Homework#1: Short course on sparse recovery and compressed sensing

1. **Answer:** Three lectures were given today. The first lecture talks about some fundamental mathematical background which will be quite useful in the subsequent lectures. It mainly focuses on how to represent the signal of interest as a discrete linear combination of basis signals. For orthogonal basis expansions, the generalized Parseval's theorem shows that all signal processing can be done by manipulating discrete sequences of expansion coefficients. This lecture also talks about the Discrete Cosine Transform (DCT) with its applications in image and video compression, and it ends with non-orthogonal bases and overcomplete frames. The second lecture reviews the theory of sparse representations and its applications in image processing. A detailed performance comparison between DCT and wavelets in image approximation is implemented, which reveals the superior of wavelets. With sparse representations, more efficient algorithms can be put forward for applications such as compression, denoising, inverse problems, and acquisition. The last lecture introduces the problem of finding a sparse representation of a signal, and gives two very different frameworks of algorithms: greedy algorithms and convex programming. Greedy algorithms are based on the idea of matching pursuit, which aims to select a subset of a dictionary with which to represent a given signal. Convex programming relaxes the combinatorial problem into a closely related convex optimization program, Basis Pursuit, which can be recast as a linear program.

2. **Answer:**

   (a) According to the definition,

   $$\begin{aligned}(\Phi[\Phi^*[\alpha]])_k &= \langle \Phi^*[\alpha], \phi_k \rangle \\ &= \sum_{m=0}^{n} \alpha_m \langle \phi_m, \phi_k \rangle.\end{aligned}$$

   Therefore, $(\Phi\Phi^*)_{k,m} = \langle \phi_m, \phi_k \rangle$. It is easy to calculate that

   $$\langle \phi_m, \phi_k \rangle = \begin{cases} 1 & m = k \\ 1/4 & 1 \le m, k \le n-1, |m-k| = 1 \\ 1/(2\sqrt{2}) & km(n-k)(n-m) = 0, |m-k| = 1 \\ 0 & \text{otherwise} \end{cases}.$$

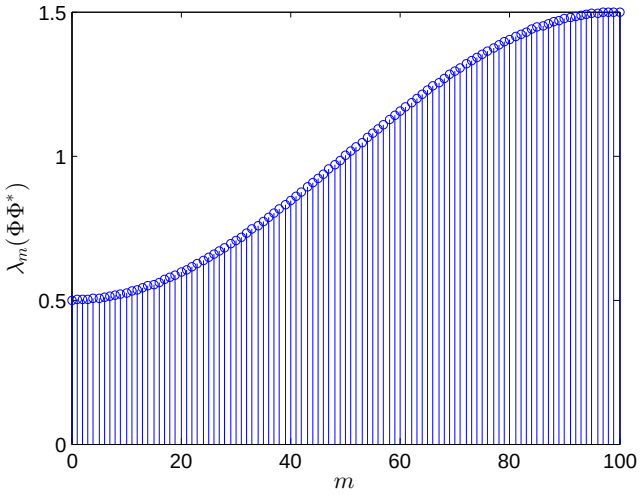
   (b) According to the properties of the adjoint operator,

   $$\begin{aligned}B &= \|\Phi\|^2 = \|\Phi\Phi^*\| = \lambda_{\max}(\Phi\Phi^*), \\ A &= \|\Phi^{-1}\|^{-2} = \|(\Phi\Phi^*)^{-1}\|^{-1} = \lambda_{\min}(\Phi\Phi^*).\end{aligned}$$

   Using this result, it is easy to calculate the frame bounds for this basis. For $n = 5$, 10, and 100, the bounds are all $A = 0.5$ and $B = 1.5$, and the printout of my code is

   ```
   for n = 5,   A = 0.5 and B = 1.5
   for n = 10,  A = 0.5 and B = 1.5
   for n = 100, A = 0.5 and B = 1.5
   ```

   For $n = 100$, the spectrum of $\Phi\Phi^*$ is



(c) According to the definition of the dual basis, the dual basis vectors satisfy

$$\langle \tilde{\phi}_k, \phi_j \rangle = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases} .$$

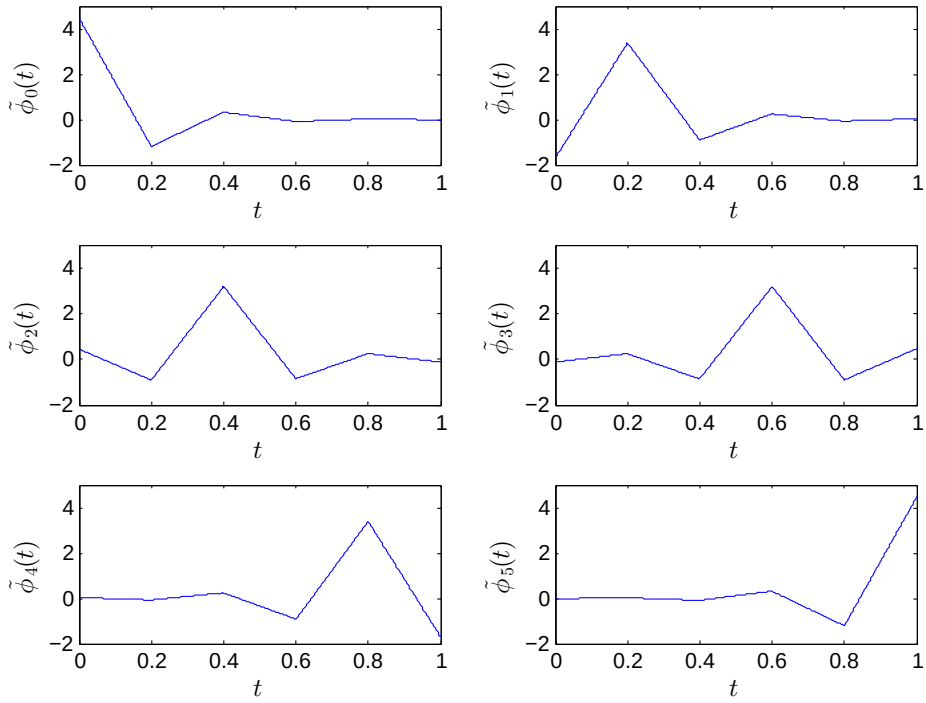
Therefore, if we set $\tilde{\phi}_k(t) = \sum_j \alpha_{k,j}\phi_j(t)$, the coefficients satisfy

$$(\Phi\Phi^*) \begin{bmatrix} \alpha_{k,0} \\ \alpha_{k,1} \\ \vdots \\ \alpha_{k,n} \end{bmatrix} = \mathbf{e}_{k+1},$$

where the entries of $\mathbf{e}_{k+1} \in \mathbb{R}^{n+1}$ are zeros expect the $(k+1)$th entry which takes value one. Since the matrix $\Phi\Phi^*$ is invertible, the coefficients can be calculated using Matlab. The coefficient vectors are

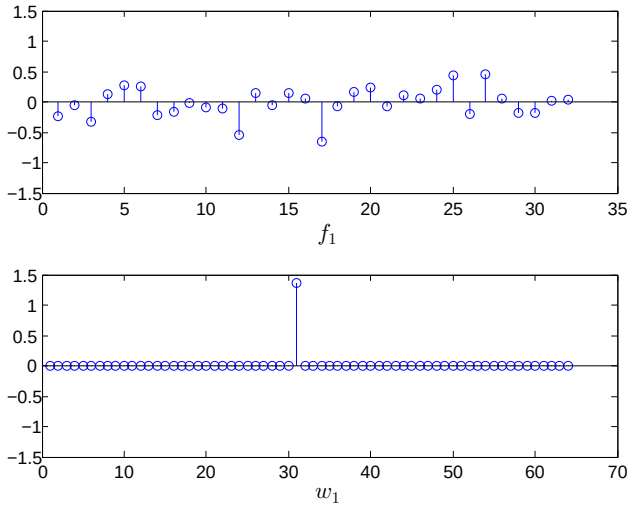
```
alpha_0 = [1.1547 -0.43757 0.11729 -0.031577 0.0090221 -0.0031898]
alpha_1 = [-0.43757 1.2376 -0.33174 0.089314 -0.025518 0.0090221]
alpha_2 = [0.11729 -0.33174 1.1611 -0.3126 0.089314 -0.031577]
alpha_3 = [-0.031577 0.089314 -0.3126 1.1611 -0.33174 0.11729]
alpha_4 = [0.0090221 -0.025518 0.089314 -0.33174 1.2376 -0.43757]
alpha_5 = [-0.0031898 0.0090221 -0.031577 0.11729 -0.43757 1.1547]
```

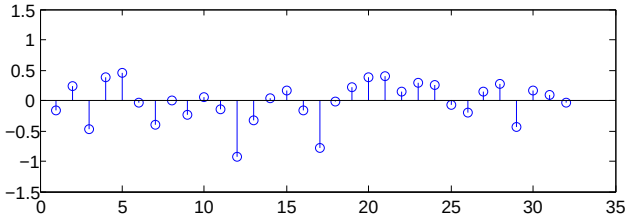
The plots of the dual basis vectors are shown as follows.



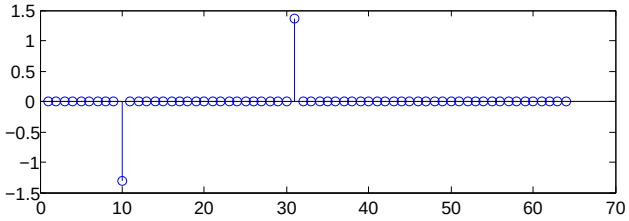
3. **Answer:** For both matching pursuit and orthogonal matching pursuit algorithms, the iterations stop when the estimate is ϵ-accurate. In Matlab code, we set $\epsilon = 1 \times 10^{-5}$.

   For matching pursuit, the estimates f_k and weights w_k for the first 10 iterations are shown as follows.

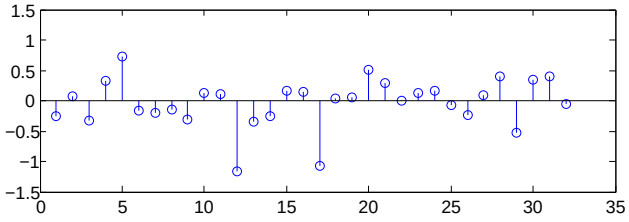




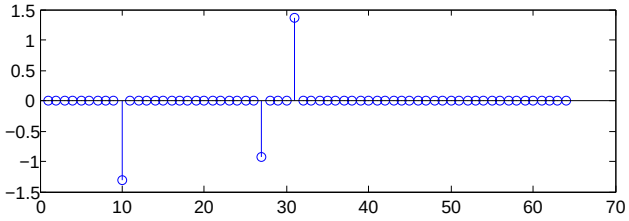
f_2



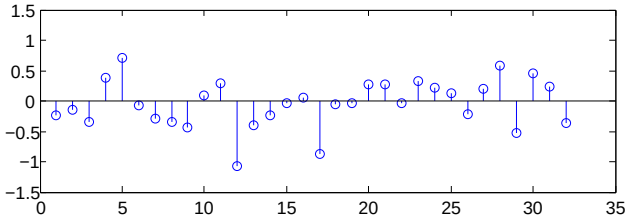
w_2



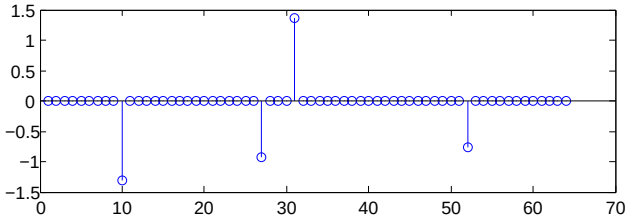
f_3



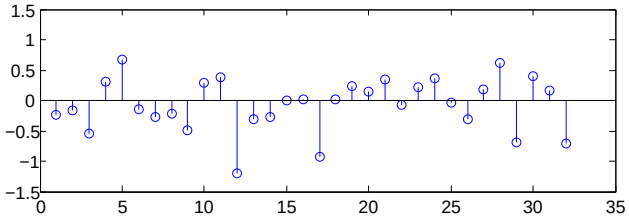
w_3



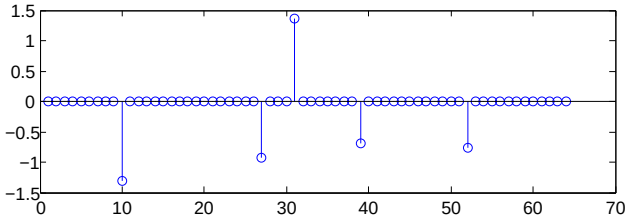
f_4



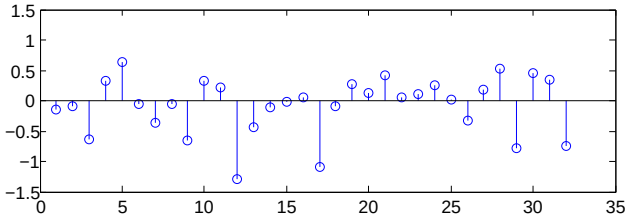
w_4



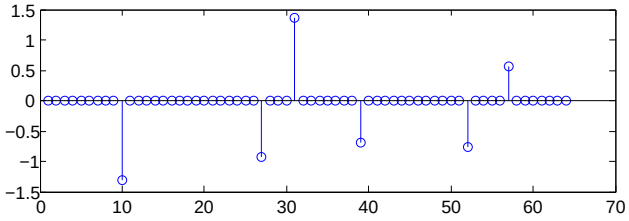
f_5



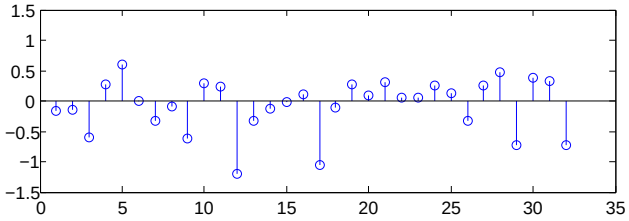
w_5



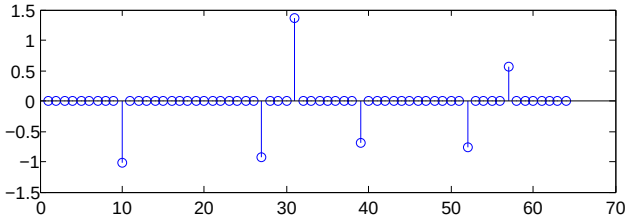
f_6



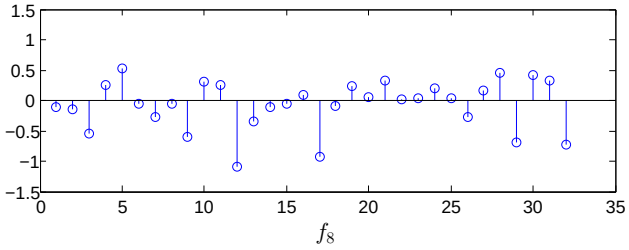
w_6



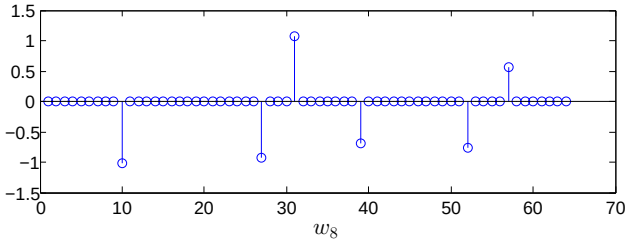
f_7



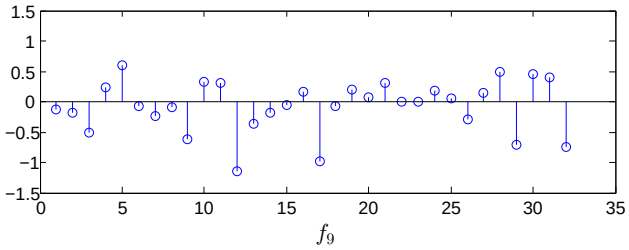
w_7



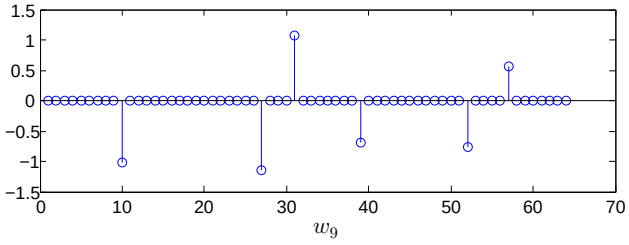
f_8



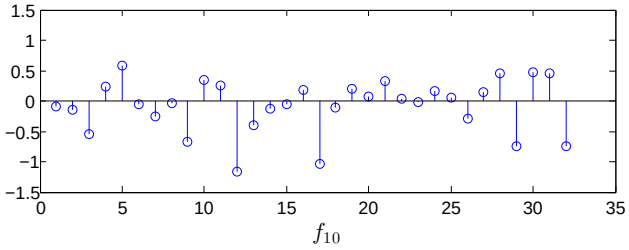
w_8



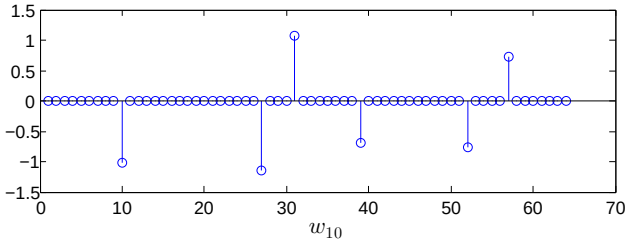
f_9



w_9

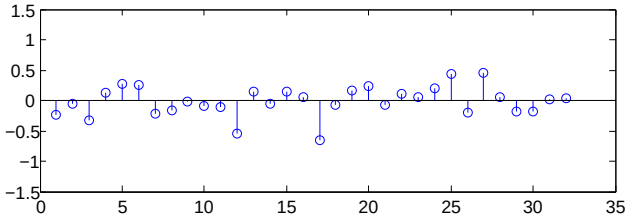


f_{10}

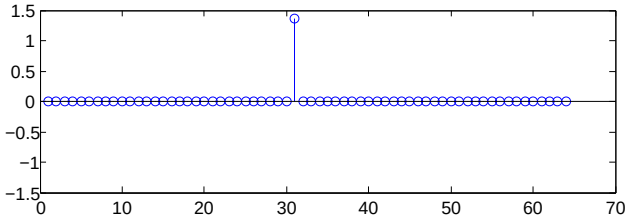


w_{10}

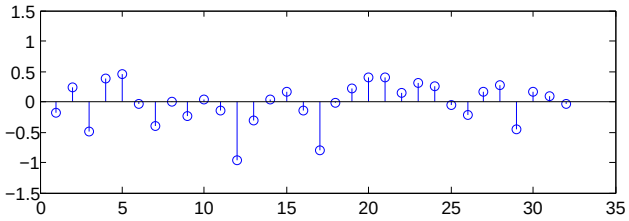
For orthogonal matching pursuit, the iterations stop after 6 iterations, and the estimates f_k and weights w_k are shown as follows.



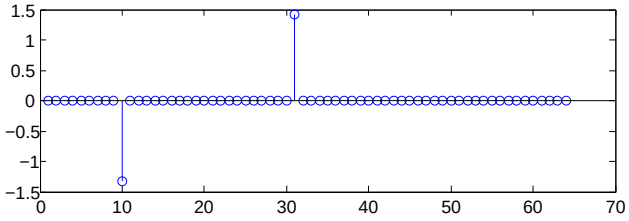
f_1



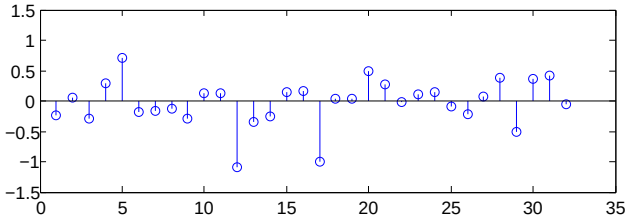
w_1



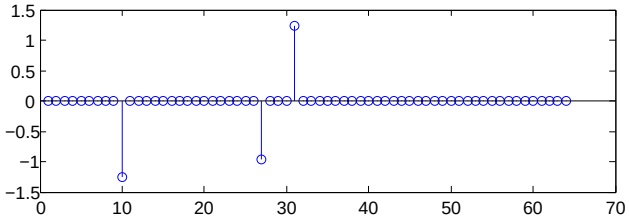
f_2



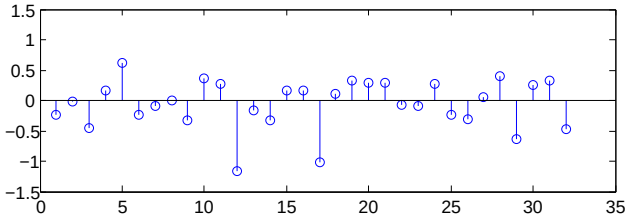
w_2



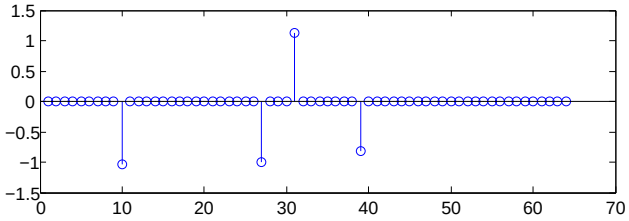
f_3



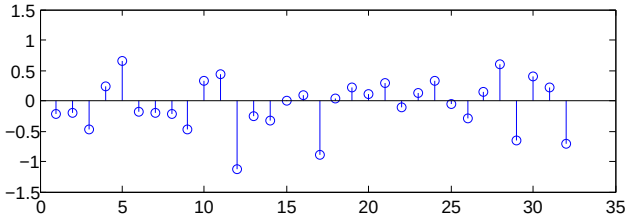
w_3



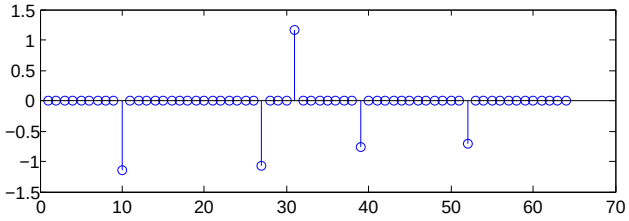
f_4



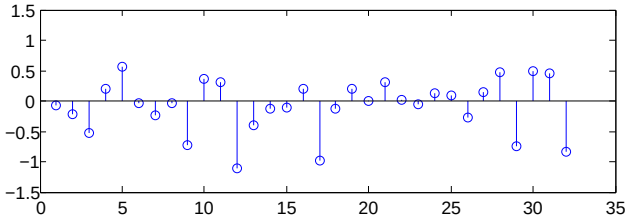
w_4



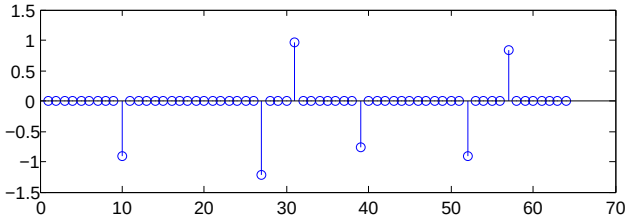
f_5



w_5



f_6



w_6