IIIT Vadodara
WINTER 2021-2022
MA202 Numerical Techniques
LAB#2 Computational Errors & Linear Equations

Computing Error, Singularity, and Ill-Condition:

$\|A\|$ is the 2-norm of a matrix A. The MATLAB command norm(A,2) returns the 2-norm of matrix A. The MATLAB command cond() tells us the degree of ill-condition for a given matrix by the size of the condition number, which is defined as $cond(A) = \|A\|\|A\|^{-1}$, where $\|A\|$ is the largest eigenvalue of AA^T i.e., the largest singular value of A. The command cond() returns the 2-norm condition number, which is the ratio of the largest singular value of A to the smallest.

Q1. Write a program docondition() to make use of MATLAB commands cond() and det() to compute the condition number and det(A)det(A^{-1}) where A is the Hilbert matrix defined by:

$$A \equiv [a_{mn}] = \frac{1}{m+n-1}.$$

Increase the dimension of the Hilbert matrix from N = 7 to 12 and see the degree of discrepancy between AA^{-1} and the identity matrix. Note: The number RCOND following the warning message about near-singularity or ill-condition given by MATLAB is a reciprocal condition number, which can be computed by the rcond() command and is supposed to get close to 1/0 for a well/badly conditioned matrix.

Various Kinds of Computing Errors:

There are various kinds of errors that we encounter when using a computer for computation.

- Truncation Error: Caused by adding up to a finite number of terms, while we should add infinitely many terms to get the exact answer in theory.
- Round-off Error: Caused by representing/storing numeric data in finite bits.
- Overflow/Underflow: Caused by too large or too small numbers to be represented/stored properly in finite bits, more specifically, the numbers having absolute values larger/smaller than the maximum (fmax)/minimum(fmin) number that can be represented in MATLAB.
- Negligible Addition: Caused by adding two numbers of magnitudes differing by over 52 bits. Loss of Significance: Caused by a “bad subtraction,” which means a subtraction of a number from another one that is almost equal in value.
- Loss of Significance: It is caused by a bad subtraction, which means a subtraction of a number from another one that is almost equal in value.

- Error Magnification: Caused and magnified/propagated by multiplying/dividing a number containing a small error by a large/small number.
- Other Errors depending on the numerical algorithms, step size, and so on.

Q2. Write a MATLAB program to evaluate the expressions $p(x) = (x-1)^5$ and $q(x) = x^5 - 5x^4 + 10x^3 - 10x^2 + 5x - 1$ for values of $x = \{0, 0.25, 0.5, 0.75, 0.90, 0.95, 0.99\}$. Plot the two expressions and comment upon the results.

Q3. Write a MATLAB program to compute the below two mathematically equivalent expressions for values $x = 1, 10, 100, 10000, 100000$. Also plot them, and based on the plots, find out which is better in terms of resisting the loss of significance.

$$\text{A) } \sqrt{2x^2+1} - 1,$$

$$\text{B) } \frac{2x^2}{\sqrt{2x^2+1}+1}.$$

Avoiding Large Errors/Overflow/Underflow:

Q4. Consider the following two expressions:

$$f(x) = \sqrt{x}\left(\sqrt{x+1} - \sqrt{x}\right),$$

$$g(x) = \frac{\sqrt{x}}{\sqrt{x+1}+\sqrt{x}}.$$

Write a MATLAB program to compute the values of the two expressions when $x = 1, 10, 100, 10000, 100000$. Observe what happens as x increases and what kind of computing error is happening upon execution.

Q5. Write a MATLAB program to compute $\dfrac{y^n}{e^{nx}}$ and $\left(\dfrac{y}{e^x}\right)^n$ for values of $x = 36$ and $y = 1e16$ for values of $n = -20, -19, 19, 20$. Which of these expressions is the right one to avoid overflow/underflow ?

Q6. For $x = 9.8^{201}$ and $y = 10.2^{199}$, evaluate the following two expressions that are mathematically equivalent and tell which is better in terms of the power of resisting the overflow:

$$z = \sqrt{x^2 + y^2},$$

$$z = y\sqrt{\left(\frac{x}{y}\right)^2 + 1}.$$

Also evaluate the following two expressions for $x = 9.8^{-201}$ and $y = 10.2^{-199}$, and comment upon the which is better in terms of nature of overflow/underflow.

System of linear equations:

There are several numerical schemes for solving a system of linear equations of the type: $Ax = b$, where A is a $M \times N$ matrix, matrix x is $n \times 1$, and matrix b is $m \times 1$. We will deal with the three cases:

(i) The case where the number (M) of equations and the number (N) of unknowns are equal (M = N) so that the coefficient matrix A is square.

(ii) The case where the number (M) of equations is smaller than the number(N) of unknowns (M < N) so that we might have to find the minimum-norm solution among the numerous solutions.

(iii) The case where the number of equations is greater than the number of unknowns (M > N) so that there might exist no exact solution and we must find a solution based on global error minimization, like the "least-squares error (LSE) solution."

Q7. Write a MATLAB routine lineqsol() to solve a given set of linear equations, covering all of the above three cases.

Q8. Solve the linear equations using command lineqsol() to solve the given set of linear equations, covering all of three cases depicted below:

1. $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ and $b = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$,

2. $A = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$ and $b = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$,

3. $A = (1 \quad 2)$ and $b = 3$,

4. $A = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $b = \begin{pmatrix} 2.1 \\ 3.9 \end{pmatrix}$.