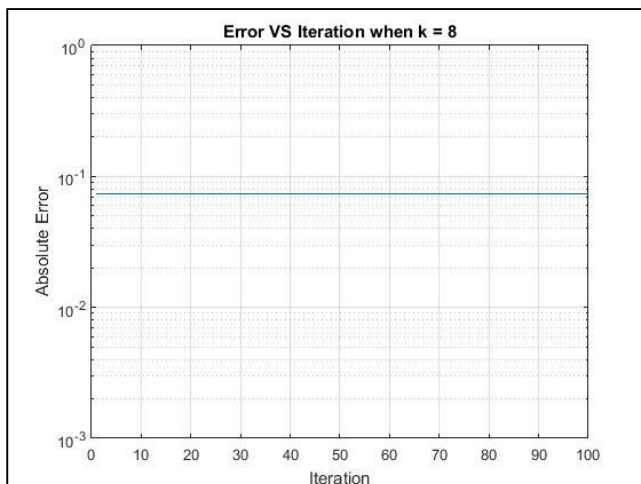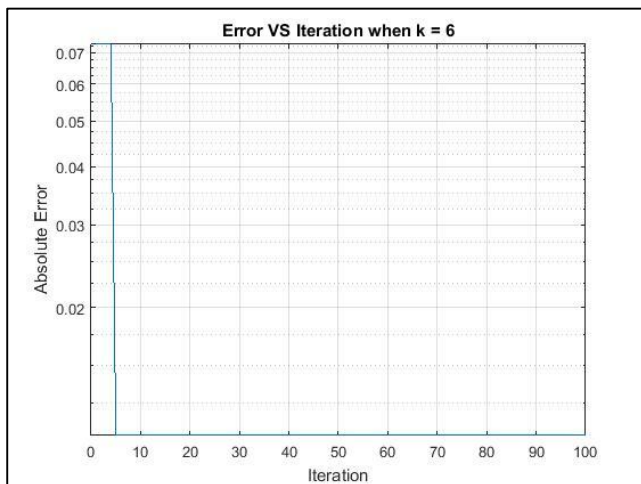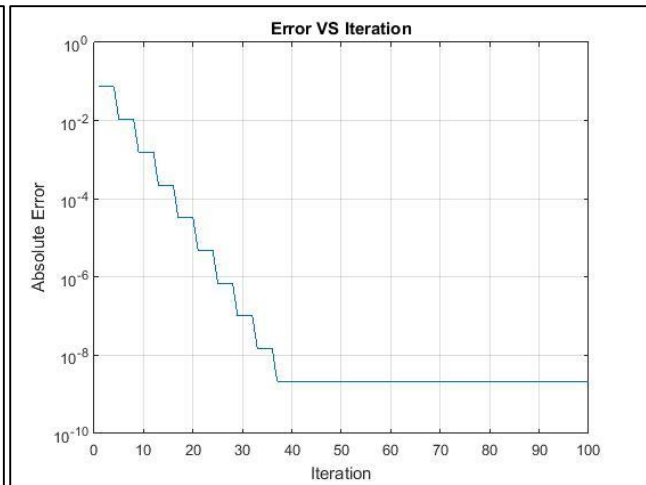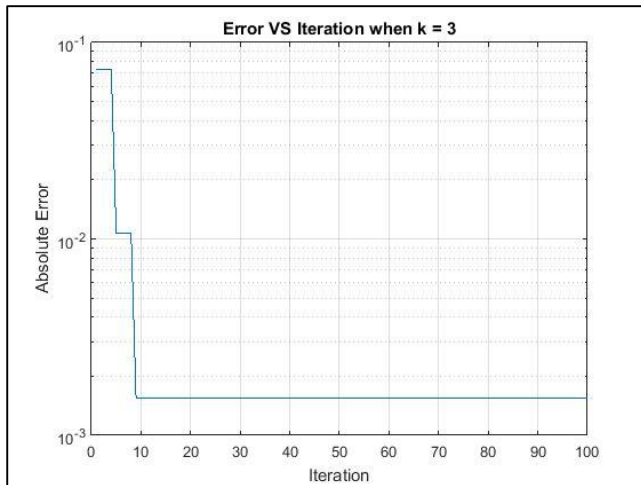Name: Jerry Chienyu Chen
Student Number: 301228216

MACM 316 – Computing Assignment 4 Report

Observing the figure on the top right, f(x) = exp(x^2) with absolute error versus the number of iteration, the error gradually decreases and eventually stabilizes at about 38 iteration with approximately 10^(-8.8) absolute error due to round off error. In terms of accuracy, it increases as the number of iteration increases reaching its max accuracy around 38th iteration. The algorithm is considered as accurate after the 38th iteration. In terms of efficiency, I would not consider the algorithm is efficient enough because the error only decreases every 5 iterations before the 38th iteration. In terms of robustness, I would not consider the algorithm robust either because its error stabilized at around 10^(-8.8) which is still 7 digits off from machine epsilon.

For the second function f(x) = exp(x^(2*k)), observing the plots on the left hand side, the error gets bigger when the value of k gets bigger and the error flattens out when k is greater than or equal to 8 at about 10^(-1.2). In terms of accuracy, the accuracy decreases when the value of k increases proven from the figures. Thus the algorithm is not accurate. In terms of efficiency, the algorithm is efficient because it executes fast and it takes value of k that is greater than or equal to 8 for the value of error to stabilize at 10^(-1.2).

In terms of robustness, the algorithm is even worse than the first function, because the greater the value of k is, the greater the error is shown in the figures, which is moving further away from the machine epsilon. Thus the algorithm is not considered robust. Considering Taylor series expansion, the more terms that are sum together, the more derivatives of f we have to take. Doing this will result small numbers close to 0, and eventually reaches 0. Thus the coefficient value gradually grows closer to the machine epsilon which results in a major round off error.