Lab Session 9

MA-581 : Numerical Computations Lab R. Alam Date: October 25, 2025

Errors, pivot growth and ill-conditioning

Sensitivity and accuracy: Suppose that $\hat{x}$ is a computed solution of $Ax = b$. Then it can be shown that $(A + E)\hat{x} = b$ for some E and the backward error of $\hat{x}$ is given by

$$\eta(\hat{x}, A) := \|E\|_2/\|A\|_2 = \frac{\|A\hat{x} - b\|_2}{\|A\|_2\|x\|_2}.$$

If $\eta(\hat{x}, A) = \mathcal{O}(\mathbf{u}))$ then the algorithm is backward stable.

The sensitivity of the system is measured by $\text{cond}(A) := \|A\|_2\|A^{-1}\|_2$ which is called the condition number of A. By perturbation theory, we have

$$\frac{\|x - \hat{x}\|_2}{\|x\|_2} \lesssim \text{cond}(A)\,\eta(\hat{x}, A).$$

Suppose that $\|\cdot\|$ is either the 1-norm, ∞-norm or the 2-norm and that x and $\hat{x}$ are two vectors such that $\|x - \hat{x}\|/\|x\| \leq 0.5 \times 10^{-p}$. Then x and $\hat{x}$ agree to p significant digits in the entries j which satisfy $|x_j| \simeq \|x\|$.

The purpose of this lab tutorial is to understand ill-conditioning of linear system, stability of an algorithm and their influence on the accuracy of computed solution.

1. Consider the Hilbert matrix H, where $H(i, j) := 1/(i + j - 1)$ (the MATLAB command `H = hilb(n)` generates H) and perform the following experiments.

   (a) Convince yourself that the condition number of H grows quickly with n. Try

   ```
   >> C=[];
   >> N= 2:2:16;
   for n=N
   H=hilb(n); C=[C; cond(H)];
   end
   >> semilogy(N,C)
   ```

   Can you guess an approximate relationship between $\text{cond}(H)$ and n based on this graph? The MATLAB `cond(H)` computes the 2-norm condition number of H. Theoretically $\text{cond}(H) \approx \left(\frac{(1+\sqrt{2})^{4n}}{\sqrt{n}}\right)$. Plot (in a single plot) the theoretical value of $\text{cond}(H)$ and $\text{cond}(H)$ computed by MATLAB. The condition number computed by MATLAB reaches the maximum when $n = 13$. The computed condition number does not continue to grow when $n > 13$. This can be explained as follows: Since H is positive definite (that is, $H = H^*$ and eigenvalues of H are all positive), $\text{cond}(H) = \lambda_{\max}(H)/\lambda_{\min}(H)$, where $\lambda_{\max}(H)$ and $\lambda_{\min}(H)$ are the largest and the smallest eigenvalues of H. It is a fact that $\lambda_{\max}(H) \to \pi$ and $\lambda_{\min}(H) \to 0$ as $n \to \infty$. Hence

   $$\text{cond}(H) = \frac{\sigma_{\max}(H)}{\lambda_{\min}(H)} \approx \frac{\pi}{\lambda_{\min}(H) + \texttt{eps}} \approx \frac{\pi}{\texttt{eps}}.$$

```
%%%%%%     Matlab program that implements growth of cond(H)
% generate Hilbert matrices and compute cond number with 2-norm

N=50; % maximum size of a matrix
condofH = []; % conditional number of Hilbert Matrix
N_it= zeros(1,N);


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

% at this point we have a vector condofH that contains the condition
% number of the Hilber matrices from 1x1 to 50x50.
% plot on the same graph the theoretical growth line.


% Theoretical growth of condofH
x = 1:50;
y = (1+sqrt(2)).^(4*x)./(sqrt(x));

% plot
plot(N_it, log(y));
plot(N_it, log(condofH),'x', N_it,log(y));


% plot labels
plot(N_it, log(condofH),'x', N_it,log(y))
title('Conditional Number growth of Hilbert Matrix: Theoretical vs Matlab')
xlabel('N', 'fontsize', 16)
ylabel('log(cond(Hn))','fontsize', 16)
lgd = legend ('Location', 'northwest')
legend('MatLab', 'Theoretical')
legend('show')

%%%%%%    end of the program
```

(b) The importance of condition number stems from the fact that if $\text{cond}(H) = 10^t$ then we may lose t digits in the computed solution of $Hx = b$. Examine this criterion by solving $Hx = b$. Here is how you can pick up the exact solution. Choose an arbitrary x and set $b := Hx$. Then x is the exact solution of $Hx = b$. The matrix H is SPD (symmetric positive definite). The matlab backslash `A \ b` command uses Cholesky factorization to solve an SPD system. There is also a matlab command `invhilb` which computes H^{-1} in a special way. You can also use GEPP (Gaussian Elimination with Partial Pivoting) to solve $Hx = b$. You may have to use `format` `long e` to see more digits. Try

```
>> n=8;
>> H=hilb(n); HI = invhilb(n);
>> x= rand(n,1);
>> b =H*x;
>> x1 = H\ b; % Call this is method1
>> x2 = HI*b; % Call this is method2
```

Compute backward error `eta`, condition number `cond` and the error `err` for method1 and method2 and display the result in the format `[ eta cond err]`.
Repeat for $n = 10$ and $n = 12$.

* List the results corresponding to $n = 8, 10, 12$, and determine correct digits in `x1, x2`.
* How many digits are lost in computing `x1` and `x2`? How does this correlate with the size of the condition number?
* Which is better among `x1` and `x2` or isn't there much of a difference? Is it fair to say that the inaccuracy resulted from a poor algorithm?

2. Wilkinson's matrix is defined as follows: 1 on the diagonal, -1 everywhere below the main diagonal, 1 in the last column, and 0 everywhere else. Write a MATLAB function `W = Wilkinson(n)` that generates Wilkinson's matrix W of size n using MATLAB functions `eye, tril` and `ones`.

   For $n = 32$, pick a random x and then compute $b := W * x$. Solve $Ax = b$ using MATLAB backslash command and compute the error $\|x - \hat{x}\|_\infty / \|x\|_\infty$ (type `help norm` for more info about computing norm). Does the size of the error confirm that GEPP is unstable for this system? Also compute cond(A). Can the poor answer be attributed to ill-conditioning of the matrix W? Repeat the test for $n = 64$.

3. Pivot growth of Gaussian elimination with partial pivoting (GEPP) is given by $PG(A) = \max_{ij} |U(i,j)| / \max_{ij} |A(i,j)|$, which influences the accuracy of computed solution. Use MATLAB function `[L, U, p] = lu(A)` for computing LU decomposition of a nonsingular matrix A and compute the pivot growth `rho` $= PG(A)$.

   [**Hint:** Given a matrix X, `max(X)` is a row vector containing the maximum element from each column.]

   It is well known that the pivot growth factor for GEPP satisfies $PG(A) \leq 2^{n-1}$ which is attained by the Wilkinson matrix. Verify this graphically by doing the following:

   First plot the graph of 2^{n-1} in $\log 10$ scale for $n = 10 : .5 : 505$ by setting `X = 2.^(n-1)` and then typing `semilogy(n,X,'r')`. Hold this plot by typing `hold on` and type the following sequence of commands (which assumes that the Wilkinson matrix of size n is generated by the function `W = Wilkinson(n)`).

```
n = 10:20:500;
m = length(n); G = zeros(m,1);
for i = 1:m
W = Wilkinson(n(i)); [L,U,p] = lu(W);
```

```
G(i) = max(max(abs(U)))/max(max(abs(W)));
end
semilogy(n,G,'b*')
```

The second plot should come in the form of blue dots that fall on the red curve produced by the earlier plot.

However, statistics suggest that for most practical examples, $PG(A) \leq n^{2/3}$ for GEPP. In fact, for random matrices, the average pivot growth for GEPP is $PG(A) \leq \frac{1}{4}n^{0.71}$. You can verify this fact for random matrices as follows.

```
N=500;
n=[10 20 30 40 50 60 70 80 90 100];
for j=1:length(n)
m=0;
for i=1:N
A=rand(n(j));
[L,U,P]=lu(A);
rho = max( max( abs(U)))/max( max( abs(A)));
m=m+rho;
end;
g(j)=m/N;
end;
plot(n,g,'--', n,0.25*n.^(0.71),'-');
legend('average growth factor', '0.25*n^{0.71}','Location','NorthWest')
xlabel('matrix size'),ylabel('growth factor \rho')
```

4. There is no strong correlation between pivot growth and the ill-conditioning of a matrix. This is illustrated by a Golub matrix. A Golub matrix A of size n is an ill-conditioned integer matrix whose LU factorization without pivoting fails to reveal that A is ill-conditioned. The matrix A is given by $A := LU$, where L unit lower triangular with random integer entries and U is unit upper triangular with random integer entries. The function `golub` given below generates a Golub matrix of size n :

```
function A = golub(n)
s = 10;
L = tril(round(s*randn(n)),-1)+eye(n);
U = triu(round(s*randn(n)),1)+eye(n);
A = L*U;
```

Compute LU factorization of A using your function `[L, U] = GENP(A)`. Also, compute the pivot growth $PG(A)$ and the condition number $\text{cond}(A) = \|A\|_\infty \|A^{-1}\|_\infty$ using MATLAB command `cond(A)`. If $\text{cond}(A)$ is large then the system $Ax = b$ is ill-conditioned and in such a case A is called ill-conditioned. Does $PG(A)$ reflect the ill-conditioning of A?

*************End************