Table of Contents

AMSC 460 - HW8

```
clear all; format compact; close all; syms f(x) x y
```

Problem 1 (finished)

Let A ∈ R2×2 be given by A = [1 1+ #; 1- # 1] Find kAk# and kA−1#∞. What is cond#(A)? Is this matrix well-conditioned or ill-conditioned? Let b = [1 + (1 + #)#, 1]^T. Find the exact solution of Ax = b. Use MATLAB's backslash command to solve Ax = b for progressively smaller values of # = 10^k for k = −5, −6, ... At which value of # does the computed solution no longer accurately represent the true solution?

```
syms e % use e to represnt #
A = [1 1+ e; 1- e 1]
A^-1
% Assume # is small and positive
% We found ||A||_#=2+# and ||A^-1||_# = 1/e
% So the condition of A will be (||A||_#)*(||A^-1||_#)= (2+e)/e
% The solution of Ax=b is
```

```
A =
[     1, e + 1]
[1 - e,      1]
ans =
[       1/e^2, -(e + 1)/e^2]
[(e - 1)/e^2,        1/e^2]
```

```
b = transpose([1 + (1 + e )*e, 1])
x = (A^-1)*b
% By processing we can see the solution of Ax= b is x = [1; e]
```

```
b =
e*(e + 1) + 1
            1
x =
    (e*(e + 1) + 1)/e^2 - (e + 1)/e^2
1/e^2 + ((e - 1)*(e*(e + 1) + 1))/e^2
```

```
k = 5;
while k < 10
    e = 10^-k; % use e to represnt #
    A=[1 1+e; 1-e 1];
    b = transpose([1 + (1 + e )*e, 1]);
    x=A\b
    fprintf(['\twhen # = 10^-%.1f\n'], k)
    k = k + 1;
```

```
end

x =
    1.0000
    0.0000
 when # = 10^-5.0
x =
    1.0000
    0.0000
 when # = 10^-6.0
x =
    1.0104
   -0.0104
 when # = 10^-7.0
Warning: Matrix is close to singular or badly scaled. Results may be
 inaccurate.
RCOND =  2.110223e-16.
x =
    1.0522
   -0.0522
 when # = 10^-8.0
Warning: Matrix is close to singular or badly scaled. Results may be
 inaccurate.
RCOND =  1.100223e-16.
x =
     0
     1
 when # = 10^-9.0
```

```
          At # = 10^-8 the computed solution no longer accurately represent the tr
```

Problem 2 (Yup)

(MATLAB) Consider the system Ax = b where b = [.254 .127]^T and A = [.913 .659; .457 .330].Use the MATLAB backslash command to find the exact solution x. Usethe command cond to find the 2-norm condition number of A. Consider the two approximate solutions x1 = [–0.0827 0.5]^T, and x2 = [0.999 _ 1.001]^T Using the norm command, compute (a) the relative forward errors for x1 and x2 using the 2-norm (b) the relative backward errors for x1 and x2

```
A = [0.913 0.659; 0.457 0.330];
b = transpose([0.254 0.127]);
x = A\b
Two_norm_condition_number_of_A = cond(A,2)

x1 = transpose([-0.0827 0.5]);
x2 = transpose([0.999 -1.001]);

relative_forward_errors_for_x1 = norm(x-x1,2)/norm(x,2)
relative_forward_errors_for_x2 = norm(x-x2,2)/norm(x,2)
```

```
x =
    1.0000
   -1.0000
Two_norm_condition_number_of_A =
```

```
   1.2485e+04
relative_forward_errors_for_x1 =
    1.3081
relative_forward_errors_for_x2 =
   1.0000e-03
```

```
%The size of relative forward error 1 is near 1 while the size of
 error 2 is very small

relative_backward_errors_for_x1 = norm(b-A*x1,2)/norm(b,2)
relative_backward_errors_for_x2 = norm(b-A*x2,2)/norm(b,2)
```

```
relative_backward_errors_for_x1 =
   7.2598e-04
relative_backward_errors_for_x2 =
    0.0062
```

```
%The size of relative backward error 1 is very small while error 2 is
 relatively larger
```

Comment on the size of your backward and forward errors. Does a small backward error imply an approximate solution is accurate? How do your observations relate to the condition number of A?

A small backward error does not imply an approximate solution is accurate. The condition number of A is equal to 1.248e+04 which is very large, which is bad and we call it ill-conditioned, and we expect to lose 4 digits of accuracy in computing x.

Published with MATLAB® R2020b