Mehak Gupta
301311972

COMPUTING ASSIGNMENT 3

a) For a sample matrix the various norms and conditions numbers are:-

- Onenorm = 2.6289 Onecond = 14.879
- Twonorm = 1.7823 Twocond = 9.7789
- Infnorm = 2.1233 Infcond = 16.0975
- Fronorm = 1.9180 Frocond = 10.9428

After running the code a couple of times I have observed that Norm values always seem to be in between [0,3] but cond values vary greatly and are between [1, Infinity).

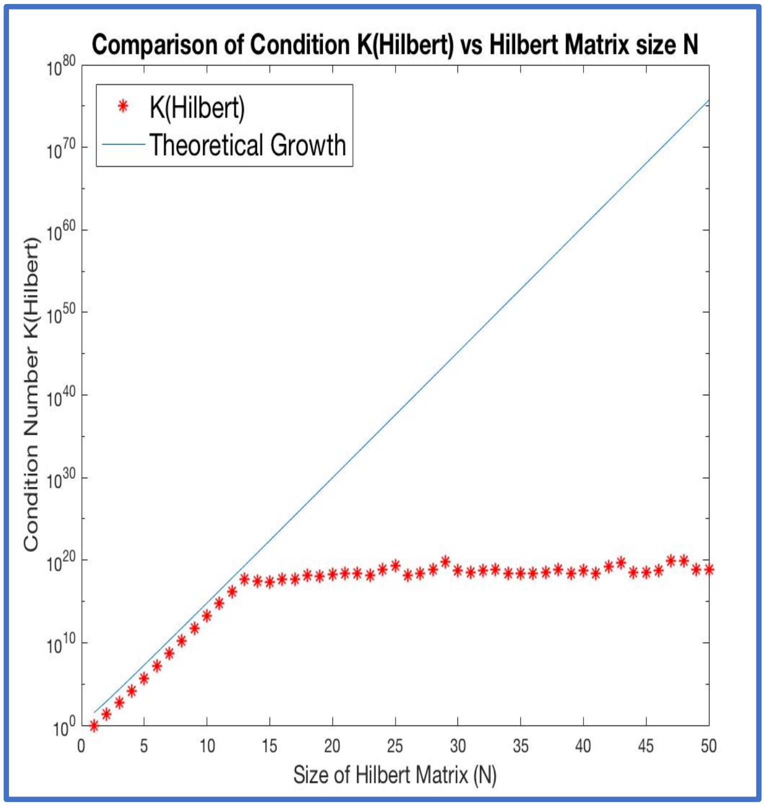

Fig (a)

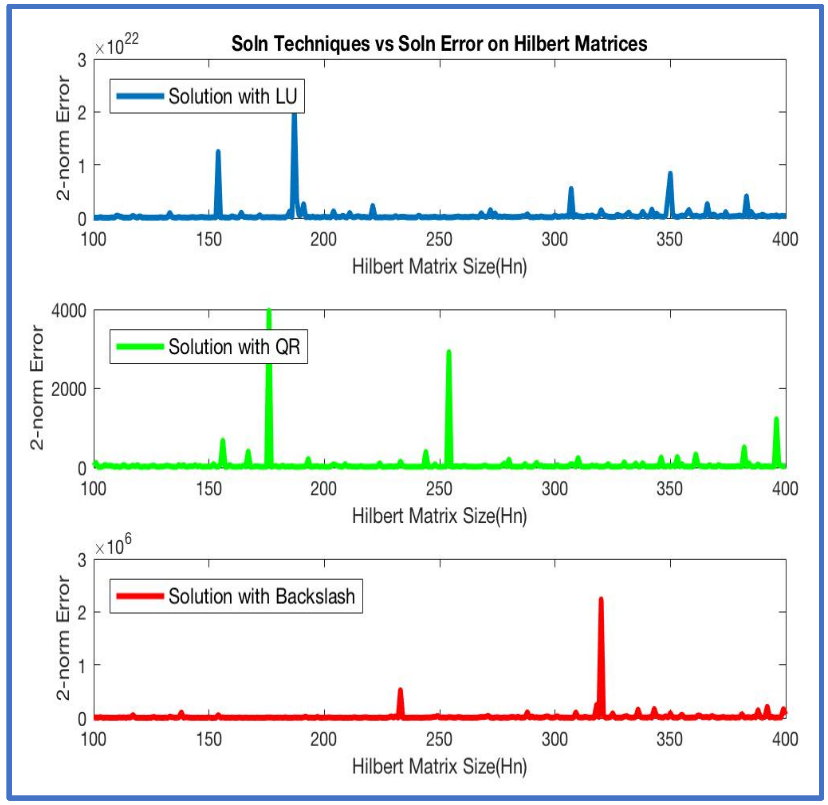

Fig (b)

b) Fig (a) gives the plot of cond(Hn) versus n with the comparison to the theoretical anticipated growth. The condition number of the Hilbert matrix grows along the theoretical growth until n =13. After n >13 the observed growth is hindered and does not follow the theoretical growth as shown in fig(a). The reasoning is explained in more detail in part (e).

c) Fig (b). Subplot command is used to plot the errors while using LU, QR and Backslash versus dimension of Hn simultaneously. As shown in the fig(b), LU is the most inaccurate with a calculated error of 10^22 approx. Backslash is definitely better than LU with an error of 10^6 and QR is the best with only 10^3 error. So definitely on the basis of accuracy QR is the best method to solve linear systems using Hilbert matrix.

d) Through the accuracy numbers that we see in part (c) it is clear that QR has best accuracy out of the three but slowest running time. LU has worst accuracy but it is the fastest among the QR, LU and Backslash. Backslash has moderate accuracy and running time.

e) Question- Behaviour of the condition number of the matrix in part(b) and comparison with the anticipated theoretical growth. Solution- The Hilbert matrix becomes very ill-conditioned, and the differences in inputs to the function used to estimate the condition number become so large that round-off error eventually hides the growth of the condition number of the Hilbert matrix, resulting in the jagged ridge starting around n = 13. A singular matrix has no inverse and as the condition number of a matrix gets larger the matrix gets closer to becoming singular. Since K(Hn) = ||A|| ||A^-1|| the inverse gets harder to calculate as it requires precision more than what MATLAB can perform which results in more error.

CODE (FIG(a))

```
 Routine for CA NUMBER

 matrix norms example.
.ear;
 = rand(3,3) % define a random 3 by 3 matrix
 compute the 2 norm and condition number using the 2 norm.
>nenorm = norm(A,1)
>necond = cond(A,1) % 2-norm

:wonorm = norm(A,2)
:wocond = cond(A,2) % cond number with 2-norm

.nfnorm = norm(A,inf)
.nfcond = cond(A,inf)

:ronorm = norm(A,'fro')
:rocond = cond(A,'fro')


 generate Hilbert matrices and
 compute cond number with 2-norm
 = 50; % total numer of matrices
>ndofH = [];
_g =[];

 compute the cond number of Hn
>r n = 1:N
   Hn = hilb(n);
   condofH = [condofH cond(Hn,2)];
   t_g(n) = ((1+sqrt(2))^(4*n))/sqrt(n);
ıd

>ndofH(1)
_g(1)

:milogy(1:N,condofH,'r*', 'DisplayName','K(Hilbert)')
>ld on
:milogy(1:N,(t_g), 'DisplayName','Theoretical Growth')
.tle('Comparison of Condition K(Hilbert) vs Hilbert Matrix size N', 'fontsize', 15)
.abel('Size of Hilbert Matrix (N)', 'fontsize', 13)
.abel('Condition Number K(Hilbert)','fontsize',13)
jd = legend('Location','northwest')
jd.FontSize =16;
:gend('show')
>ld off

 add here similar plots for QR and backslash
```

CODE (FIG(b))

```
 Routine for CA NUMBER

 matrix norms example.
.ear;
 = rand(3,3) % define a random 3 by 3 matrix
 compute the 2 norm and condition number using the 2 norm.
/onorm = norm(A,2) % 2-norm
```

```
vocond = cond(A,2) % cond number with 2-norm

 Add code here for the computation of the 1-norm, the infinity norm
 and the Frobenius norm and condition number based on those norms.



 generate Hilbert matrices and
 compute cond number with 2-norm
= 50; % total numer of matrices
ondofH = [];

 compute the cond number of Hn
or n = 1:N
   Hn = hilb(n);
   condofH = [condofH cond(Hn,2)];
nd

 at this point you have a vector condofH that contains the condition
 number of the Hilber matrices from 1x1 to 50x50.
 Figure out how to plot this (regular plot?, log log plot?, semilog plot?)
 and also plot on the same graph the theoretical growth line. Include and
 explain this graph in your report.



 Third part - compare the performance of solving an ill-conditioned linear
 system using LU, QR and backslash.

.ndim = 100; % minimum number of rows and columns of Hilbert matrix
axdim = 400; % maximum number of rows and columns of Hilbert matrix
 errors in 2-norm for 3 methods
rrorlu = [];
rrorqr = [];
rrorbackslash = [];

or k = mindim:maxdim
   Hk = hilb(k); % generate Hilbert matrix
   x = ones(k,1); % give the solution of the system
   b = Hk*x; % % compute RHS
   % get solution back by using different methods
   [P,L,U] = lu(Hk); % lu factorization of Hk
   [Q,R] = qr(Hk); % qr factorization of Hk
   xlu = U \ (L \ (P * b)); % solution with LU
   xqr = R \ Q \ b; % solution with QR
   xbackslash = Hk \ b; % solution with backslash command
   % computing errors
   errorlu = [errorlu norm(xlu-x,2)];
   errorqr = [errorqr norm(xqr-x,2)];
   errorbackslash = [errorbackslash norm(xbackslash-x,2)];
nd

total errors
otalerrorlu = sum(errorlu)
otalerrorqr = sum(errorqr)
otalerrorbackslash = sum(errorbackslash)

 plot solutions
ubplot(3,1,1)
.ot(mindim:maxdim,errorlu,'LineWidth',3)
.abel('Hilbert Matrix Size(Hn)', 'fontsize', 11)
.abel('2-norm Error','fontsize',11)
gd = legend('Location','northwest')
gd.FontSize =11;
egend('Solution with LU')
```

```
.tle('Soln Techniques vs Soln Error on Hilbert Matrices','FontSize', 11)


ıbplot(3,1,2)
.ot(mindim:maxdim,errorqr,'g','LineWidth',3)
.abel('Hilbert Matrix Size(Hn)', 'fontsize', 11)
.abel('2-norm Error','fontsize',11)
jd2 = legend('Location','northwest')
jd2.FontSize =11;
egend('Solution with QR')

ıbplot(3,1,3)
.ot(mindim:maxdim,errorbackslash,'r','LineWidth',3)
.abel('Hilbert Matrix Size(Hn)', 'fontsize', 11)
.abel('2-norm Error','fontsize',11)
jd3 = legend('Location','northwest')
jd3.FontSize =11;
egend('Solution with Backslash')


 add here similar plots for QR and backslash
```