In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy
import math
```

In [ ]:
```
def RBFkernel(x1: float, x2: float, sigma_sq: float = 1):
    """
    RBF kernel.
    """
    pow = -1/(2*sigma_sq) * (x1 - x2)**2
    return math.exp(pow)


def cov_mat(x1: np.ndarray, x2: np.ndarray, ker: callable, sigma_sq: float = 1) -> np.ndarray:
    """
    Returns the Covraiance matrix for the given kernel.
    """
    n = max(x1.shape)
    cov = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            cov[i][j] = ker(x1[i], x2[j], sigma_sq)

    return cov
```

In [ ]:
```
plt.figure(figsize=[9, 12])
plt.rcParams.update({
    'font.size': 8,
    'text.usetex': True,
    'text.latex.preamble': r'\usepackage{amsfonts}'
})
```

In [ ]:
```
n_points = 100
n_draws = 20
interval = (0, 1)
xs = np.linspace(interval[0], interval[1], n_points)#.reshape(-1, 1)
Si = cov_mat(xs, xs, ker = RBFkernel, sigma_sq=0.12)
mean = np.zeros(n_points)
ys = np.random.multivariate_normal(mean=mean, cov=Si, size=n_draws)#.reshape(-1, 1)

for i in range(n_draws):
    plt.plot(xs, ys[i])

plt.grid()
plt.gca().set_facecolor((0.9, 0.9, 0.9))
plt.title(r"20 Sampled Functions From $\mathcal{G}(\mu, k)$")
```