Aim : To determine the eigenvalue and eigenvector of a square matrix using Power Method

Theory:

Power Method :

It is iteration method , in which we use that converging property of the eigenvector ,this mean that if we multiply a vector that close to the eigenvector and then take output and multiply again it with the square matrix then after some iteration the vector converge to the most dominant eigenvector of matrix.

$$AX(1) = \lambda(1)X(2)$$
$$AX(2) = \lambda(2)X(3)$$
$$AX(3) = \lambda(3)X(4)$$
$$\vdots$$
$$AX(N) = \lambda(N-1)X(N)$$

Her A is the square matrix X is the vector and X(N) is the eigenvector and λ is eigenvalue.

How to program it :

To program it we first need to allocate a square matrix and two vectors and then we take the guess vector and the square matrix. Then we multiply both then get the new vector then check the maximum difference between the new vector and guess vector if the difference is less than error then print the vector if not then change the guess vector equal to new vector then we get new guess vector and then repeat the process.

Program in FORTRAN 95:

```
program main
  implicit none ! declaring the variable
  real , allocatable :: matrix(:,:) , vector(:) , pervious_vector(:)
  real :: user_error , eigenvalue , error = 1074E12
  integer :: i ,j,order , iteration

  ! getting the input from the user and also verified the input and allocating the space

  print *, "Enter the order of matrix :: "
  read *, order
  if (order <=0) stop "Invalid order of the matrix"
  allocate(matrix(order,order),vector(order),pervious_vector(order))
  print *, "Enter the Matrix :: "
  do i = 1,order
    do j = 1,order
      print *, "Enter the ",i,j,"th element :: "
      read *, matrix(i,j)
    enddo
  enddo
  print *, "Enter the guess for the eigenvector :: "
  read *, (pervious_vector(i),i=1,order)
  print *, "Enter the tolerance :: "
  read *, user_error
  print *, "Enter the maximum iteration :: "
  read *, iteration
  if (iteration <=0 ) stop "Invalid number of the iteration"

  ! making a loop until the error is less than tolerance or if iteration is complete
```

```
   do j = 1,iteration
      eigenvalue = 0
      do i = 1,order     ! multiplying the matrix and also find the largest value save in eigenvalue
         vector(i) = sum(matrix(i,:)*pervious_vector(:))
         if (abs(vector(i)) > eigenvalue) eigenvalue = vector(i)
      enddo
      error = 0
      do i = 1,order      ! Calculation the error and making vector equal to pervious vector
         vector(i) = vector(i)/eigenvalue
         error = max(abs(vector(i) - pervious_vector(i)),error)
         pervious_vector(i) = vector(i)
      enddo
      if (user_error > abs(error)) goto 1
   enddo
1   print *, "The eigenvector correspond to matrix in ",j,"is ",(vector(i), i= 1,order)
   eigenvalue = 0 ! here we calculating the eigenvalue by using the ratio of pervious vector and
new vector
   do i = 1,order
      vector(i) = sum(matrix(i,:)*pervious_vector(:))
      eigenvalue = eigenvalue + vector(i)/pervious_vector(i)
   enddo
   eigenvalue = eigenvalue/real(order)
   print *, "The eigenvalue is " , eigenvalue
   deallocate(vector,pervious_vector,matrix) ! de-allocating the all variables
   stop
end program
```

Output :

```
 Enter the order of matrix ::
3
 Enter the Matrix ::
 Enter the          1          1 th element ::
1
 Enter the          1          2 th element ::
2
 Enter the          1          3 th element ::
0
 Enter the          2          1 th element ::
-2
 Enter the          2          2 th element ::
1
 Enter the          2          3 th element ::
2
 Enter the          3          1 th element ::
1
 Enter the          3          2 th element ::
3
 Enter the          3          3 th element ::
1
 Enter the guess for the eigenvector ::
1          1          1
 Enter the tolerance ::
0.0001
```

Enter the maximum iteration ::
50
The eigenvector correspond to matrix in 10 is 0.499992937 0.499992937 1.00000000
The eigenvalue is 3.00000954

Flow Chart: