

CE640 / OC512 – Week 3 Assignment
Functions and Program Control

1. Write a function, called `multGauss`, to compute the sum of multiple Gaussian (normal distribution) profiles. See the figure [here](#) to illustrate what Gaussians of different locations, amplitudes and ‘widths’ (standard deviations) look like. Your function should accept, as input, a matrix with three columns, and a vector `x`. The three columns should be the amplitude, the width, and the location of the individual Gaussian profiles. The rows of the matrix contain these parameters for each of the Gaussian profiles that you wish to compute. Your function should return, as output, the SUM of the individual Gaussian profiles. In other words, the individual profiles will be $y_1(x)$, $y_2(x)$, $y_3(x)$, and so on. What you want to return is $y = y_1 + y_2 + y_3 + \dots$. Please set up your function to perform ‘error checking’ to ensure that the input matrix has the needed three columns. **HINT: This is a great place for an ‘if’ statement.** If it does not, you should report an error to the command window and terminate the function call. Also set up your function so that it will work no matter how many Gaussians are requested. For example, the user might wish to compute the sum of 2 profiles, or 5, or 10. Your code needs to be flexible and be able to handle an arbitrary number of rows in the input matrix. **HINT: this is a great place for a ‘for’ statement.**

Next, your function, to accomplish its goal, your function should call the function `gaussmf` (built-in matlab function) to do the basic calculation. If you look at the help for this function, you will see that `gaussmf` accepts only width and location, not amplitude. Thus, `gaussmf` always returns Gaussians with an amplitude of 1, and your function will have to scale them with the desired amplitudes before computing the sum.

Great, now that you have written your function, and called it `multGauss.m`, write an m-file called `mygauss.m` that we will use to ‘test’ your function. Your script will specify the input variables, issue a call to `multGauss` and plot the result. For your test, let the domain range from 0 to 20. Let there be four Gaussians. The amplitudes are to be 0.1, 0.5, 1, and 2. The standard deviations (sigma) are to be 0.5, 1, 2 and 3. The positions are to be 2, 4, 6, and 8. Since your function returns the SUM of the gaussians, your m-file should plot this output, i.e. the sum.