## Geometrisierung von Flächenmomenten

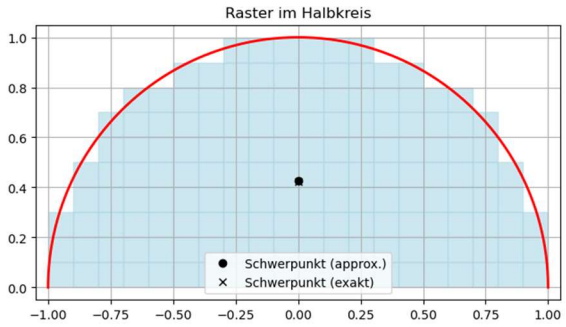
Mögliches Vorgehen, um Python-Code ONLINE laufen zu lassen
https://jupyter.org/try-jupyter/lab/
dort: file -> new notebook -> eine *.ipynb-Datei entsteht -> Python-Quellcode hineinkopieren -> run!

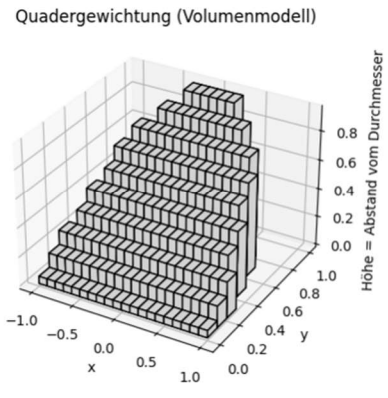
alternativ: Matplotlib Code Online Runner – Create and Visualize Python Charts Online
https://matplotlib.codeutility.io/ dort Python-Datei hochladen -> run

---

Geometrisierung_von_Flaechenmomenten.py

```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

def halbkreis_quader(n):
    r = 1.0
    dx = r / n
    dy = dx
    A_q = dx * dy

    x_vals = np.arange(-r, r, dx)
    y_vals = np.arange(0, r, dy)

    A_sum = 0.0
    V_sum = 0.0

    quader = []

    for x in x_vals:
        for y in y_vals:
            xm = x + dx / 2
            ym = y + dy / 2

            # Mittelpunkt im Halbkreis?
            if xm**2 + ym**2 <= r**2:
                A_sum += A_q
                V_sum += ym * A_q
                quader.append((x, y, ym))

    # Exakte Werte
    A_exakt = 0.5 * np.pi * r**2
    y_s_exakt = 4 / (3 * np.pi)

    # Approximationen
    y_s_approx = V_sum / A_sum

    # Prozentwerte
    A_prozent = 100 * A_sum / A_exakt
    y_prozent = 100 * y_s_approx / y_s_exakt
```



```
    # -------- Ausgabe --------
    print(f"n = {n}")
    print(f"dx = dy = {dx:.6f}")
    print("\n--- Fläche ---")
    print(f"A approx = {A_sum:.6f}")
    print(f"A exakt  = {A_exakt:.6f}")
    print(f"Verhältnis = {A_prozent:.2f} %")

    print("\n--- Schwerpunkt / Quader ---")
    print(f"y_S approx = {y_s_approx:.6f}")
    print(f"y_S exakt  = {y_s_exakt:.6f}")
    print(f"Verhältnis = {y_prozent:.2f} %")

    # -------- 2D-Grafik --------
    fig = plt.figure(figsize=(12, 4))

    ax1 = fig.add_subplot(1, 2, 1)
    theta = np.linspace(0, np.pi, 400)
    ax1.plot(np.cos(theta), np.sin(theta), 'r', lw=2)

    for (x, y, _) in quader:
        rect = plt.Rectangle((x, y), dx, dy,
                             color='lightblue', alpha=0.6)
        ax1.add_patch(rect)

    ax1.plot(0, y_s_approx, 'ko', label="Schwerpunkt (approx.)")
    ax1.plot(0, y_s_exakt, 'kx', label="Schwerpunkt (exakt)")
    ax1.set_aspect('equal')
    ax1.set_xlim(-1.05, 1.05)
    ax1.set_ylim(-0.05, 1.05)
    ax1.set_title("Raster im Halbkreis")
    ax1.grid(True)
    ax1.legend()


    # -------- 3D-Grafik (Quader) --------
    ax2 = fig.add_subplot(1, 2, 2, projection='3d')

    #for (x, y, h) in quader:
    #    ax2.bar3d(x, y, 0, dx, dy, h,
    #              shade=True, alpha=0.7)
    for (x, y, h) in quader:
        ax2.bar3d(
            x, y, 0,
            dx, dy, h,
            #color='gray',
            color='lightgray',
            edgecolor='black',
            #alpha=0.8,
            alpha=1.0,
            #shade=True
            shade=False
        )

    ax2.set_xlabel("x")
    ax2.set_ylabel("y")
    ax2.set_zlabel("Höhe = Abstand vom Durchmesser")
    ax2.set_title("Quadergewichtung (Volumenmodell)")

    plt.tight_layout()
    plt.show()


# ----------------------------
# Programmeinstieg
# ----------------------------
if __name__ == "__main__":
    n = int(input("Gib n ein (Anzahl der Unterteilungen von r): "))
    halbkreis_quader(n)
```

```
Gib n ein (Anzahl der Unterteilungen von r):
```

```
Gib n ein (Anzahl der Unterteilungen von r): 10
n = 10
dx = dy = 0.100000

--- Fläche ---
A approx = 1.580000
A exakt  = 1.570796
Verhältnis = 100.59 %

--- Schwerpunkt / Quader ---
y_S approx = 0.425949
y_S exakt  = 0.424413
Verhältnis = 100.36 %
```