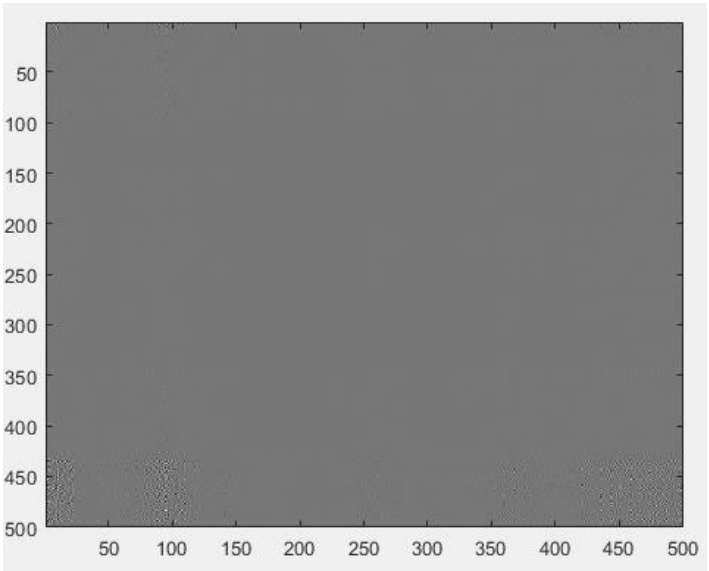

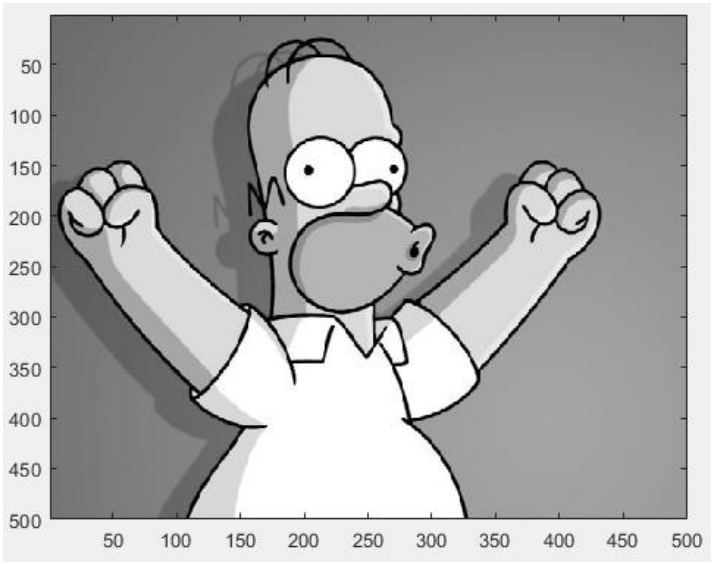

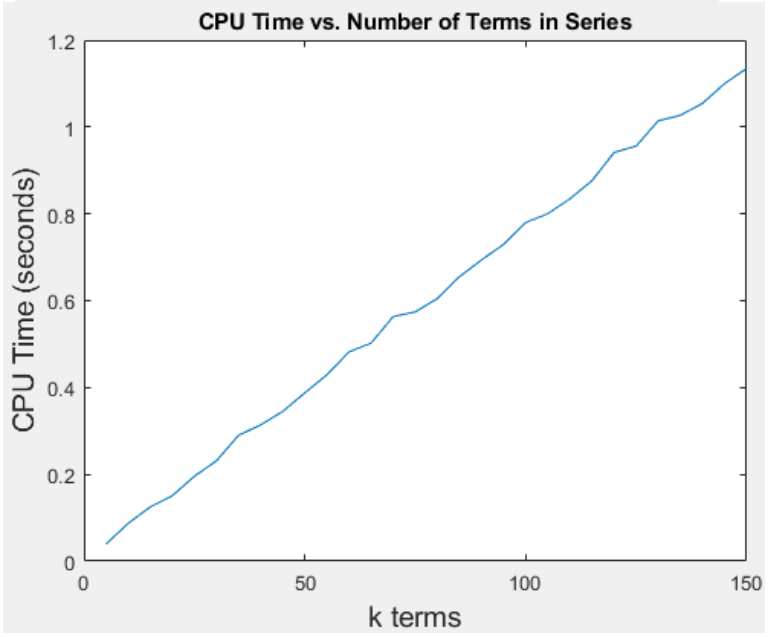

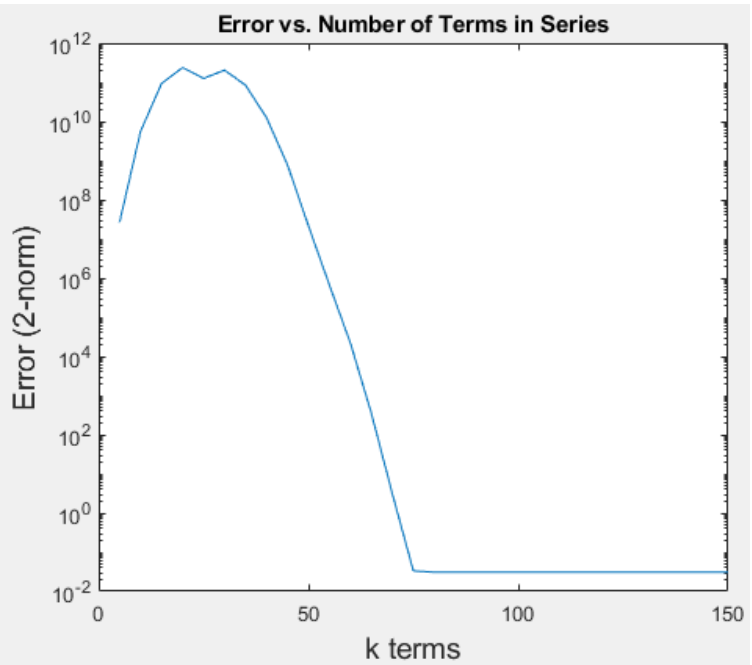

a) The first plot (top left) used k = 50 and has terrible image quality with relatively small entries causing most of the image to be dark gray. The image with k = 150 (top right) is much higher in quality and creates a very clear image.

b) Flops estimation for my algorithm:

- $\frac{A}{i}\ has\ n^2\ operations$
- Matrix multiplication of 2 N x N matrices has n multiplications and n -1 additions for each element. Therefore, 2n -1 operations per element. Operations will then be $n^2(2n-1) = 2n^3 - n^2$
- Adding 2 matrices together has n^2 operations.
- **Total flops for my algorithm for k iterations is:** $k(2n^3 - n^2 + n^2 + n^2) = k(2n^3 + n^2)$
- Big O is $O(kn^3)$. This is not consistent with my plot, where CPU time linearly depends on k. This is possibly because CPU time can be influenced by my PC's performance or MATLAB making operations more efficient in the background.

c) CPU time for expm is much faster and consistently at around 0.14 seconds. Error increases with k, maxing at 20 ≤ k ≤ 30, then dropping and plateauing at k = 75. The accuracy of my algorithm is at its best when the number of terms k ≥ 75.

d) **Accuracy**: At its lowest, error is around 0.03 and below which, considering the size of the matrices, is acceptable. This only applies to when k is greater than 75, however. Although expm is still more accurate, my algorithm has an acceptable accuracy when there is a high number of terms.

Efficiency: The CPU time of my algorithm is maxed at 1.17 seconds at k = 150 and increases linearly. Expm is around 0.14 seconds meaning it is much more efficient.

Robustness: My algorithm is not guaranteed to return a good result and has large errors when k ≤ 75. Therefore, it is not robust and heavily depends on k for good outputs.

```
% Computing Assignment #4: CA4.m
% Author: Andy Liu
% ID: 301472847

load('CA4matrix2.mat');

expA50 = exp_approx(A, 50);
expA150 = exp_approx(A, 150);
imagesc(real(expA50));
colormap gray;
imagesc(real(expA150));
colormap gray;

k_values = 5:5:150;
cpu_time = zeros(length(k_values), 1);
for i = 1:length(k_values)
    tic;
    exp_approx(A, k_values(i));
    cpu_time(i) = toc;
end

plot(k_values, cpu_time);
xlabel('k terms', 'FontSize', 14);
ylabel('CPU Time (seconds)', 'FontSize', 14);
title('CPU Time vs. Number of Terms in Series');

tic;
expm(A);
expm_time = toc;

err = zeros(length(k_values), 1);
expAexact = expm(A);
for i = 1:length(k_values)
    expAk = exp_approx(A, k_values(i));
    err(i) = norm(expAexact-expAk, 2);
end
semilogy(k_values, err);
xlabel('k terms', 'FontSize', 14);
ylabel('Error (2-norm)', 'FontSize', 14);
title('Error vs. Number of Terms in Series');
```

```
% Computing Assignment #4: exp_approx.m
% Author: Andy Liu
% ID: 301472847

function expA = exp_approx(A, k)
    n = size(A, 1);
    expA = eye(n);
    Ak = eye(n);
    for i = 1:k
        Ak = Ak * A / i;
        expA = expA + Ak;
    end
end
```