ICT Tool:- 'Scilab' Program for Power Method

Prabhavathi M[1]

[1]*Assistant Professor, Dept. Of Computer Science, Field Marshal K M Cariappa College, Madikeri, Karnataka, India*

Abstract - *In Mathematics, Power method is used to find the dominant eigenvalue and the corresponding eigenvector. Eigen value problems occur in many areas of science and engineering, such as structural analysis and dynamics problems. Eigen values are also important in analyzing numerical methods. Power method is generally used to calculate these Eigen value and corresponding eigenvector of the given matrix.*

In the era of Information and Communication Technology (ICT), ICT programming technique is easier task. One of the important numerically oriented programming languages is Scilab. In this paper we discuss Power method using Scilab with source code, method and output.

Key Words: Power Method, Eigenvector, Scilab, ICT, Eigenvalue.

1. INTRODUCTION

Scilab is a high-level, numerically oriented programming language. The language provides an interpreted programming environment, with matrices as the main data type. By using matrix-based computation, many numerical problems may be expressed in a reduced number of code lines, as compared to similar solutions using traditional languages, such as Fortran, C, or C++. This allows users to rapidly construct models for a range of mathematical problems [3]. The power method can be used for computing the dominant eigenvalue with the largest magnitude and the corresponding eigenvector of a matrix. It is a simple algorithm which does not compute matrix decomposition, and hence it can be used in cases of large sparse matrices. Power method gives the largest eigenvalue and it converges slowly [1].

1.1 POWER METHOD

Assume that the n x n matrix A has the eigenvalues λ_j, j = 1, 2, .., n and that

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n| \qquad \ldots(1)$$

Also assume that the eigenvectors x_j, j = 1, 2, 3 . . . , n are linearly independent. This means that any vector z can be written as

$$z = \sum_{j=1}^{n} c_j\, x_j$$

and

$$Az = \sum_{j=1}^{n} c_j\, A\, x_j = \sum_{j=1}^{n} c_j\, \lambda_j\, x_j$$

Repeated multiplication with A leads to

$$A^k Z = \sum_{j=1}^{n} c_j\, \lambda_j^k\, x_j$$

$$= \lambda_1^k \left(c_1 x_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k x_2 + \ldots + c_n \left(\frac{\lambda_n}{\lambda_1}\right)^k x_n \right)$$

Because of (1) this behaves asymptotically as $\lambda_1^k\, c_1\, x_1$ as $(k \to \infty)$, i.e., a multiple of the eigenvector x_1.

Let's look at a simple mathematical formulation of eigen values and eigen vector. For this, consider a matrix A. We have to find the column vector X and the constant L (L=lamda) such that:

[A]{X} = L{X}

Now, consider these three set of equations:

a11x1 + a12x2 + a13x3 = Lx1
a12x1 + a22x2 + a23x3 = Lx2
a31x1 + a32x2 + a33x3 = Lx3

These equations can be written as:

(a11-L)x1 + a12x2 + a13x3 = 0
a21x1 + (a22-L)x2 + a23x3 = 0
a31x1 + a32x2 + (a33-L)x3 = 0

1.2 ALGORITHM AND FLOW CHART

Algorithm for the Power method to find the Eigen value and the corresponding Eigen vector is based on the method of "Characteristic polynomial".

For the Power Method algorithm and flowchart, we will consider a given matrix A for which we have to find the column vector X and a constant L (read as Lamda) such that:

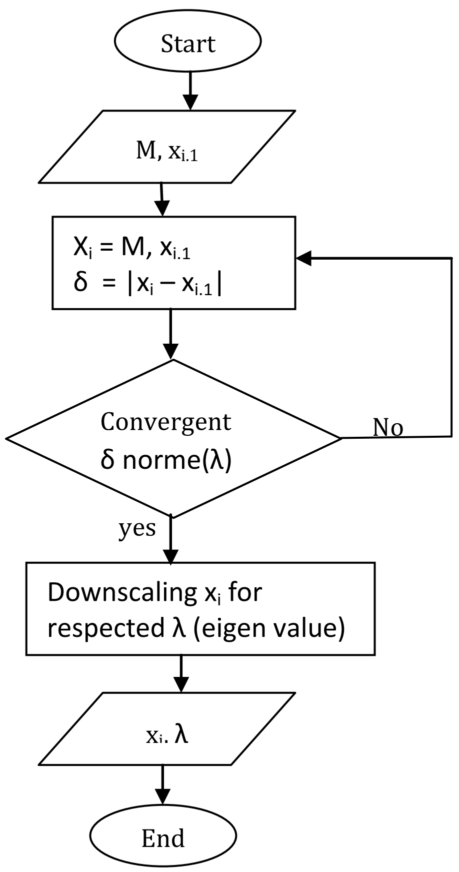
[A] [X] = L [X] which gives
{ [A] – L [I] } [X] = [0]

Now, for a system with 3 linear algebraic equations, we get
– a set of {X1, X2, X3} column vector for L1
– a set of {X1, X2, X3} column vector for L2
– a set of {X1, X2, X3} column vector for L3

Algorithm:

1. Start
2. Define matrix X
3. Calculate Y = AX
4. Find the largest element in magnitude of matrix Y and assign it to K.
5. Calculate fresh value X = (1/K) * Y
6. If [Kn – K(n-1)] > delta, goto step 3.
7. Stop

Flow chart:



2. SCILAB PROGRAM FOR POWER METHOD

```
clc
clear
A=input('Enter the matrix : ');
disp(A)
x=input('Enter the initial approximation to the eigenvector : ');
disp(x)
[nA,mA]=size(A)
[nx,mx]=size(x)

if(nA<>mA)then
  mprintf(" matrix must be square\n")
  abort
elseif (mA<>nx) then
  mprintf("matrix compatible dimension b/w A & b\n")
  abort
end
n=nA
e=zeros(1,n)

while(1) do
  for i=1:n
    z(i)=0
    for j=1:n
      z(i)=z(i)+A(i,j)*x(j)
    end
  end
  zmax=abs(z(1))
  for i=2:n
    if abs(z(i))>zmax then
      zmax=abs(z(i))
    end
  end
  for i=1:n
    z(i)=z(i)/zmax
  end
  for i=1:n
    e(i)=0
    e(i)=abs((abs(z(i)))-(abs(x(i))))
  end
  emax=e(1)
  for i=2:n
    if (e(i)>emax)then
      emax=e(i)
    end
  end
  for i=1:n
    x(i)=z(i)
  end
  if(emax<0.001) then// 0.001 is allowed error
    break
  end
end

mprintf("The required eigenvalue is : %f \n",zmax)
mprintf("The required eigenvector is \n");
for i=1:n
  mprintf("%f\t",z(i));
end
```

3. OUTPUT

OUTPUT1:

Enter the matrix : [2, -1, 0; -1, 2, -1; 0, -1, 2]

```
  2.  -1.   0.
 -1.   2.  -1.
  0.  -1.   2.
Enter the initial approximation to the eigenvector : [1; 0; 0]

  1.
  0.
  0.
The required eigenvalue is : 3.414214
The required eigenvector is
0.708459          -1.000000          0.705754
```

OUTPUT2:

```
Enter the matrix : [2, -1, 0; -1, 2, -1; 0, -1, 2]

  2.  -1.   0.
 -1.   2.  -1.
  0.  -1.   2.
Enter the initial approximation to the eigenvector : [1, 0, 0]

  1.   0.   0.
matrix compatible dimension b/w A & b
```

REFERENCES

[1] P R Kolhe, M H Tharkar, Pradip Kolhe, S Gawande, "ICT Tool: - 'C' Language Program for Power Method," International Journal of Emerging Technologies in Engineering Research , vol. 5, Issue – 6, June. 2017, pp.96-97.

[2] Dr. B.S Grewal "Numerical Methods in Engineering & Science" Book.

[3] https://en.wikipedia.org/wiki/Scilab

[4] Qi Lei, Kai Zhong, Inderjit S. Dhillon. "Coordinate-wise Power Method", 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.