(1) Translate the pseudo code given in box 2.2.2 in the Course Notes to `MatLab`.

The linear fits we've developed thus far can be used in many physical applications. These algorithms all depend on the resulting minimized normal equations being linear. In many cases this will not be true and we need some tactics to use in these cases. It should be noted that non-linear problems are notoriously difficult to handle and no single approach exists that can handle all nonlinear problems that arise.

Let us define the sum of square differences as

$$S(a_1, a_2, \ldots, a_m) = \sum_{k=1}^{N} [y_k - Y(x_k; a_1, a_2, \ldots, a_m)]^2 \tag{1}$$

where the y_k are the data located at x_k and $Y(x_k; a_1, a_2, \ldots, a_m)$ is the function to be fit to the y_k. Thus we are looking for the coefficients, a_n, that minimize S in the case in which the partials result in a set of nonlinear equations

Method 1.

(2) The interpolating polynomial for S near the minimum is

$$a_1^1 = a_1^0 - \tfrac{h_1}{2} \frac{S(a_1^0 + h_1) - S(a_1^0 - h_1)}{S(a_1^0 + h_1) - 2S(a_1^0) + S(a_1^0 - h_1)}$$

where where a_1^0 is the initial guess for the parameter a_1. What are the next steps in the algorithm.

- Repeat with all other params
- Then Repeat with set of new params

or

- Taylor expand around values $a_1^0, a_2^0, \ldots a_m^0$ and only keep the linear terms
- write as a set of linear equations
- Continue till enough S's are processed
- Find matrix of second partial derivatives (Hessian)

The code, `minimize.m`, starts the process of implementing this algorithm for you. It is located in the course Teams page along with a data set called `nonlinear_1.dat`. Modify/complete the code to find the best fit of the data set to the function,

$$f(x) = \frac{a}{1 + be^{-cx}}$$

where a, b, and c are the parameters to be found. This function is the solution to a type of population growth called *logistic growth*. The term a is called the carrying capacity, b is the initial population, and c the growth rate. Use the data to guide your initial guess for a, guess both $b, c < 1$.

The method just discussed can be very slow to converge and can sometimes go very much awry if the initial guesses are not "close" to the minimum. A more robust method can be developed by Taylor expanding the condition for S to be a minimum. We address this next