Listing 1 ”Kernel Fourier transforms”

```python
import matplotlib.pyplot as plt
import numpy as np
import math

from matplotlib import rc
rc('font',**{'family':'sans-serif','sans-serif':['Helvetica']})
## for Palatino and other serif fonts use:
#rc('font',**{'family':'serif','serif':['Palatino']})
rc('text', usetex=True)

# del matplotlib.font_manager.weight_dict['roman']
# matplotlib.font_manager._rebuild()

def Epanechnikov_Kernel_FT(u):
    result = []
    for _u in u:
        if abs(_u) < 0.01:
            v = 1
        else:
            v = 3 * ( np.sin(_u) - _u * np.cos(_u) ) / pow(_u, 3)
        result.append(v)
    return result

def Spline_Kernel_FT(u, beta):
    result = []
    for _u in u:
        v =  1 / ( 1 + pow(abs(_u), beta))
        result.append(v)
    return result

def Pinsker_Kernel_FT(u, beta):
    result = []
    for _u in u:
        v = max(0, ( 1 - pow(abs(_u), beta)))
        result.append(v)
    return result

def Silverman_Kernel_FT(u):
    result = []
    for _u in u:
        v =  1 / ( 1 + pow(_u, 4))
        result.append(v)
    return result

def Pinsker_Kernel(u):
    result = []
    for _u in u:
        if abs(_u) < 0.01:
            v = 2 / (3 * np.pi)
        else:
            v = 2 * ( np.sin(_u) - _u * np.cos(_u) ) / ( np.pi * pow(_u, 3) )
        result.append(v)
    return result


def main():
    plt.style.use('_mpl-gallery')

    x_min = -20
    x_max =  20
    x_pitch = 0.01

    u = np.arange(x_min, x_max, x_pitch)

    ft_Epane = Epanechnikov_Kernel_FT(u)
    ft_Silverman = Silverman_Kernel_FT(u)

    beta = 2
    ft_Spline = Spline_Kernel_FT(u, beta)
    ft_Pinsker = Pinsker_Kernel_FT(u, beta)

    fig, ax = plt.subplots()

    ax.set_title('Kernel FT', fontsize=18)
    ax.set_xlabel(r'$\omega$', fontsize=14)
    ax.set_ylabel(r'$\hat{K}(\omega)$', fontsize=14)

    ax.plot(u, ft_Epane, color='b', linewidth=3.0, label="Epanechnikov Kernel [p25]")
    ax.plot(u, ft_Silverman, color='g', linewidth=3.0, label="Silverman Kernel [p27]")
    ax.plot(u, ft_Spline, color='r', linewidth=3.0, label="Spline Kernel (1.55) [p27]")
    ax.plot(u, ft_Pinsker, color='m', linewidth=3.0, label="Pinsker Kernel (1.56) [p27]")

    x_tick = 1
    y_tick = 0.1

    ft = ft_Epane + ft_Silverman + ft_Spline + ft_Pinsker

    y_min = math.floor(min(ft)*10) / 10 - y_tick
    y_max = math.ceil(max(ft)*10) / 10 + y_tick

    ax.set(xlim=(x_min, x_max), xticks=np.arange(x_min, x_max, x_tick), ylim=(y_min, y_max), yticks=np.arange(y_min, y_max, y_tick))
    plt.subplots_adjust(left=0.1, bottom=0.1, top=0.9)

    ax.axhline(0, color='k', linewidth=1.0)
    ax.axvline(0, color='k', linewidth=1.0)

    plt.legend(fontsize=14)

    ep_tex = r'\begin{eqnarray*}\hat{K}(\omega) = \frac{3}{\omega^3}(\sin\omega - \omega\cos\omega)\end{eqnarray*}'
    ax.text(x_min + ( x_max - x_min) / 2 + 7 * x_tick, y_min + (y_max - y_min )/ 2 + y_tick, ep_tex, color="b", fontsize=20)

    slv_tex = r'\begin{eqnarray*}\hat{K}(\omega) = \frac{1}{1 + \omega^4}\end{eqnarray*}'
    ax.text(x_min + ( x_max - x_min) / 2 + 7 * x_tick, y_min + (y_max - y_min )/ 2, slv_tex, color="g", fontsize=20)

    spl_tex = r'\begin{eqnarray*}\hat{K}(\omega) = \frac{1}{1 + |\omega|^{\beta(=2)}}\end{eqnarray*}'
    ax.text(x_min + ( x_max - x_min) / 2 + 7 * x_tick, y_min + (y_max - y_min )/ 2 - y_tick, spl_tex, color="r", fontsize=20)

    pin_tex = r'\begin{eqnarray*}\hat{K}(\omega) = (1 - |\omega|^{\beta(=2)})_{+}\end{eqnarray*}'
    ax.text(x_min + ( x_max - x_min) / 2 + 7 * x_tick, y_min + (y_max - y_min )/ 2 - 2 * y_tick, pin_tex, color="m", fontsize=20)

    plt.show()

if __name__ == "__main__":
    main()
```