

Part 1: Report

Question 1:A

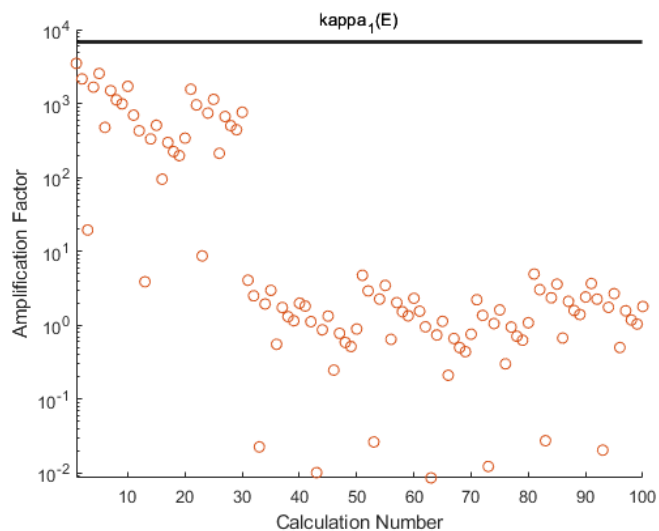
Note: all required values are displayed in table form for each section prior to their discussion.

Values Calculated for Matrix A = E (with $Ax = b + \Delta b$)

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 $Ax = b$

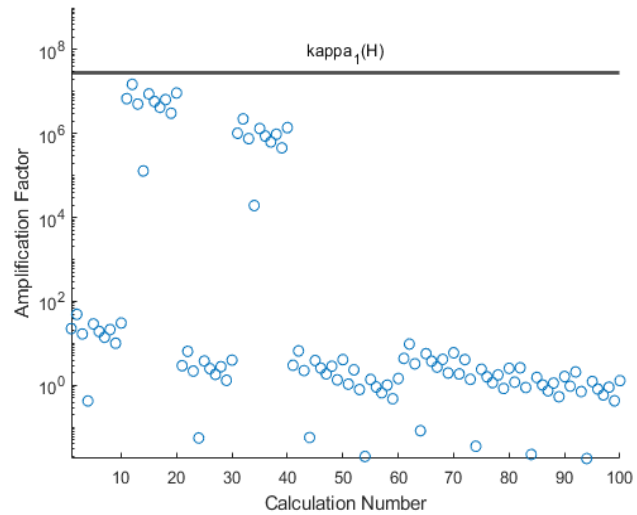
(where $\Delta b = b + \epsilon \cdot d$) is very large and give poor estimations to x ; however, by looking at the median (and the graph), we see that in most cases the error estimation is < 1 , with many very close to 0 (7 estimations are $< .1$), thus the perturbed b can give a fairly good estimation to x .



Values Calculated for Matrix A = H (with $Ax = b + \Delta b$)

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 .



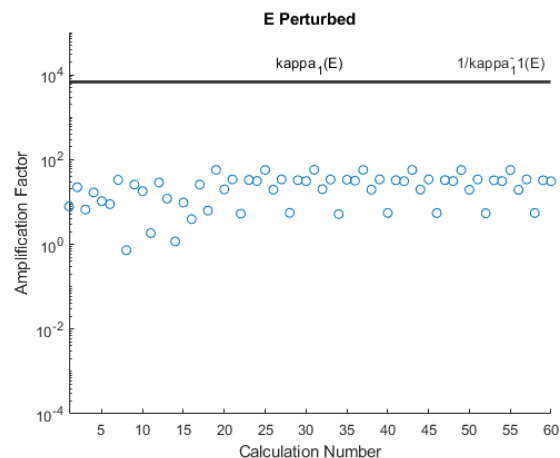
Question 1:B

Values Calculated for Matrix A = E (with $(A + \Delta A)x = b$)

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.

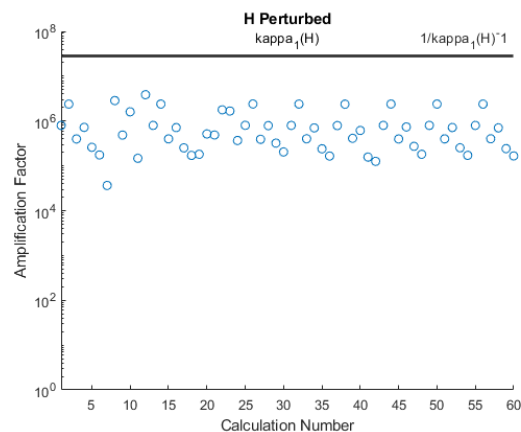
There is only 1 value that is less than 1 in this case, so very few estimations are good approximations to the x vectors. This shows that perturbations that originate on the left-side of the equation are more consistent than those on the right-side, as seen in part A, but cause much worse approximations,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



Values Calculated for Matrix A = H (with $(A + \Delta A)x = b$)

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)
$\text{norm}((A \cdot \text{AINV}) - I)$	9.8276e-14	3.1838e-10	1.3746e-07
$\text{norm}(AC - A)$	1.4949e-12	7.3695e-12	3.0106e-09
$\text{norm}(\text{AINV} - H I)$	-	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 H18 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.

```

1  %%
2  %-----Question 1a-----
3  %First Matrix
4
5  %extract data from file
6  data = load('Data.mat');
7  A = data.E;
8  B = data.B;
9  D = data.D;
10 epsilon = data.epsilon;
11
12 %calculate amplification upper bound
13 k1 = cond(A,1);
14 x = [];
15 y = [];
16
17 %Amplification Factors and index i
18 Amp = [];
19 i = 0;
20 %Calculate
21 for j = 1:10
22     b = B(:,j);
23     x = A\b;
24     for k = 1:10
25         i=i+1;
26         d = D(:,k);
27         y = A\b + epsilon*d;
28         Amp(i) = norm(y-x,1) / (epsilon*norm(x,1));
29     end
30 end
31
32 %Display Values
33 disp('Average of Amplification Factors for E:');
34 disp(mean(Amp));
35 disp('Median of Amplification Factors for E:');
36 disp(median(Amp));
37 disp('Max of Amplification Factors for E:');
38 disp(max(Amp));
39 disp('kappa_{1}(E)');
40 disp(k1);
41
42 %plot
43 yl = yline(k1, '-', 'kappa_{1}(E)', 'LineWidth', 2);
44 yl.LabelHorizontalAlignment = 'center';
45 yl.Color = 'black';
46 hold on
47 num = [1:1:100];
48 scatter(num,Amp)
49 set(gca, 'YScale', 'log')
50 xlabel('Calculation Number')
51 ylabel('Amplification Factor')
52 axis([1 100 0 10^4])
53 hold off
54

```

```

55 %%
56 %Second Matrix
57
58 %extract data from file
59 - data = load('Data.mat');
60 - A = data.H;
61 - B = data.B;
62 - D = data.D;
63 - epsilon = data.epsilon;
64
65 %calculate amplification upper bound
66 - k1 = cond(A,1);
67 - x = [];
68 - y = [];
69
70 %Amplification Factors and index i
71 - Amp = [];
72 - i = 0;
73 %Calculate
74 - for j = 1:10
75 -     b = B(:,j);
76 -     x = A\b;
77 -     for k = 1:10
78 -         i=i+1;
79 -         d = D(:,k);
80 -         y = A\b + epsilon*d;
81 -         Amp(i) = norm(y-x,1) / (epsilon*norm(x,1));
82 -     end
83 - end
84
85 %Display Values
86 - disp('Average of Amplification Factors for H:');
87 - disp(mean(Amp));
88 - disp('Median of Amplification Factors for H:');
89 - disp(median(Amp));
90 - disp('Max of Amplification Factors for H:');
91 - disp(max(Amp));
92 - disp('kappa_{1}(H)');
93 - disp(k1);
94
95 %plot
96 - yl = ylabel(k1, '-', 'kappa_{1}(H)', 'LineWidth', 2);
97 - yl.LabelHorizontalAlignment = 'center';
98 - yl.Color = 'black';
99 - hold on
100 - num = [1:1:100];
101 - scatter(num,Amp)
102 - set(gca, 'YScale', 'log')
103 - axis([1 100 0 10^9])
104 - xlabel('Calculation Number')
105 - ylabel('Amplification Factor')
106 - hold off
107

```

```

108 %%
109 %-----Question 1b-----
110 %Matrix 1
111 %extract data from file
112 data = load('Data.mat');
113 A = data.E;
114 B = data.B;
115 BIGC = data.BIGC;
116 epsilon = data.epsilon;
117
118 %calculate amplification upper bound & the reciprocal of its reciprocal
119 kl = cond(A,1);
120 kl_r = 1/rcond(A);
121 x = [];
122 y = [];
123
124 %Amplification Factors and index i
125 Amp = [];
126 i = 0;
127 %Calculate
128 for j = 1:10
129     b = B(:,j);
130     x = A\b;
131     for n = 1:6
132         C = BIGC(:, :, n);
133         i = i+1;
134         z = (A + epsilon*C)\b;
135         Amp(i) = norm(z-x,1) / (epsilon*norm(x,1));
136     end
137 end
138
139
140 %Display Values
141 disp('Average of Amplification Factors for E (Perturbed):');
142 disp(mean(Amp));
143 disp('Median of Amplification Factors for E (Perturbed):');
144 disp(median(Amp));
145 disp('Max of Amplification Factors for E (Perturbed):');
146 disp(max(Amp));
147 disp('kappa_{1}(perturbed E)');
148 disp(kl);
149 disp('1/kappa_{1}(perturbed E)^{-1}');
150 disp(kl_r);
151 %plot
152 y1 = yline(kl, '-', 'kappa_{1}(E)', 'LineWidth', 2);
153 y1.LabelHorizontalAlignment = 'center';
154 y1.Color = 'black';
155 y2 = yline(kl_r, '-', '1/kappa_{1}^{-1}(E)', 'LineWidth', 2);
156 y1.LabelHorizontalAlignment = 'center';
157 y1.Color = 'black';
158
159 hold on
160 num = [1:60];
161 scatter(num,Amp)
162 set(gca, 'YScale', 'log')
163 axis([1 60 10^{-4} 10^5])
164 title('E Perturbed')
165 xlabel('Calculation Number')
166 ylabel('Amplification Factor')
167 hold off
168

```

```

169 %%
170 %-----Question 1b-----
171 %Matrix 2
172 %extract data from file
173 data = load('Data.mat');
174 A = data.H;
175 B = data.B;
176 BIGC = data.BIGC;
177 epsilon = data.epsilon;
178
179 %calculate amplification upper bound & the reciprocal of its reciprocal
180 k1 = cond(A,1);
181 k1_r = 1/rcond(A);
182 x = [];
183 y = [];
184
185 %Amplification Factors and index i
186 Amp = [];
187 i = 0;
188 %Calculate
189 for j = 1:10
190     b = B(:,j);
191     x = A\b;
192     for n = 1:6
193         C = BIGC(:,n);
194         i=i+1;
195         z = (A + epsilon*C)\b;
196         Amp(i) = norm(z-x,1) / (epsilon*norm(x,1));
197     end
198 end
199
200 %Display Values
201 disp('Average of Amplification Factors for H(Perturbed):')
202 disp(mean(Amp))
203 disp('Median of Amplification Factors for H(Perturbed):')
204 disp(median(Amp))
205 disp('Max of Amplification Factors for H(Perturbed):')
206 disp(max(Amp))
207 disp('kappa_{1}(perturbed H)')
208 disp(k1)
209 disp('1/kappa_{1}(perturbed H)^{-1}')
210 disp(k1_r)
211
212 %plot
213 y1 = yline(k1, '-', 'kappa_{1}(H)', 'LineWidth', 2);
214 y1.LabelHorizontalAlignment = 'center';
215 y1.Color = 'black';
216 y2 = yline(k1_r, '-', '1/kappa_{1}(H)^{-1}', 'LineWidth', 2);
217 y1.LabelHorizontalAlignment = 'center';
218 y1.Color = 'black';
219 hold on
220 num = [1:60];
221 scatter(num,Amp)
222 set(gca, 'YScale', 'log')
223 axis([1 60 10^0 10^8])
224 title('H Perturbed')
225 xlabel('Calculation Number')
226 ylabel('Amplification Factor')
227 hold off
228
229

```



```

230 %%
231 %-----Question 2 a-----
232 %extract data from file
233 data = load('Data.mat');
234 %Set up matrices
235 A = data.E;
236 AINV = inv(A);
237 AC = inv(AINV);
238 I = eye(6);
239 %Calculate norms
240 x = norm((A*AINV)-I);
241 y = norm(AC - A);
242 %Results
243 output = ['For A = E ->   norm((A*AINV)-I):', num2str(x), '   norm(AC - A): ', num2str(y)];
244 disp(output);
245
246

```

```

247 %%
248 %-----Question 2 b-----
249 %extract data from file
250 data = load('Data.mat');
251 %Set up matrices
252 A = data.H;
253 HI = data.HI;
254 AINV = inv(A);
255 AC = inv(AINV);
256 I = eye(6);
257 %Calculate norms
258 x = norm((A*AINV)-I);
259 y = norm(AC - A);
260 distToExact = norm(AINV - HI);
261 %Results
262 output = ['For A = H ->   norm((A*AINV)-I):', num2str(x), '   norm(AC - A): ', num2str(y), '   norm(AINV - HI):', num2str(distToExact)];
263 disp(output);
264

```

```

265 %%
266 %-----Question 2 c-----
267 %extract data from file
268 data = load('Data.mat');
269 %Set up matrices
270 A = data.H8;
271 HI8 = data.HI8;
272 AINV = inv(A);
273 AC = inv(AINV);
274 I = eye(8);
275 %Calculate norms
276 x = norm((A*AINV)-I);
277 y = norm(AC - A);
278 distToExact = norm(AINV - HI8);
279 kl = cond(A,1);
280 %Results
281 output = ['For A = H8->   norm((A*AINV)-I):', num2str(x), '   norm(AC - A): ', num2str(y), '   norm(AINV - HI8):', num2str(distToExact), '   kappa_1(A):', num2str(kl)];
282 disp(output);

```