A NEW EIGENVALUE SOLVER FOR SOLVING EIGENVALUE PROBLEMS IN STRUCTURAL ENGINEERING

Ezeh J.C., Ibearugbulem O. M., Nwaokorie M. C., Njoku K.O.

Abstract— **This paper presents a computer program written with visual basic that is capable of handling any n x n eigenvalue matrix problem that uses lump mass matrix. The program outputs the characteristic polynomial and eigenvalues of any n x n matrix eigenvalue problem. The program is written base on the application of Householder's technique and Newton-Raphson iteration technique. Householder's technique was used to generate the characteristics polynomial of an eigenvalue problem while the eigenvalues of the characteristics equation are generated using Newton-Raphson's procedure .The effectiveness of this method was demonstrated by comparing the eigenvalues and the characteristics polynomial obtained in the present study with existing eigenvalues obtained from previous works.**

Index Terms— visual basic program, Eigenvalues, Characteristic polynomial, Householder's technique, Newton-Raphsons iteration technique

INTRODUCTION

Eigenvalue problems arise when solving problems of mathematical physics. Such problems give rise to matrix eigenvalue problems which according to Yousef (2011) are classified in three different categories. The first category consists of problems related to the analysis of vibrations, which typically generate symmetric generalized eigenvalue problems. The second is the class of problems related to stability analysis and this second class of problems generates non symmetric matrices. The third category comprises physical applications related to quantum mechanical systems. This means that the knowledge of eigenvalue is very important to Structural, Electrical and Mechanical Engineers, who must solve such problems in order to make headway in their work. It is because of this that many researchers have spent most of their time and energy in this area of research and by so doing; many methods of solving such problems have been developed. Gene and Henk (2000) studied on the main research developments in the area of computational methods for eigenvalue problems during the 20th century. Klans and Edward (1973) surveyed on the most efficient solution methods currently in use for eigenvalue problems. The optimization method, called deflation-accelerated conjugate gradient was used by Luca et al (2000) to sequentially compute the smallest eigenpairs of a symmetric, positive definite, generalized eigen problem. Ibearugbulem et al (2013) developed a method known as matrix iterative-inversion that can solve all types of generalized eigenvalue problems for all matrix sizes which is efficient in convergence to exact solutions of eigenvalues. Xiao and Tong (2013) proposed a simple but yet effective solution to eigenvalue problems called truncated power method that can approximately solve non convex optimization problems. Other research works in this area of study includes the works of Edoardo and Mario (2003), Matthias et al (2013), Yunkai et al (2006), Arbenz and Greus (2005), Quillen and Ye (2010) and Bergamaschi et al (2000). The different methods used by these researchers are the Jacobi method, power iteration method, householder – QR – inverse iteration method, polynomial method, Lanczos method, Arnoldi method, Block inverse-free preconditioned Krylov subspace method and inner-outer iterative method. These methods can solve problems that have lump mass matrices. Also problems that have consistent mass matrix have been solved using matrix iterative – inversion method. Computer programs that can solve eigenvalue problems exit. Such programs are written in order to simplify the use of all these methods. To the best of the authors' knowledge, it is rare to find a program that solves problems of any n x n matrix eigenvalue problem, which outputs the characteristic polynomial and eigenvalues at the same time. What are in existence are programs that either outputs the eigenvalues or the characteristic

Manuscript received June 29, 2014.

Ezeh, J.C., Department of Civil Engineering, Federal University of Technology, Owerri, Nigeria

Ibearugbulem O.M., Department of Civil Engineering, Federal University of Technology, Owerri, Nigeria

Nwaokorie M. C., Department of Civil Engineering, Federal University of Technology, Owerri, Nigeria

Njoku K.O., Department of Civil Engineering, Federal University of Technology, Owerri, Nigeria

polynomials. So it is important to develop a program that can do this. In this present work, such a program is presented and it is written based on the application of Householder's technique and Newton – Raphson iteration technique. The Householder's technique was Use to generate the characteristics polynomial of any n x n matrix eigenvalue problem and the roots of the characteristics equation where generated using Newton – Raphson's procedure. The effectiveness of the program was demonstrated by comparing the eigenvalues obtained using this program with existing eigenvalues obtained from previous works.

EIGENVALUE PROBLEMS IN STRUCTURAL ENGINEERING

Eigenvalues that arise when studying vibration problems are very important to structural engineers, who must analyze such problems in other to obtain the fundamental frequencies of vibration.

$$Ax + \omega Bx + \omega^2 Cx = 0 \quad (1)$$

Equation (1) is a quadratic eigenvalue problem and it represents the matrix equation form of a typical structural dynamic problem. Where A is the stiffness matrix, C is the mass matrix, B is the damping matrix, ω is vibration frequency and x is the eigenvector.

For the case of no damping, where $B = 0$,we have:

$$Ax + \omega^2 Cx = 0 \quad (2)$$

Equation (2) can be of this form shown as equation (3)

$$Ax = \omega^2 Cx \quad (3)$$

Equation (3) is a generalized eigenvalue problem use for finding the natural frequencies of free vibration structural dynamic problems. Another generalized eigenvalue problem in structural Engineering arises in buckling analysis. The equation governing buckling of an assemblage of structural element is given as equation (4).

$$Ax - \omega A_g x = 0 \quad (4)$$

Where A is the small deflection stiffness matrix and A_g which is always banded, is the geometric stiffness matrix of the element system. The eigenvalues give the buckling loads and the eigenvectors represent the corresponding buckling modes.

If the size of a square matrix is more than 3 x 3, it is tedious to determine the eigenvalues of equations (3) and (4). In other to overcome such difficulties, the used of lump mass have been recommended by analysts and according to them, would help facilitate condensation of the structural matrix. Making C an identity matrix reduces equation (3) to:

$$(A - \lambda I)x - 0 \quad (5)$$

The eigenvalues λ (or ω^2) and eigenvectors x are the free vibration frequencies $(rad/sec)^2$ and corresponding mode shape repectively.

HOUSEHOLDER'S CHARACTERISTIC EQUATION

Householder's technique of eigenvalue solving as given by Szilard (2004) was used in this work. This eigen value technique was used to determine the characteristic polynomial from the expression given as equation (6) or equation (7):

$$[A - \omega^2 C] = [K] = 0 \quad (6)$$

$$[A - \lambda I] = [K] = 0 \quad (7)$$

Where K is the matrix eigenvalue.
The characteristic polynomial representing equation (6) or equation (7), according to Szilard (2004) is given as equation (8):

$$\lambda^n + \alpha_1 \lambda^{n-1} + \alpha_2 \lambda^{n-2} + \alpha_3 \lambda^{n-3} + \cdots \ldots \alpha_{n-1} \lambda + \alpha_n = 0 \quad (8)$$

Equation (8) is the Householder's characteristic equation of any n x n matrix eigenvalue problem.
In this expression, we have that;

$\alpha_1 = -T_1$, $\alpha_2 = -\frac{1}{2}(\alpha_1 T_1 + T_2)$, $\alpha_3 = -\frac{1}{3}(\alpha_2 T_1 + \alpha_1 T_2 + T_3)$, $\alpha_n = -\frac{1}{n}(\alpha_{n-1} + \alpha_1 T_2 + \ldots\ldots\ldots\ldots \alpha_1 T_{n-1} + T_n)$.
Where T_1 = Trace [K], T_2 = Trace $[K]^2$, T_n = Trace $[K]^n$

EVALUATING CHARACTERISTICS POLYNOMIAL TO EIGENVALUES

In this work, Newton-Raphson's iterative method as given by Bird (2010) was use to evaluate the characteristic polynomial to obtain the eigenvalues as:

$$a_2 = a_1 - \frac{f(a_1)}{f'(a_1)} \quad (9)$$

where a_1 is the approximate value of a real root of the expression $f(x) = 0$
F (a_1) is the function of a_1 and f ' (a_1) is the first derivative of function of a_1.

VISUAL BASIC PROGRAM FOR THE METHOD

A visual basic program was written in order to simplify the use of this method. This program outputs the characteristic polynomial and also goes on to give the

eigenvalues of any n x n matrix eigenvalue problem. The program is as shown in appendix to this work and for verification purposes; this program was used to test the following problems.

1 $\left\| \begin{bmatrix} 2 & 1 & -1 \\ -3 & 2 & -3 \\ 3 & 1 & -2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\|$ (Alan, 2002)

2 $\left\| \begin{bmatrix} 3 & 2 & 1 \\ 4 & 5 & -1 \\ 2 & 3 & 4 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\|$ (Glyn, 2011)

3 $\left\| \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\|$ (Michael,1998)

4 $\left\| \begin{bmatrix} 2 & -1 & 1 & 2 \\ 0 & 1 & 1 & 0 \\ -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right\|$ (Glyn,2011)

RESULTS, DISCUSSION, AND CONCLUSION

Visual basic program output data obtained for the above four eigenvalue problems are shown on Table 1. Also Table 1 shows the results obtained from previous works of Alan (2002), Glyn (2011) and Michael (1998). From the table, it can be seen that the solution of this study as obtained from the program are the same with that given by Alan (2002) for problem 1,this means that the eigenvalues and characteristics polynomial are the same. For problems 2 and 4 the characteristics polynomial as obtained using the program are the same with that given by Glyn (2011). The same is applicable to problem 3. From the above results, it can be said that the visual basic program written to solve any n x n matrix eigenvalve problem is recommended for users for solving eigenvalue problems in structural engineering.

Table 1: Result of Eigenvalue problem

Problem		1st eigenvalue	2nd eigenvalue	3rd eigenvalue	4th eigenvalue	Characteristics polynomial
1	V Basic Program (present study)	-1	1	2		$\lambda^3-2\lambda^2-\lambda+2$
	Alan(2002)	-1	1	2		$\lambda^3-2\lambda^2-\lambda+2$
2	V Basic Program (present study)	1.38	3.62	7		$\lambda^3-12\lambda^2+40\lambda-35$
	Glyn(2011)					$\lambda^3-12\lambda^2+40\lambda-35$
3	V Basic Program (present study)	1	1	1		$\lambda^3-3\lambda^2+3\lambda-1$
	Michael(1998)	1	1	1		
4	V Basic Program (present study)	3.061	3.061	3.061	3.061	$\lambda^4-4\lambda^3+2\lambda^2+5\lambda+2$
	Glyn(2011)					$\lambda^4-4\lambda^3+2\lambda^2+5\lambda+2$

APPENDIX (VISUAL BASIC PROGRAME)

```
Private Sub mnustart_Click()
  Text1.Text = ""
   N = InputBox("WHAT IS THE SIZE OF MATRIX"): N = N * 1
   ReDim A(N, N), B(N, N), A1(N, N), B1(N, N), C(N, N), Eig(N)
   ReDim D(N + 1), T(N + 1), L(N)
    Z = 2: T(1) = 0
   For X = 1 To N
   For Y = 1 To N
   A(X, Y) = 0: C(X, Y) = 0
   B(X, Y) = InputBox([Y], [X], "ENTER A"): A1(X, Y) = B(X, Y)
* 1
   Next Y
   C(X, X) = 1
   Next X
2000 For I = 1 To N
   For J = 1 To N
   For K = 1 To N
   A(I, J) = A(I, J) + C(I, K) * B(K, J)
   Next K
   Next J
   Next I
   For X = 1 To N
   T(Z) = T(Z) + A(X, X)
   Next X
   For X = 1 To N
   For Y = 1 To N
   C(X, Y) = A(X, Y): A(X, Y) = 0
   Next Y
   Next X
   If Z = N + 1 Then GoTo 3000
   Z = Z + 1: GoTo 2000
3000    Z = 3: D(1) = 1: D(2) = -T(2): M = 2: MM = M: X = 2: D(Z)
= 0
4000    D(Z) = D(Z) + D(M) * T(X)
      If X < Z - 1 Then X = X + 1: M = M - 1: GoTo 4000
      If X = Z - 1 Then D(Z) = D(Z) + T(X + 1): D(Z) = D(Z) / -(Z -
1)
      If Z = N + 1 Then GoTo 5000
      Z = Z + 1: M = MM + 1: MM = M: D(Z) = 0: X = 2: GoTo
4000
5000
      Z = N - 1: X = 2
      Text1.Text = Text1.Text + CStr("L" & N)
6000    For X = 2 To N
      Text1.Text = Text1.Text + (" + (" & D(X) & "L" & Z & ")")
      Z = Z - 1
      Next X
      Text1.Text = Text1.Text + (" + (" & D(X) & ")")
      Text1.Text = Text1.Text + (" ") & vbCrLf: Text1.Text =
Text1.Text + (" ") & vbCrLf
      Text1.Text = Text1.Text + (" ") & vbCrLf: Text1.Text =
Text1.Text + (" ") & vbCrLf
      For J = 1 To N + 1
      Text1.Text = Text1.Text + ("    D(" & J & ") =" & D(J)) &
vbCrLf
      Next J
      Text1.Text = Text1.Text + (" ") & vbCrLf: Text1.Text =
Text1.Text + (" ") & vbCrLf
      Text1.Text = Text1.Text + (" ") & vbCrLf: Text1.Text =
Text1.Text + (" ") & vbCrLf
  ' ROOT OF THE CHARACTERISTIC EQUATION BEGINS
HERE
   F = 0: FP = 0: LL = -5: M = N: Z = 0: CC = 1
   For X = 1 To N
   For Y = 1 To N + 1
   F = F + D(Y) * LL ^ M
   M = M - 1
   Next Y
   If F < 0 Or Abs(F) < 0.001 Then FF = 0 Else FF = 1
   KH = FF
7000  If Z = 100 Then L(X) = L(X - 1): GoTo 9000
   LL = LL + 0.1
   For Y = 1 To N + 1
   F = F + D(Y) * LL ^ M
   M = M - 1
   Next Y
   Z = Z + 1
   If F < 0 Or Abs(F) < 0.001 Then FF = 0 Else FF = 1
   If Abs(F) < 0.001 Then GoTo 8000
   M = N: F = 0: FP = 0
   If FF = KH Then GoTo 7000
8000  L(X) = LL
9000
   F = 0: FP = 0:  M = N: Z = 0:  LL = LL + 0.1
   Next X
1000 Text1.Text = Text1.Text + ("    ") & vbCrLf: Text1.Text =
Text1.Text + ("    ") & vbCrLf
   Text1.Text = Text1.Text + ("    ") & vbCrLf: Text1.Text =
Text1.Text + ("    ") & vbCrLf
   M = N: F = 0: FP = 0
    For J = 1 To N
   For X = 1 To 30
   For Y = 1 To N + 1
   F = F + D(Y) * L(J) ^ M
   If Y = N + 1 Then GoTo 20000
   FP = FP + M * D(Y) * L(J) ^ (M - 1)
20000 M = M - 1
   Next Y
   If FP = 0 Then GoTo 30000
   L(J) = L(J) - F / FP
30000 M = N: F = 0: FP = 0
   Next X
   M = N: F = 0: FP = 0
   Text1.Text = Text1.Text + ("    R(" & J & ") =" & L(J)) & vbCrLf
    Next J
   End Sub
```

REFERENCES

1) Arbenz, P., and Geus, R. (2005). Multilevel Preconditioned Iteractive eigensolvers for maxwell eigenvalue Problems. Applied numerical Mathematics, vol.54, issue 2, Pp 107-121.
2) Alan. J. (2002). Advanced Engineering Mathematic. San Diego: Harcourt/Academic press.
3) Bergamaschi, L., Pini, G. and Sartoretto, F. (2000). Approximate inverse preconditioning in the parallel solution of sparse eigen problems, Numer. Lin. Appl. 7 (3) Pp 99-116.
4) Bird .J. (2010). Higher Engineering Mathematics: UK: Elsevier Ltd.
5) Edoardo. D.N and Mario .B. (2013). Block Iterative Eigensolvers for Sequences of correlated Eigenvalue Problems. Computer Physics Communications, Vol. 184, Issue 11, Pp 2478-2488.
6) Gene, H.G. and Henk, A.V. (2000). Eigenvalue Computation in the 20th Century. Journal of Computational and Applied Mathematics, Vol.123, Pp 35-65.

7) Glyn. G. (2011). Advanced Modern Engineering Mathematics, 4th ed. Harlow: Pearson Education Limited

8) Ibearugbulem, O.M, Ettu, L.O., Ezeh. J.C. and Anya, U.C. (2013). Application of Matrix Iterative-Inversion in Solving Eigenvalue Problems in Structural Engineering. International Journal of Computational Engineering Research, Vol.3, Issue.4, Pp 17-22.

9) Klaus, J.B and Edward, L.W. (1973). Solution Methods for Eigenvalue Problems in Structural Mechanics.International journal for numerical methods in Engineering Vol.6, Pp 213-226.

10) Luca.B., Giorgio, P. and Flavio, S. (2000). Parallel. Preconditioning of a Sparse Eigensolver**.** Parallel Computing. Vol. 27, Pp 963-976.

11) Matthias, P., Elmar, P. and Paolo, B. (2013). High-Performance Solvers for Dense Hermitian Eigen Problems. SIAM Journal on Scientific Computing, Vol.35, Issue. 1 Pp 1-22.

12) Michael, D. (1998).Advanced Engineering Mathematics 2nd Ed. New Jersey: Prentice hall.

13) Quillen, P. and Ye, Q. (2010). A block Inverse - Free Preconditioned Krylov Subspace Method for Symmetric Generalized Eigenvalue Problems. Journal of Computational and Applied Mathematics, Vol. 233. Issue (5): Pp 1298-1313.

14) Szilard, R. (2004). Theories and Applications of Plate Analysis. New Jersey: John Wiley and Sons Inc

15) Yousef (2011). Numerical Methods for Large Eigenvalue Problems……, Society for Industrial and Applied Mathematics.

16) Yunkai, Z., Yousef, S., Murilo, L.T. and James, R.C. (2006). Self- Consistent- Field Calculations Using Chebyshv- Filtered Subspace Iteration. Journal of Computational Physics, Vol. 219 Issue (1), Pp 172-184.

17) Xiao, T.Y. and Tong. Z. (2013). Truncated Power Method for Sparse Eigenvalue Problems. Journal of Machine Learning Research, Vol. 14, Pp 899-925.