Chapter 8: Numerical Methods for Linear Algebra

Prof. Alok Shukla

Department of Physics
IIT Bombay, Powai, Mumbai 400076

Course Name: Introduction to Numerical Analysis (PH 307)

- In this chapter we we will discuss numerical techniques broadly aimed at solving the following two problems:
  - Solving a system of linear equations
  - Obtaining the eigenvalues and eigenvectors of matrices. This is also called diagonalization of matrices.
- We will start our discussion by reviewing methods for solving a linear system of equtations.

System of Linear Equations

- Let us consider a system of n linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots &= \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \tag{1}$$

- which can be expressed compactly in the matrix form

$$AX = b, \tag{2}$$

where

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ldots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}; \quad X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}; \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \tag{3}$$

- Our aim is to develop numerical methods for solving such equations.

Gaussian Elimination method

- First we discuss Gaussian elimination method, which is also the method that we humans use to solve simultaneous linear equations
- The approach is based on the concept of successive elimination of variables until the solution is found
- Let us consider the following example to demonstrate this approach

$$
\begin{aligned}
3x_1 - x_2 + x_3 + 2x_4 &= 8 \\
6x_1 - 4x_2 + 3x_3 + 5x_4 &= 13 \\
3x_1 - 13x_2 + 9x_3 + 3x_4 &= -19 \\
-6x_1 + 4x_2 + x_3 - 18x_4 &= -34
\end{aligned} \tag{4}
$$

- Let us first make the coefficient of x_1 1 in the first equation by dividing it by 3, leading to

$$
\begin{aligned}
x_1 - \frac{1}{3}x_2 + \frac{1}{3}x_3 + \frac{2}{3}x_4 &= \frac{8}{3} \\
6x_1 - 4x_2 + 3x_3 + 5x_4 &= 13 \\
3x_1 - 13x_2 + 9x_3 + 3x_4 &= -19 \\
-6x_1 + 4x_2 + x_3 - 18x_4 &= -34
\end{aligned}
$$

Gaussian Elimination Method

- Next we eliminate x_1 from the lower three equations $(2 \leq j \leq 4)$ by subtracting $a_{j1}*$first-equation from remaining three, leading to

$$\begin{aligned}
x_1 - \tfrac{1}{3}x_2 + \tfrac{1}{3}x_3 + \tfrac{2}{3}x_4 &= \tfrac{8}{3} \\
-2x_2 + x_3 + x_4 &= -3 \\
-12x_2 + 8x_3 + x_4 &= -27 \\
2x_2 + 3x_3 - 14x_4 &= -18
\end{aligned}$$

- Next, we eliminate x_2 from lower two equations by first dividing the second equation by -2, and then by subtracting $a_{j2}*$second-equation $(j = 3, 4)$ from the lower two equations leading to

$$\begin{aligned}
x_1 - \tfrac{1}{3}x_2 + \tfrac{1}{3}x_3 + \tfrac{2}{3}x_4 &= \tfrac{8}{3} \\
x_2 - \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4 &= \tfrac{3}{2} \\
2x_3 - 5x_4 &= -9 \\
4x_3 - 13x_4 &= -21
\end{aligned}$$

- First we divide Eq. 3 by two and subtract 4*third-equation from the fourth one to obtain

$$\begin{aligned} x_1 - \tfrac{1}{3}x_2 + \tfrac{1}{3}x_3 + \tfrac{2}{3}x_4 &= \tfrac{8}{3} \\ x_2 - \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4 &= \tfrac{3}{2} \\ x_3 - \tfrac{5}{2}x_4 &= -\tfrac{9}{2} \\ -3x_4 &= -3 \end{aligned}$$

- Now we divide the last equation by -3 to finally obtain

$$\begin{aligned} x_1 - \tfrac{1}{3}x_2 + \tfrac{1}{3}x_3 + \tfrac{2}{3}x_4 &= \tfrac{8}{3} \\ x_2 - \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4 &= \tfrac{3}{2} \\ x_3 - \tfrac{5}{2}x_4 &= -\tfrac{9}{2} \\ x_4 &= 1 \end{aligned} \tag{5}$$

- This equations can be expressed as the following matrix equation

$$UX = b' \tag{6}$$

- where U denotes an "upper triangular" in which only the diagonal and elements above it are non-zero, while those placed lower than the diagonal are zero
- In this case

$$U = \begin{pmatrix} 1 & -\frac{1}{3} & \frac{1}{3} & \frac{2}{3} \\ 0 & 1 & -\frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 & -\frac{5}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}; \qquad b' = \begin{pmatrix} \frac{8}{3} \\ \frac{3}{2} \\ -\frac{9}{2} \\ 1 \end{pmatrix} \tag{7}$$

- In other words, by performing a series of elementary row operations, we have managed to reduce the original coefficient matrix A to an upper-triangular matrix U, and the constants vector b to b'.

Gaussian elimination...

- The final step in solving the equations, i.e., obtaining the values of x_i, is back substitution
- In this we start with the lowest equation, i.e., the 4th one in this case, to obtain the value of x_4

$$\boxed{x_4 = 1}$$

- We substitute this in the third equation

$$x_3 - \frac{5}{2} \times 1 = -\frac{9}{2}$$
$$\implies \boxed{x_3 = -2}$$

- Next we substitute the values of x_3 and x_4 in the second equation

$$x_2 - \frac{1}{2} \times (-2) - \frac{1}{2} \times 1 = \frac{3}{2}$$
$$\implies \boxed{x_2 = 1}$$

- In the final step, we substitute the values of x_2, x_3, and x_4 in the first equation to obtain the value of x_1, completing the solution

$$x_1 - \frac{1}{3} \times 1 + \frac{1}{3} \times (-2) + \frac{2}{3} \times 1 = \frac{8}{3}$$
$$\implies \boxed{x_1 = 3}$$

- Thus, in the matrix language, the process of Gaussian elimination consists of reduction the coefficient matrix to an upper–triangular form, followed by the process of back substitution

- One can verify that for a general $n \times n$ coefficient matrix A, the reduction to U form is achieved by

$$a_{ij} \leftarrow a_{ij} - \left(\frac{a_{ik}}{a_{kk}}\right) a_{kj}$$
$$b_i \leftarrow b_i - \left(\frac{a_{ik}}{a_{kk}}\right) b_k$$

where $1 \le k \le n-1$, $k+1 \le i \le n$, and $k+1 \le j \le n$

- The back substitution step is implemented as

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^{n} a_{ij} x_j \right) \qquad \text{for } i = n, n-1, n-2, \ldots, 1$$

Pseudo-code for Gaussian Elimination Method

```
Real array: a(n:n), b(n),x(n)
Integer: i, j, k, n
real: sum,xmult
#Code begins
read n and allocate arrays a, b, and x
Initialize/read the values of a(i,j)
initialize/read the values of b(i)
for k=1 to n-1
for i=k+1 to n
xmult=a(i,k)/a(k,k)
a(i,k)=xmult
for j = k+1 to n
a(i,j)=a(i,j)-xmult*a(k,j)
end for loop of j
b(i)=b(i)-xmult*b(k)
end for loop of i
end for loop of k
```

Gaussian elimination code....

```
#Gaussian elimination complete, now start back substitution
x(n)=b(n)/a(n,n)
#Now run the for loop in the reverse order
for i=n-1 to 1
sum=b(i)
for j=i+1 to n
sum=sum-a(i,j)*x(j)
end for loop of j
x(i)=sum/a(i,i)
end for loop of i
# solution is complete, array x contains those, which one can print
print (x(i),i=1,n)
end program
```

Instabilities of the Gaussian Elimination Method

- Let us consider the following 2×2 system of linear equations

$$\begin{aligned} 0x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2 \end{aligned} \tag{8}$$

- If we apply the Gaussian elimination method (GEM) naively to this problem, we will have to divide the first equation by 0, leading to a failure of the procedure in the first step itself
- However, if we flip the equations

$$\begin{aligned} x_1 + x_2 &= 2 \\ 0x_1 + x_2 &= 1 \end{aligned} \tag{9}$$

we immediately obtain the solution because the coefficient matrix is already in the UT form.

Instabilities of GEM...

- Thus GEM is unstable whenever the coefficients are small, however, the problem in most cases can be solved interchanging equations, which is called pivoting
- For example, in Eq. 8, suppose the coefficient of x_1 in the first equation is a small number ε instead of zero

$$\begin{aligned} \varepsilon x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2 \end{aligned} \tag{10}$$

- If we apply GEM to this system, we will obtain

$$\begin{aligned} x_2 &= \frac{2 - \frac{1}{\varepsilon}}{1 - \frac{1}{\varepsilon}} \\ x_1 &= \frac{1 - x_2}{\varepsilon} \end{aligned}$$

GEM instabilities...

- In the limit $\varepsilon \to 0$, we obtain $x_2 \approx 1$ and $x_1 \approx 0$
- The correct solutions are

$$
\begin{aligned}
x_1 &= \frac{1}{1-\varepsilon} \approx 1 \\
x_2 &= \frac{1-2\varepsilon}{1-\varepsilon} \approx 1
\end{aligned}
$$

  which are quite different from the numerical solutions
- It is easy to verify that, as before, if we swap the two equations, the instability goes away, and we obtain the correct solution by using GEM.
- Let us demonstrate the pivoting approach for a larger set of equations

$$
\begin{aligned}
x_1 + 2x_2 + 3x_3 + x_4 &= -2 \\
3x_1 + x_2 + 0x_3 + 2x_4 &= 1 \\
x_1 - x_2 - x_3 - x_4 &= 5 \\
2x_1 + 0x_2 + 2x_3 + 3x_4 &= 3
\end{aligned}
\tag{11}
$$

- First we make an $n = 4$ dimensional row vector S consisting of the element of maximum magnitude from each each row

$$S_i = \max\{|a_{ij}|\},\ j = 1, \ldots n$$

- With this, we obtain

$$S = (3, 3, 1, 3)$$

- Next, we create a row vector r obtained by dividing the magnitude of the first element of i-th row by S_i

$$r_i = \frac{|a_{i1}|}{S_i}$$

- For this case

$$r = \left(\frac{1}{3}, 1, 1, \frac{2}{3}\right)$$

GEM with pivoting...

- If we define r_{max} as the maximum element of vector r, we swap row 1, with the first row corresponding to r_{max}
- In this case, we interchange rows 1 and 2 to obtain

$$\begin{aligned} 3x_1 + x_2 + 0x_3 + 2x_4 &= 1 \\ x_1 + 2x_2 + 3x_3 + x_4 &= -2 \\ x_1 - x_2 - x_3 - x_4 &= 5 \\ 2x_1 + 0x_2 + 2x_3 + 3x_4 &= 3 \end{aligned} \tag{12}$$

- Next, we perform the following row operations

$$row(2) = row(2) - \frac{1}{3} row(1)$$
$$row(3) = row(3) - \frac{1}{3} row(1)$$
$$row(4) = row(4) - \frac{2}{3} row(1)$$

- And obtain

$$
\begin{aligned}
3x_1 + x_2 + 0x_3 + 2x_4 &= 1 \\
\frac{5}{3}x_2 + 3x_3 + \frac{1}{3}x_4 &= -\frac{7}{3} \\
-\frac{4}{3}x_2 - x_3 - \frac{5}{3}x_4 &= \frac{14}{3} \\
-\frac{2}{3}x_2 + 2x_3 + \frac{5}{3}x_4 &= \frac{7}{3}
\end{aligned} \qquad (13)
$$

- Now, for the lower 3 rows, $S = (3, 5/3, 2)$, and $r = (5/9, 4/5, 1/3)$, which implies we must interchange rows 2 and 3

$$
\begin{aligned}
3x_1 + x_2 + 0x_3 + 2x_4 &= 1 \\
-\frac{4}{3}x_2 - x_3 - \frac{5}{3}x_4 &= \frac{14}{3} \\
\frac{5}{3}x_2 + 3x_3 + \frac{1}{3}x_4 &= -\frac{7}{3} \\
-\frac{2}{3}x_2 + 2x_3 + \frac{5}{3}x_4 &= \frac{7}{3}
\end{aligned} \qquad (14)
$$

- Now we need to perform the row operations $row(3) = row(3) + \frac{5}{4}row(2)$, and $row(4) = row(4) - \frac{1}{2}row(2)$

- With this we have

$$\begin{aligned} 3x_1 + x_2 + 0x_3 + 2x_4 \quad &= 1 \\ -\tfrac{4}{3}x_2 - x_3 - \tfrac{5}{3}x_4 \quad &= \tfrac{14}{3} \\ \tfrac{7}{4}x_3 - \tfrac{7}{4}x_4 \quad &= \tfrac{7}{2} \\ \tfrac{5}{2}x_3 + \tfrac{5}{2}x_4 \quad &= 0 \end{aligned} \tag{15}$$

- For the last two rows we have $S = \left(\frac{7}{4}, \frac{5}{2}\right)$ and $r = (1, 1)$ which means that no swapping is necessary
- Now only one row operation is needed on row 4, $row(4) = row(4) - \frac{10}{7}row(3)$, leading to

$$\begin{aligned} 3x_1 + x_2 + 0x_3 + 2x_4 \quad &= 1 \\ -\tfrac{4}{3}x_2 - x_3 - \tfrac{5}{3}x_4 \quad &= \tfrac{14}{3} \\ \tfrac{7}{4}x_3 - \tfrac{7}{4}x_4 \quad &= \tfrac{7}{2} \\ 5x_4 \quad &= -5 \end{aligned} \tag{16}$$

GEM with pivoting...

- Now that the coefficient matrix is in the UT form, the back substitution begins leading to the final solution

$$\boxed{x_4 = -1}$$

$$\boxed{x_3 = \frac{4}{7}\left(\frac{7}{2} - \frac{7}{4}\right) = 1} \implies \boxed{x_3 = 1}$$

$$-\frac{4}{3}x_2 - 1 + \frac{5}{3} = \frac{14}{3} \implies \boxed{x_2 = -3}$$

$$3x_1 - 3 - 2 = 1 \implies \boxed{x_1 = 2}$$

- So far we have discussed direct methods of solving a system of linear equations
$$AX = b,$$
where A an $n \times n$ matrix, as discussed earlier
- These methods are based on successive elimination of variables
- GEM approach belongs to this class of methods
- Other methods that belong to this class are GEM with pivoting and Gauss-Jordan elimination method
- These methods are quite efficient when n is not too large
- The computational time involved in these methods scales typically as n^3
- Therefore, when n is very large say $n \sim 10^5 - 10^9$, these methods will be come intractable

Iterative methods...

- There is a class of methods called iterative methods which don't have this problem
- In these methods, the main mathematical operation is the multiplication of an n-dimensional vector by an $n \times n$ matrix
- This operation scales as n^2, which reduces the computational effort by an order of magnitude as compared to the elimination methods.
- Furthermore, their computer implementation is also quite easy and straightforward
- We will discuss two methods in this class, namely Point-Jacobi and Gauss-Seidel methods
- Next, we briefly discuss the iterative methods, in general, followed by a discussion of the two above-mentioned methods

Iterative Methods: General Discussion

- Let us split the $n \times n$ coefficient matrix A as

$$A = I - B, \tag{17}$$

  where I is the $n \times n$ identity matrix, and $B = I - A$ is also an $n \times n$ matrix

- If we substitute Eq. 17 in our linear equations $AX = b$, we obtain

$$X = BX + b \tag{18}$$

- But, this equation cannot be treated as the solution, because the solution X appears on both sides of the equation
- However, it can be used in iterative schemes for obtaining X as follows

$$X_{i+1} = BX_i + b, \tag{19}$$

  where X_i and X_{i+1} are the solution vectors obtained after i-th and $i+1$-th iterations

Iterative Methods...

- Therefore, one can start with a guess solution (trial vector) for $i = 0$, X_0, and then continue with the iterative scheme of Eq. 19
- But, what is the guarantee that the iterative scheme will converge to the true solution in a reasonably small number of iterations?
- Actually, there is a theorem which states that the iterative scheme of Eq. 19 will converge to the true solution X, if and only if, the spectral radius of matrix B, $\rho(B) < 1$
- Spectral radius $\rho(B)$ of a matrix B is nothing but the maximum magnitude of its eigenvalues
- If the eigenvalues of B are $\{b_i, i = 1, \ldots, n\}$, $\rho(B) = \max\{|b_i|, i = 1, \ldots, n\}$
- Therefore, the success of the iterative approaches depends on finding a suitable matrix B which satisfies the condition $\rho(B) < 1$
- Next, we discuss two methods that are based on this

Point-Jacobi Approach

- In this approach we partition the coefficient matrix A is

$$A = D + LU, \tag{20}$$

where D is the diagonal matrix consisting of the diagonal elements of A

$$D = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & 0\cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0\cdots & 0 & a_{nn} \end{pmatrix} \tag{21}$$

and LU contains the off-diagonal elements of A

$$LU = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2}\cdots & a_{nn-1} & 0 \end{pmatrix} \tag{22}$$

Point-Jacobi Method...

- Substituting Eq. 20 in our linear equation $AX = b$, and rearranging

$$\begin{aligned} DX &= b - LUX \\ \implies X &= D^{-1}b - D^{-1}LUX \end{aligned} \tag{23}$$

- Using Eq. 23, we set up the iteration scheme

$$X_{i+1} = JX_i + D^{-1}b, \tag{24}$$

  where $J = -D^{-1}LU$, is called the Jacobi matrix
- And the iterative scheme defined by Eq. 24 is called the Point-Jacobi method.
- Note that this method involves the inverse of a diagonal matrix which can be trivially computed

- D^{-1} is also a diagonal matrix with non-zero elements a_{ii}^{-1}

$$D^{-1} = \begin{pmatrix} a_{11}^{-1} & 0 & \cdots & 0 \\ 0 & a_{22}^{-1} & 0\cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0\cdots & 0 & a_{nn}^{-1} \end{pmatrix} \tag{25}$$

- As a result

$$D^{-1}b = \begin{pmatrix} a_{11}^{-1}b_1 \\ a_{22}^{-1}b_2 \\ \vdots \\ a_{nn}^{-1}b_n \end{pmatrix} \tag{26}$$

and

$$J = -D^{-1}LU = \begin{pmatrix} 0 & -a_{11}^{-1}a_{12} & \cdots & -a_{11}^{-1}a_{1n} \\ -a_{22}^{-1}a_{21} & 0 & \cdots & -a_{22}^{-1}a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{nn}^{-1}a_{n1} & -a_{nn}^{-1}a_{n2} & \cdots & 0 \end{pmatrix} \tag{27}$$

Point-Jacobi Method: An Example

- I wrote a Fortran 90 code implementing the Point-Jacobi method, and tried it on a linear system taken from the Wikipedia

$$A = \begin{pmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{pmatrix}, \; b = \begin{pmatrix} 6 \\ 25 \\ -11 \\ 15 \end{pmatrix}$$

with the solution

$$X = \begin{pmatrix} 1 \\ 2 \\ -1 \\ 1 \end{pmatrix}$$

Point-Jacobi Method; example...

- I took the starting vector to be the null vector in all the trials

$$X_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- The convergence was assumed when the magnitude of the difference vector *eps* was less than a user specified tolerance *tol*

$$eps = \sqrt{\sum_{j=1}^{n} (X_{i+1}(j) - X_i(j))^2}$$

$$eps < tol$$

Point-Jacobi covergence...

- If *itr* is the total number of iterations needed to achieve convergence, for the given problem, the convergence was as follows

$$tol = 1.0 \times 10^{-4}, itr = 13, X = \begin{pmatrix} 0.9999897 \\ 2.0000158 \\ -1.0000125 \\ 1.0000192 \end{pmatrix}$$

$$tol = 1.0 \times 10^{-5}, itr = 16, X = \begin{pmatrix} 1.0000008 \\ 1.9999987 \\ -0.9999990 \\ 0.9999986 \end{pmatrix}$$

Gauss-Seidel Method

- In Gauss-Seidel approach, the partitioning of the coefficient matrix A is done slightly differently

$$AX = b$$
$$\implies (DL + U)X = b$$
$$\implies X = -DL^{-1}UX + DL^{-1}b$$

- The last equation above can be converted into a formal iteration scheme

$$X_{i+1} = -DL^{-1}UX_i + DL^{-1}b, \tag{28}$$

which is called the Gauss-Seidel iterative method.

- Above DL is the matrix containing the elements of the diagonal and lower triangle

$$DL = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0\cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2}\cdots & a_{nn-1} & a_{nn} \end{pmatrix} \tag{29}$$

- while U contains the elements of the upper triangle of A, while rest of its elements are zeros.

$$U = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \tag{30}$$

- Eq. 28 involves calculation of the inverse of a non-diagonal matrix, which will be time consuming for large n
- Therefore, the iterative scheme is reformulated as

$$DLX_{i+1} = -UX_i + b \tag{31}$$

- Equation above can be treated like a linear equation in the unknown X_{i+1}, whose value can be quickly obtained by the forward substitution because the DL matrix is a lower-triangular one
- Also note that everything on the RHS of this equation is known and easy to compute
- Thus we can completely avoid the calculation of the inverse

Convergence of Point-Jacobi and Gauss-Seidel Approaches

- As discussed earlier, these approaches will converge provided the matrix involved, B, in the iterative process has the spectral radius $\rho(B) < 1$
- One can show that it will always be the case provided the coefficient matrix A is diagonally dominant
- A matrix is said to be diagonally dominant provided the magnitude of the diagonal element of each row is larger than the sum of the magnitudes of the off-diagonal elemets of that row

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}| \qquad \text{for } i = 1, 2, \ldots, n \tag{32}$$

- Therefore, before using iterative methods to solve a linear system, one should check whether or not it is diagonally dominant using Eq. 32