Optimization-Based Data Analysis
Homework assignment 3

Luca Venturi

October 27, 2017

Exercise 1

(a) - i. Since f is real, it holds

$$F[k] = \langle f, h_k \rangle = \langle \overline{f}, h_k \rangle = \overline{\langle f, \overline{h_k} \rangle} = \overline{\langle f, h_{-k} \rangle} = \overline{F[-k]}.$$

Moreover, since f is even, it holds that

$$F[-k] = \int_{-1/2}^{1/2} f(t) e^{i2\pi kt}\, dt = \int_{-1/2}^{1/2} f(-t) e^{i2\pi kt}\, dt \overset{s=-t}{=} \int_{-1/2}^{1/2} f(s) e^{-i2\pi ks}\, dt = F[k].$$

This proves that $F[k] = \overline{F[k]}$, i.e. $F[k]$ is real.

(a) - ii. By the previous result, it holds that

$$\begin{aligned} S\{f\}(t) &= F[0] + \sum_{k=1}^{\infty} \Big(F[k] e^{i2\pi kt} + F[-k] e^{-i2\pi kt} \Big) = F[0] + \sum_{k=1}^{\infty} F[k] \Big(e^{i2\pi kt} + e^{-i2\pi kt} \Big) \\ &= F[0] + 2 \sum_{k=1}^{\infty} F[k] \cos(2\pi kt) = \sum_{k=0}^{\infty} a_k \cos(2\pi kt), \end{aligned}$$

where $a_0 = F[0]$ and $a_k = F[k]$ for $k \geq 1$.

(b) Notice that

$$f(t) = \frac{1}{2} \big(e^{i2\pi t} e^{i2\pi\varphi} + e^{-i2\pi t} e^{-i2\pi\varphi} \big).$$

If follows that $F[1] = \frac{1}{2} e^{i2\pi\varphi}$, $F[-1] = \frac{1}{2} e^{-i2\pi\varphi}$ and $F[k] = 0$ if $k \neq 1, -1$.

(c) Notice that

$$\int_{-1/2}^{1/2} D_n(t) f(t)\, dt = \langle D_n, \overline{f} \rangle = \sum_{k=-n}^{n} \sum_{j=-\infty}^{\infty} F[j] \langle h_k, h_j \rangle = \sum_{k=-n}^{n} F[k] = S_n\{f\}(0),$$

where S_n denotes the partial Fourier sum. Since $S_n\{f\}(t) \to f(t)$ for every t as $n \to \infty$, it follows that

$$\int_{-1/2}^{1/2} D_n(t) f(t)\, dt \to S_n\{f\}(0)$$

as $n \to \infty$.

Exercise 2

The code for part (a) and (b) is contained in the file `timedata/freq.py`.

(a) Figures 1, 2 and 3 show the plots of the magnitudes of the discrete Fourier coefficients for, respectively, $N = 2049, 4097, 8193$. We can observe that the coefficients for any frequency k with $|k| > 2048$ are zero. Moreover the largest magnitude coefficients appear at frequency $k = 0$.

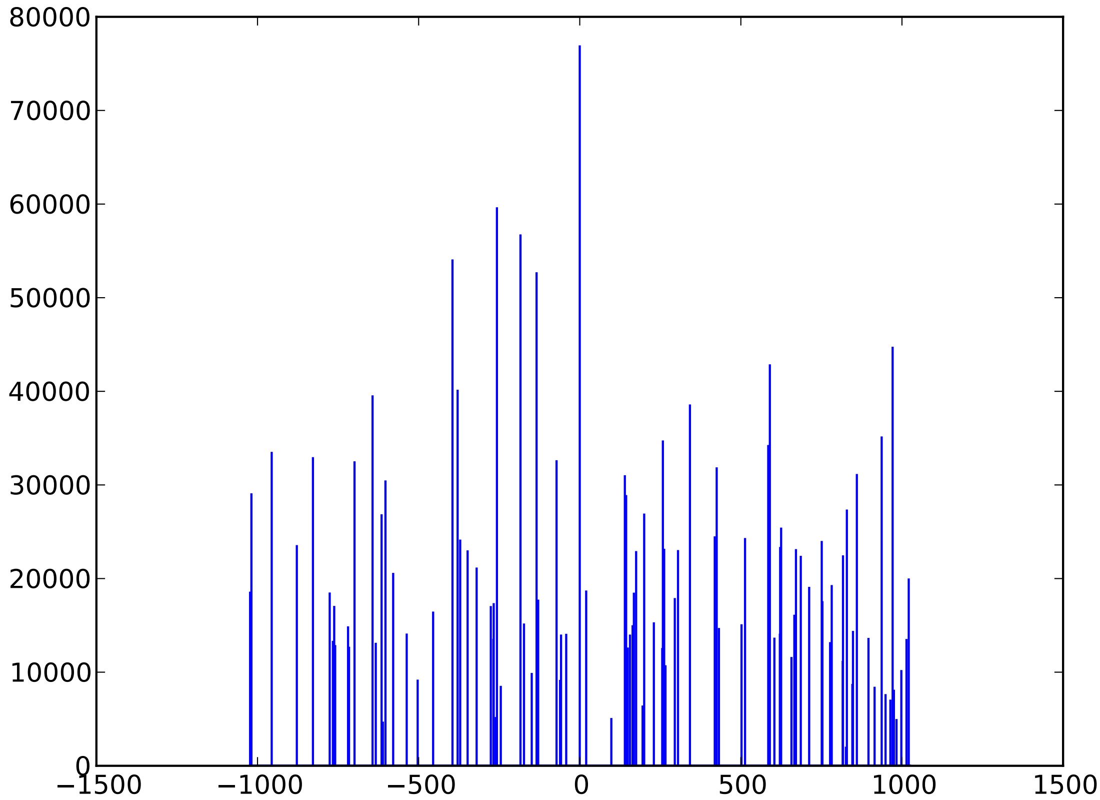


Figure 1

(b) The values of a_k with largest magnitude are

$$37.500000000000078, \quad 29.060260498971161, \quad 27.645665925813127$$

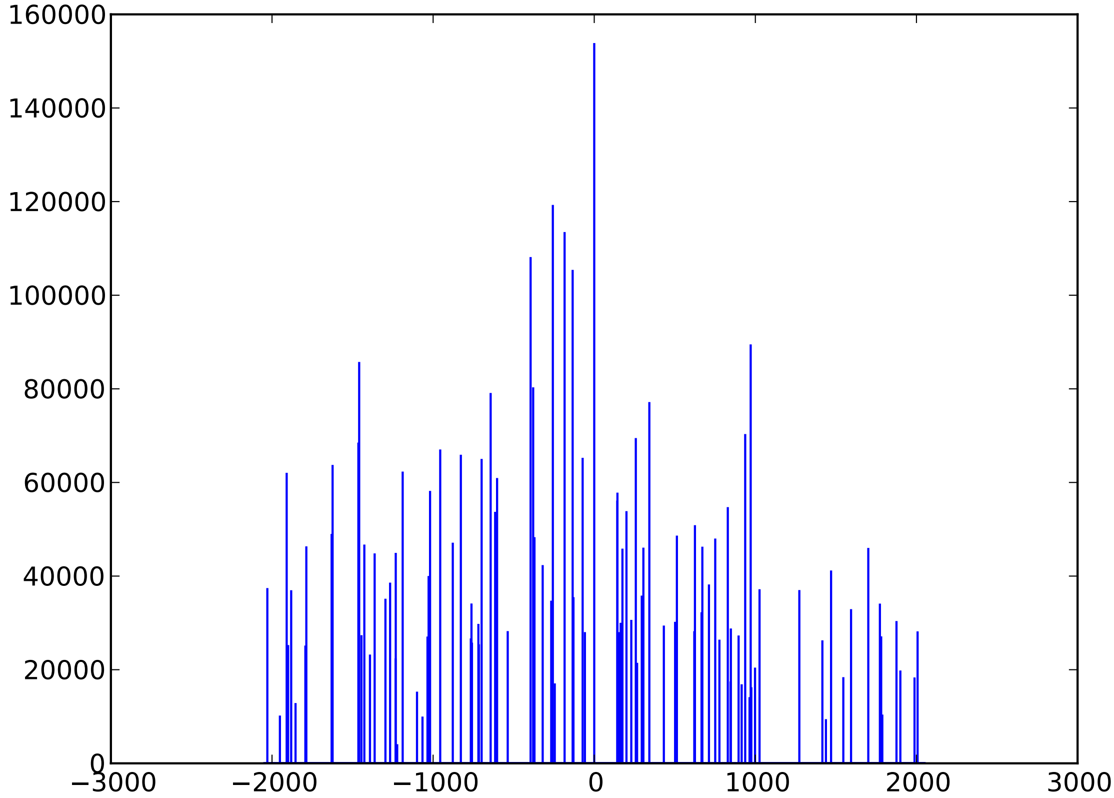
corresponding to the frequencies $0, 257, 184$.

(c) Let's call $f^{(n)} = (f(0), f(1/n), \ldots, f((n-1)/n))^T$. By definition

$$F^{(n)} = F_n^* f^{(n)}, \quad \text{where} \quad F_n \doteq \left[h_0^{[n]} \cdots h_{n-1}^{[n]} \right].$$

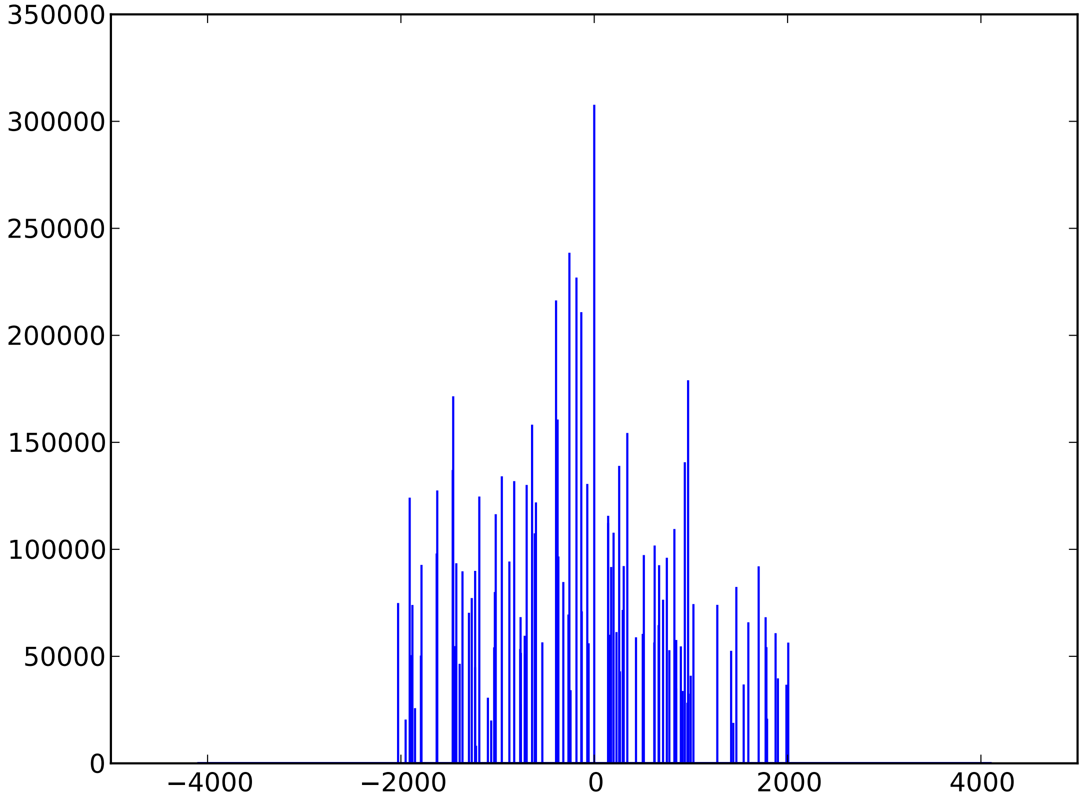
Since $k_c \leq 4096$, all the Fourier coefficients of f are $F = (F[-4096], \ldots, F[4096])^T$. In particular we know that

$$f^{(n)} = \left[h_{-4096}^{[n]} \cdots h_{4096}^{[n]} \right] F.$$



160000
140000
120000
100000
80000
60000
40000
20000
0
−3000
−2000
−1000
0
1000
2000
3000

Figure 2



350000
300000
250000
200000
150000
100000
50000
0
−4000
−2000
0
2000
4000

Figure 3

In particular

$$F^{(8193)} = F_{8193}^* f^{(8193)} = \begin{bmatrix} \left(\overline{h_0^{[8193]}}\right)^T \\ \vdots \\ \left(\overline{h_{8192}^{[8193]}}\right)^T \end{bmatrix} \begin{bmatrix} h_{-4096}^{[8193]} \cdots h_{4096}^{[8193]} \end{bmatrix} F = \begin{bmatrix} \mathbf{0} & I_{4097} \\ I_{4096} & \mathbf{0} \end{bmatrix} F,$$

where I_n is the identity matrix of dimension n. It follows that

$$F^{(2049)} = F_{2049}^* f^{(2049)} = \begin{bmatrix} \left(\overline{(h_0^{[2049]}}\right)^T \\ \vdots \\ \left(\overline{h_{2048}^{[2049]}}\right)^T \end{bmatrix} \begin{bmatrix} h_{-4096}^{[2049]} \cdots h_{4096}^{[2049]} \end{bmatrix} F$$

$$= \begin{bmatrix} \left(\overline{(h_0^{[2049]}}\right)^T \\ \vdots \\ \left(\overline{h_{2048}^{[2049]}}\right)^T \end{bmatrix} \begin{bmatrix} h_0^{[2049]} \cdots h_{2048}^{[2049]} h_0^{[2049]} \cdots h_{2048}^{[2049]} h_0^{[2049]} \cdots h_{2048}^{[2049]} h_0^{[2049]} \cdots h_{2048}^{[2049]} \end{bmatrix} \tilde{F}$$

where $\tilde{F} \doteq (0, 0, F[-4096], \ldots, F[4096], 0)^T = \begin{bmatrix} \mathbf{0} | \mathbf{0} | I_{8913} | \mathbf{0} \end{bmatrix}^T F$. The above formulas then imply that

$$F^{(2049)} = \begin{bmatrix} I_{2049} | I_{2049} | I_{2049} | I_{2049} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ I_{8913} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{0} & I_{4096} \\ I_{4097} & \mathbf{0} \end{bmatrix} F^{(8193)}.$$

The above formula can be probably written in a smarter way.

(d) False: it is true if we already know that $k_c \leq 4096$, otherwise we can not conclude anything from this fact.

(e) True: using the result taken from the notes, it holds that

$$\begin{aligned} \int_0^1 f(t) e^{-2\pi i k t}\, dt &= \frac{1}{M} \sum_{j=0}^{M-1} f(j/M) \int_0^1 d_{[j/M]}(t) e^{-2\pi i k t}\, dt \\ &= \frac{1}{M} \sum_{j=0}^{M-1} f(j/M) \int_0^1 d(t - j/M) e^{-2\pi i k t}\, dt \\ &= \frac{1}{M} \sum_{j=0}^{M-1} f(j/M) \int_0^1 d(s) e^{-2\pi i k s} e^{-2\pi i k j/M}\, ds \\ &= \frac{1}{M} \sum_{j=0}^{M-1} f(j/M) e^{-2\pi i k j/M} \langle d, h_k \rangle = \frac{1}{M} \sum_{j=0}^{M-1} f(j/M) e^{-2\pi i k j/M}. \end{aligned}$$

where d is a Dirichlet kernel with cut-off frequency k_c and we used the change of variable $s = t - j/M$.

Exercise 3

(a) Be $j = 0, \ldots, m-1$. Then

$$W^{[m]}_{j,2k+1} = e^{-i2\pi j(2k+1)/m} = e^{-i2\pi j/m} e^{-i2\pi j(2k)/m} = e^{-i2\pi j/m} \cdot W^{[m]}_{j,2k}.$$

It follows that (using the notation from the notes)

$$W^{[m]}_{:,2k+1} = \overline{h^{[m]}_1} * W^{[m]}_{:,2k}$$

where $*$ denotes the component-wise product.

(b) Given $j = 0, \ldots, m/2 - 1$, we have

$$W^{[m]}_{j+m/2,2k} = e^{-i2\pi(j+m/2)(2k)/m} = e^{-i2\pi j(2k)/m} e^{-i2\pi mk} = e^{-i2\pi j(2k)/m} = W^{[m]}_{j,2k}.$$

Therefore $W^{[m]}_{0:m/2-1,2k} = W^{[m]}_{m/2:,2k}$.

(c) For every $j = 0, \ldots, m/2 - 1$, we have

$$W^{[m]}_{j,2k} = e^{-i2\pi j(2k)/m} = e^{-i2\pi jk/(m/2)} = W^{[m/2]}_{j,k}.$$

Therefore $W^{[m]}_{0:m/2-1,2k} = W^{[m/2]}_{:,k}$.

Exercise 4

(a) Given a vector x we denote its (discrete) Fourier transform by $\mathcal{F}(x)$. It holds that

$$\begin{aligned}\mathcal{F}((x * y)_{[m]})[k] &= e^{-i2\pi km} \mathcal{F}((x * y))[k] = e^{-i2\pi km} \mathcal{F}(x)[k]\mathcal{F}(y)[k] \\ &= \mathcal{F}(x_{[m]})[k]\mathcal{F}(y)[k] = \mathcal{F}(x_{[m]} * y)[k] \\ &= \mathcal{F}(x)[k]\mathcal{F}(y_{[m]})[k] = \mathcal{F}(x * y_{[m]})[k].\end{aligned}$$

Since their (discrete) Fourier transforms are equal, it follows that

$$(x * y)_{[m]} = x_{[m]} * y = x * y_{[m]}.$$

(b) The method is the following. Given $x, y \in \mathbb{R}^n$, let us define $\tilde{x}, \tilde{y} \in \mathbb{R}^{2n-1}$ as

$$\tilde{x}[k] = \begin{cases} x[k] & \text{if } k = 0, \ldots, n-1 \\ 0 & \text{if } k \geq n \end{cases}, \quad \tilde{y}[k] = \begin{cases} y[k] & \text{if } k = 0, \ldots, n-1 \\ 0 & \text{if } k \geq n \end{cases}.$$

Then the claim is that

$$(x *_L y)[k] = (\tilde{x} * \tilde{y})[k], \quad \text{for } k = 0, \ldots, n-1. \tag{1}$$

If (1) holds, then one can compute $x *_L y$ by computing $\tilde{x} * \tilde{y}$, whose cost is $O((2n-1)\log(2n-1)) = O(n \log n)$ (using DFT). We only need to show that (1) holds. Given $k = 0, \ldots, n-1$, we have

$$\begin{aligned}(\tilde{x} * \tilde{y})[k] &= \sum_{j=0}^{k} \tilde{x}[j]\tilde{y}[k-j] + \sum_{j=k+1}^{2n-1} \tilde{x}[j]\tilde{y}[k-j+2n-1] \\ &= \sum_{j=0}^{k} x[j]y[k-j] + \sum_{j=k+1}^{n-1} \tilde{x}[j]\tilde{y}[k-j+2n-1] + \sum_{j=n}^{2n-1} \tilde{x}[j]\tilde{y}[k-j+2n-1] \\ &= (x *_L y)[k] + \sum_{j=k+1}^{n-1} x[j] \cdot 0 + \sum_{j=n}^{2n-1} 0 \cdot \tilde{y}[k-j+2n-1] = (x *_L y)[k].\end{aligned}$$

(c) My implementation is in the file `convolution/convolve.py`. Figure 4 shows the results obtained.

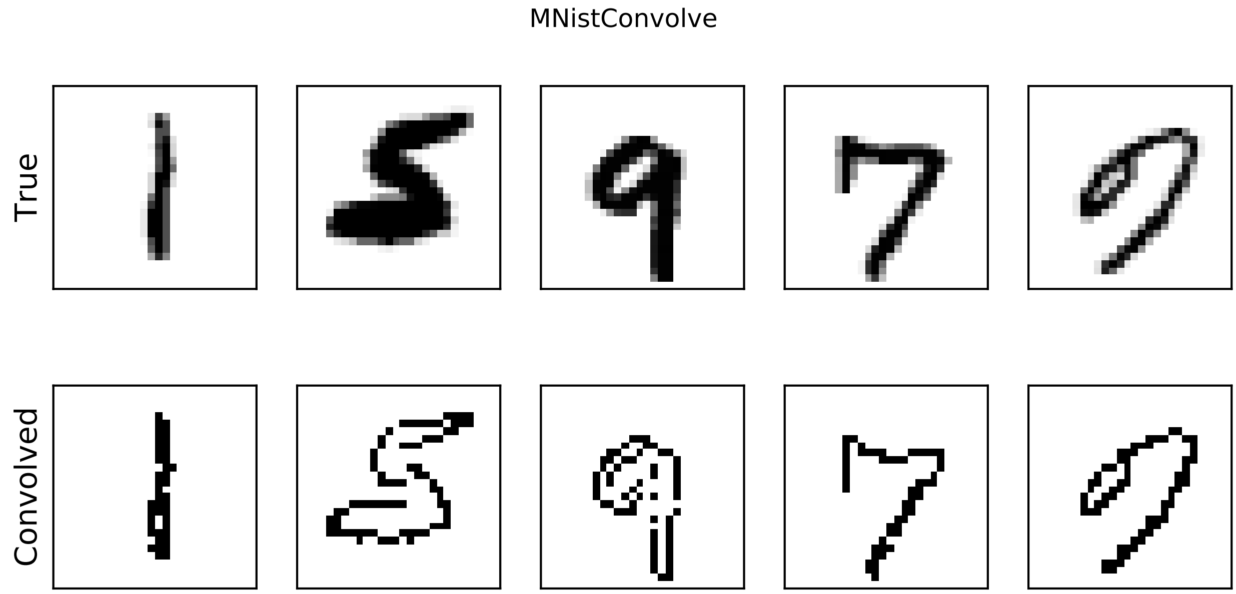


Figure 4

(d) The kernel x used in the previous part can assist in edge detection for the following reason. Suppose $y[i,j]$ is a pixel which not part of the edge (and suppose for simplicity that the image pixels have values $0,1$). Then $x[0,0]$ will give value one to $y[i,j]$ minus $1/8$ the sum of the values that surround $y[i,j]$. If $y[i,j]$ is not on edge (and it is part of the digit) then all the neighbors will also have value 1. Therefore x will give value 0 to y. On the other hand if $y[i,j]$ is on the edge its value will be bigger than 0, more explicitly 1 - 1/8 times the number of neighbors which are part of the digit.

Exercise 5

(a-b-c) The code of my implementation is contained in the file `wiener/wiener_filter.py`.

(d) Figures 5, 6 and 7 show the plots generated by the program.

Figure 5

Figure 6

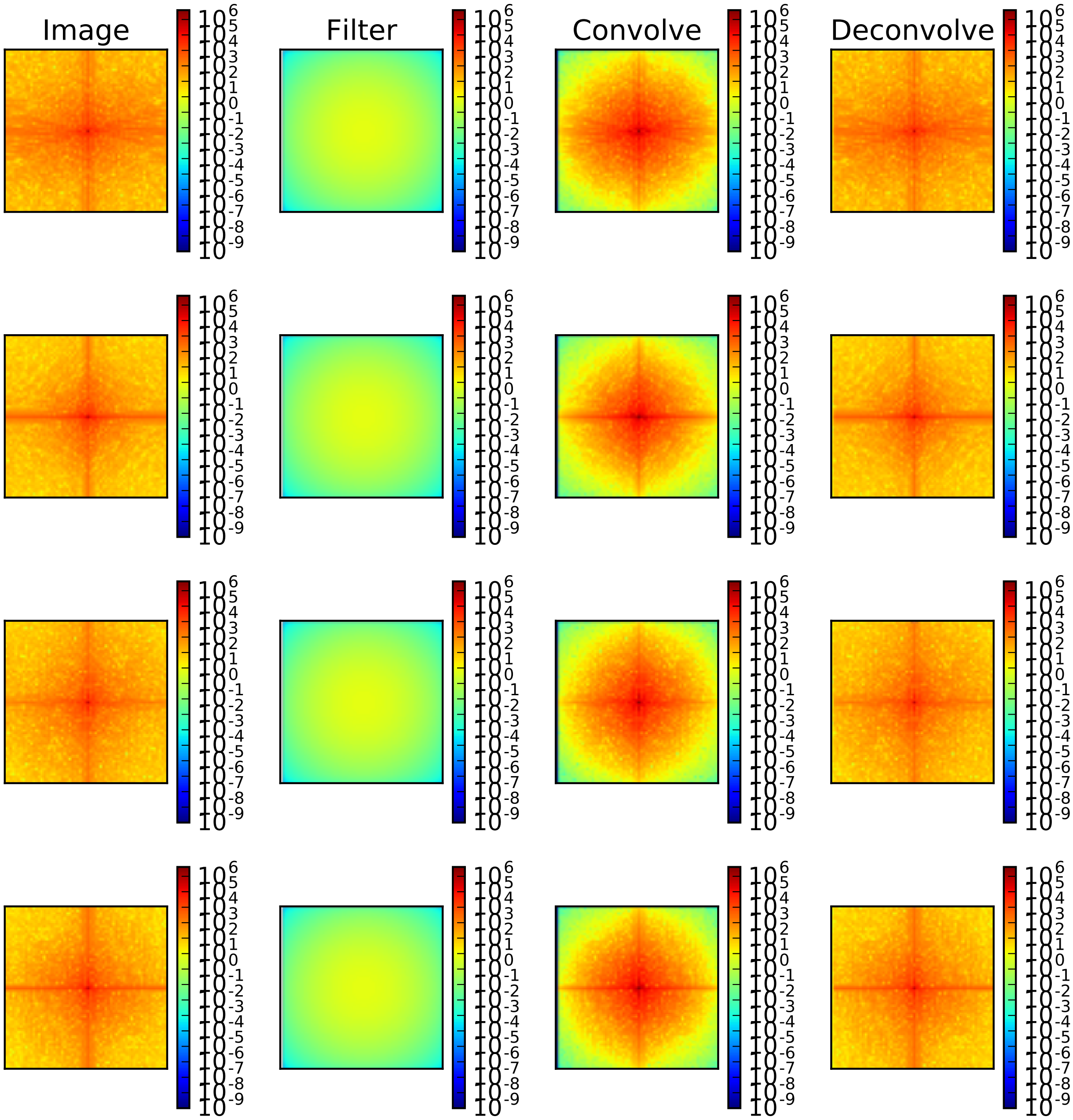


Figure 7