```
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([
    [ 0,  0,   1,  0],
    [ 0,  0,   0,  1],
    [-2, -1,  1j, 5j],
    [-1, -2, -5j, 1j]
], dtype=complex)

# ------------------------------------------------------------
# 2. Fonction de calcul du Range Numerique (Generalise)
# ------------------------------------------------------------
def compute_numerical_range(matrix, num_samples=200000):
    n = matrix.shape[0]
    z = (np.random.randn(n, num_samples) + 1j * np.random.randn(n, num_samples))
    z /= np.linalg.norm(z, axis=0)
    az = np.dot(matrix, z)
    points = np.sum(np.conj(z) * az, axis=0)
    return points

def compute_block_numerical_range(matrix, block_sizes, num_samples=150000):
    n = matrix.shape[0]
    points = []
    cum_sizes = np.insert(np.cumsum(block_sizes), 0, 0)
    for _ in range(num_samples):
        parts = []
        for i in range(len(block_sizes)):
            size = block_sizes[i]
            z_part = np.random.randn(size) + 1j * np.random.randn(size)
            z_part /= np.linalg.norm(z_part)
            parts.append(z_part)
        V = np.zeros((n, len(block_sizes)), dtype=complex)
        for i in range(len(block_sizes)):
            V[cum_sizes[i]:cum_sizes[i+1], i] = parts[i]
        A_compressed = np.dot(np.conj(V).T, np.dot(matrix, V))
        eigvals = np.linalg.eigvals(A_compressed)
        points.extend(eigvals)
    return np.array(points)

print("Calcul des points en cours... Veuillez patienter quelques secondes.")
#points_A5 = compute_block_numerical_range(A5, [2,2], num_samples=300000)
points_A5 = compute_block_numerical_range(A5, [2,1], num_samples=150000)
vals_A5 = np.linalg.eigvals(A5)

fig, (ax) = plt.subplots(1, 1, figsize=(14, 6))

ax.scatter(points_A5.real, points_A5.imag, s=0.05, c='red', alpha=0.5)
ax.scatter(vals_A5.real, vals_A5.imag, color='black', s=20, zorder=5,
label='Valeurs propres')
ax.set_title(r"Nuage du Range de $\mathcal{A}_5$")
#ax.set_xlim([-3, 5])
#ax.set_ylim([-3, 5])
ax.grid(True, linestyle='--', alpha=0.5)
ax.set_aspect('equal')

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