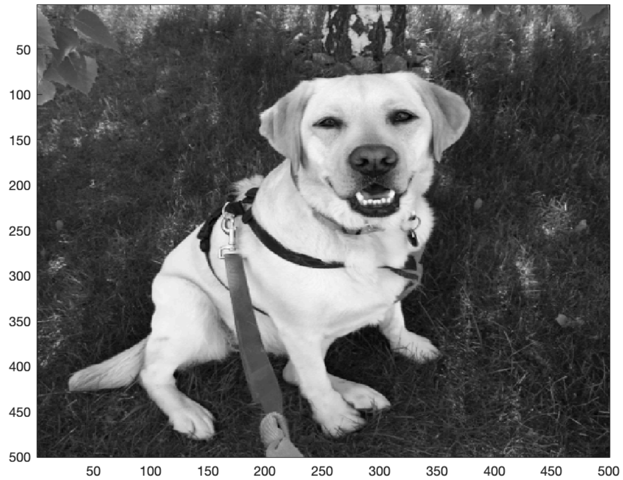
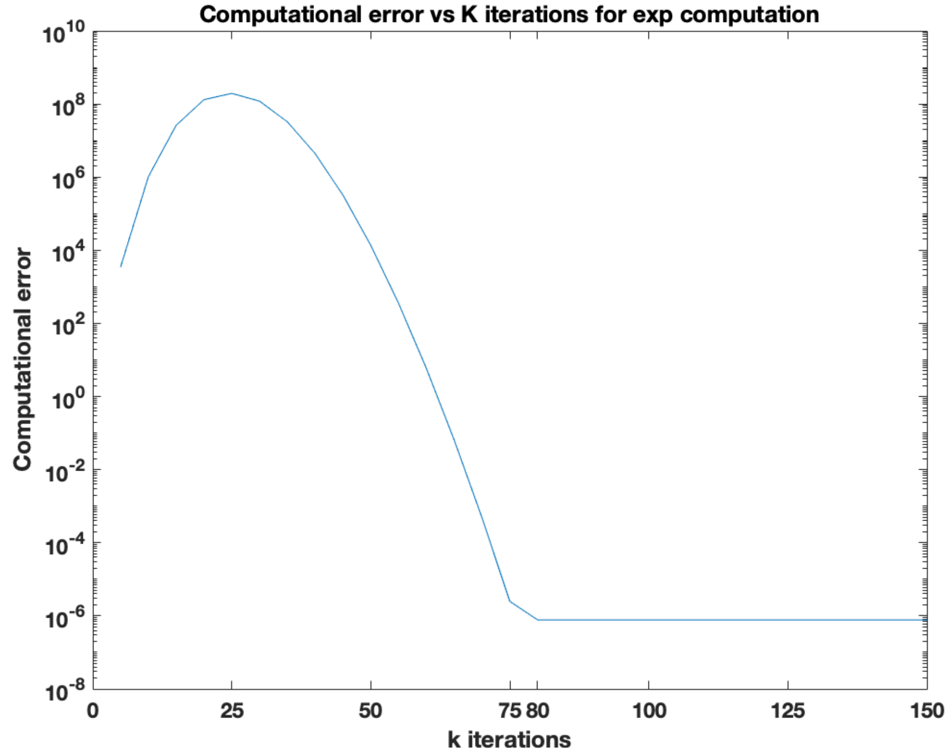
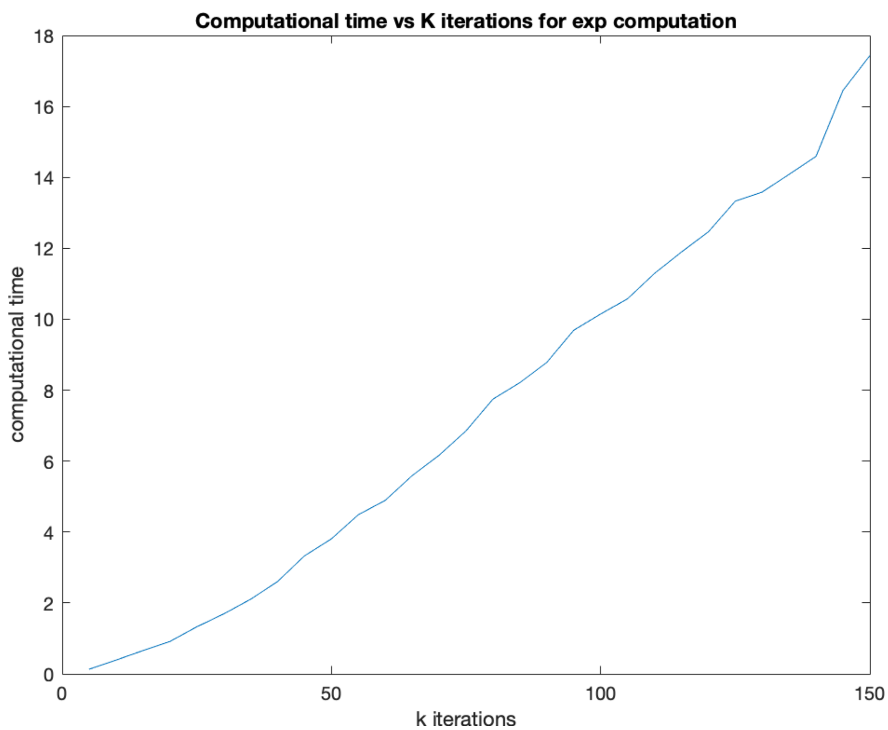
MACM316 Assignment 3 – Ngawang Kyirong 301312227. *discussions with fellow classmates/TA's occurred

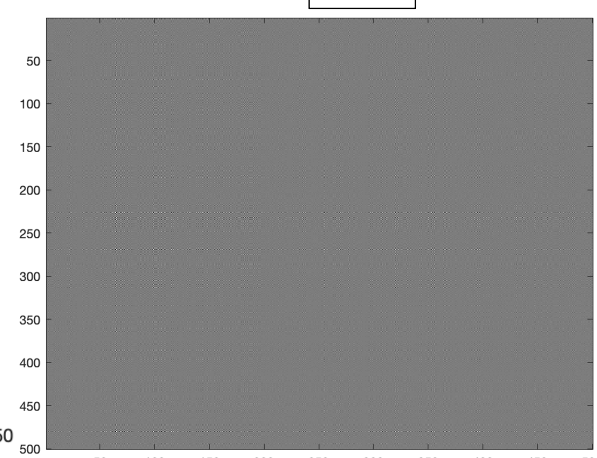
K = 150







K=50



B.) As seen in the "Computational time vs K iterations for exp computation" plot, there is a **linear** relationship between the increasing values of **k** iterations and the increasement in **time**. Contrastingly, there is not a linear relationship between the **flop** count and k. However, the amount of **time** it takes to complete is linear to k because for every increase in k by 5, the number of loops inside is also increased by 5 and the tic is before the inside loop with the toc immediately following afterwards (this is $O(k)$ because of the single for loop being used for timing). While the number of flops increases quadratically to k because of the amount of matrix multiplications (O(n^3)) is also dependant on k: $O(number\ of\ flops\ *\ k)$. Therefore, they are not in agreement with each other according to my algorithm.

C.) As seen in the "Computational error vs K iterations for exp computation" plot, the error increases until k=25 and then starts to decrease until k=80 and then stays constant. The **accuracy** of the algorithm is good as the error decreases to 10^{-6} and that the error is decreasing successfully with respect to parameter increase. However, it does not get down to **machine epsilon** because of floating point arithmetic. The **robustness** of the algorithm is not good as the calculations do seem to be affected by floating point arithmetic. As we can see at k=80, *expAk* is now being added to a matrix filled with very small numbers. Therefore, *expAk* ends up unchanged in the addition of matrices (ex. $c\ +\ 1/100000\ =\ c$) due to **roundoff error** (not enough **mantissa**) and **machine epsilon**. Therefore, the error calculations remain constant for the remaining k because *expAk* is never changing anymore.

```matlab
A = load('CA3matrix.mat');
I = eye(500);
k_50 = 50;
k_150 = 150;
expAk = I;
expAk2 = I;
expAk3 = I;

% part a (k=50)
for n = 1:k_50
    expAk = expAk + ((1/factorial(n)) * A.A^n);
end
imagesc(real(expAk));
colormap gray

% part a (k=150)
for n = 1:k_150
    expAk = expAk + ((1/factorial(n)) * A.A^n);
end
imagesc(real(expAk));
colormap gray

% part b
times = [];
for n = 5:5:150
    tic;
    for i = 1:n
        expAk2 = expAk2 + ((1/factorial(i)) * A.A^i);
    end
    times = [times toc];
    expAk2 = I;
end

plt = plot(5:5:150, times);
xlabel("k iterations");
ylabel("computational time");
title("Computational time vs K iterations for exp computation");

% part c
error = [];
expA = expm(A.A);
for n = 5:5:150
    for i = 1:n
        expAk3 = expAk3 + ((1/factorial(i)) * A.A^i);
    end
    error = [error norm(expA - expAk3)/norm(expA)];
    expAk3 = I;
end

plt = semilogy(5:5:150, error);
xlabel("k iterations");
ylabel("Computational error");
title("Computational error vs K iterations for exp computation");
```