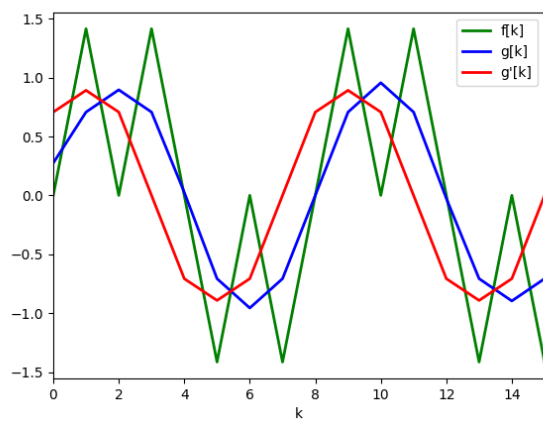
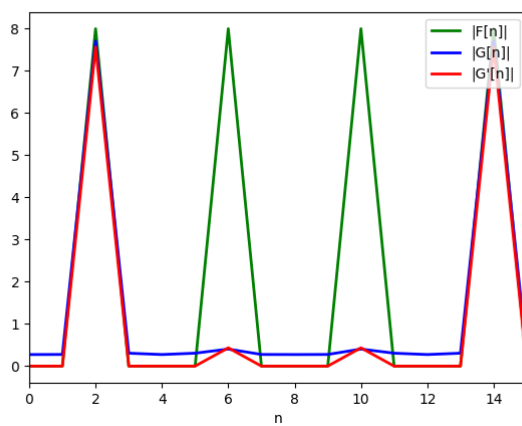
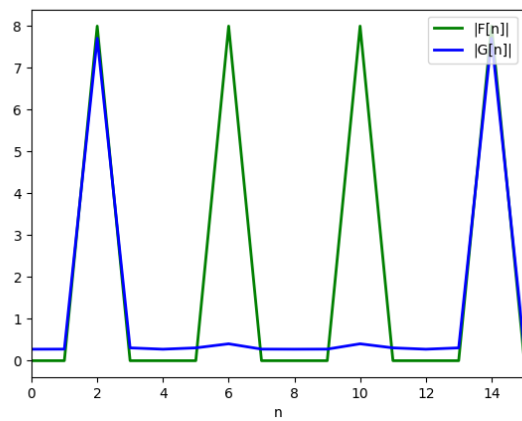


## デジタル信号処理課題

氏名：園山佳典

学籍番号：26002201991

結果



ソースコード

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

# パラメータ
T = 1
w1 = 0.25 * np.pi / T
w2 = 0.75 * np.pi / T
wc = 0.50 * np.pi / T
L = 7
N = 15
M = 16
k1 = np.arange(start=0 - 7, stop=N - 6, step=1) # h[k]のとり範囲
k2 = np.arange(start=0, stop=N + 1, step=1) # f[k]のとり範囲(1 つ目の plot
のとり範囲)
h = (wc / np.pi) * np.sinc(wc * k1 * T / np.pi) # h[k]の関数
f = np.sin(w1 * k2 * T) + np.sin(w2 * k2 * T) # f[k]の関数
g = [0] * (2 * N)
g1 = [0] * (N + 1)
# 畳み込みの公式再現
for i in range(2 * N):
    g0 = 0
    for j in range(2 * N):
        if (i - j < 0) or (i - j > N) or (j > N):
            g0 += 0
        if ((i - j < 0) and (i - j > N)) or ((j > N) and (i - j < 0)):
            g0 += 0
        if (i - j >= 0) and (j <= N) and (i - j <= N):
            g0 += f[j] * h[i - j]
        g[i] = g0
    if i >= 7 and i < 23:
        g1[i - 7] = g0

# DFT 行列を計算
n = np.arange(start=0, stop=M, step=1)
```

```

w0 = np.exp(-1j * 2 * np.pi / M)
mtx_dft = np.zeros((M, M)) + 1j * np.zeros((M, M))
for i in range(M):
    for j in range(M):
        mtx_dft[i, j] = w0 ** (i * j)

# f[k]の長さがNより短い場合、ゼロを詰める
f_fft = f
f_ex = np.zeros((M - f.size))
if not len(f_ex) == 0:
    f_fft = np.concatenate([f, f_ex])
# 行列演算でDFTを行う(9回目の講義資料に参考)
F = np.dot(mtx_dft, f_fft)
# スペクトル F[n]は複素数
# 絶対値で振幅スペクトル|F[n]|を計算
F_amp = abs(F)

# g1[k]の長さがNより短い場合、ゼロを詰める
g_ggt = g1
g_ex = np.zeros((M - len(g1)))
if not len(g_ex) == 0:
    g_ggt = np.concatenate([g1, g_ex])
# 行列演算でDFTを行う(9回目の講義資料に参考)
G = np.dot(mtx_dft, g_ggt)
# スペクトル G[n]は複素数
# 絶対値で振幅スペクトル|G[n]|を計算
G_amp = abs(G)

# h[k]の長さがNより短い場合、ゼロを詰める
h_hht = h
h_ex = np.zeros((M - h.size))
if not len(h_ex) == 0:
    h_hht = np.concatenate([h, h_ex])
# 行列演算でDFTを行う(9回目の講義資料に参考)
H = np.dot(mtx_dft, h_hht)
# スペクトル F[n]は複素数

```

```

# 絶対値で振幅スペクトル $|F[n]|$ を計算
H_amp = abs(H)

Gd = F * H
Gd_amp = F_amp * H_amp
gd = np.fft.ifft(Gd)

# f[k],g[k],g'[k]の波形を描画
plt.figure()
# 横軸 k、縦軸 f[k]で描画
plt.plot(k2, f, color="g", linewidth=2, label="f[k]")
# 横軸 k、縦軸 g[k]で描画
plt.plot(k2, g1, color="b", linewidth=2, label="g[k]")
# 横軸 k、縦軸 gd[k]で描画
plt.plot(k2, gd, color="r", linewidth=2, label="g'[k]")
# 軸の範囲を設定
plt.xlim((0, N))
# 横軸のラベルを k に設定
plt.xlabel("k")
# 凡例
plt.legend(loc="upper right")
plt.show()

# F[n],G[n]の波形を描画
plt.figure()
# 横軸 n、縦軸  $|F[n]|$ ,  $|G[n]|$ で描画
plt.plot(n, F_amp, color="g", linewidth=2, label="|F[n]|")
plt.plot(n, G_amp, color="b", linewidth=2, label="|G[n]|")
# 横軸の範囲を 0~M-1 に設定
plt.xlim((0, M - 1))
# 横軸のラベルを n に設定
plt.xlabel("n")
# 凡例
plt.legend(loc="upper right")
plt.show()

```

```

# |F[n]|, |G[n]|, |G'[n]|の波形を描画
plt.figure()
# 横軸 n、縦軸 |F[n]|, |G[n]|, |G'[n]|で描画
plt.plot(n, F_amp, color="g", linewidth=2, label="|F[n]|")
plt.plot(n, G_amp, color="b", linewidth=2, label="|G[n]|")
plt.plot(n, Gd_amp, color="r", linewidth=2, label="|G'[n]|")
# 横軸の範囲を 0~M-1 に設定
plt.xlim((0, M - 1))
# 横軸のラベルを n に設定
plt.xlabel("n")
# 凡例
plt.legend(loc="upper right")
plt.show()

```

## 考察

$G'[n]$  と  $|G[n]|$  の誤差や  $g[k]$  と  $g'[k]$  の位相のずれは離散フーリエ変換 (DFT) における近似誤差と、畳み込みの近似による誤差だと考えられる。DFT は連続時間信号に対する離散時間近似なので、無限長の信号に対して厳密な周波数成分を求めることはできず、有限長にして計算するため周波数成分の振幅や位相に誤差が生じたと考えられる。また畳み込みも無限長の信号同士での計算だが、実際の計算では有限長の信号を扱っているため、無限長の信号を有限長で近似する必要があり、畳み込みの結果には誤差が生じたと推測する。

