

# MATLAB Assignment 8

Spring 2019, Section B

Please submit this homework as a `.m` file, with suppressed output. Remember that all lectures and homeworks may be found at [github.com/guybaryosef/ECE210-materials](https://github.com/guybaryosef/ECE210-materials). Homework is due on ——— to [guybymatlab@gmail.com](mailto:guybymatlab@gmail.com).

**1. One plus one makes...** The fourier transform allows us to transform signals back and forth between the time and frequency domains. While this can be pretty clearly understood for time series signals (such as sound, for example), what this means for other forms of data might be more ambiguous. In this problem we will create a cool hybrid picture phenomena that is possible through the fourier transform and hopefully as we go through the different parts, you will get a sense of what the frequency domain means for images. If you are in need of a reference, much of the this question is similar to the image part shown in lesson 7.

- a. To begin, please download the two `.png` images located at [github.com/guybaryosef/ECE210-materials/tree/master/homeworks/hw8](https://github.com/guybaryosef/ECE210-materials/tree/master/homeworks/hw8) and import them into your MATLAB script file.
- b. Notice that these images are not the same size! As a preprocessing step, shrink the images so that they have the same pixel dimensions. Note that to do this, you do not want to just delete the first `X` pixels from the larger picture, because this will un-center the image. Instead if one picture is, for example, 50 pixels larger than the other along the `x` axis, remove its first and last 25 pixels along the `x` axis, that way equalizing its size while still keeping it centered. At this point, also convert your pixel values into a double type and normalize the values. Plot the two images beside one another using ***subplot*** and ***imshow***. You should currently be seeing the fab-tastic faces of Marilyn Monroe and Albert Einstein looking out at you.
- c. Next we will like to create a 2-d lowpass filter. One popular option is called a gaussian filter function, which unsurprisingly looks like a 3-d gaussian distribution. The equation for this filter is:

$$g(x, y) = \exp - \frac{(x - a)^2 + (y - b)^2}{2\sigma^2}$$

where  $\sigma$  is the standard deviation and  $(a, b)$  is the center of the function. Using a standard deviation of 14, create this filter so that it has the same dimensions as the two images (**Hint:** Use ***meshgrid***). Note that you want the center of the filter,  $(a, b)$ , to be the center of the image. Quickly plot your results using ***surf*** (titles and the like unnecessary), so that you can see how this resembles a lowpass filter in the frequency domain. If your graph is black, you should call your graph a variable name add the following line after it: ***set(your-variable-name, 'LineStyle', 'none')***.

- d. Just as the above can be a lowpass filter, we can quickly make a highpass filter by calculating  $g2(x, h) = 1 - g(x, y)$ . Do so here.

- e. Take the fast fourier transform of each image using `fftshift(fft2(img))`. Lets now apply the filters to the images. Because we want to combine these two images together, let's emphasize for each image its characteristics. What I mean here is that Marilyn has smooth, silky hair, and so a lowpass filter, which has the effect of *smoothing* an image, is appropriate. In the same vain, Albert's image has a lot more edges, creases, and lines to it (I will avoid saying anything meaner) and so you should apply the highpass filter to it. Remembering that convolution in the time domain is multiplication in the frequency domain, and so all that is necessary here is element-wise multiplication. Create another 2x2 subplot, with the original images in the frequency domain as well as the filtered images in the frequency domain.
- f. Using the inverse fft, return each image back to the time domain, and subplot them side by side. As a bonus question, where is Albert?
- g. Average the two images together, and we have our hybrid photo! Gah! A little scary at first sight! However look at it up close. Now back away from the screen and look at it again. Do you see different faces peering up at you? If not, feel free to tweak around with the standard deviation of the filters as well as the other parameters. This question was taken directly from <https://jeremykun.com/2014/09/29/hybrid-images/>, which is an awesome blog full of cool and informative ideas and concepts.

**2. Girl look at that Bode!** A continuous time system is characterized by the following equation:

$$f = t^5 e^{2t} \sin(5t) + t^3 e^3 t \cos(4t) + t \cos(10t)$$

- a. Take the Laplace Transform of the above equation. (Hint: Use the symbolic toolbox!)
- b. Use `numden` and `coeffs` to extract the coefficients from the Laplace transform. Use `double` to convert them to numeric form.
- c. Create a bode plot for the equation you obtained above. As is customary with log-scaled axis (to differentiate from the more standard linear spacing), turn the grid on.