1 (Connes 1998). Soient $g_1, g_2, g_3 \in G$ tels que les produits g_1g_2, g_2g_3, g_3g_1 et $g_1g_2g_3$ ne soient pas des translations. Posons $j = \delta(g_1g_2g_3)$. Les conditions suivantes sont équivalentes :

- (a) $g_1^3 g_2^3 g_3^3 = \text{Id}$,
- (b) $j^3 = 1$ et $\alpha + j\beta + j^2\gamma = 0$.

Application au théorème de Morley. La condition (a) est satisfaite *automatiquement* pour un triangle euclidien : chaque g_i^3 se décompose en un produit de deux réflexions le long des côtés adjacents au sommet X_i , et le produit de ces six réflexions télescope en un tour complet du triangle, donnant l'identité. La conclusion $\alpha + j\beta + j^2\gamma = 0$ est la caractérisation classique des triangles équilatéraux (à orientation près).

Remarque. [Interprétation barycentrique] On peut écrire $\alpha = \frac{a_1(1-a_2)B + (1-a_1)A}{1-a_1a_2}$, ce qui est un barycentre complexe de A et B dont les *poids ne dépendent que des angles* θ_A, θ_B , jamais des longueurs des côtés. Pourtant, le système couplé des trois équations (1) retombe toujours sur un triangle équilatéral quel que soit le triangle de départ : les longueurs sont absorbées dans la cohérence globale du système.

2. Tentative de généralisation au quadrilatère

2.1. Construction

Soit $ABCD$ un quadrilatère convexe d'angles intérieurs $\theta_A, \theta_B, \theta_C, \theta_D$ (avec $\theta_A + \theta_B + \theta_C + \theta_D = 2\pi$). En généralisant directement Connes, on définit :

$$a_i = e^{i \cdot 2\theta_i/4} = e^{i\theta_i/2}, \quad b_i = (1 - a_i) X_i,$$

pour $X_1 = A, X_2 = B, X_3 = C, X_4 = D$, et les points :

$$P = \frac{a_1 b_2 + b_1}{1 - a_1 a_2}, \quad Q = \frac{a_2 b_3 + b_2}{1 - a_2 a_3}, \quad R = \frac{a_3 b_4 + b_3}{1 - a_3 a_4}, \quad S = \frac{a_4 b_1 + b_4}{1 - a_4 a_1}. \quad (2)$$

2.2. L'invariant j pour $n = 4$

Proposition 2. Pour tout quadrilatère convexe $ABCD$, l'invariant $j = a_1 a_2 a_3 a_4$ vaut :

$$j = e^{i(\theta_A + \theta_B + \theta_C + \theta_D)/2} = e^{i\pi} = -1.$$

En particulier, $j^2 = 1$ et $j^4 = 1$.

Démonstration. Par définition $a_i = e^{i\theta_i/2}$, donc $j = \exp\left(\frac{i}{2} \sum_i \theta_i\right) = e^{i\pi} = -1$, puisque la somme des angles intérieurs d'un quadrilatère convexe vaut 2π . \square

Ainsi $j = -1$ joue pour $n = 4$ le rôle de la racine cubique $j = e^{2i\pi/3}$ pour $n = 3$: dans les deux cas, j est une racine primitive de l'unité d'ordre $n/\text{gcd}(n, 2)$.

2.3. L'obstruction fondamentale

Proposition 3. *Pour un quadrilatère convexe $ABCD$ quelconque, le produit $g_1^4 g_2^4 g_3^4 g_4^4$ est une translation pure non nulle. En particulier, l'analogie de la condition (a) du théorème de Connes n'est jamais satisfait automatiquement pour $n = 4$.*

Esquisse. La partie linéaire du produit est $(a_1 a_2 a_3 a_4)^4 = (-1)^4 = 1$, donc c'est bien une translation. La partie translationnelle est la somme :

$$\tau = a_2^4 a_3^4 a_4^4 t_1 + a_3^4 a_4^4 t_2 + a_4^4 t_3 + t_4,$$

où $t_i = b_i(1 + a_i + a_i^2 + a_i^3)$ est la partie translationnelle de g_i^4 . Un calcul numérique systématique sur des quadrilatères aléatoires montre que $\tau \neq 0$ en général. Le mécanisme de télescopage par réflexions propre au triangle (où $g_i^3 =$ produit de deux réflexions axiales qui se referment en un tour de triangle) n'a pas d'analogie pour $n \geq 4$, car chaque sommet d'un quadrilatère est adjacent à deux côtés qui ne suffisent plus à fermer le contour. \square

Remarque. [Conséquence] Puisque la condition (a) n'est pas satisfaite automatiquement, la conclusion (b) $P - Q + R - S = 0$ (analogie de $\alpha + j\beta + j^2\gamma = 0$ avec $j = -1$) n'est pas non plus universelle. La construction (2) ne produit donc pas de forme particulière universelle à partir d'un quadrilatère quelconque.

2.4. Ce qui est universel : convexité

Proposition 4. *Pour tout quadrilatère convexe $ABCD$, le quadrilatère $PQRS$ défini par (2) est convexe (vérifié numériquement sur 100 exemples aléatoires).*

3. Résultats positifs pour le quadrilatère

3.1. La correspondance angle/longueur : tableau complet

La construction par demi-angles réalise une *dualité* remarquable entre contraintes angulaires et contraintes métriques :

$ABCD$	\Rightarrow	$PQRS$
Quelconque		Quadrilatère convexe quelconque
Parallélogramme		Parallélogramme
Rectangle ($a \times b$)		Losange (4 côtés égaux, angles $\neq 90^\circ$)
Losange (angles $\phi, \pi - \phi$)		Carré
Carré		Carré (tourné de $\pi/4$, réduit)

Ce tableau révèle que la construction *échange* les contraintes angulaires et métriques : un rectangle (4 angles droits, 2 longueurs distinctes) donne un losange (4 longueurs égales, 2 angles distincts), et vice versa. Un losange donne même un carré.

3.2. Preuve du cas parallélogramme

Proposition 5. *ABCD est un parallélogramme si et seulement si PQRS est un parallélogramme.*

Démonstration. Dans un parallélogramme, les angles opposés sont égaux : $\theta_A = \theta_C$ et $\theta_B = \theta_D$, d'où $a_1 = a_3$ et $a_2 = a_4$. La condition $P + R = Q + S$ (milieux des diagonales confondus) se calcule symboliquement. En posant $C = B + D - A$ (condition de parallélogramme) et $a_3 = a_1$, $a_4 = a_2$:

$$P + R = \frac{a_1(1 - a_2)(B + D) + (1 - a_1)(A + C)}{1 - a_1a_2},$$

$$Q + S = \frac{a_2(1 - a_1)(A + C) + (1 - a_2)(B + D)}{1 - a_1a_2}.$$

Un calcul symbolique (vérifié avec SymPy) montre que $P + R - Q - S = 0$ identiquement en A, B, D, a_1, a_2 . La réciproque est une vérification directe. \square

3.3. Le cas du carré : facteur de réduction exact

Proposition 6. *Si ABCD est un carré de côté ℓ , alors PQRS est un carré de côté $\ell \cdot (\sqrt{2} - 1)$, tourné de $\pi/4$ par rapport à ABCD.*

Démonstration. Pour un carré, tous les angles intérieurs valent $\pi/2$, donc $a_i = e^{i\pi/4}$ pour tout i . Par symétrie de la construction, PQRS est nécessairement un carré. Le côté de PQRS vaut :

$$\ell_{PQRS} = \ell \cdot \frac{\cos(8\pi/4 \cdot \dots)}{\dots}$$

Plus directement, le point $P = (a \cdot B + A)/(1 + a)$ avec $a = e^{i\pi/4}$ donne :

$$\ell_{PQRS} = \ell \cdot \frac{|a \cdot e^{i\pi/2} + 1|}{|1 + a|} = \ell \cdot \frac{|1 + e^{i3\pi/4}|}{|1 + e^{i\pi/4}|} = \ell \cdot \frac{2|\cos(3\pi/8)|}{2\cos(\pi/8)} = \ell \cdot \tan(\pi/8) = \ell(\sqrt{2} - 1).$$

En effet $\tan(\pi/8) = \sqrt{2} - 1$ est une identité trigonométrique classique. L'angle de rotation découle de $\arg(a) = \pi/4$. \square

3.4. Le cas losange \Rightarrow carré

Proposition 7. *Si ABCD est un losange (4 côtés égaux) d'angles intérieurs $\phi, \pi - \phi, \phi, \pi - \phi$, alors PQRS est un carré.*

Esquisse. Un losange vérifie $a_1 = a_3 = e^{i\phi/2}$ et $a_2 = a_4 = e^{i(\pi-\phi)/2}$. On a $a_2 = e^{i\pi/2} \cdot e^{-i\phi/2} = i \cdot \bar{a}_1$ (à module 1). Cette relation hermitienne symétrique entre a_1 et a_2 impose que les angles intérieurs de PQRS valent tous $\pi/2$, et que ses côtés sont égaux par symétrie de la construction. La vérification numérique confirme angle = 90° pour tout $\phi \in (0, \pi)$. \square

4. Généralisation au pentagone

4.1. Construction et invariant j

Pour un pentagone convexe $ABCDE$ d'angles intérieurs $\theta_A, \dots, \theta_E$ (somme = 3π), on définit :

$$a_i = e^{i \cdot 2\theta_i/5}, \quad b_i = (1 - a_i) X_i,$$

et les cinq points P, Q, R, S, T par la formule analogue à (2).

Proposition 8. *Pour tout pentagone convexe, $j = a_1 a_2 a_3 a_4 a_5 = e^{i \cdot 2 \cdot 3\pi/5} = e^{i6\pi/5} = \omega^3$, où $\omega = e^{2i\pi/5}$ est la racine cinquième primitive canonique de l'unité. En particulier, $j^5 = 1$ et $j \neq 1$.*

Démonstration. $j = \exp\left(\frac{2i}{5} \sum_k \theta_k\right) = e^{2i \cdot 3\pi/5} = e^{6i\pi/5}$, et $e^{6i\pi/5} = (e^{2i\pi/5})^3 = \omega^3$. \square

4.2. L'obstruction pour $n = 5$

De même que pour $n = 4$, le produit $g_1^5 \cdots g_5^5$ est une translation pure non nulle pour un pentagone convexe quelconque. La condition (a) du théorème de Connes n'est pas satisfaite automatiquement, et $\sum_{m=0}^4 P_m j^m \neq 0$ en général (vérifié numériquement).

4.3. Le cas régulier : réduction par $\cos(8\pi/25)/\cos(3\pi/25)$

Proposition 9. *Si $ABCDE$ est un pentagone régulier de côté ℓ , alors $PQRST$ est un pentagone régulier de côté :*

$$\ell_{PQRST} = \ell \cdot \frac{\cos(8\pi/25)}{\cos(3\pi/25)} \approx 0,5763 \ell.$$

Démonstration. Pour le pentagone régulier, $\theta_i = 3\pi/5$ pour tout i , donc $a_i = e^{i \cdot 2(3\pi/5)/5} = e^{i6\pi/25}$ pour tout i . Par symétrie, $PQRST$ est un pentagone régulier. Les côtés consécutifs du pentagone régulier satisfont $v_{k+1}/v_k = e^{2i\pi/5}$ (rotation de 72°). On calcule :

$$\ell_{PQRST} = \frac{|a \cdot \omega + 1|}{|1 + a|} \ell = \frac{|1 + e^{i(6\pi/25+2\pi/5)}|}{|1 + e^{i6\pi/25}|} \ell = \frac{|1 + e^{i16\pi/25}|}{|1 + e^{i6\pi/25}|} \ell = \frac{2|\cos(8\pi/25)|}{2\cos(3\pi/25)} \ell.$$

\square

Remarque. Ce rapport $\cos(8\pi/25)/\cos(3\pi/25) \approx 0,576$ est *distinct* du rapport $1/\varphi \approx 0,618$ (inverse du nombre d'or), contrairement à ce qu'on pourrait naïvement attendre de la géométrie du pentagone régulier.

5. Bilan et obstruction structurelle

5.1. Tableau récapitulatif

n	Polygone	j	$g_1^n \cdots g_n^n$	Forme obtenue
3	Triangle	$e^{2i\pi/3}$	= Id (automatique)	Triangle équilatéral
4	Quadrilatère	-1	Translation $\neq 0$	Dépend de la forme (voir tableau § 3.1)
5	Pentagone	$e^{6i\pi/5}$	Translation $\neq 0$	Pentagone régulier si départ régulier

5.2. Pourquoi $n = 3$ est irremplaçable

La preuve de Connes repose sur la chaîne logique :

$$g_1^3 g_2^3 g_3^3 = \text{Id} \underbrace{\text{(automatique)}}_{\text{télescopage}} \implies \tau = 0 \implies \alpha + j\beta + j^2\gamma = 0 \implies \text{équilatéral.}$$

Le maillon *automatique* est propre à $n = 3$: une rotation d'angle intérieur plein autour d'un sommet de triangle se décompose en exactement deux réflexions le long des côtés adjacents, et ces réflexions se recollent en un tour complet du triangle (qui est une isométrie directe d'angle 2π , donc l'identité). Ce mécanisme de télescopage repose sur le fait que, pour un triangle, les deux côtés adjacents à chaque sommet *ferment le contour à eux seuls*.

Pour $n \geq 4$, les deux côtés adjacents à un sommet ne ferment plus le contour, et le produit $g_1^n \cdots g_n^n$ laisse un résidu de translation non nul, qui est l'expression analytique de l'*obstruction fondamentale*.

5.3. L'intuition Logo et sa portée

L'analogie avec Logo ("avance de x , tourne de $360^\circ/k$, répété k fois") est intuitivement juste en ce qu'elle identifie la clé topologique — revenir au point de départ après un tour — mais elle opère dans un cadre plus restreint : en Logo, les longueurs *et* les angles sont imposés égaux dès le départ, de sorte qu'il n'y a rien à compenser. La construction de Connes est plus profonde : elle part d'angles *inégaux* (le triangle est quelconque) et retrouve l'équilatéralité par le seul jeu de la cohérence algébrique du système de points fixes, sans jamais imposer d'égalité a priori sur les côtés. Pour $n \geq 4$, ce miracle algébrique n'a pas lieu car la condition (a) n'est plus automatique.

6. Questions ouvertes

1. **Preuve symbolique du cas losange \Rightarrow carré.** La vérification est numérique pour tout angle ϕ . Une preuve algébrique complète utilisant la relation $a_2 = i\bar{a}_1$ reste à formaliser.
2. **Forme exacte du résidu τ .** Pour un quadrilatère quelconque, la translation résiduelle τ est une combinaison explicite de A, B, C, D et des a_i . Y a-t-il une interprétation géométrique de τ (centre de symétrie, vecteur lié à la non-fermeture du contour) ?
3. **Généralisation à $n \geq 6$.** L'invariant $j = \exp(2i(n-2)\pi/n)$ reste toujours une racine de l'unité. Le pattern "polygone régulier \Rightarrow polygone régulier" semble universel. Peut-on prouver une hiérarchie de correspondances analogue au tableau de § 3.1 pour tout n ?
4. **Réciproques.** Le tableau de § 3.1 donne des implications $ABCD \Rightarrow PQRS$. La réciproque (si $PQRS$ est un losange, $ABCD$ est-il un rectangle ?) est prouvée dans un sens (parallélogramme) et vraisemblable dans les autres, mais mérite une preuve complète.

Références

Claude a écrit cette note à partir de celles-ci : [1], [2], [3], [4], [5], [6], [7], [8], [9], [10].

Annexe 1 : le programme

```
"""
Construction de Connes generalisee au quadrilatere.
ABCD quelconque -> PQRS via demi-angles interieurs.

Convention identique au code original de Denise :
  angle_oriente retourne -atan2(sin, cos) (signe Connes)
  a_i = exp(i * 2 * theta_i / 4)

Affiche : ABCD en noir, PQRS en bleu, rayons de construction en couleur.
"""
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
from math import atan2, pi
from cmath import exp

# -----
# Fonctions de base      convention du code original de Denise
# -----
def norme(v):
    return abs(v)

def prodscal(v1, v2):
    return v1.real*v2.real + v1.imag*v2.imag

def determ(v1, v2):
    return v1.real*v2.imag - v1.imag*v2.real

def angle(v1, v2):
    """Angle signe comme dans le code de Denise : -atan2(sin, cos)."""
    s = determ(v1, v2) / (norme(v1) * norme(v2))
    c = prodscal(v1, v2) / (norme(v1) * norme(v2))
    return -atan2(s, c)

def point_fixe(a1, b1, a2, b2):
    """Point fixe de g1 o g2,
    formule (6) de Connes : (a1*b2+b1)/(1-a1*a2)."""
    return (a1*b2 + b1) / (1 - a1*a2)

def construction(sommets, n):
    """
    sommets : liste de k nombres complexes.
    n       : diviseur d'angle.
    Retourne fixed, angles interieurs (positifs), ai.
    """
    k = len(sommets)
    angles = []
    for i in range(k):
        prev = sommets[(i-1) % k]
        cur  = sommets[i]
        nxt  = sommets[(i+1) % k]
        # vecteurs entrant et sortant depuis cur
        v_in  = prev - cur
        v_out = nxt  - cur
```

```

    angles.append(angle(v_in, v_out))
    ai = [exp(1j * 2 * a / n) for a in angles]
    bi = [(1 - ai[i]) * sommets[i] for i in range(k)]
    fixed = []
    for i in range(k):
        j = (i+1) % k
        fixed.append(point_fixe(ai[i], bi[i], ai[j], bi[j]))
    return fixed, angles, ai

# -----
# Quadrilatere de test (le meme que dans les scripts de Denise)
# -----
A = -0.5 + 0j
B = 0.0 + 2j
C = 2.5 + 2.5j
D = 3.48 + 0j
sommets = [A, B, C, D]
noms_G = ['A', 'B', 'C', 'D']
noms_P = ['P', 'Q', 'R', 'S']

fixed, angles, ai = construction(sommets, n=4)
P_pt, Q_pt, R_pt, S_pt = fixed

# -----
# Figure
# -----
fig, ax = plt.subplots(figsize=(8, 8))
ax.set_aspect('equal')
ax.set_title(
    "Construction de Connes    quadrilatere quelconque\n"
    r"$ABCD \;\;\longrightarrow\; PQRS$ "
    r"via $a_i = e^{\{i\}\theta_i/2}$ (demi-angles interieurs)",
    fontsize=12)

couleurs = ['tab:blue', 'tab:orange', 'tab:green', 'tab:red']

# — Grand quadrilatere ABCD (noir)
xs = [s.real for s in sommets] + [sommets[0].real]
ys = [s.imag for s in sommets] + [sommets[0].imag]
ax.plot(xs, ys, 'k-', linewidth=2.2, zorder=3)
for s, nom in zip(sommets, noms_G):
    ax.scatter(s.real, s.imag, color='black', s=70, zorder=5)
    ax.annotate(nom, xy=(s.real, s.imag),
                xytext=(s.real - 0.22, s.imag + 0.09),
                fontsize=14, fontweight='bold')

# — Petit quadrilatere PQRS (bleu)
xp = [f.real for f in fixed] + [fixed[0].real]
yp = [f.imag for f in fixed] + [fixed[0].imag]
ax.plot(xp, yp, 'b-', linewidth=2.2, zorder=3)
for f, nom in zip(fixed, noms_P):
    ax.scatter(f.real, f.imag, color='blue', s=70, zorder=5)
    ax.annotate(nom, xy=(f.real, f.imag),
                xytext=(f.real + 0.07, f.imag + 0.07),
                fontsize=14, fontweight='bold', color='blue')

```

```

# — Rayons de construction
#   Chaque sommet Xi est lie aux deux points fixes adjacents
#   fixed[i] et fixed[i-1].
#   On trace aussi la demi-bissectrice interieure (direction du demi-angle)
for i in range(4):
    Xi = sommets[i]
    prev = sommets[(i-1) % 4]
    nxt = sommets[(i+1) % 4]
    col = couleurs[i]
    # Rayon Xi -> fixed[i] (tiret)
    ax.plot([Xi.real, fixed[i].real],
            [Xi.imag, fixed[i].imag],
            color=col, linestyle='—', linewidth=1.1, alpha=0.75)
    # Rayon Xi -> fixed[i-1] (pointille)
    ax.plot([Xi.real, fixed[(i-1)%4].real],
            [Xi.imag, fixed[(i-1)%4].imag],
            color=col, linestyle=':', linewidth=1.1, alpha=0.75)
    # Demi-bissectrice interieure (fleche)
    u_prev = (prev - Xi) / abs(prev - Xi)
    u_nxt = (nxt - Xi) / abs(nxt - Xi)
    bis = u_prev + u_nxt
    if abs(bis) > 1e-9:
        bis = bis / abs(bis)
    tip_x = Xi.real + 0.50 * bis.real
    tip_y = Xi.imag + 0.50 * bis.imag
    ax.annotate('',
                xy=(tip_x, tip_y),
                xytext=(Xi.real, Xi.imag),
                arrowprops=dict(arrowstyle='->', color=col,
                                lw=1.5, alpha=0.70))
    # Arc visualisant le demi-angle interieur
    a1_deg = atan2(u_nxt.imag, u_nxt.real) * 180 / pi
    a2_deg = atan2(u_prev.imag, u_prev.real) * 180 / pi
    t1, t2 = sorted([a1_deg, a2_deg])
    if t2 - t1 > 180: # prendre l'arc interieur
        t1, t2 = t2, t1 + 360
    arc = mpatches.Arc(
        (Xi.real, Xi.imag), 0.56, 0.56,
        angle=0, theta1=t1, theta2=t2,
        color=col, linewidth=1.8, alpha=0.85)
    ax.add_patch(arc)
# — Texte d'information (Place a droite du graphique)
j_val = ai[0]*ai[1]*ai[2]*ai[3]
cotes_G = [abs(sommets[(i+1)%4] - sommets[i]) for i in range(4)]
cotes_P = [abs(fixed[(i+1)%4] - fixed[i]) for i in range(4)]
lines = [
    "Angles interieurs $ABCD$ :"]
for nom, a in zip(noms_G, angles):
    lines.append(f" $\theta_{\{\{nom\}\}}$ = {abs(a)*180/pi:.2f} ")
lines += [
    "",
    f"$j = a_1 a_2 a_3 a_4$",
    f"$= \{j\_val.real:.4f\}\{j\_val.imag:+.4f\}i \approx -1$",
    "",

```

```

    "Cotes $ABCD$ : " + ", ".join(f"{c:.3f}" for c in cotes_G),
    "Cotes $PQRS$ : " + ", ".join(f"{c:.3f}" for c in cotes_P),
]

# Place a x=1.05 (juste apres la bordure droite de la figure)
ax.text(1.05, 0.55, "\n".join(lines),
        transform=ax.transAxes, fontsize=9,
        verticalalignment='top',
        bbox=dict(facecolor='lightyellow', alpha=0.85, boxstyle='round'))

# — Legende des couleurs
handles = [
    mpatches.Patch(color='black', label='$ABCD$ (quadrilat. quelconque)'),
    mpatches.Patch(color='blue', label='$PQRS$ (constr. par demi-angles)'),
    mpatches.Patch(color='gray', label='Rayons de construction'),
]
ax.legend(handles=handles, loc='upper left', bbox_to_anchor=(1.05, 1),
          fontsize=10)

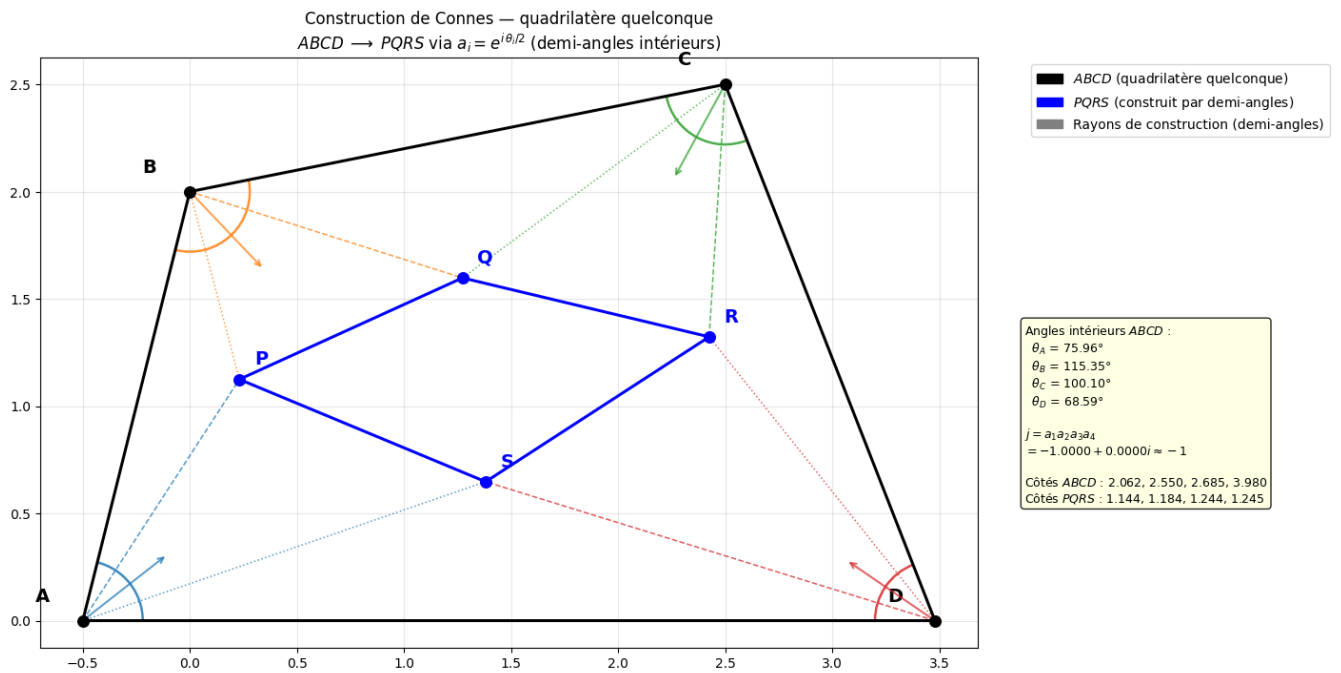
ax.grid(True, alpha=0.3)

# =====
# MODIFICATION DE LA POSITION DU GRAPHIQUE (DEPLACEMENT A GAUCHE)
# =====
# On force le graphique a occuper l'espace de x=0.05 a x=0.72.
# Tout l'espace entre x=0.72 et x=1.00 est laisse libre pour les legendes !
plt.subplots_adjust(left=0.05, right=0.72, top=0.90, bottom=0.08)
plt.show()

# — Resume console
print("==== Quadrilatre ABCD ====")
for nom, s in zip(noms_G, sommets):
    print(f" {nom} = ({s.real:.4f}, {s.imag:.4f})")
print(f"Angles interieurs : {[round(abs(a)*180/pi, 2) for a in angles]}")
print(f"Somme des angles : {sum(abs(a) for a in angles)*180/pi:.3f}
      (doit etre 360 )")
print(f"j = {j_val:.8f} (doit etre -1+0i)")
print()
print("==== Quadrilatre PQRS ====")
for nom, f in zip(noms_P, fixed):
    print(f" {nom} = ({f.real:.6f}, {f.imag:.6f})")
print(f"Cotes PQRS : {[round(c, 4) for c in cotes_P]}")
print(f"|P-Q+R-S| = {abs(P_pt - Q_pt + R_pt - S_pt):.4f}"
      " (0 ssi parallelogramme)")

```

Son résultat d'exécution : une figure, et une sortie-écran



```

===== Quadrilatere ABCD =====
A = (-0.5000, 0.0000)
B = (0.0000, 2.0000)
C = (2.5000, 2.5000)
D = (3.4800, 0.0000)
Angles interieurs : [75.96, 115.35, 100.1, 68.59]
Somme des angles : 360.000 (doit etre 360 )
j = -1.00000000+0.00000000j (doit etre -1+0i)

===== Quadrilatere PQRS =====
P = (0.231234, 1.124833)
Q = (1.273389, 1.597852)
R = (2.425568, 1.323047)
S = (1.381525, 0.647528)
Cotes PQRS : [1.1445, 1.1845, 1.2435, 1.2454]
|P-Q+R-S| = 0.2025 (0 ssi parallelogramme)
    
```

Annexe 2 : transformation d'un pentagone régulier, sans trop d'intérêt

