Visual Report of Task 3, 5, 7
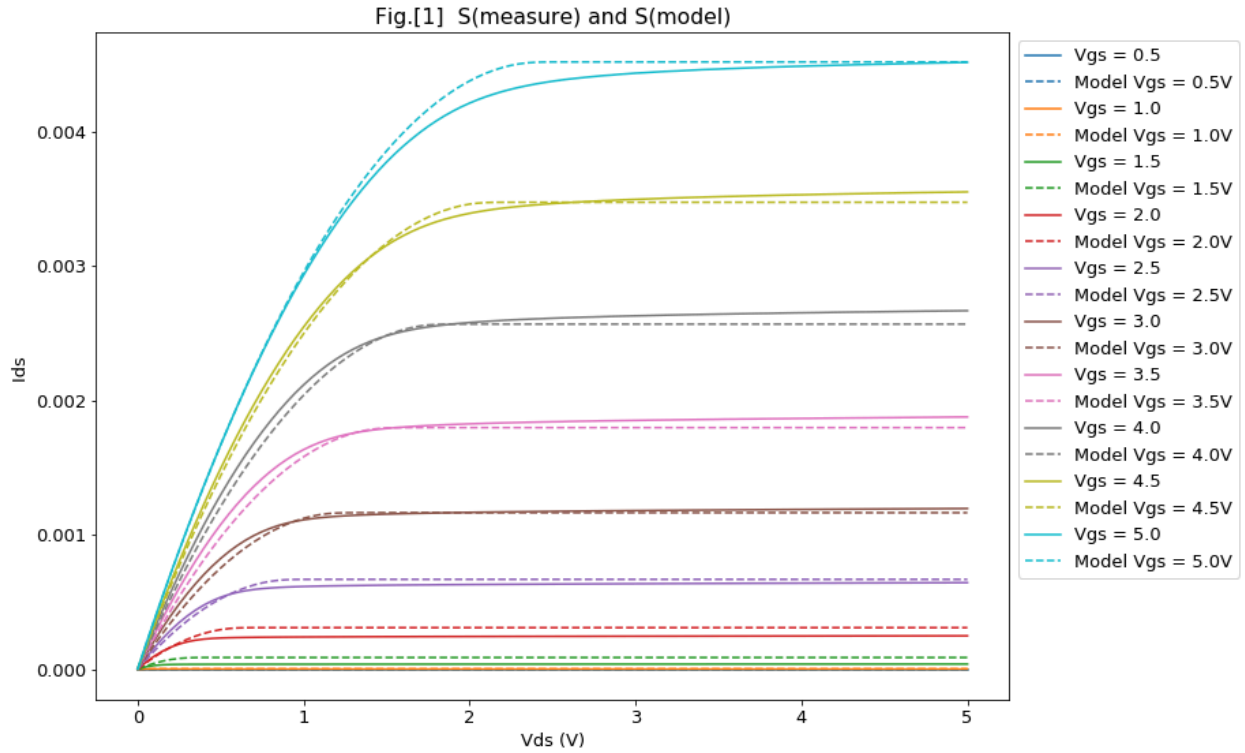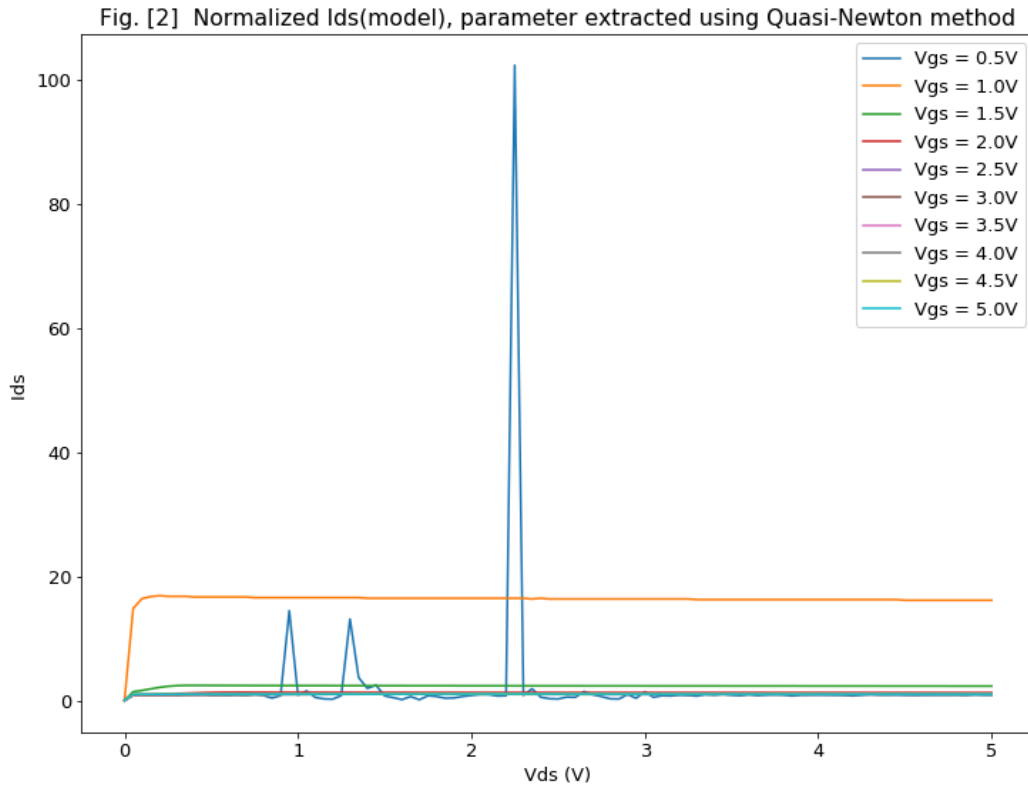
- **Task 3:** *Plot of S(measure), co-plotted with S(model) using Quasi-Newton method*

Fig.[1] S(measure) and S(model)

The dotted curves are generated from parameters extracted in Task 4.

- **Task 5.1:** *Plot of S(model) = I_{ds}(model) / I_{ds}(measure)*

Fig. [2] Normalized Ids(model), parameter extracted using Quasi-Newton method

It could be seen that for $V_{gs} = 0.5V$ and $V_{gs} = 1.0V$, I_{ds}(model) is a lot larger than I_{ds}(measure), too large that they even zoom out the other curves. Therefore, another plot that zoom in with $V_{gs} > 1.0V$ is generated as below:

Fig. [3] Normalized Ids(model), parameter extracted using Quasi-Newton method
Zoom in Vgs in range(1.5, 5.0)
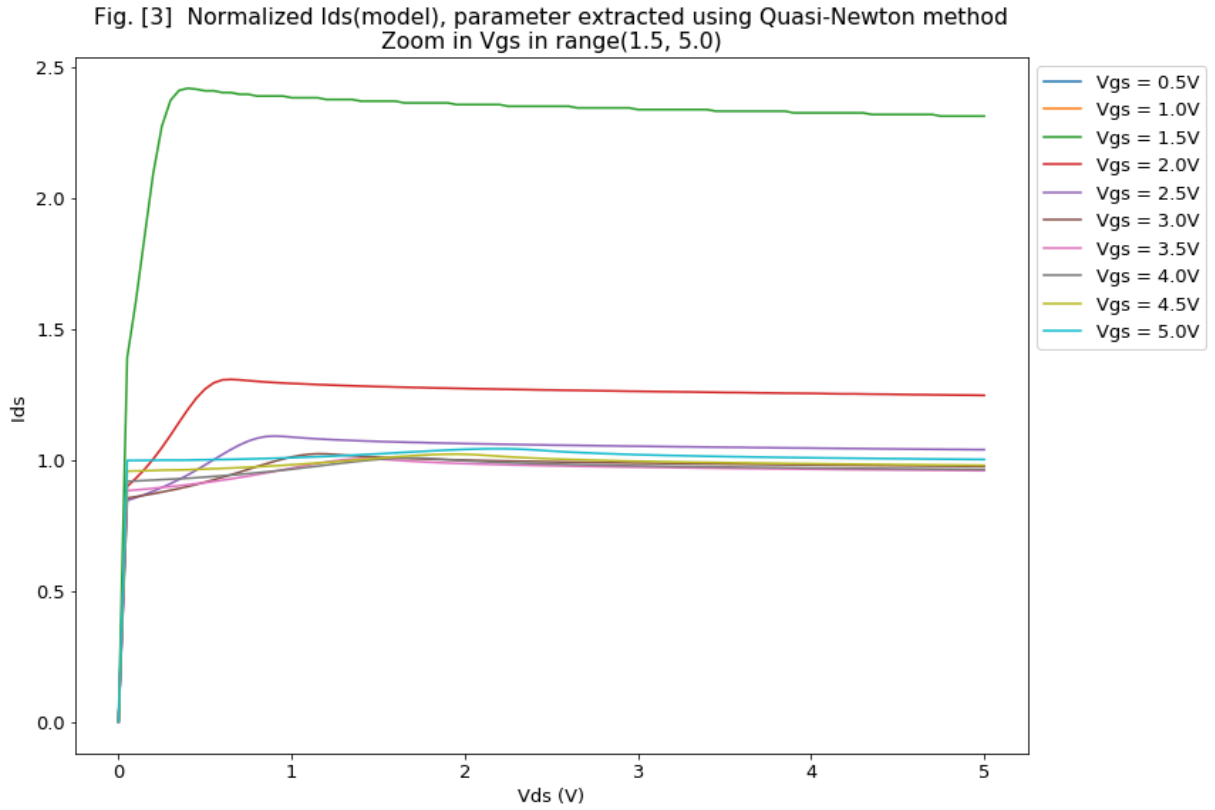
From Fig. [2] and [3], it could be concluded that the smaller I_{ds} data points have larger errors. Especially when $V_{gs} = 0.5V$, the I_{ds}(model) even oscillates to as large as 100 times of I_{ds}(measure), although with un-normalized data, loss function's value V is extremely small and within a magnitude of 10^{-6}.

The reason could be that with smaller I_{ds}(also smaller V_{gs}), the loss function is less dominated by the errors of the small I_{ds}, therefore $||V||$ is not affected much by the small I_{ds} data points.

As a result, the larger I_{ds} points would be fit better by Least-Square-Estimation, because their errors weigh more, and would thus be optimized more by the iterative solver; Vice versa, the smaller I_{ds} points are "ignored" by the model, and thus their errors are magnified greatly.

Without normalization, the larger I_{ds} data points have better performance than the smaller I_{ds} data points. Therefore, normalization would enhance smaller I_{ds} data points' performance by assigning equal weight of errors as the larger I_{ds} data points, and at the same time harm larger I_{ds} data's performance. Therefore, normalization or not should depend on specific requirements.

● **Task 7.1:** Visual Validation of Iterative Solver Using Quasi-Newton Method

a) *Plot of $S(measure)$ and $S(model)$ using Quasi-Newton method (As in Fig. [1])*
b) *Plot of $\log(I_{ds}(measure))$ vs. V_{gs} with two V_{ds}: (As in Fig. [4])*

Fig. [4]   Task 7 - Validation for EKV parameter extraction
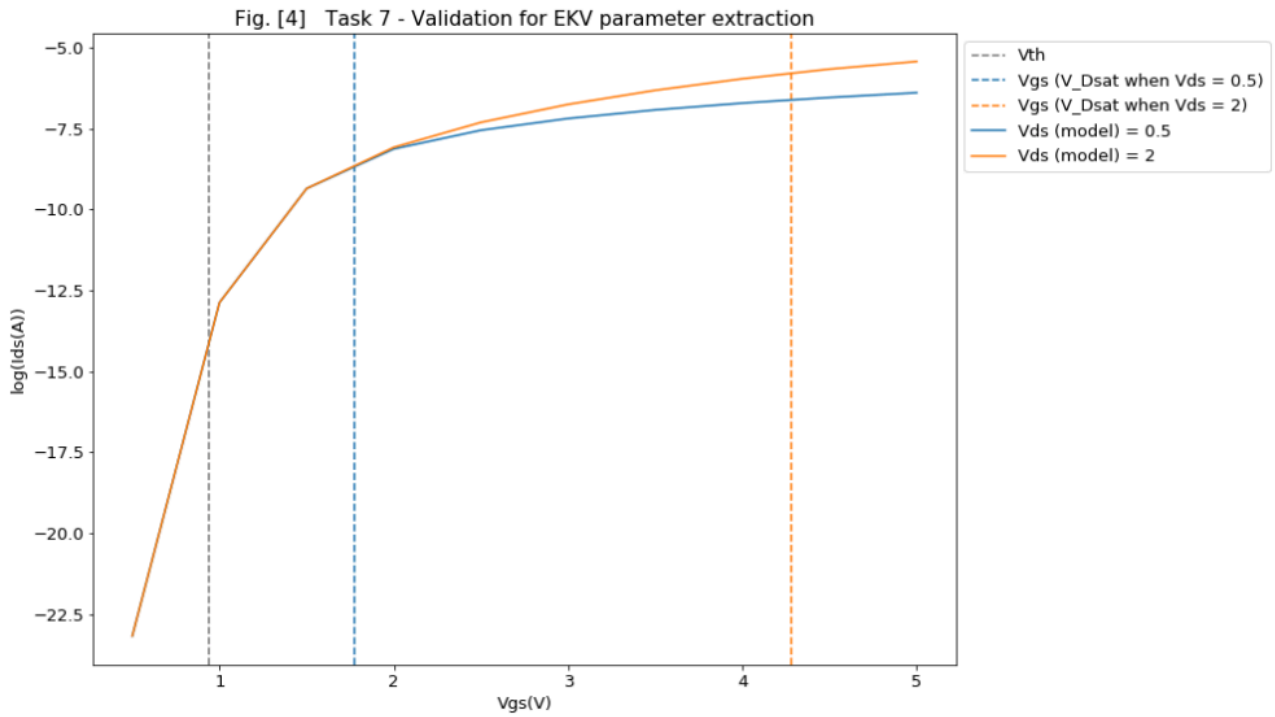
To observe how Ids is influenced with different V_{gs} and V_{ds} values, several vertical lines could be drawn on of Fig. [4] to better delimit the boundaries of the $V_{gs} = V_{th}$, $V_{ds} > V_{dsat}$, and $V_{ds} < V_{dsat}$.

To better relate to the approximation validation in Task 7, Fig. [4] is re-plotted as I_{ds} vs. V_{gs} to get rid of the logarithm form, as in Fig. [5]:

Fig. [5]   Task 7 - Validation for EKV parameter extraction
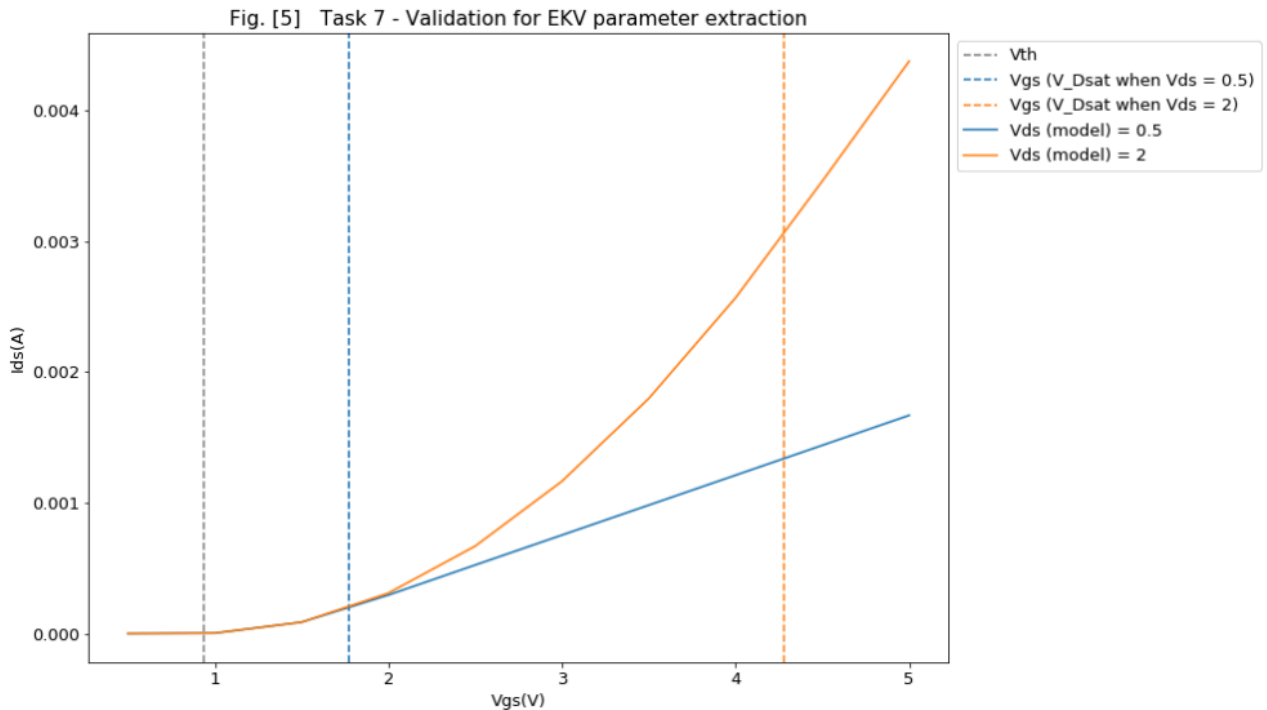
a)   **For $V_{gs} < V_{th}$** (the grey dotted vertical line at around $V_{gs} = 1$):
It could be concluded that the I_{ds} is almost completely determined by V_{gs}, and insensitive to V_{ds}, because the two curves are overlapping with each other before V_{th}.
Also According to Auto_Report's Task 7's numerical validation, I_{ds} is exponential to V_{gs}.

b)   **For $V_{gs} > V_{th}$, and $V_{ds} > V_{dSat}$:**

The largest V_{gs} that guarantees $V_{ds} > V_{dSat}$ would be calculated from $V_{gs} = \frac{V_{ds}}{\kappa} + V_{th}$.

The V_{dSat} delimiters for $V_{ds} = 0.5$, and $V_{ds} = 2.0$ are respectively plotted as the blue and orange vertical lines in Fig. [5]):

It could be seen that when $V_{ds} > V_{dSat}$ (before the blue dotted line), the I_{ds} is almost completely determined by V_{gs}, and insensitive to V_{ds}, because two curves overlaps and changes with V_{gs} before the blue dotted line.

According to the Auto_Report's Task 7's numerical validation, I_{ds} is quadratic to V_{gs}.

c) **For $V_{gs} > V_{th}$, and $V_{ds} < V_{dSat}$:**

Right after the blue dotted line, these two curves begin to diverge, indicating that V_{ds} starts to impact on I_{ds}.

According to the Auto_Report's Task 7's numerical validation, I_{ds} is quadratic to V_{ds}.

It could be concluded from both of the visual and numerical (in Auto_Report.txt) validations, that the implementation of iterative solver using Quasi-Newton method is correct.