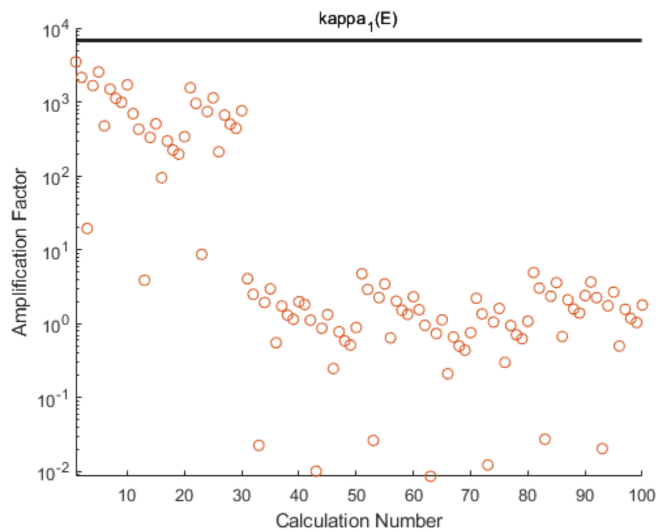
Part 1: Report

Question 1:A

Note: all required values are displayed in table form for each section prior to their discussion.

Values Calculated for Matrix A = E (with Ax = b + Δb)

Average	258.2577
Median	1.9622
Maximum	3.4880e+03
k_1(E)	6.8296e+03

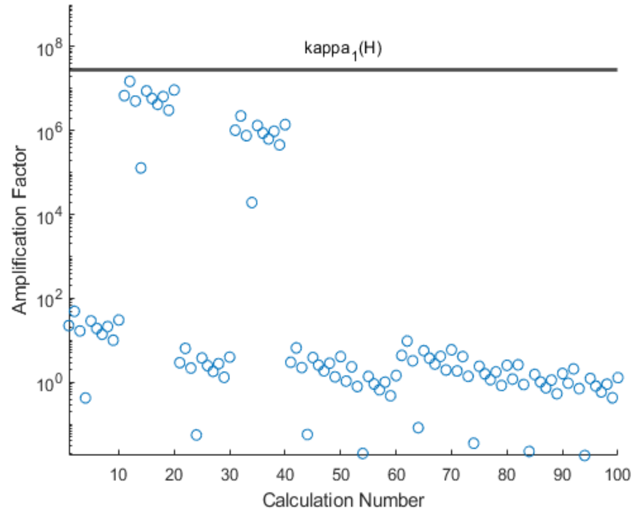


As expected, k_1(E) is an upper bound for the relative errors in the system of equations, with the maximum close to half as large as the upper bound. The average value is much higher than the median, which can be contributed to the high maximum (the first value calculated) and many of the first 30 values, which are > 100. The median, a more resistant estimate of the midpoint of the graph, lies closer to 1. This appears to show that the average error introduced from Δb in the right-hand side of the equation of Ax = b (where Δb = b+ epsilon*d) is very large and give poor estimations to x; however, by looking at the median (and the graph), we see that in most cases the error estimation is < 1, with many very close to 0 (7 estimations are < .1), thus the perturbed b can give a fairly good estimation to x.

Values Calculated for Matrix A = H (with Ax = b + Δb)

Average	7.4294e+05
Median	2.5611
Maximum	1.4951e+07
k_1(H)	2.9070e+07

For matrix H the 1-condition number is much higher than matrix E, and like matrix E is still close to double the maximum relative error calculated. The graph shows more of the values are clustered near the median, though the extreme values still cause the average to be extremely high. The median is higher for this graph than for Matrix E, but most values are near or below the median, with many approaching 0. Thus, this matrix is more resistant to the perturbations from Δb, and the solutions in Δx are good approximations to x.

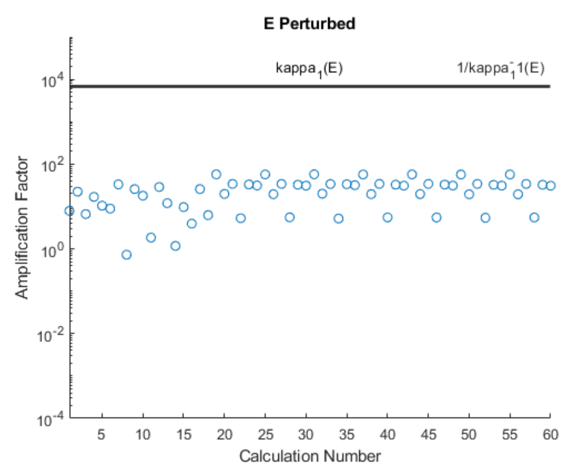


Question 1:B

Values Calculated for Matrix A = E (with (A + ΔA)x = b)

Average	25.0106
Median	27.2603
Maximum	57.4606
k_1(E)	6.8296e+03
1/rcond(E)	6.8296e+03

The estimations to x are much more consistent in this series, with the median and average much closer to each other than in the previous calculation for Matrix E, and the 1-condition number much higher than the maximum here. The calculated value 1/rcond(E) = (k_1(E)^-1)^-1 = k_1(E) was consistent with expectations.

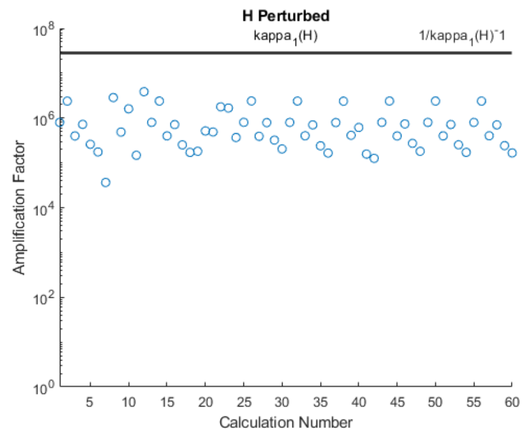


There is only 1 value that is less than 1 in this case, so very few estimations are good approximations to the x vectors. This shows that perturbations that originate on the left-side of the equation are more consistent than those on the right-side, as seen in part A, but cause much worse approximations, with few estimations coming close to the actual values of x.

Values Calculated for Matrix A = H (with (A + ΔA)x = b)

Average	8.6303e+05
Median	5.0130e+05
Maximum	3.8664e+06
k_1(H)	2.9070e+07
1/rcond(H)	2.9070e+07

Though in the graph it appears that the estimations are consistent for the perturbed Matrix H, this is due to the logarithmic scale. The 1-condition, maximum and midpoints are, in fact, orders of magnitude apart. This is shown by the average and median, of which the difference is over 360,000. The calculated value 1/rcond(E) = (k_1(E)^-1)^-1 = k_1(E) is extremely high (approaching infinity) and thus the error in approximation to x is highly amplified, which is consistent with the findings seen in the graph, where there are no error estimations that are remotely close to 0.



Question 2

Results of required computations are summarized in the table below:

	A = E (Part A)	A = H (Part B)	A = H8 (Part C)
norm((A*AINV)-I)	9.8276e-14	3.1838e-10	1.3746e-07
norm(AC - A)	1.4949e-12	7.3695e-12	3.0106e-09
norm(AINV - HI)	-	3.7458e-4	673.7377 (HI = HI8)
k1(A)	-	-	3.3873e+10

It is not surprising that though mathematically these norms should all be 0 they are slightly off; this can be contributed to the error that arises from storing values in a machine, as we have learned that this is often a source of error. What is quite surprising is that these values vary so much depending on how we try to evaluate the norm and the matrices involved. The distance between the calculated inverses and the exact inverses are particularly high, for HI8 in part c it is extremely high at 673.7 (which can be expected based on the extremely high 1-condition number).

Part 2: Code

Go to Matlab Code

*Note: scatterplots that are output when code is run altogether have overlapping data (as seen when link is followed), whereas when each section is evaluated separately the scatterplots are displayed as above.

```
%%
%---------------------------------Question 1a-----------------------
%First Matrix

%extract data from file
data = load('Data.mat');
A = data.E;
B = data.B;
D = data.D;
epsilon = data.epsilon;

%calculate amplification upper bound
k1 = cond(A,1);
x = [];
y = [];

%Amplification Factors and index i
Amp = [];
i = 0;
%Calculate
for j = 1:10
    b = B(:,j);
    x = A\b;
    for k = 1:10
        i=i+1;
        d = D(:,k);
        y = A\(b + epsilon*d);
        Amp(i) = norm(y-x,1) / (epsilon*norm(x,1));
    end
end

%Display Values
disp('Average of Amplication Factors for E:');
disp(mean(Amp));
disp('Median of Amplication Factors for E:');
disp(median(Amp));
disp('Max of Amplication Factors for E:');
disp(max(Amp));
disp('kappa_{1}(E)');
disp(k1);

%plot
y1 = yline(k1, '-','kappa_{1}(E)', 'LineWidth', 2);
y1.LabelHorizontalAlignment = 'center';
y1.Color = 'black';
hold on
num = [1:1:100];
scatter(num,Amp)
set(gca, 'YScale', 'log')
xlabel('Calculation Number')
ylabel('Amplification Factor')
axis([1 100 0 10^4])
hold off
```

```
%%
%Second Matrix

%extract data from file
data = load('Data.mat');
A = data.H;
B = data.B;
D = data.D;
epsilon = data.epsilon;

%calculate amplification upper bound
k1 = cond(A,1);
x = [];
y = [];

%Amplification Factors and index i
Amp = [];
i = 0;
%Calculate
for j = 1:10
    b = B(:,j);
    x = A\b;
    for k = 1:10
        i=i+1;
        d = D(:,k);
        y = A\(b + epsilon*d);
        Amp(i) = norm(y-x,1) / (epsilon*norm(x,1));
    end
end

%Display Values
disp('Average of Amplication Factors for H:');
disp(mean(Amp));
disp('Median of Amplication Factors for H:');
disp(median(Amp));
disp('Max of Amplication Factors for H:');
disp(max(Amp));
disp('kappa_{1}(H)');
disp(k1);

%plot
yl = yline(k1, '-','kappa_{1}(H)', 'LineWidth', 2);
yl.LabelHorizontalAlignment = 'center';
yl.Color = 'black';
hold on
num = [1:1:100];
scatter(num,Amp)
set(gca, 'YScale', 'log')
axis([1 100 0 10^9])
xlabel('Calculation Number')
ylabel('Amplification Factor')
hold off
```

```matlab
%%
%-------------------------------Question 1b------------------------------
%Matrix 1
%extract data from file
data = load('Data.mat');
A = data.E;
B = data.B;
BIGC = data.BIGC;
epsilon = data.epsilon;

%calculate amplification upper bound & the reciprocal of its reciprocal
k1 = cond(A,1);
k1_r = 1/rcond(A);
x = [];
y = [];

%Amplification Factors and index i
Amp = [];
i = 0;
%Calculate
for j = 1:10
    b = B(:,j);
    x = A\b;
    for n = 1:6
        C = BIGC(:,:,n);
        i = i+1;
        z = (A + epsilon*C)\b;
        Amp(i) = norm(z-x,1) / (epsilon*norm(x,1));
    end

end

%Display Values
disp('Average of Amplication Factors for E (Perturbed):');
disp(mean(Amp));
disp('Median of Amplication Factors for E (Perturbed):');
disp(median(Amp));
disp('Max of Amplication Factors for E (Perturbed):')
disp(max(Amp));
disp('kappa_{1}(perturbed E)');
disp(k1);
disp('1/kappa_{1}(perturbed E)^-1');
disp(k1_r);
%plot
y1 = yline(k1, '-','kappa_{1}(E)', 'LineWidth', 2);
y1.LabelHorizontalAlignment = 'center';
y1.Color = 'black';
y2 = yline(k1_r, '-','1/kappa_{1}^-1(E)', 'LineWidth', 2);
y1.LabelHorizontalAlignment = 'center';
y1.Color = 'black';

hold on
num = [1:60];
scatter(num,Amp)
set(gca, 'YScale', 'log')
axis([1 60 10^-4 10^5])
title('E Perturbed')
xlabel('Calculation Number')
ylabel('Amplification Factor')
hold off

```

```
%%
%--------------------------------Question 1b-------------------------------
%Matrix 2
%extract data from file
data = load('Data.mat');
A = data.H;
B = data.B;
BIGC = data.BIGC;
epsilon = data.epsilon;

%calculate amplification upper bound & the reciprocal of its reciprocal
k1 = cond(A,1);
k1_r = 1/rcond(A);
x = [];
y = [];

%Amplification Factors and index i
Amp = [];
i = 0;
%Calculate
for j = 1:10
    b = B(:,j);
    x = A\b;
    for n = 1:6
        C = BIGC(:,:,n);
        i=i+1;
        z = (A + epsilon*C)\b;
        Amp(i) = norm(z-x,1) / (epsilon*norm(x,1));
    end
end

%Display Values
disp('Average of Amplication Factors for H(Perturbed):')
disp(mean(Amp))
disp('Median of Amplication Factors for H(Perturbed):')
disp(median(Amp))
disp('Max of Amplication Factors for H(Perturbed):')
disp(max(Amp))
disp('kappa_{1}(perturbed H)')
disp(k1)
disp('1/kappa_{1}(perturbed H)^-1')
disp(k1_r)

%plot
y1 = yline(k1, '-','kappa_{1}(H)', 'LineWidth', 2);
y1.LabelHorizontalAlignment = 'center';
y1.Color = 'black';
y2 = yline(k1_r, '-','1/kappa_{1}(H)^-1', 'LineWidth', 2);
y1.LabelHorizontalAlignment = 'center';
y1.Color = 'black';
hold on
num = [1:60];
scatter(num,Amp)
set(gca, 'YScale', 'log')
axis([1 60 10^0 10^8])
title('H Perturbed')
xlabel('Calculation Number')
ylabel('Amplification Factor')
hold off
```

```
%%
%---------------------------------Question 2 a-------------------------------
%extract data from file
data = load('Data.mat');
%Set up matrices
A = data.E;
AINV = inv(A);
AC = inv(AINV);
I = eye(6);
%Calculate norms
x = norm((A*AINV)-I);
y = norm(AC - A);
%Results
output = ['For A = E ->    norm((A*AINV)-I):', num2str(x), '    norm(AC - A): ', num2str(y)];
disp(output);

```

```
%%
%--------------------------------Question 2 b------------------------------
%extract data from file
data = load('Data.mat');
%Set up matrices
A = data.H;
HI = data.HI;
AINV = inv(A);
AC = inv(AINV);
I = eye(6);
%Calculate norms
x = norm((A*AINV)-I);
y = norm(AC - A);
distToExact = norm(AINV - HI);
%Results
output = ['For A = H ->    norm((A*AINV)-I):', num2str(x), '    norm(AC - A): ', num2str(y), '    norm(AINV - HI):', num2str(distToExact)];
disp(output);

```

```
%%
%-------------------------------Question 2 c------------------------------
%extract data from file
data = load('Data.mat');
%Set up matrices
A = data.H8;
HI8 = data.HI8;
AINV = inv(A);
AC = inv(AINV);
I = eye(8);
%Calculate norms
x = norm((A*AINV)-I);
y = norm(AC - A);
distToExact = norm(AINV - HI8);
k1 = cond(A,1);
%Results
output = ['For A = H8->    norm((A*AINV)-I):', num2str(x), '    norm(AC - A): ', num2str(y), '    norm(AINV - HI8):', num2str(distToExact), '    kappa_{1}(A):', num2str(k1)];
disp(output);
```