Module 5 : Solving Nonlinear Algebraic Equations

Section 1 : Introduction

1 Introduction

Consider set of n nonlinear simultaneous equations of type

$$f_i(\mathbf{x}) = 0 \quad for\ i = 1, 2, \ldots, n \qquad \text{-------(1)}$$

$$\mathbf{F}(\mathbf{x}) = \overline{\mathbf{0}} \qquad \text{-------(2)}$$

where $\mathbf{x} \in R^n$ and $\mathbf{F}(.) : R^n \rightarrow R^n$ represents a $n \times 1$ function vector. This problem may have no solution, an infinite number of solutions or any finite number of solutions. In the module on Problem Discretization using Approximation Theory, we have already introduced a basic version of the Newton's method, in which a sequence of approximate linear transformations is constructed to solve equation (Fx). In this module, we develop this method further and also discuss the conditions under which it converges to the solution. In addition, we discuss the following two approaches that are frequently used for solving nonlinear algebraic equations: (a) method of successive substitutions and (b) unconstrained optimization. Towards the end of the module, we briefly touch upon two fundamental issues related to nonlinear algebraic equations, namely (a) the (local) existence uniqueness of the solutions and (b) the notion of conditioning of nonlinear algebraic equations.

Module 4 : Solving Linear Algebraic Equations

Section 7 : Matrix Conditioning and Behavior of Solutions

7 Matrix Conditioning and Behavior of Solutions

One of the important issue in computing solutions of large dimensional linear system of equations is the round-off errors caused by the computer. Some matrices are *well conditioned* and the computations proceed smoothly while some are inherently *ill conditioned,* which imposes limitations on how accurately the system of equations can be solved using any computer or solution technique. We now introduce measures for assessing whether a given system of linear algebraic equations is inherently *ill conditioned* or *well conditioned*.

Normally any computer keeps a fixed number of significant digits. For example, consider a computer that keeps only first three significant digits. Then, adding

$$0.234 + 0.00231 \rightarrow 0.236$$

results in loss of smaller digits in the smaller number. When a computer can commits millions of such errors in a complex computation, the question is, how do these individual errors contribute to the final error in computing the solution? Suppose we solve for $\mathbf{Ax} = \mathbf{b}$ using LU decomposition, the elimination algorithm actually produce approximate factors $\mathbf{L}'$ and $\mathbf{U}'$. Thus, we end up solving the problem with a wrong matrix, i.e.

$$\mathbf{A} + \delta\mathbf{A} = \mathbf{L}'\mathbf{U}' \quad \text{--------(141)}$$

instead of right matrix $\mathbf{A} = \mathbf{LU}$. In fact, due to round off errors inherent in any computation using computer, we actually end up solving the equation

$$(\mathbf{A} + \delta\mathbf{A})(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b} \quad \text{--------(142)}$$

The question is, how serious are the errors $\delta\mathbf{x}$ in solution $\mathbf{x}$, due to round off errors in matrix $\mathbf{A}$ and vector $\mathbf{b}$? Can these errors be avoided by rearranging computations or are the computations inherent ill-conditioned? In order to answer these questions, we need to develop some quantitative measure for **matrix conditioning**.

The following section provides motivation for developing a quantitative measure for matrix conditioning. In order to develop such a index, we need to define the concept of norm of a $m \times n$ matrix. The formal definition of matrix condition number and methods for computing it are presented in the later sub-sections.

7.1 Motivation [3]

In many situations, if the system of equations under consideration is **numerically well conditioned,** then it is possible to deal with the menace of round off errors by re-arranging the computations. If the system of equations is inherently an **ill conditioned** system, then the rearrangement trick does not help. Let us try and understand this by considering to simple examples and a computer that keeps only three significant digits.

Consider the system (**System-1**)

$$\begin{bmatrix} 0.0001 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad \text{--------(143)}$$

If we proceed with Gaussian elimination without maximal pivoting , then the first elimination step yields

$$\begin{bmatrix} 0.0001 & 1 \\ 0 & -9999 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -9998 \end{bmatrix} \quad \text{--------(144)}$$

and with back substitution this results in

$$x_2 = 0.999899 \quad \text{--------(145)}$$

which will be rounded off to

$$x_2 = 1 \quad \text{--------(146)}$$

in our computer which keeps only three significant digits. The solution then becomes

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T = \begin{bmatrix} 0.0 & 1 \end{bmatrix} \quad \text{--------(147)}$$

However, using maximal pivoting strategy the equations can be rearranged as

$$\begin{bmatrix} 1 & 1 \\ 0.0001 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \text{--------(148)}$$

and the Gaussian elimination yields

$$\begin{bmatrix} 1 & 1 \\ 0 & 0.9999 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0.9998 \end{bmatrix} \quad \text{--------(149)}$$

and again due to three digit round off in our computer, the solution becomes

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

Thus, when A is a well conditioned numerically and Gaussian elimination is employed, the main reason for blunders in calculations is wrong pivoting strategy. If maximum pivoting is used then natural resistance of the system of equations to *round-off errors* is no longer compromised.

Now, to understand difficulties associated with **ill conditioned** systems, consider another system (**System-2**)

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad \text{--------(150)}$$

By Gaussian elimination

$$\begin{bmatrix} 1 & 1 \\ 0 & 0.0001 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad \text{--------(151)}$$

If we change R.H.S. of the system 2 by a small amount

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix} \quad \text{--------(152)}$$

$$\begin{bmatrix} 1 & 1 \\ 0 & 0.0001 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0.0001 \end{bmatrix} \Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{--------(153)}$$

Note that change in the fifth digit of second element of vector $\mathbf{b}$ was amplified to change in the first digit of the solution. Here is another example of an illconditioned matrix [Gour]. Consider the following system

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix} \quad \text{--------(154)}$$

whose exact solution is $\mathbf{x} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$. Now, consider a slightly perturbed system

$$\left[\mathbf{A} + \begin{bmatrix} 0 & 0 & 0.1 & 0.2 \\ 0.08 & 0.04 & 0 & 0 \\ 0 & -0.02 & -0.11 & 0 \\ -0.01 & -0.01 & 0 & -0.02 \end{bmatrix} \right] \mathbf{x} = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix} \quad \text{--------(155)}$$

This slight perturbation in $\mathbf{A}$ matrix changes the solution to

$$\mathbf{x} = \begin{bmatrix} -81 & 137 & -34 & 22 \end{bmatrix}^T$$

Alternatively, if vector $\mathbf{b}$ on the R.H.S. is changed to

$$b = \begin{bmatrix} 31.99 & 23.01 & 32.99 & 31.02 \end{bmatrix}^T$$

then the solution changes to

$$\mathbf{x} = \begin{bmatrix} 0.12 & 2.46 & 0.62 & 1.23 \end{bmatrix}^T$$

Thus, matrices A in System 2 and in equation (A4) are **ill conditioned**. Hence, no numerical method can avoid sensitivity of these systems of equations to small permutations, which can result even from truncation errors. The ill conditioning can be shifted from one place to another but it cannot be eliminated.

7.2 Condition Number [3]

Condition number of a matrix is a measure to quantify matrix Ill-conditioning. Consider system of equations given as $\mathbf{A}\mathbf{x} = \mathbf{b}$. We examine two situations: (a) errors in representation of vector $\mathbf{b}$ and (b) errors in representation of matrix $\mathbf{A}$.

7.2.1 Case: Perturbations in vector b [3]

Consider the case when there is a change in $\mathbf{b}$, i.e., $\mathbf{b}$ changes to $\mathbf{b} + \delta\mathbf{b}$ in the process of numerical computations. Such an error may arise from experimental errors or from round off errors. This perturbation causes a change in solution from $\mathbf{x}$ to $\mathbf{x} + \delta\mathbf{x}$, i.e.

$$\mathbf{A}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b} \quad \text{--------(156)}$$

By subtracting $\mathbf{A}\mathbf{x} = \mathbf{b}$ from the above equation we have

$$\mathbf{A}\delta\mathbf{x} = \delta\mathbf{b} \quad \text{--------(157)}$$

To develop a measure for conditioning of matrix $\mathbf{A}$, we compare relative change/error in solution,i.e. $\|\delta\mathbf{x}\|/\|\mathbf{x}\|$ to relative change in $\mathbf{b}$,i.e. $\|\delta\mathbf{b}\|/\|\mathbf{b}\|$. To derive this relationship, we consider the following two inequalities

$$\delta\mathbf{x} = \mathbf{A}^{-1}\delta\mathbf{b} \Rightarrow \|\delta\mathbf{x}\| \leq \|\mathbf{A}^{-1}\| \, \|\delta\mathbf{b}\|$$

--------(158)

$$\mathbf{Ax} = \mathbf{b} \Rightarrow \|\mathbf{b}\| = \|\mathbf{Ax}\| \leq \|\mathbf{A}\| \, \|\mathbf{x}\| \quad \text{--------(159)}$$

which follow from the definition of induced matrix norm. Combining these inequalities, we can write

$$\|\delta\mathbf{x}\| \, \|\mathbf{b}\| \leq \|\mathbf{A}^{-1}\| \, \|\mathbf{A}\| \, \|\mathbf{x}\| \, \|\delta\mathbf{b}\| \quad \text{--------(160)}$$

$$\Rightarrow \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq (\|\mathbf{A}^{-1}\| \, \|\mathbf{A}\|) \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} \quad \text{--------(161)}$$

$$\Rightarrow \frac{\|\delta\mathbf{x}\|/\|\mathbf{x}\|}{\|\delta\mathbf{b}\|/\|\mathbf{b}\|} \leq \|\mathbf{A}^{-1}\| \, \|\mathbf{A}\| \quad \text{--------(162)}$$

It may be noted that the above inequality holds for any vectors $\mathbf{b}$ and $\delta\mathbf{b}$. The number

$$c(\mathbf{A}) = \|\mathbf{A}^{-1}\| \, \|\mathbf{A}\| \quad \text{--------(163)}$$

is called as **condition number** of matrix $\mathbf{A}$. The condition number gives an upper bound on the possible amplification of errors in $\mathbf{b}$ while computing the solution Strang.

7.2.2 Case: Perturbation in matrix A[3]

Suppose ,instead of solving for $\mathbf{Ax} = \mathbf{b}$, due to truncation errors, we end up solving

$$(\mathbf{A} + \delta\mathbf{A})(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} \quad \text{--------(164)}$$

Then, by subtracting $\mathbf{Ax} = \mathbf{b}$ from the above equation we obtain

$$\mathbf{A}\delta\mathbf{x} + \delta\mathbf{A}(\mathbf{x} + \delta\mathbf{x}) = \overline{\mathbf{0}} \quad \text{--------(165)}$$

$$\Rightarrow \delta\mathbf{x} = -\mathbf{A}^{-1}\delta\mathbf{A}(\mathbf{x} + \delta\mathbf{x}) \quad \text{--------(166)}$$

Taking norm on both the sides, we have

$$\|\delta\mathbf{x}\| = \|\mathbf{A}^{-1}\delta\mathbf{A}(\mathbf{x} + \delta\mathbf{x})\| \quad \text{--------(167)}$$

$$\|\delta\mathbf{x}\| \leq \|\mathbf{A}^{-1}\| \, \|\delta\mathbf{A}\| \, \|\mathbf{x} + \delta\mathbf{x}| \quad \text{--------(168)}$$

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x} + \delta\mathbf{x}\|} \leq (\|\mathbf{A}^{-1}\| \, \|\mathbf{A}\|) \frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|} \quad \text{--------(169)}$$

$$\frac{\|\delta\mathbf{x}\|/\|\mathbf{x} + \delta\mathbf{x}\|}{\|\delta\mathbf{A}\|/\|\mathbf{A}\|} \leq c(\mathbf{A}) = \|\mathbf{A}^{-1}\| \, \|\mathbf{A}\| \quad \text{--------(170)}$$

Again,the condition number gives an upper bound on % change in solution to % error A.

In simple terms, the condition number of a matrix tells us how serious is the error in solution of $\mathbf{Ax} = \mathbf{b}$ due to the truncation or round off errors in a computer. These inequalities mean that round off error comes from two sources

- Inherent or natural sensitivity of the problem,which is measured by $c(\mathbf{A})$
- Actual errors $\delta\mathbf{b}$ *or* $\delta\mathbf{A}$.

It has been shown that the maximum pivoting strategy is adequate to keep ($\delta\mathbf{A}$) in control so that the whole burden of round off errors is carried by the condition number $c(\mathbf{A})$. If condition number is high (>1000), the system is ill conditioned and is more sensitive to round off errors. If condition

number is low (<100) system is well conditioned and you should check your algorithm for possible source of errors.

7.2.3 Computations of condition number

Let λ_n denote the largest magnitude eigenvalue of matrix $\mathbf{A}$ and λ_1 denote the smallest magnitude eigen value of $\mathbf{A}$. Then, we know that

$$\|\mathbf{A}\|_2^2 = \rho(\mathbf{A}^T\mathbf{A}) = \lambda_n \quad \text{--------(171)}$$

Also,

$$\|\mathbf{A}^{-1}\|_2^2 = \rho[(\mathbf{A}^{-1})^T\mathbf{A}^{-1}] = \rho\left[(\mathbf{A}\mathbf{A}^T)^{-1}\right] \quad \text{--------(172)}$$

This follows from identity

$$(\mathbf{A}^{-1}\mathbf{A})^T = \mathbf{I}$$
$$\mathbf{A}^T(\mathbf{A}^{-1})^T = \mathbf{I}$$
$$(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T \quad \text{--------(173)}$$

Now, if λ is eigenvalue of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{v}$ is the corresponding eigenvector, then

$$(\mathbf{A}^T\mathbf{A})\mathbf{v} = \lambda\mathbf{v} \quad \text{--------(174)}$$
$$\mathbf{A}\mathbf{A}^T(\mathbf{A}\mathbf{v}) = \lambda(\mathbf{A}\mathbf{v}) \quad \text{--------(175)}$$

λ is also eigenvalue of $\mathbf{A}\mathbf{A}^T$ and $(\mathbf{A}\mathbf{v})$ is the corresponding eigenvector. Thus, we can write

$$\|\mathbf{A}^{-1}\|_2^2 = \rho\left[(\mathbf{A}\mathbf{A}^T)^{-1}\right] = \rho\left[(\mathbf{A}^T\mathbf{A})^{-1}\right] \quad \text{--------(176)}$$

Also, since $\mathbf{A}\mathbf{A}^T$ is a symmetric positive definite matrix, we can diagonalize it as

$$\mathbf{A}^T\mathbf{A} = \Psi\Lambda\Psi^T \quad \text{--------(177)}$$

$$\Rightarrow (\mathbf{A}^T\mathbf{A})^{-1} = [\Psi\Lambda\Psi^T]^{-1} = (\Psi^T)^{-1}\left[\Lambda^{-1}\right]\Psi^{-1} = \Psi\Lambda^{-1}\Psi^T$$

as Ψ is a unitary matrix. Thus, if λ is eigen value of $\mathbf{A}^T\mathbf{A}$ then $1/\lambda$ is eigen value of $(\mathbf{A}^T\mathbf{A})^{-1}$. If λ_1 smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$ then $1/\lambda_1$ is largest magnitude eigenvalue of $\mathbf{A}^T\mathbf{A}$

$$\Rightarrow \rho[(\mathbf{A}^T\mathbf{A})^{-1}] = 1/\lambda_1$$

Thus, the condition number of matrix $\mathbf{A}$ can be computed using 2-norm as

$$c_2(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2 = (\lambda_n/\lambda_1)^{1/2}$$

where λ_n and λ_1 are largest and smallest magnitude eigenvalues of $\mathbf{A}^T\mathbf{A}$

The condition number can also be estimated using any other norm. For example, if we use $\infty - norm$, then

$$c_\infty(\mathbf{A}) = \|\mathbf{A}\|_\infty \|\mathbf{A}^{-1}\|_\infty$$

Estimation of condition number by this approach, however, requires computation of $\mathbf{A}^{-1}$, which can be unreliable if $\mathbf{A}$ is ill conditioned.

Example 10 TaylorPhilllipsConsider the Hilbert matrix discussed in the module Problem Discretization using Approximation Theory. These matrices, which arise in simple polynomial approximation are notoriously ill conditioned and $c(\mathbf{H}_n) \to \infty$ as $n \to \infty$. For example, consider

$$\mathbf{H}_3 = \begin{bmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{bmatrix} ; \quad \mathbf{H}_3^{-1} = \begin{bmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{bmatrix}$$

$$\|\mathbf{H}_3\|_1 = \|\mathbf{H}_3\|_\infty = 11/6 \quad and \quad \left\|\mathbf{H}_3^{-1}\right\|_1 = \left\|\mathbf{H}_3^{-1}\right\|_\infty = 408$$

Thus, condition number can be computed as $c_1(\mathbf{H}_3) = c_\infty(\mathbf{H}_3) = 748.$ For n = 6, $c_1(\mathbf{H}_3)$ $= c_\infty(\mathbf{H}_3) = 29 \times 10^6,$ which is extremely bad.

Even for n = 3, the effects of rounding off can be quite serious. For, example, the solution of

$$\mathbf{H}_3\mathbf{x} = \begin{bmatrix} 11/6 \\ 13/12 \\ 47/60 \end{bmatrix}$$

is $\mathbf{x} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T.$ If we round off the elements of $\mathbf{H}_3$ to three significant decimal digits, we obtain

$$\begin{bmatrix} 1 & 0.5 & 0.333 \\ 0.5 & 0.333 & 0.25 \\ 0.333 & 0.25 & 0.2 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1.83 \\ 1.08 \\ 0.783 \end{bmatrix}$$

then the solution changes to $\mathbf{x} + \delta\mathbf{x} = \begin{bmatrix} 1.09 & 0.488 & 1.491 \end{bmatrix}^T.$ The relative perturbation in

elements of matrix $\mathbf{H}_3$ does not exceed 0.3%. However, the solution changes by 50%! The main indicator of ill-conditioning is that the magnitudes of the pivots become very small when Gaussian elimination is used to solve the problem.

Example 11Consider matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

This matrix is near singular with eigen values (computed using *Scilab*)

$$\lambda_1 = 16.117 \; ; \lambda_2 = -1.1168 \; ; \; \lambda_3 = -1.3 \times 10^{-15}$$

has the condition number of $c_2(\mathbf{A}) = 3.8131 \times 10^{16}.$ If we attempt to compute inverse of this matrix using Scilab, we get following result

$$\mathbf{A}^{-1} = 10^{16} \times \begin{bmatrix} -0.4504 & 0.9007 & -0.4504 \\ 0.9007 & -1.8014 & 0.9007 \\ -0.4504 & 0.9007 & -0.4504 \end{bmatrix}$$

with a warning: '*Matrix is close to singular or badly scaled.*' The difficulties in computing inverse of this matrix are apparent if we further compute product $\mathbf{A} \times \mathbf{A}^{-1},$ which yields

$$\mathbf{A} \times \mathbf{A}^{-1} = \begin{bmatrix} 2 & 0 & 2 \\ 8 & 0 & 0 \\ 16 & 0 & 8 \end{bmatrix}$$

On the other hand, consider matrix

$$\mathbf{B} = 10^{-17} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 1 & 3 \end{bmatrix}$$

with eigen values

$$\lambda_1 = 4.73 \times 10^{-17}; \lambda_2 = -1 \times 10^{-17}; \lambda_3 = 1.26 \times 10^{-17}$$

The eigenvalues are 'close to zero' the matrix is almost like a null matrix. However, the condition number of this matrix is $c_2(\mathbf{B}) = 5.474.$ If we proceed to compute of $\mathbf{B}^{-1}$ using Scilab, we get

$$\mathbf{B}^{-1} = 10^{16} \times \begin{bmatrix} -1.67 & 8.33 & -5 \\ 6.67 & -3.33 & 0 \\ -1.67 & -1.67 & 5 \end{bmatrix}$$

and $\mathbf{B} \times \mathbf{B}^{-1}$ yields $\mathbf{I}$, i.e. identity matrix.

Thus, it is important to realize that each system of linear equations has a inherent character, which can be quantified using the condition number of the associated matrix. The best of the linear equation solvers cannot overcome the computational difficulties posed by inherent ill conditioning of a matrix. As a consequence, when such ill conditioned matrices are encountered, the results obtained using any computer or any solver are unreliable.