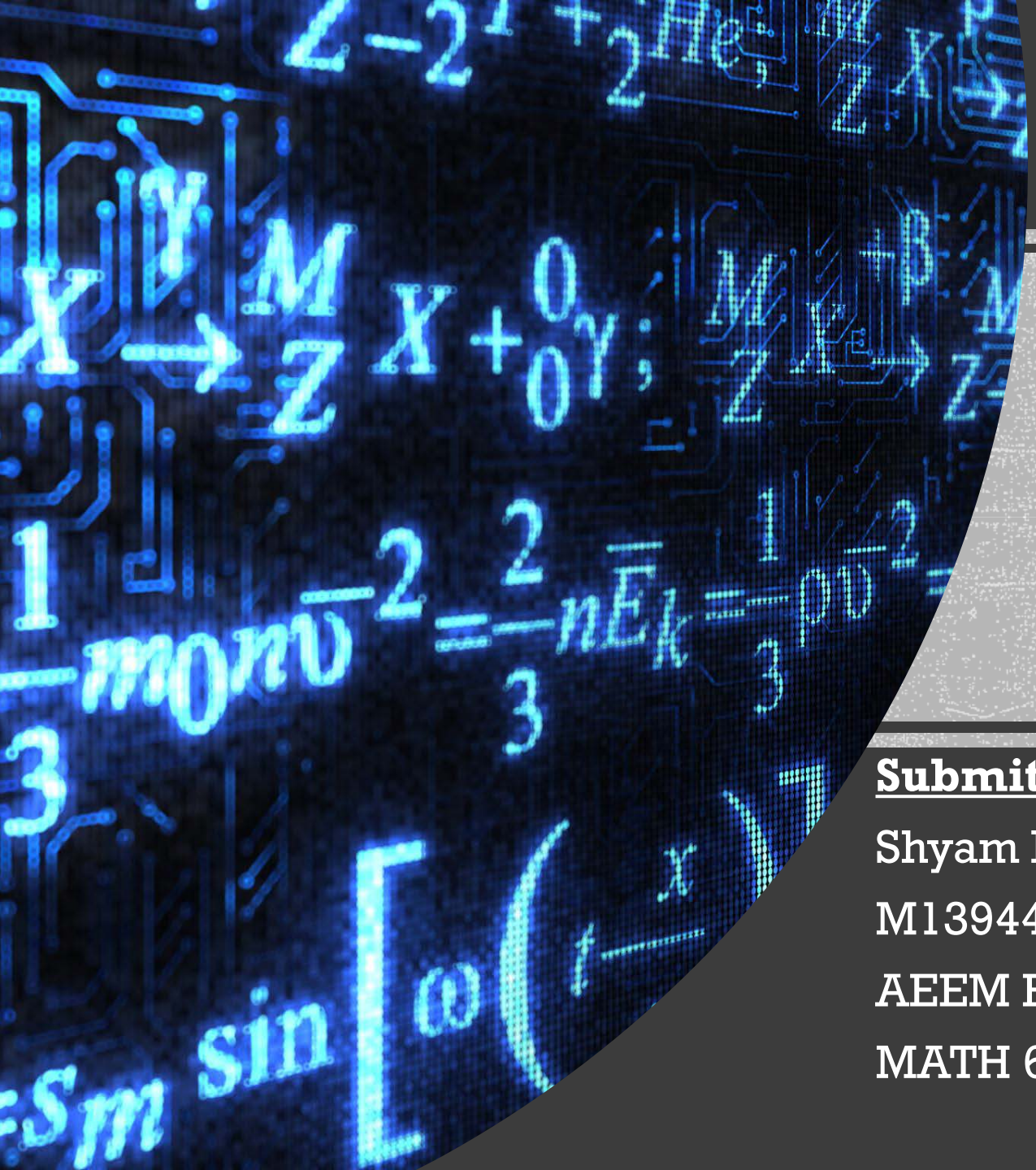


NUMERICAL METHODS FOR ILL-CONDITIONED MATRICES -A PROJECT WITH HILBERT MATRIX

Submitted By:

Shyam Rauniyar

M13944385

AEEM Fall 2020

MATH 6012

Submitted To:

Dr. Yizao Wang

Faculty of Applied Linear Algebra

MATH 6012

University of Cincinnati

OUTLINE OF PRESENTATION

- About Ill-Conditioned Matrices
- About Hilbert Matrices and its Ill-Conditioned Nature
- QR Decomposition of Hilbert Matrix of Order $n = 2, 3, 4, 10, 20$
- Solving $H_n \boldsymbol{x} = \boldsymbol{y}^*$, where $\boldsymbol{y}^* = H_n \boldsymbol{x}^*$ and $x_i^* = (-1)^i i/(i+1)$
  - Using Gaussian Elimination (GE)
  - Using Stable Gram-Schmidt (GS) Process
  - Using Householder's (HH) Method
- Checking Orthogonality of Q's of GS and HH using Frobenius Norm
- Comparing the Methods using $\|\boldsymbol{x}^* - \boldsymbol{x}\|_\infty$ for Errors
- Comments on Each Numerical Method based on Results

ILL-CONDITIONED MATRICES

- $Ax = y$ is a linear system where A is ill-conditioned matrix when a small change in the constant coefficients results in a large change in the solution.
- The condition number associated with the linear equation $Ax = y$ gives a bound on how inaccurate the solution x will be after approximation
- Conditioning is a property of the matrix, not the algorithm or floating-point accuracy of the computer used to solve the corresponding system
- If the condition number is large, even a small error in b may cause a large error in x

HILBERT MATRICES

- $a_{ij} = \frac{1}{i+j-1}$
- Hilbert matrices are ill-conditioned
- They have large condition numbers
- Hilbert matrices are nearly singular
- Inverting Hilbert matrices is numerically unstable
- There are some matrices that are hard to handle even with sophisticated pivoting strategies. Such ill-conditioned matrices are typically characterized by being "almost" singular. A famous example of an ill-conditioned matrix is the n × n Hilbert matrix.

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1} \end{bmatrix}$$

USE OF MATLAB FOR FURTHER CALCULATIONS

- Programmed following functions for solving linear systems
  - $Hilbert\_n(n)$ for Hilbert Matrix of order n.
  - $solverGauss(A, \boldsymbol{b})$ with Gaussian Elimination algorithm
  - $QRfactor\_GS(A)$ with Gram-Schmidt algorithm for QR decomposition
  - $QRfactor\_HH(A)$ with Householder Method for QR decomposition
  - $backsub(U, \boldsymbol{c})$ *with backward substitution algorithm*

- Step to $\boldsymbol{y}^* = H_n \boldsymbol{x}^*$
  - Input Order(s) of Hilbert Matrix (H_n)
  - Calculate Hilbert Matrix with $Hilbert\_n(n)$
  - Calculate $\boldsymbol{x}^*$ with $x_i^* = \frac{(-1)^i i}{i+1}$
  - Calculate $\boldsymbol{y}^* = H_n \boldsymbol{x}^*$

SOLVING $H_n \boldsymbol{x} = \boldsymbol{y}^*$

- Gaussian Elimination
  - Determined $(U, \boldsymbol{c})$ with: *solveGauss* $(H_n, \boldsymbol{y}^*)$
  - Solved for $\boldsymbol{x}$ using: $backsub(U, \boldsymbol{c})$

- Stable Gram-Schmidt (GS)
  - Determined $[Q, R]$ of H_n by function: $QRfactor\_GS(H_n)$
  - Calculate: $norm(QTQ - eye(n), 'fro')$
  - Calculated: $\boldsymbol{c} = Q^T * \boldsymbol{b}$
  - Solved for $\boldsymbol{x}$ using: $backsub(R, \boldsymbol{c})$

- Householder Method (HH)
  - Determined $[Q, R]$ of H_n by function: $QRfactor\_HH(H_n)$
  - Calculate: $norm(QTQ - eye(n), 'fro')$
  - Calculated: $\boldsymbol{c} = Q^T * \boldsymbol{b}$
  - Solved for $\boldsymbol{x}$ using: $backsub(R, \boldsymbol{c})$

Note: $eye(n)$ produces Identity Matrix of order n

$[Q, R]$ FOR $n = 2, 3, 4$ USING STABLE GS

$n = 2$

$$Q = \begin{pmatrix} 0.8944 & -44772 \\ 0.4472 & 0.8944 \end{pmatrix}$$

$$R = \begin{pmatrix} 1.1180 & 0.5963 \\ 0.0000 & 0.0745 \end{pmatrix}$$

$n = 3$

$$Q = \begin{pmatrix} 0.8571 & -0.5016 & 0.1170 \\ 0.4286 & 0.5685 & -0.7022 \\ 0.2857 & 0.6521 & 0.7022 \end{pmatrix}$$

$$R = \begin{pmatrix} 1.667 & 0.6429 & 0.45 \\ 0.0000 & 0.1017 & 0.1053 \\ 0.0000 & 0.0000 & 0.0039 \end{pmatrix}$$

$n = 4$

$$Q = \begin{pmatrix} 0.8381 & -0.5226 & 0.1540 & -0.0263 \\ 0.4191 & 0.4417 & -0.7278 & 0.31577 \\ 0.2794 & 0.5288 & 0.1395 & -0.7892 \\ 0.2095 & 0.5021 & 0.6536 & 0.5261 \end{pmatrix}$$

$$R = \begin{pmatrix} 1.1932 & 0.6705 & 0.4749 & 0.3698 \\ 0.0000 & 0.1185 & 0.1257 & 0.1175 \\ 0.0000 & 0.0000 & 0.0062 & 0.0096 \\ 0.0000 & 0.0000 & 0.0000 & 0.0002 \end{pmatrix}$$

$[Q, R]$ FOR $n = 2, 3, 4$ USING HH

$n = 2$

$$Q = \begin{pmatrix} -0.8944 & -44772 \\ -0.4472 & 0.8944 \end{pmatrix}$$

$$R = \begin{pmatrix} -1.1180 & -0.5963 \\ 0.0000 & 0.0745 \end{pmatrix}$$

$n = 3$

$$Q = \begin{pmatrix} -0.8571 & 0.5016 & 0.1170 \\ -0.4286 & -0.5685 & -0.7022 \\ -0.2857 & -0.6521 & 0.7022 \end{pmatrix}$$

$$R = \begin{pmatrix} -1.667 & -0.6429 & -0.45 \\ 0.0000 & -0.1017 & -0.1053 \\ 0.0000 & 0.0000 & 0.0039 \end{pmatrix}$$

$n = 4$

$$Q = \begin{pmatrix} -0.8381 & 0.5226 & -0.1540 & -0.0263 \\ -0.4191 & -0.4417 & 0.7278 & 0.31577 \\ -0.2794 & -0.5288 & -0.1395 & -0.7892 \\ -0.2095 & -0.5021 & -0.6536 & 0.5261 \end{pmatrix}$$

$$R = \begin{pmatrix} -1.1932 & -0.6705 & -0.4749 & -0.3698 \\ 0.0000 & -0.1185 & -0.1257 & -0.1175 \\ -0.0000 & -0.0000 & -0.0062 & -0.0096 \\ -0.0000 & -0.0000 & 0.0000 & 0.0002 \end{pmatrix}$$

$[Q, R]$ FOR $n = 10, 20$ USING STABLE GS

$n = 10$

$$Q = \begin{pmatrix} 0.8033 & -0.5503 & \dots & 0.0001 \\ 0.4016 & 0.2544 & \dots & 0.0001 \\ \vdots & \vdots & \ddots & \vdots \\ 0.0803 & 0.2049 & \dots & 0.0710 \end{pmatrix}$$

$$R = \begin{pmatrix} 1.2449 & 0.7303 & \dots & 0.1973 \\ 0.0000 & 0.1574 & \dots & 0.1181 \\ \vdots & \vdots & \ddots & \vdots \\ 0.0000 & 0.0000 & \dots & 0.0000 \end{pmatrix}$$

$n = 20$

$$Q = \begin{pmatrix} 0.7915 & -0.5565 & \dots & -0.3191 \\ 0.3958 & 0.2015 & \dots & -0.0947 \\ \vdots & \vdots & \ddots & \vdots \\ 0.0396 & 0.1024 & \dots & 0.0802 \end{pmatrix}$$

$$R = \begin{pmatrix} 1.2634 & 0.7538 & \dots & 0.1205 \\ 0.0000 & 0.1737 & \dots & 0.0975 \\ \vdots & \vdots & \ddots & \vdots \\ 0.0000 & 0.0000 & \dots & 0.0000 \end{pmatrix}$$

$[Q, R]$ FOR $n = 10, 20$ USING HH

$n = 10$

$$Q = \begin{pmatrix} -0.8033 & 0.5503 & \dots & 0.0000 \\ -0.4016 & -0.2544 & \dots & 0.0001 \\ \vdots & \vdots & \ddots & \vdots \\ -0.0803 & -0.2049 & \dots & -.0710 \end{pmatrix}$$

$$R = \begin{pmatrix} -1.2449 & -0.7303 & \dots & -0.1973 \\ 0.0000 & -0.1574 & \dots & -0.1181 \\ \vdots & \vdots & \ddots & \vdots \\ -0.0000 & -0.0000 & \dots & 0.0000 \end{pmatrix}$$

$n = 20$

$$Q = \begin{pmatrix} -0.7915 & 0.5565 & \dots & 0.0000 \\ -0.3958 & -0.2015 & \dots & -0.0000 \\ \vdots & \vdots & \ddots & \vdots \\ -0.0396 & -0.1024 & \dots & 0.0050 \end{pmatrix}$$

$$R = \begin{pmatrix} -1.2634 & -0.7538 & \dots & -0.1205 \\ 0.0000 & -0.1737 & \dots & -0.0975 \\ \vdots & \vdots & \ddots & \vdots \\ 0.0000 & 0.0000 & \dots & 0.0000 \end{pmatrix}$$

SOLVING $H_n \boldsymbol{x} = \boldsymbol{y}^*$; WHERE $n = 2, 3, 4$

- The solution for $\boldsymbol{x}$ are same for all numerical methods and equivalent to actual x:
  - $\boldsymbol{x} = \begin{pmatrix} -0.5000 \\ 0.6667 \end{pmatrix}; n = 2$
  - $\boldsymbol{x} = \begin{pmatrix} -0.5000 \\ 0.6667 \\ -0.75 \end{pmatrix}; n = 3$
  - $\boldsymbol{x} = \begin{pmatrix} -0.5000 \\ 0.6667 \\ -0.75 \\ 0.8 \end{pmatrix}; n = 3$

SOLVING $H_n \boldsymbol{x} = \boldsymbol{y}^*$; $n = 10,20$; USING GE

$n = 10$

$$\boldsymbol{x} = \begin{pmatrix} -0.5000 \\ 0.6667 \\ -0.7500 \\ 0.8000 \\ -0.8333 \\ 0.8571 \\ 0.8750 \\ 0.8889 \\ 0.9000 \\ 0.9091 \end{pmatrix}$$

$n = 20$

$$\boldsymbol{x} = \begin{pmatrix} -0.5000 \\ 0.6667 \\ -0.7500 \\ 0.8013 \\ -0.8517 \\ 1.002 \\ -1.5439 \\ 2.8369 \\ \vdots \\ 1.4508 \end{pmatrix}$$

SOLVING $H_n \boldsymbol{x} = \boldsymbol{y}^*$; $n = 10,20$; USING STABLE GS

$n = 10$

$$\boldsymbol{x} = 10^7 \begin{pmatrix} -0.0000 \\ 0.0023 \\ -0.0458 \\ 0.4424 \\ -2.1169 \\ -5.8347 \\ 9.5940 \\ -9.2887 \\ 4.8842 \\ -1.0756 \end{pmatrix}$$

$n = 20$

$$\boldsymbol{x} = 10^{15} \begin{pmatrix} 0.0000 \\ -0.000 \\ -0.0001 \\ 0.0027 \\ -0.0298 \\ 0.8683 \\ -2.4871 \\ 4.6005 \\ \vdots \\ 1.4508 \end{pmatrix}$$

SOLVING $H_n \boldsymbol{x} = \boldsymbol{y}^*$; $n = 10,20$; USING HH

$n = 10$

$$\boldsymbol{x} = \begin{pmatrix} -0.5000 \\ 0.6667 \\ -0.7500 \\ 0.8000 \\ -0.8333 \\ 0.8571 \\ 0.8750 \\ 0.8889 \\ 0.9000 \\ 0.9091 \end{pmatrix}$$

$n = 20$

$$\boldsymbol{x} = \begin{pmatrix} -0.5000 \\ 0.6667 \\ -0.7501 \\ 0.8007 \\ -0.8318 \\ 0.7898 \\ -0.3769 \\ -1.0012 \\ \vdots \\ 0.4472 \end{pmatrix}$$

'n'	'GE_Err'	'GS_Err'	'HH_Err'
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
10	0.00	95939567.74	0.00
20	6.97	9492541210849210.00	12.59

ERRORS ($||x^* - x||_\infty$) FOR DIFFERENT ALGORITHMS

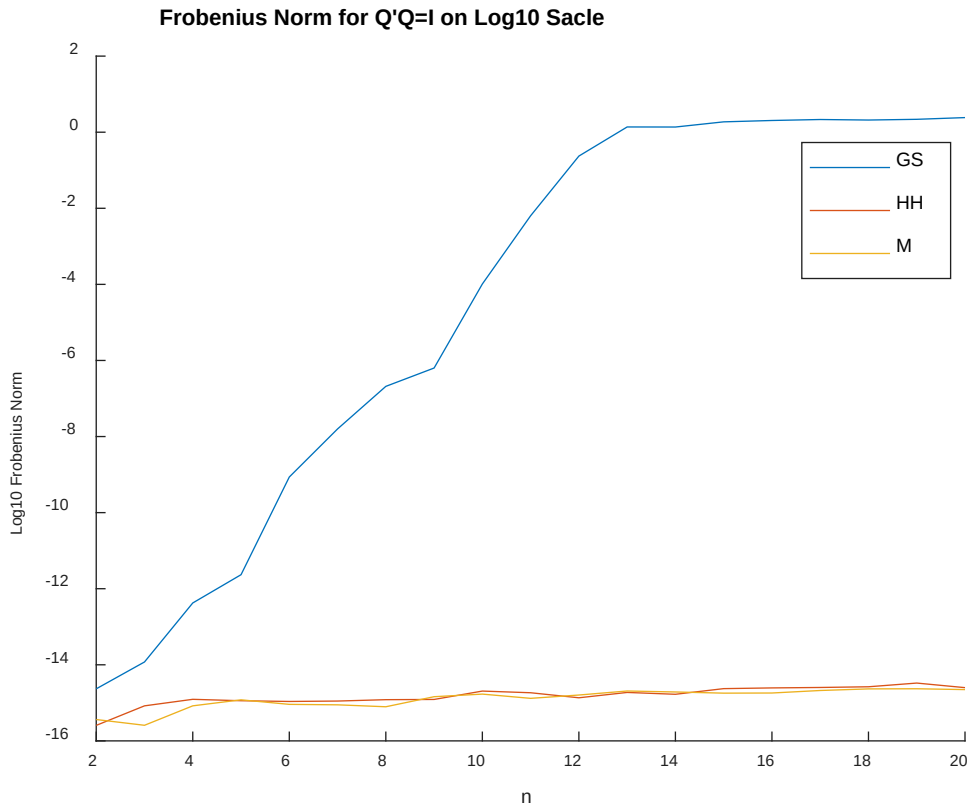
Log10 Scale will be used in graphical representation to work with large values and their deviation

FROBENIUS NORM CHECKING $Q^TQ = I$ FOR DIFFERENT NUMERICAL METHODS

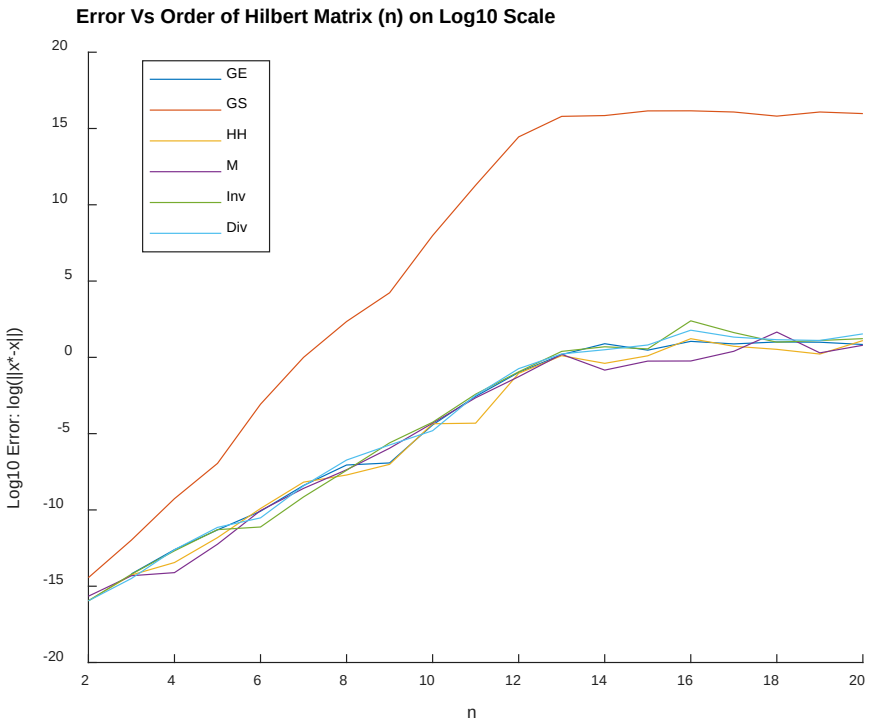
'n'	'GS_norm'	'HH_norm'	'M_norm'
2	2.34E-15	2.56E-16	3.64E-16
3	1.19E-14	8.34E-16	2.57E-16
4	4.22E-13	1.24E-15	8.34E-16
10	0.00010235	2.04E-15	1.70E-15
20	2.42091296	2.51E-15	2.23E-15

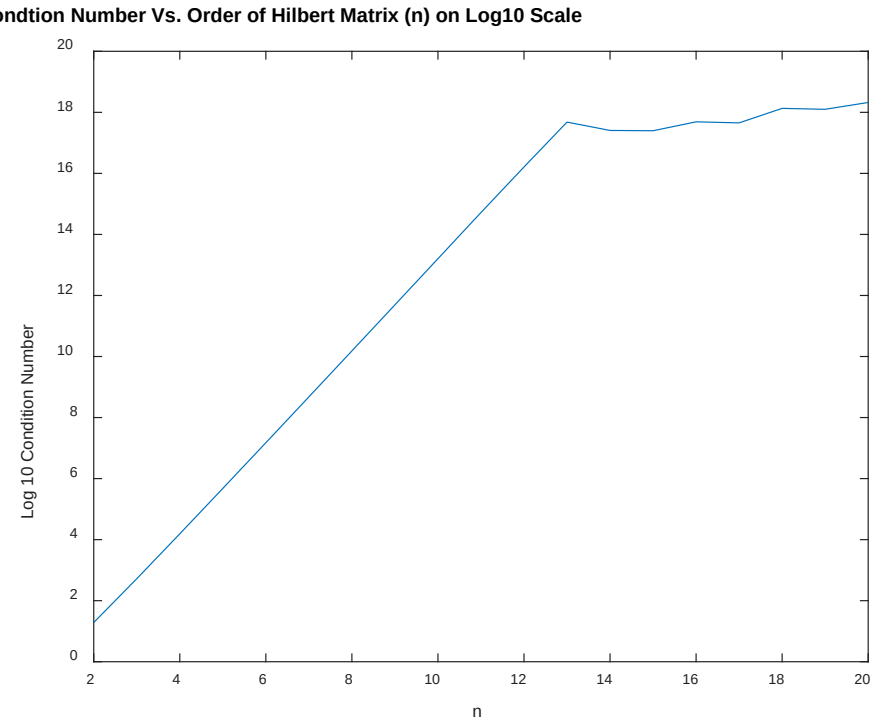
Log10 Scale will be used in graphical representation to work with large values and their deviation

FROBENIUS NORM TO CHECK $Q^TQ = I$ ON LOG10 SCALE



CONDITION NUMBERS AND ERRORS ($\|x^* - x\|_\infty$) ON LOG (BASE 10) SCALE





Note: 'M' is a method that uses $[\mathrm{Q}, \mathrm{R}] = qr()$ of MATLAB;
'Inv' is a method using $\boldsymbol{x} = inv(A) * \boldsymbol{b}$;
'Div' is a method $\boldsymbol{x} = A \backslash \boldsymbol{b}$

RESULTS FROM COMPARISONS

- Frobenius Norms on the $Q^TQ - I$ from Stable GS and HH methods suggests that, HH method is superior to Stable GS considering it provides a better QR decomposition where Q shows more orthogonality.
- Solving Linear Systems with Hilbert Matrix as coefficient shows that drastically better results are derived from HH and GE methods.
- MATLAB's $qr()$ function uses LAPACK routines to compute the QR decomposition and thus provides more consistent and accurate results.
- Condition number of Hilbert Matrices increases with its order.
- Accuracy of Solving Linear System is a result of condition number of coefficient matrix (here, Hilbert Matrix) and thus the accuracy for all the methods diminishes with increase in condition number.

CONCLUDING REMARKS

- Stable Gram Schmidt Process overcomes flaws of its prior versions, Non-normalized and Classical, which are subject to numerical instabilities, which accumulate round-off errors that corrupt computations, give inaccurate results and non-orthogonal vectors.
- Stable GS process is numerically robust algorithm used for more practical computations and computing QR factorization of mildly ill-conditioned matrices.
- Most computers use floating point arithmetic and the degree of arithmetic precision used by the computer hardware affects the accuracy of the numerical solution algorithm(s) in the software.
- Hence, the solution of a linear system whose coefficient matrix is an ill-conditioned Hilbert matrix H_n, even for moderately large n, is a very challenging problem, even using high precision computer arithmetic. Hence, Stable GS fails at higher n.
- Thus, Householder's Method is the method of choice for moderately ill-conditioned systems. But severe ill-conditioning will defeat even this ingenious approach, and accurately solving such systems can be an extreme challenge.

CONCLUDING REMARKS (CONTD.)

- Gauss-Jordan Method and in computer algebra systems such as Maple and Mathematica use exact rational arithmetic to perform computations resulting in greater accuracy. Thus, GE method has greater accuracy here than Stable GS.
- Inverse of H_n can be calculated as follows which can give more accurate results for solving linear systems than a computer using floating point arithmetic.

$$H_n^{-1} = (-1)^{i+j}(i + j - 1)\binom{n+i-1}{n-j}\binom{n+j-1}{n-i}\binom{i+j-2}{i-1}^2 \text{, where } \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

- Partial Pivoting Strategy can be used, whereby the largest (in absolute value) legitimate (i.e., in the pivot column on or below the diagonal) element is selected as the pivot, even if the diagonal element is nonzero, to suppress the undesirable effects of round-off errors during the computation.

THANK YOU!!