NJESR/July 2021/ Vol-2/Issue-7 DOI - 10.53571/NJESR.2021.2.7.56-71 E-ISSN-2582-5836

THE MATRIX FORMULATIONS AND ALGEBRAIC EQUATIONS USING SCILAB

Dr. Ravindra Singh[1*], Dr. Ashok Kumar Shukla[1], Dr. Shiv Shankar Gaur[1], Dr. Priyanka Verma[1], Polly Biswas[2], Dr.Omwati Rana[3]

[1]Department of Physics, Shivaji College (University of Delhi), Raja Garden, New Delhi-110027.

[2]Department of Physics, Maitreyi College (University of Delhi), Chanakyapuri New Delhi-110021.

[3]Department of Physics, Daulat Ram College (University of Delhi), New Delhi-110007.

Corresponding author: ravidelhi06@yahoo.com

ABSTRACT

The study of matrix formulations and algebraic equations have been done through Scilab software. In this paper the commands for matrixoperations have been used.The matrix operationslike, identity matrix, diagonal matrix, matrix with all element ones & zeros, eigen values & eigen vectors, random matrix (elements of a matrix are found randomly) have been applied. Besides these, some other parameters such as rank, determinant, diagonal, inverse, transpose, adjoint, adding & deleting rows or columns of a matrix, sum& product of rows or columns of a matrix have been discussed. So the role of matrix is so important in Scilab to perform or execute the programs. At last the algebraic equations have been solved by using matrix form.

Keywords: Scilab, Matrix Formulations

INTRODUCTION

Scilab is an open source platform and it is free of cost to everyone need, easily available everywhere.It having large no. of applications in all branches.The mathematical problems &operations can be done with the help of Scilab. In scilab everything is a matrix. We are familiar about the matrix and its properties, in mathematics branch, branch of sciences and engineering we use the matrices. In this paper we have done the mathematical formulation and algebraic equation. A complete environment is provided by it as well as all the operations either which are mathematical or programable. Eighty to ninety percent operations of Scilabare similar to Matlab. It is so easy to install it on the device and need not of high RAM,

it easily runs on low RAM device. It can also run online on android devices with full operations. Some operations on differential equations have been done for Euler's method and modify Euler's method with Scilab [1]. For different operating systems there are different versions for installation. Window version cannot be installed in Mac Version and Macversion can't be installed in Linux version; in other words we can say that for Windows operating system we must install Windows version only because Linux or Mac version of Scilab will not work on Windows version.Scilab is available world wide. Some numerical techniques were opted to study least square fitting and the results were verified by manually as well as by the Scilab; it was found that in both the results were same [2]. In quantum physics there is also important role of Scilab which capable to explain the properties like atomic phenomena and the behaviour of elementary particles [3]. The study of the three statistics had been done at different temperature by Biswas et.al. and the comparison of the results took place through Scilab [4]. Scilab is a strong software like others that is capable to solve the problems based on X-cos simulations or based on programming. It frequently and widely has been used in research institutions, industries and companies [5]. To plot graphs and solving the differential equations Scilab is very friendly; an application to solve the differential equations had been discussed with different methods and under some conditions constraints as well asthe comparison of the results were done and analyzed with Laplace transform and with D'operator method, both methods gave the same result when the study had been done [6]. Scilab programming took place for the estimation of parameters in Gompertz curve using Gauss-Newton method of least squares was done which is statistical analysis [7]. Some authors proposed and discussed their views for the comprehensive study of the equations [8-9]. If we talk about themodeling we can say that much more numerical methods had been used mathematical modelling that's an importantcourse in mathematics and engineering [10]; the mathematical problems in science and engineering frequently solved by Scilab but due to its limitations some ODEs can be solved analytically. Some algebraic & mathematical operations like matrices, problems based on ODEs and integration were performed; it generates the approximate solution for tangent line method or the Euler method [11].

MATRIX FORMULATIONS WITH COMMANDS:

Formation of a row matrix'A':

The row matrix can be constructed by the command

```
--> A=[12   13  14  22]
```

Or

--> A=[12, 13, 14, 22]

A = 12. 13. 14. 22.***// It is the output***

Formation of a column matrix'B':

The column matrix can be made by the command

--> B=[12;13;14;22]

B =

12.

13.

14.

22.***// It is the output***

Formation of any 3 x 3 matrix'A':

--> A=[12 13 11;22 65 77;55 44 22]

A =

12. 13. 11.

22. 65. 77.

55. 44. 22.***// It is the output***

Formation of any n x n matrix (n is any positive integer)

We can prepare any square matrix of order n*n with random elements. Here we are taking any 5 x 5 matrix. The command is following

B=testmatrix('magi',5)

B =

17. 24. 1. 8. 15.

23. 5. 7. 14. 16.

4. 6. 13. 20. 22.

10. 12. 19. 21. 3.

11. 18. 25. 2. 9.***// It is the output***

Next we will make some operations using matrix. For convenience we are taking any 3 x 3 matrix; then we make some important operations.

--> A=[2 2 4 ;6 8 5;8 6 7]

A =

2. 2. 4.

6. 8. 5.

8. 6. 7.***// It is the output***

Length of a matrix'A':(the number of elements in matrix)

The command is

--> length(A)

ans =

9.***// It is the output***

Size of a matrix'A':(the number of rows and columns in a matrix)

The command is

--> *size(A)*

ans =

3. 3.***// It is the output***

Determinant of a matrix 'A'

The command is

--> det(A)

ans =

-64.***// It is the output***

Rank of a matrix 'A'

The command is

--> rank(A)

ans =

3.***// It is the output***

Diagonal of a matrix 'A':(diagonal elements of a matrix A)

The command is

-->diag(A)

ans =

2.

8.

7. ***// It is the output***

Trace of a matrix 'A':(trace is the sum of diagonal elements)

The command is

--> trace(A)

ans =

17.***// It is the output***

Transpose of a matrix 'A'

The command is

--> A'

ans =

2. 6. 8.

2. 8. 6.

4. 5. 7.***// It is the output***

Inverse of a matrix 'A'

The command is

--> inv(A)

ans =

-0.40625 -0.15625 0.34375

0.03125 0.28125 -0.21875

0.4375 -0.0625 -0.0625 ***// It is the output***

Adjoint of a matrix 'A'

The command is

-->det(A)*inv(A)

ans =

26. 10. - 22.

- 2. - 18. 14.

- 28. 4. 4.***// It is the output***

SPECIAL MATRICES

IDENTITY MATRIX:

The command is

-->eye(A)

ans =

1. 0. 0.

0. 1. 0.

0. 0. 1.***// It is the output***

ALL ELEMENTS ZERO

The command is

--> zeros(A)

ans =

0. 0. 0.

0. 0. 0.

0. 0. 0.***// It is the output***

ALL ELEMENTS WITH ONES

The command is

--> ones(A)

ans =

1. 1. 1.

1. 1. 1.

1. 1. 1.*// It is the output*

EIGENVALUES AND EIGENVECTORS OF MATRIX 'A'

-->[V,E]=spec(A)

E =

16. 0 0

0 - 1.5615528 0

0 0 2.5615528 *// The diagonal values are the eigen values*

V =

- 0.2926279 - 0.7766956 0.3570898

- 0.6549290 0.1702700 - 0.8144420

- 0.6967330 0.6064256 0.4573522*// Corresponding eigen vectors*

E=eigenvalues, V=eigenvectors

Use of 'gsort' command. It pin out the matrix element in decreasing order and return them column by column. Now takin a new 4x4 matrix 'A'

--> A=[1 -5 6 10; 5 -4 7 11;30 12 -1 9;11 -11 6 7]

A =

1. -5. 6. 10.

5. -4. 7. 11.

30. 12. -1. 9.

11. -11. 6. 7. *// It is the output*

-->gsort(A)

ans =

30. 10. 6. -1.

12. 9. 6. -4.

11. 7. 5. -5.

11. 7. 1. -11.***// It is the output***

RANDOM: (bydefault rand command produces a matrix of random number with uniformly distribution between 0 to 1)

-->rand(A)

ans =

0.3321719 0.4368588 0.4051954 0.4818509

0.5935095 0.2693125 0.9184708 0.2639556

0.5015342 0.6325745 0.0437334 0.4148104 ***// It is the output***

(bydefault rand command produces a matrix of random number with uniformly distribution between 0 to 1)

-->A=rand(3,4)

A =

0.0683740 0.7263507 0.2320748 0.8833888

0.5608486 0.1985144 0.2312237 0.6525135

0.6623569 0.5442573 0.2164633 0.3076091 ***// It is the output***

RANDOM ELEMENTS:

--> A =rand(3,4,'normal')

A =

- 1.4061926 0.5546874 2.5891773 - 1.6565358

- 1.0384734 - 0.2143931 0.0424792 1.7224151

- 1.7350313 - 2.0735088 - 0.4534087 - 1.2317066***// It is the output***

Suppose we again take a 4x4 matrix A

-->A=testmatrix('magi',4)

A =

16. 2. 3. 13.

5. 11. 10. 8.

9. 7. 6. 12.

4. 14. 15. 1.

-->A(2,3)*// representing in the second row, the third element of the above matrix 'A'*

ans =

10.

Use of Colon Operator(:)

-->A(3,:) *// All elements of third row of above matrix A*

ans =

9. 7. 6. 12.

-->A(:,2) *// All elements of second row of above matrix A*

ans =

2.

11.

7.

14.

-->A(:,2:3)*// formation of new matrix from old (above) matrix or pick up 2nd to 3rd column of matrix A*

ans =

2. 3.

11. 10.

7. 6.

14. 15.

-->A(1:2,2:3)***//pick the new matrix from 1st to2nd row and 3rd to 4th column from matrix'A'***

ans =

2. 3.

11. 10.

-->A(3,:)=[0 9 90 900] ***// Replacing 3rd row of above matrix 'A' by new elements 0 9 90 900***

A =

16. 2. 3. 13.

5. 11. 10. 8.

0. 9. 90. 900.

4. 14. 15. 1.

-->A(:,4)=[1 10 100 1000]' ***// Replacing 4th column of a matrix A by new elements1 10 100 1000***

OR

-->A(:,4)=[1;10;100;1000]

A =

16. 2. 3. **1.**

5. 11. 10. **10.**

0. 9. 90. **100.**

4. 14. 15. **1000.*// It is the output.Now it is new matrix 'A'***

-->A(2,:)=[] ***// Delete 2nd row of above matrix 'A'***

A =

16. 2. 3. 1.

0. 9. 90. 100.

4. 14. 15. 1000. ***// It is the output. Now it is new matrix 'A'***

-->A(:,3)=[]***//Delete 3rd column of a matrix 'A'***

A =

16. 2. 1.

0. 9. 100.

4. 14. 1000. ***// It is the output. Now it is new matrix 'A' you can take some different matrix also***

SUM OF ROW & COLUMN ELEMENTS:

--> sum(A,'r') ***// Sum of column elements of above matrix A or column wise sum i.e for first column sum is (16+0+4 = 20), for second column sum is (2+9+14 = 25) and for the last column (1+100+1000 =1101)***

ans =

20. 25. 1101.

--> sum(A,'c')***// sum of row elements of above matrix A***

ans =

19.

109.

1018.***// It is the output***

--> sum(A) ***//sum of row and column elements***

ans =

1046.*// It is the output*

PRODUCT OF ROW & COLUMN ELEMENTS:

--> prod(A,'r') *// product of row elements of new formed matrix 'A'*

ans =

0. 252. 100000.*// It is the output*

--> prod(A,'c') *// product of columns elements*

ans =

32.

0.

56000. *// It is the output*

Minimum or Maximum values for a Matrix 'A'

As we know that new formed matrix 'A' is

A =

16. 2. 1.

0. 9. 100.

4. 14. 1000.

-->min(A,'r')*// represents the minimum value in columns of a matrix 'A'*

ans =

0. 2. 1. *// It is the output*

-->max(A,'c')*// represents the minimum value in rowsof a matrix 'A'*

ans =

16.

100.

1000. ***// It is the output***

-->min(A) ***// minimum value of a matrix 'A'***

ans =

0. ***// It is the output***

--> max(A) ***// maximum value of a matrix 'A'***

-->max(A)

ans =

1000.***// It is the output***

Singularity of Matrix:

--> clean(det(A))= =0***// represents that the matrix is singular or non- singular. If answer is F (false) it means it is not a singular matrix.***

ans =

F

--> clean(det(A))***// It gives the determinant***

ans =

122364.

ALGEBRIC EQUATIONS: Consider a set of algebraic equations given below

$$5x - 2y + 9z = 2 \quad \ldots\ldots\ldots (1)$$

$$10x - 8y - 2z = 5 \quad \ldots\ldots\ldots (2)$$

$$8x + 12y - 4z = 3 \quad \ldots\ldots\ldots (3)$$

To solve the above equations we have to write in matrix form using AX = B;
where A is the matrix(3x3) formed by the coefficients with x, y and z i.e

A=

5. - 2. 9.

10. - 8. - 2.

- 8. 12. - 4.

And X is the column matrix i.e

X=

x

y

z

Right hand side is B, a column matrix again

B =

2.

5.

3

For solving we use following commands

-->A=[5,-2,9; 10,-8,-2; -8,12,-4]

A =

5. - 2. 9.

10. - 8. - 2.

- 8. 12. - 4.

-->B=[2; 5; 3]***// It is a column matrix***

B =

2.

5.

3.

-->X=A\B

X =

1.25

1.

- 0.25 ***// It is the output*** i.e x=1.25, y=1 and z=-0.25.

RESULTS &CONCLUSIONS

The matrix formulation and algebraic equations have been successfully done with the help of Scilab. Using Scilabcommands we have successfully determined eigen values & eigen vectors, random matrix (elements of a matrix are found randomly),rank, determinant, diagonal, inverse, transpose, adjoint, adding & deleting rows or columns of a matrix, sum and product of rows or columns of a matrix,identity matrix, matrix with all element ones & zeros etc. The algebraic equations also have been solved by matrix form. In electrical circuits we can see such types of equations. These formulations frequently uses in mathematical, science & engineering branches.

REFERENCES

[1] N. M. M. Yusop et.al., "Comparison New Algorithm Modified Euler in Ordinary Differential Equation Using Scilab Programming", Lecture Notes on Software Engineering, **3**(3), 199-202 (2015).

[2] P. Biswas et.al.,"Weighted Least Square Fitting Through Matrix Form Using Scilab Programming", **2**(6), 259-264 (2021).

[3] A.C. Phillips, "Introduction to Quantum Mechanics", John Wiley & Sons, New York, 16 –17 (2003).

[4] P. Biswas et.al., "The Study of Fermi-Dirac, Bose-Einstein & Maxwell-Boltzmann Distribution with Energy at Different Temperatures with Scilab Software", NJESR **2**(6), 52-59 (2021).

[5] S. Sarnaik& S. Savant, "International Conference on Advances in Information Technology and Management ICAIM", Int. J. of Comp. Appls. (0975 – 8887) 5-8 (2016).

[6] P. Biswas et.al., "The Equations of Motion of The Electron Under Certain Conditions with an Electric Field & Path Comparison", Expl. the Sci. Research vol 2 International Book of Multidisciplinary Studies, **2**, 41-50ISBN 978-93-91216-06-1 (2021).

[7] S.G. Dastidar, "A Scilab Program for Estimating Gompertz curve Using Gauss-Newton of Least Squares, Journal of Statistical Software", **15**(12), 1-12 (2006).

[8] D. York et.al., "Unified equations for the slope, intercept, and standard errors of the best straight line", Am J. Phys., **72**(3), 367–375 (2004).

[9] I. Markovsky& S. V. Huffel, "Overview of total least-squares methods", Sig. Proc., **87,** 2283–2302 (2007).

[10] S.Rao, Applied Numerical Methods for Engineers and Scientist, Pearson Prentice Hall Education, (2002).

[11] Z. Salleh, "ODE using Euler's Technique andSCILAB Programming", Math. Models and Methods in Mod. Sci. 264-269 (2012) ISBN: 978-1-61804-106-7.