EXPERIMENT–10

AIM–Calculation of Eigen Values and Eigen Vectors of matrices using Power method.

THEORY–

Eigen Values and Eigen Vectors:

In linear algebra, an eigenvector or characteristic vector of a linear transformation is a nonzero vector that changes at most by a scalar factor when that linear transformation is applied to it. The corresponding Eigen Value is the factor by which the eigenvector is scaled.

Eigen Values and eigenvectors feature prominently in the analysis of linear transformations. Originally utilized to study principal axes of the rotational motion of rigid bodies, Eigen Values and eigenvectors have a wide range of applications, for example in stability analysis, vibration analysis, atomic orbitals, facial recognition, and matrix diagonalization.

In essence, an eigenvector v of a linear transformation T is a nonzero vector that, when T is applied to it, does not change direction. Applying T to the eigenvector only scales the eigenvector by the scalar value λ, called an Eigen Value. This condition can be written as the equation

$$T(\mathbf{v}) = \lambda\mathbf{v},$$

referred to as the Eigen Value equation or Eigen Equation. In general, λ may be any scalar. For example, λ may be negative, in which case the eigen vector reverses direction as part of the scaling, or it may be zero or complex.

POWER METHOD :-

The power method is very good at approximating the extremal Eigen Values of the matrix, that is, the eigenvalues having largest and smallest module, denoted by λ1 and λn respectively, as well as their associated eigen vectors.Solving such a problem is of great interest in several real-life applications (geosysmic, machine and structural vibrations, electric network analysis,quantum mechanics,...) where the

EXPERIMENT-10

computation of λn (and its associated eigenvector xn) arises in the determination of the proper frequency (and the corresponding fundamental mode) of a given physical system. Having approximations of λ1 and λn can also be useful in the analysis of numerical methods.

APPORXIMATION OF EIGENVALUES:-

Let A ∈ Cn×n be a diagonalizable matrix and let X ∈ Cn×n be the matrix of its right eigenvectors xi, for i = 1, . . . , n. Let us also suppose that the eigenvalues of A are ordered as |λ1| > |λ2| ≥ |λ3| . . . ≥ |λn|,

where λ1 has algebraic multiplicity equal to 1. Under these assumptions, λ1 is called the dominant eigenvalue of matrix A. Given an arbitrary initial vector q(0) ∈ Cn of unit Euclidean norm, consider for k = 1, 2, . . . the following iteration based on the computation of powers of matrices, commonly known as the power method:

$$Av = \lambda v$$

$$b_{k+1} = \frac{Ab_k}{||Ab_k||}$$

$$b_{k+1} = \frac{(A - \mu I)^{-1} b_k}{||(A - \mu I)^{-1} b_k||}$$

EXPERIMENT-10

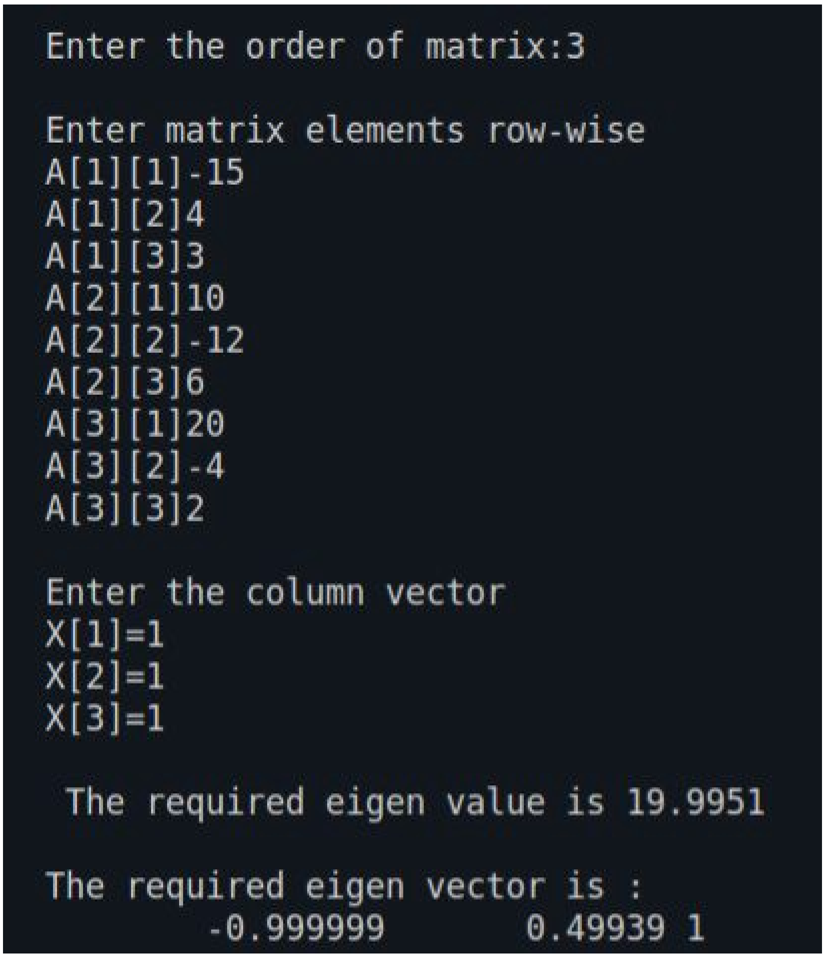
CODE-

```
#include<iostream>
#include<math.h>
using namespace std;
int main()
{
    int i,j,n;
    float A[40][40],x[40],z[40],e[40],zmax,emax;
    cout<<"\nEnter the order of matrix:";
    cin>>n;
    cout<<"\nEnter matrix elements row-wise\n";
    for(i=1; i<=n; i++)
        {
            for(j=1; j<=n; j++)
            {
            cout<<"A["<<i<<"]["<<j<<"]";
                scanf("%f",&A[i][j]);
            }
        }
    cout<<"\nEnter the column vector\n";
    for(i=1; i<=n; i++)
        {
        cout<<"X["<<i<<"]=";
        cin>>x[i];
        }
    do
    {
        for(i=1; i<=n; i++)
        {
            z[i]=0;
            for(j=1; j<=n; j++)
            {
                z[i]=z[i]+A[i][j]*x[j];
            }
        }
        zmax=fabs(z[1]);
        for(i=2; i<=n; i++)
        {
            if((fabs(z[i]))>zmax)
```

```
                zmax=fabs(z[i]);
        }
        for(i=1; i<=n; i++)
        {
            z[i]=z[i]/zmax;
        }
        for(i=1; i<=n; i++)
        {
            e[i]=0;
            e[i]=fabs((fabs(z[i]))-(fabs(x[i])));
        }
        emax=e[1];
        for(i=2; i<=n; i++)
        {
            if(e[i]>emax)
                emax=e[i];
        }
        for(i=1; i<=n; i++)
        {
            x[i]=z[i];
        }
    }
    while(emax>0.001);
    cout<<"\n The required eigen value is "<<zmax;
    cout<<"\n\nThe required eigen vector is :\n";
    for(i=1; i<=n; i++)
    {
        cout<<"\t"<<z[i];
    }
    cout<<"\n";
    return 0;
}
```

EXPERIMENT-10

OUTPUT-



```
Enter the order of matrix:3

Enter matrix elements row-wise
A[1][1]-15
A[1][2]4
A[1][3]3
A[2][1]10
A[2][2]-12
A[2][3]6
A[3][1]20
A[3][2]-4
A[3][3]2

Enter the column vector
X[1]=1
X[2]=1
X[3]=1

 The required eigen value is 19.9951

The required eigen vector is :
        -0.999999       0.49939 1
```