

```

import numpy as np
import matplotlib.pyplot as plt

# -----
# 1. Definition des matrices de l'exemple 1.11.11 de Tretter
# -----

# Matrice complete A5 (4x4) partitionnee en blocs 1x1, 2x2, 1x1
A5 = np.array([
    [ 1,      3 + 1j,  2,      1j],
    [ 3 + 1j,  1,      1j,      2],
    [-2,      1j,      1,      3 + 1j],
    [ 1j,     -2,      3 + 1j,  1]
], dtype=complex)

# Matrice A'5 (3x3) - Minor principal extrait de A5
Ap5 = np.array([
    [ 1,  1j,  2],
    [ 1j,  1,  3 + 1j],
    [-2,  3 + 1j,  1]
], dtype=complex)

# -----
# 2. Fonction de calcul du Range Numerique (Generalise)
# -----
def compute_numerical_range(matrix, num_samples=200000):
    """
    Calcule le range numerique par echantillonnage de vecteurs aleatoires
    uniformes
    sur la sphere unite complexe.
    """
    n = matrix.shape[0]

    # Generation de vecteurs complexes aleatoires normes
    # (Parties reelles et imaginaires tirees d'une distribution normale)
    z = (np.random.randn(n, num_samples) + 1j * np.random.randn(n, num_samples))
    z /= np.linalg.norm(z, axis=0)

    # Produit scalaire <Ax, x> = x^* * A * x
    # On effectue le calcul de maniere optimisee pour les gros volumes de
    donnees
    az = np.dot(matrix, z)
    points = np.sum(np.conj(z) * az, axis=0)

    return points

# -----
# 3. Calcul et generation des graphiques
# -----
print("Calcul des points en cours... Veuillez patienter quelques secondes.")
points_A5 = compute_numerical_range(A5, num_samples=300000)
points_Ap5 = compute_numerical_range(Ap5, num_samples=300000)

# Calcul des valeurs propres pour afficher les points noirs de controle
vals_A5 = np.linalg.eigvals(A5)
vals_Ap5 = np.linalg.eigvals(Ap5)

# Trace des figures
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))

# Graphique de gauche (Structure liee a A5)
ax1.scatter(points_A5.real, points_A5.imag, s=0.05, c='red', alpha=0.5)
ax1.scatter(vals_A5.real, vals_A5.imag, color='black', s=20, zorder=5,
label='Valeurs propres')

```

```
ax1.set_title(r"Nuage du Range de  $A_5$ ")
#ax1.set_xlim([-3, 5])
#ax1.set_ylim([-3, 5])
ax1.grid(True, linestyle='--', alpha=0.5)
ax1.set_aspect('equal')

# Graphique de droite (L'ovale pour A_5)
ax2.scatter(points_Ap5.real, points_Ap5.imag, s=0.05, c='red', alpha=0.5)
ax2.scatter(vals_Ap5.real, vals_Ap5.imag, color='black', s=20, zorder=5,
label='Valeurs propres')
ax2.set_title(r"Ovale du Range de  $A_5$ ")
#ax2.set_xlim([-3, 5])
#ax2.set_ylim([-3, 5])
ax2.grid(True, linestyle='--', alpha=0.5)
ax2.set_aspect('equal')

plt.tight_layout()
plt.show()
```