COMPUTING REPORT 3 MACM 316
Khang Vu
301255281

The four types of norm and condition calculation in Matlab produce different values than each other; though, 1-norm, 2-norm and Frobenius norm produces similar values of norm and its respective condition number, whereas infinity norm produces vastly different values than these three.

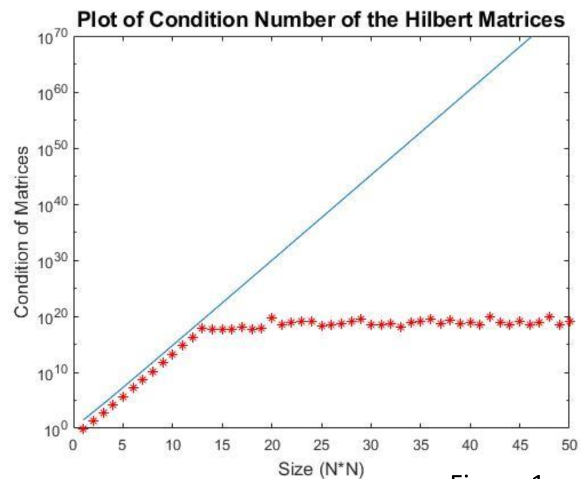

Figure 1

Figure (1) shows the graph of the condition numbers associated with matrices of dimensions ranging from 1 to 50. The red dots show the computed condition numbers and the blue line shows the anticipated growth of O ($\frac{(1+\sqrt{2})^{4n}}{\sqrt{(2)}}$). The computed values show that the anticipated growth is indeed true, although the values do plateau when n>13, possibly due to finite precision calculation error.

Figure (2) shows the error cumulated while solving Hibert matrices of dimensions ranging from 100 to 400. As shown by the plots, solving Hilbert matrices with LU decomposition produces the largest errors among the three methods, while solving using QR factorization produces the least errors. Thus, it seems that for solving Hilbert matrices, QR factorization method is recommended.

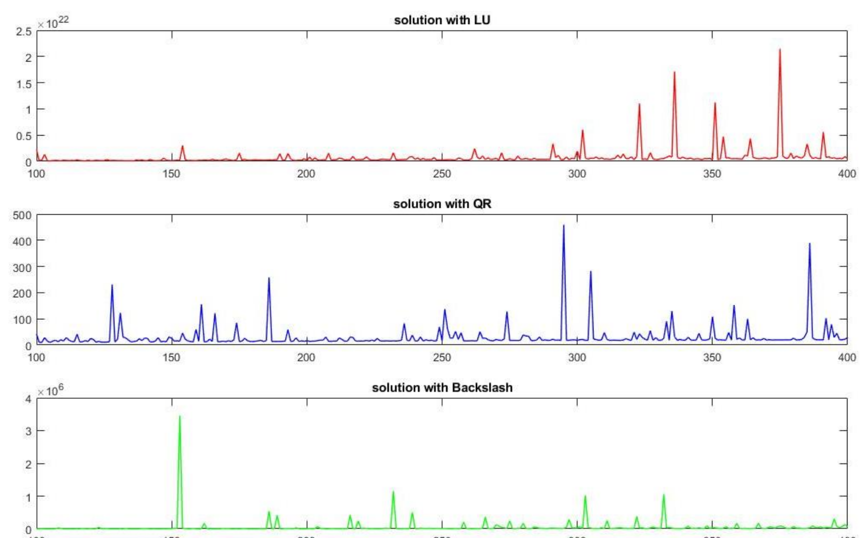

Figure 2

```
%Computing Report 3
%Ill-conditioned matrix, Hilbert Matrix


A = rand(3,3); % define a random 3 by 3 matrix
% compute the 2 norm and condition number using the 2 norm.
twonorm = norm(A,2) % 2-norm
twocond = cond(A,2) % cond number with 2-norm

% Code for the computation of the 1-norm, the infinity norm
% and the Frobenius norm and condition number based on those norms.
onenorm=norm(A,1)
onecond=cond(A,2)

infnorm=cond(A,inf)
infcond=cond(A,inf)

fronorm=norm(A,'fro')
frocond=cond(A,'fro')

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


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

% 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.

figure(1)
semilogy((1:N), (condofH), 'r*')
title(['Plot of Condition Number of the Hilbert Matrices'],'fontsize',14)
ylabel('Condition of Matrices','fontsize',12)
xlabel('Size (N*N)','fontsize',12)

hold on
fplot (@(x) (1+sqrt(2))^(4*x)/ sqrt(x), [1 50])
hold off

% Third part - compare the performance of solving an ill-conditioned linear
% system using LU, QR and backslash.
%
mindim = 100; % minimum number of rows and columns of Hilbert matrix
```

```matlab
maxdim = 400; % maximum number of rows and columns of Hilbert matrix
% errors in 2-norm for 3 methods
errorlu = [];
errorqr = [];
errorbackslash = [];

for 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)];
end

%total errors
totalerrorlu = sum(errorlu)
totalerrorqr = sum(errorqr)
totalerrorbackslash = sum(errorbackslash)

% plot solutions
figure (2)
subplot(3,1,1)
plot(mindim:maxdim,errorlu, 'r','LineWidth',1)
title('solution with LU')

% add here similar plots for QR and backslash
subplot(3,1,2)
plot(mindim:maxdim,errorqr,'b','LineWidth',1)
title('solution with QR')

subplot(3,1,3)
plot(mindim:maxdim,errorbackslash,'g','LineWidth',1)
title('solution with Backslash')
```