Ariana Bruno (amb633) & Haritha Murali (hm535)

Overall Analysis

This programming assignment focuses on optimal parameter extraction using the least squares fitting method. Two methods (Quasi-Newton and Secant) were tested. Due to the large nature of this project, it was necessary to maximize code reuse and manage the memory of the program efficiently

Memory Management

Due the immense size of datasets, it was necessary to ensure that there was as minimal "garbage collection" as possible. This was achieved in several ways:

i) First, by ensuring significant processing was done outside the main through modular functions. Temporary variables (required for calculation, but not the final results) declared in each function are automatically deallocated when the function is exited. This minimizes the memory occupied required by these temporary variables. For instance, if an instance of the $I_{d,model}$ were to be stored for each iteration of the convergence, the memory required by the program will accumulate quickly to be in range of gigabytes. However, by declaring each instance of $I_{d,model}$ in a function, we are able to clear the memory, and only store the scalar value of V (which is the sum of squares).

ii) Secondly, all helper functions used the pointers to vectors and matrices. Even when the size of the vector is small (e.g. if it is rank 3), if each helper were to create a new copy of vector, the amount of memory required for processing could increase exponentially. To minimize this, explicit copies of vectors and matrices were created only when necessary, otherwise pointers were passed between functions and helper functions.

iii) Thirdly, for larger vectors that were initialized in the main, we maximized reuse of variables by explicitly erasing the allocated memory.

Code Reuse

i) We defined a library of utility functions for common uses (e.g. for calculating the sum of squares, the norms, and the solvers). These utility functions could then be utilized throughout our code in various parent functions.

ii) By adding a bool to the sum of squared differences function, we were able to convert the calculation needed in task 4 and to the calculation needed in task 5. This allowed for more code reuse by using the same functions in both tasks and changing the bool to true for the normalized calculation in task 5 or false for the calculation in task 4.

iii) Additionally, the previous hacker practices were reused/modified for many of the functions. With verifying the smaller cases in the hacker practices, it ensured that the overall function behavior was correct and then could be further built upon to achieve the needed applications in this assignment.

Task 3 Results:

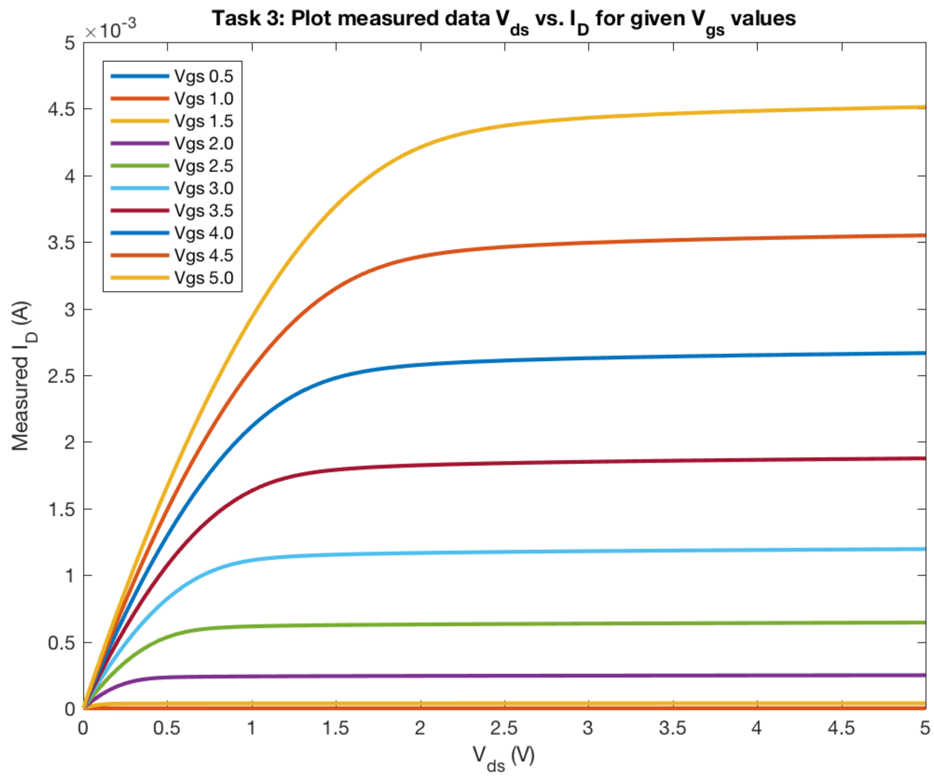

Figure 0: Ids Measured vs. Vds for given Vgs

Task 7 Results:

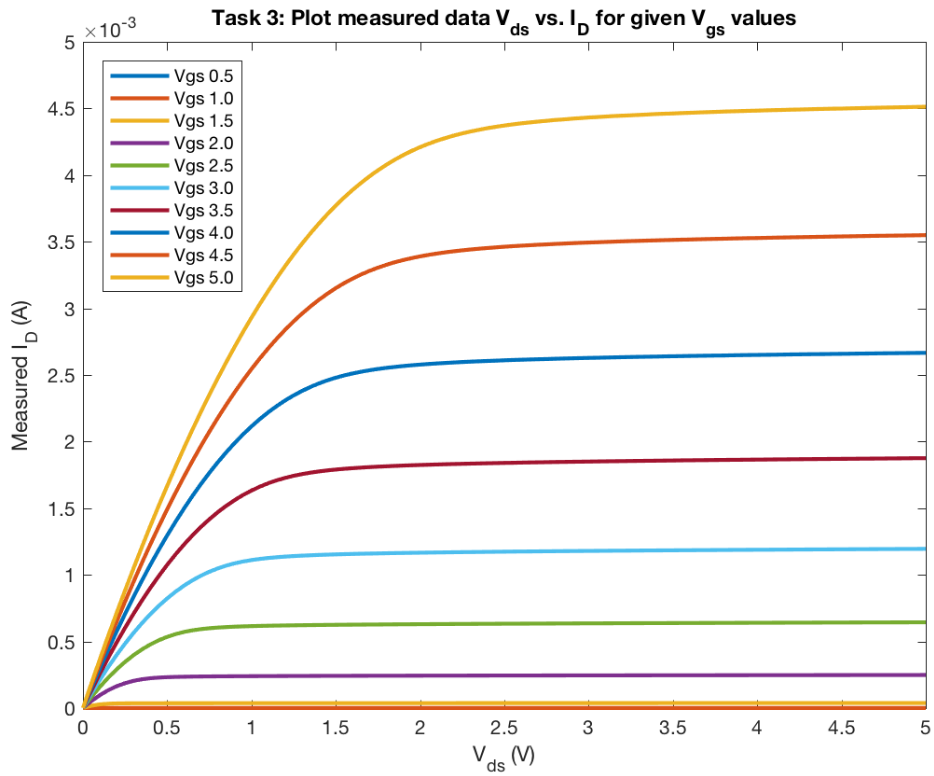

Figure 1: Ids Measured vs. Vds for given Vgs

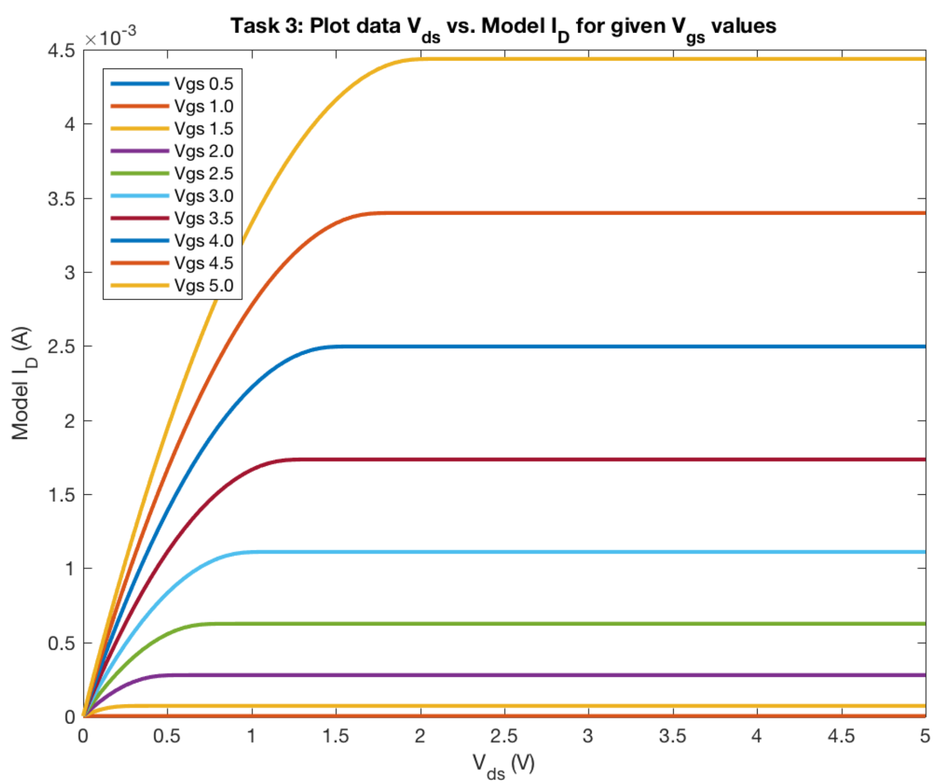

Figure 2: Ids Model vs. Vds for given Vgs

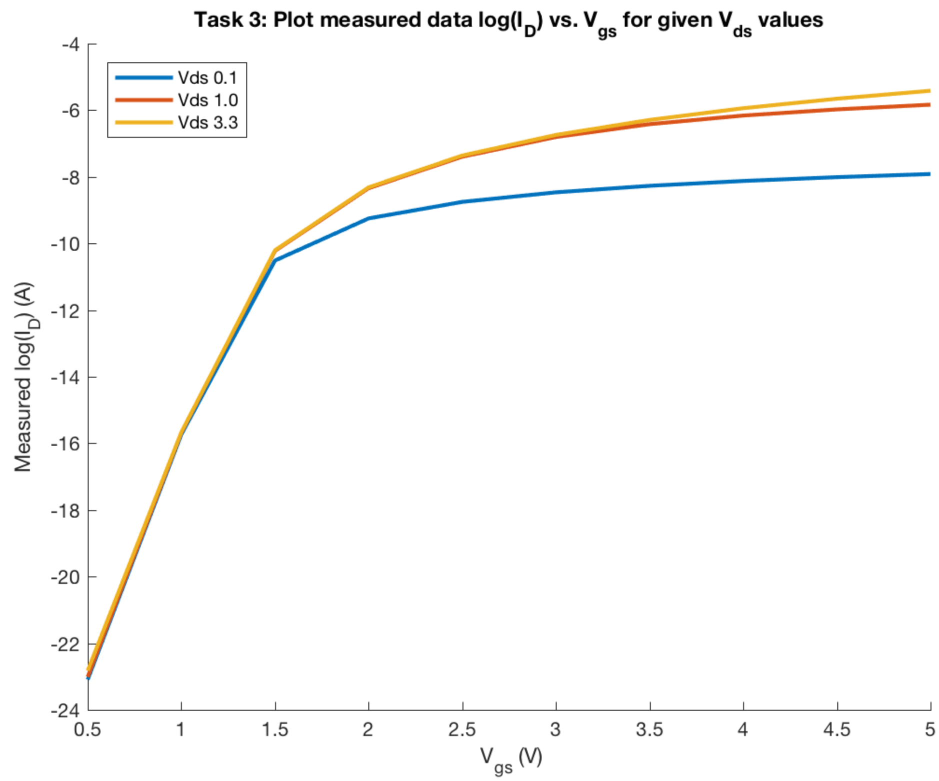

Figure 3: log(Ids) Measured vs. Vgs for given Vds

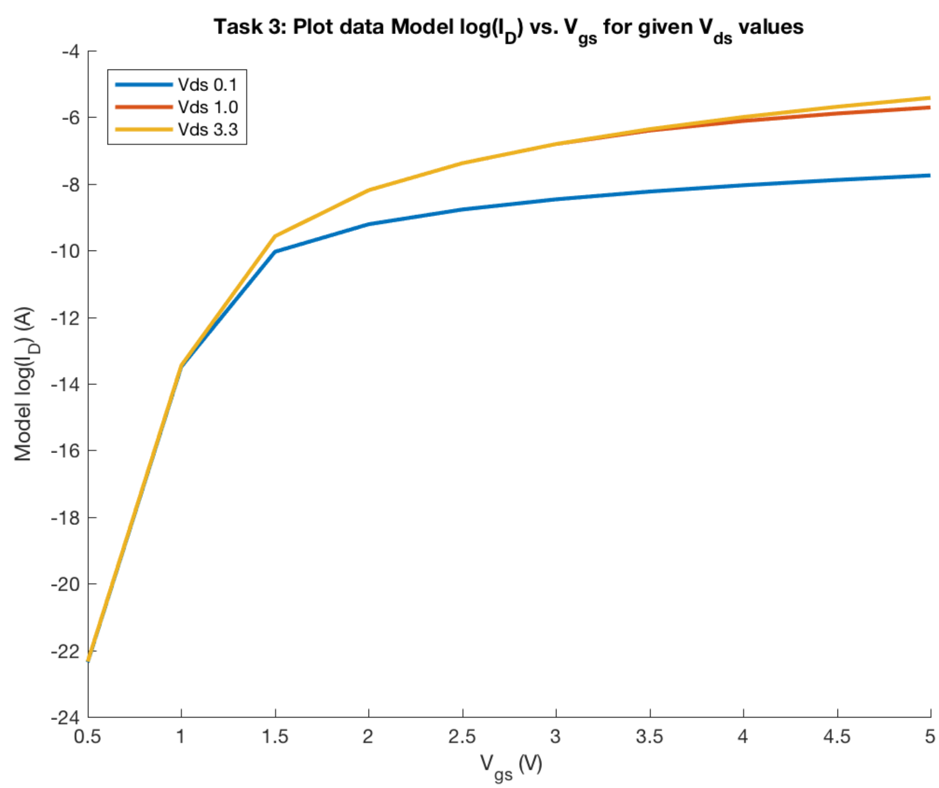

Figure 4: log(Ids) Model vs. Vgs for given Vds

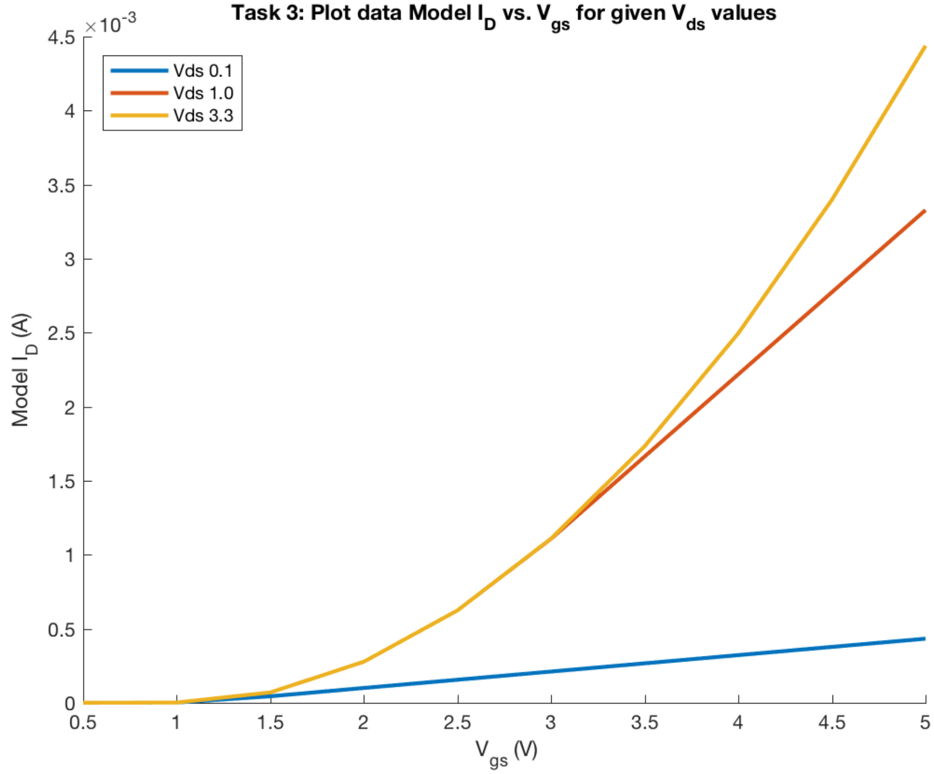

Figure 5: Ids Model vs. Vgs for given Vds

As requested in Task 7, figures 1 and 3 display the measured Ids values with respect to V_{DS} and V_{GS} values (respectively). The graphs appear to be very similar to figures 2 and 4, which take the Ids values generated from the Ids model function using the optimal parameters found from task 6 using the various methods of parameter extraction.

Figure 5, takes the same data used in figure 4 and plots the I_{DS} model data without taking the log of it. This allows us to see the quadratic behavior between I_{DS} and V_{GS} better.

Given that kappa = 0.5 and V_{th} = 1.0 (generalized values form the optimal parameters) and $V_{Dsat} = \kappa(V_G - V_{th}) - V_{SB}$ We can calculate V_{Dsat} for all V_{GS} values:

V_{GS}	V_{dsat}
0.5	-0.25
1.0	0
1.5	0.25
2.0	0.5
2.5	0.75
3.0	1.0

3.5	1.25
4.0	1.5
4.5	1.75
5.0	2.0

- For $V_{GS} < V_{th}$, I_{Dmodel} should be an exponential function of V_{GS} with $\kappa < 1$ and nearly insensitive to V_{DS}.

In figure 4, it shows the model log(I_{DS}) values with respect to V_{GS}. Since the assumed optimal kappa value is less than 1.0, then for all $V_{GS} < V_{th}$ (assumed to be 1.0), then I_{DS} is assumed to be exponential with respect to V_{GS}, which is confirmed from this graph. Since all V_{DS} values are undistinguishable when $V_{GS} < 1.0$, and log(I_{DS}) appears to be linear while $V_{GS} < 1.0$, then it is the same as saying I_{DS} is exponential for $V_{GS} < 1.0$.

- For $V_{GS} > V_{th}$ and $V_{DS} > V_{Dsat} = \kappa(V_G - V_{th}) - V_{SB}$, I_{Dmodel} should be quadratic to V_{GS} and insensitive to V_{DS}, or $\left|\frac{\left(\frac{\partial I_D}{\partial V_{GS}}\right)}{\left(\frac{\partial I_D}{\partial V_{DS}}\right)}\right| >> 1$. This is the shape of the family curve in $I_D(V_{DS})$ with V_{GS} as parameters.

Based on these values we can see that:

$V_{GS} > V_{th}$ and $V_{DS} > V_{Dsat}$, I_{Dmodel} should be quadratic to V_{GS} and insensitive to V_{DS}.

For figure 5, consider the instance when $V_{GS} > 1.0$, when V_{DS} is 3.3 V (the yellow line) it is always having a quadratic behavior because it is always greater than V_{Dsat} for all $V_{GS} > 1.0$.

Now still considering figure 5, consider the instance when $V_{GS} > 1.0$, when V_{DS} is 1.0 V (the red line) then the $V_{DS} > V_{Dsat}$ true until $V_{GS} > 3.0$. The blue line is quadratic until V_{GS} is 3.0 and then seems to become linear as V_{GS} continues to grow. Between $1.0 < V_{GS} < 3.0$, the I_{DS} value appears to be insensitive to V_{DS} meaning that they achieve the same I_D for the same values of V_{GS} at different V_{DS} values.

- For $V_{GS} > V_{th}$ and $V_{DS} < V_{Dsat}$, I_{Dmodel} should be quadratic to V_{DS}.

Consider figure 3, it can be seen that there is quadratic behavior between I_{DS} and V_{DS} that occurs when $V_{GS} > V_{th} = 1.0$ V and $V_{DS} < V_{Dsat}$. When $V_{GS} = 1.0$ (see the red-orange line), we can see that the behavior never becomes quadratic and appears to be linear. When $V_{GS} = 3.0$ (see the light blue line), $V_{Dsat} = 1.0$, thus, When $V_{DS} < 1.0$, it can be seen that I_{DS} has quadratic behavior to V_{DS} for all V_{DS} less than 1.0, when V_{DS} is greater than 1, the I_{DS} model values appear to become linear with respect to V_{DS}.