Finite Element Analysis of Continuous Plates Using a High-Performance Programming Language (MATLAB)

Amaefule Excel Obumneme [1], **Abuka Owen Chukwuebuka** [1], **Nwachukwu Nnaemeka Chukwudi** [1], **Nnadi Chukwuebuka Oscar** [1], **Ogbonna Chibueze Nelson** [1], **Adeyemo Habeeb Keji** [1], **Ogundu Meshack Ibeamaka** [1], **Okechukwu Sunday** [1], **Chinomso Don-Ugbaga** [1], **Gregory Ezeokpube** [1]

[1] *Michael Okpara University of Agriculture, Umudike*
PMB 7267, Umuahia Umudike, Abia State, Nigeria

DOI: 10.22178/pos.84-9

LCC Subject Category: L7-991

Received 21.07.2022
Accepted 25.08.2022
Published online 31.08.2022

Corresponding Author:
olipas.cris@gmail.com

Abstract. This paper uses MATLAB, a finite element software program to compare the results of several finite analysis methods for continuous plates and check the degree of correlation with the exact values obtained by Timoshenko (1959) and Cheung (1996). The results showed little or no significant difference between plates in finite elements. Two different finite Numerical techniques are used. The Finite Strip and Exact methods and their results are compared to the results from the MATLAB program. Finite element analysis (FEA) workflow using MATLAB includes generation of meshes, geometry creation, defining physics of load, initial conditions and boundary problems, calculation, and results from visualization. FEA is a very general approach for solving Equations in science and engineering. This work offers solutions to the increasing errors associated with several other numerical methods when solving any equations of plates (continuous). It makes it easier to calculate and design larger structures through geometry discretization of plates and plains into more minor elements.

Keywords: Finite Element Analysis; Finite Element Method; MATLAB; Continuous Plates; Finite Strip Method; Exact Method.

INTRODUCTION

The finite element method (FEM) is a technique that is used to analyze finite elements of any given phenomenon. For any physical phenomena to be quantified and comprehended, it is necessary to use mathematics; such physical phenomena can include thermal transport, structural or fluid behaviour, biological cell growth, and wave propagation. Partial differential equations (PDEs) describe most of these processes. However, the finite element method and other numerical techniques developed over the past decades are used to solve partial differential equations using a computer.

Finite element analysis is a numerical technique based on the finite element method (FEM). This technique uses computer programs to predict behaviours of several physical systems, such as the deformation of solids, fluid flow, and heat conduction. Engineers and physicists use FEA as it allows the application of physical laws to practical scenarios. This allows for precision, versatility, and practicality.

FEM is widely used to get solutions for complex problems in the engineering field that are formulated using PDEs.

The development of reliable computer programs is required for numerical simulations based FEM to get correct solutions. This undergraduate thesis aims to help students understand how finite element processes solve problems through computer programs. The foundations of FEM and other computational problems have been taught and discussed in class lectures. Although, a class alone is not sufficient to grasp deep knowledge of how the operations like matrix assembly, boundary conditions, etc. all work effectively.

The role of plate structures in every structural system that man has ever built cannot be overemphasized. Closed-form solutions to specific plate problems are seen in the literature and are an asset to this century's designers.

These solutions, however, are generally for plates with relatively simple support and loading conditions and restricted to elastic, homogenous, isotropic plates of constant thickness. During the early periods of engineering, plates were designed with much more strength than was required. With the introduction of digital computers, investigators have all focused on numerical techniques and their study, like the FEA, used to solve the problem of plate structures.

This work simply tries to offer solutions linked to the other numerical methods in solving continuous plate's equations through FEA. It aims at:

1. Reducing errors in calculation.

2. Aiding calculations of larger structures.

3. Aiding in the design of larger structures.

4. Develop programs that tackle issues on any form of continuous plates.

METHODOLOGY

Matrix Laboratory is a programming software used for quick and easy scientific calculations and Input/Output (I/O).

It has hundreds of built-in functions for a wide range of computations and many toolboxes set for different research disciplines, for example, data analysis, optimization, and statistics. Among much strength:

1) MATLAB may behave as a calculator or as a programming language.

2) MATLAB combines calculations and efficiently plots graphs.

3) MATLAB is not challenging to learn.

4) MATLAB is interpreted (not compiled), and errors are not hard to fix.

5) MATLAB is optimized to perform matrix operations quickly and efficiently.

6) MATLAB has some object-oriented elements.

Finite element analysis of continuous plates. Outline of steps. In any region of space where a particular phenomenon is occurring (continuum), a problem of any dimension, the field variables (stress, displacement, etc.) can assume an infinite set of values because it is a function of each generic point on the continuum. The finite element discretization procedure decreases the issues to one with a limited number of unknowns by dividing the solution region into smaller bits and expressing the unknown field variable in terms of an appropriate approximating function within each element. The values of the field variables gotten for specific pints which let on the element boundaries every other point in which the details are completely defined using the interpolation (approximating) function.

Step 1. Discretization of plate. The discretization is done accordingly, and if the number of divisions increases, it is done similarly. Sixty-four elements were discretized. The node and element numbers are also shown below [35].

Figure 1 shows the figures denoting the number of nodes in which the graphs are plotted in y and x directions using MATLAB for a 64-element division.

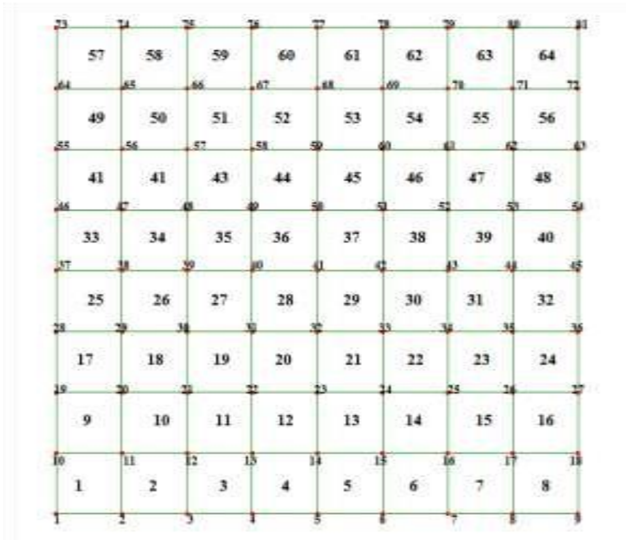

Figure 1

Figure 2 shows the node numbering done in the MATLAB program for the x-axis for 64 elements. The same procedure is used to number nodes. When the node number increases, the total will start from the centre of the plate along the x axis from left to right.

Figure 3 shows the numbering of nodes done in the MATLAB program for the y-axis for 64 elements. The same procedure will follow in the number of factors; increased node numbering will be done along the centre line of the plate from top to bottom.

The first thing to do in FEA is to divide the rectangular plate into more minor elements. The discretization can be achieved using a variety of factors, which depends on the degree of accuracy needed and the nature of the solution region.

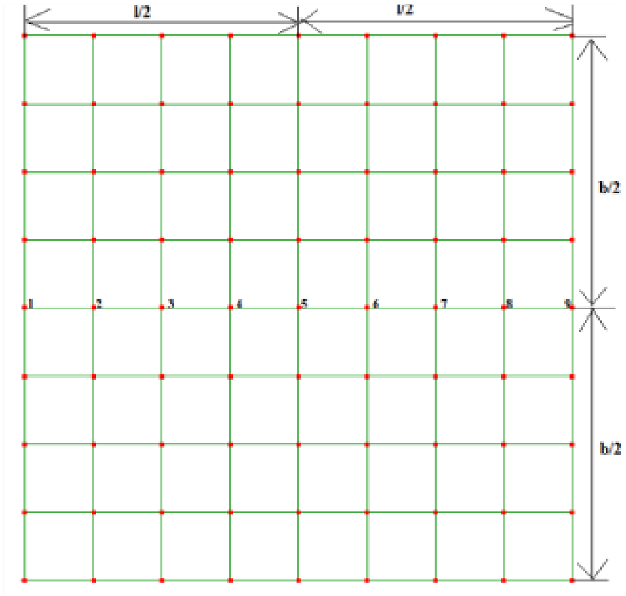

Figure 2

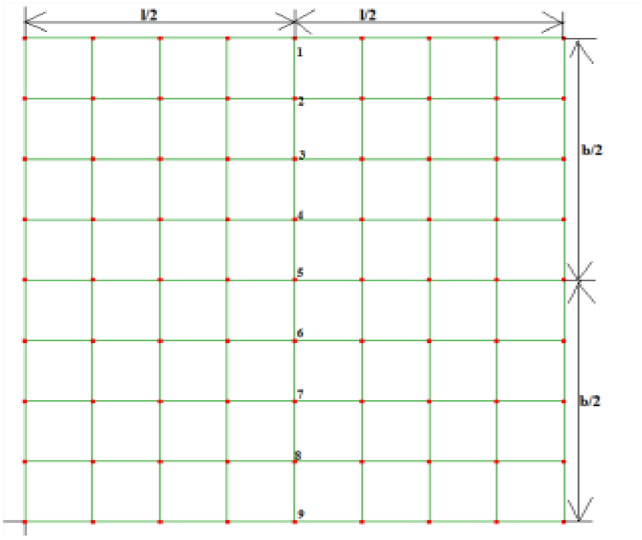

Figure 3

During the analysis of plates by FEM, the choice of the number and the kind of element used is only a matter of good judgment and experience. The number of finite elements in a portion of the solution region where the variable is being modelled (displacement) differs. An essential part of the discretization process is node identification and numbering. And the total of nodes should be orderly.

This is important because the finite element equation generated for each element takes cognizance of the node numbers around that element.

Step 2. Develop or write the MATLAB program for the plate.

```
% FiniteElementParkings1.m Program for Finite Element Analysis of
% Rectangular Plates subjected to Bending using Automatic Mesh Generation
% Data Programs
% FiniteElemMeshGen5a
% FiniteElemMeshGen5b
% FiniteElemMeshGen7a
% FiniteElemMeshGen7b
% FiniteElemMeshGen7c
% FiniteElemMeshGen7d
tic
FiniteElemMeshGen3aRebecca
% Ex = E;
% Ey = E;
% Vx = v;
% Vy = v;
% Gxy = E/(2*(1+v));
% Dx = Ex*t^3/(12*(1-Vx*Vy));
% Dy = Ey*t^3/(12*(1-Vx*Vy));
% D1 = Vx*Ey*t^3/(12*(1-Vx*Vy));
% Dxy = Gxy*t^3/12;
Ks=zeros(3*f,3*f);
K=zeros(12,12,s);
% k=zeros(12,12,s);
% ELEMENT STIFFNESS MATRIX CALCULATION
disp ('ELEMENT STIFFNESS MATRIX CALCULATION:');
for a1=1:s
%   FiniteElemStiffnessMatrix1
  p = a/b;
  s1 = 20*a^2*Dy + 8*b^2*Dxy;
  s2 = 15*a*b*D1;
  s3 = 20*b^2*Dx + 8*a^2*Dxy;
  s4 = 30*a*p*Dy + 15*b*D1 + 6*b*Dxy;
  s5 = 30*b*p^(-1)*Dy + 15*a*D1 + 6*a*Dxy;
  s6 = 60*p^(-2)*Dx + 60*p^2*Dy + 30*D1 + 84*Dxy;
  s7 = 10*a^2*Dy - 2*b^2*Dxy;
  s8 = -30*a*p*Dy - 6*b*Dxy;
  s9 = 10*b^2*Dx - 8*a^2*Dxy;
  s10 = 15*b*p^(-1)*Dx - 15*a*D1 - 6*a*Dxy;
  s11 = 30*p^(-2)*Dx - 60*p^2*Dy - 30*D1 - 84*Dxy;
  s12 = 10*a^2*Dy - 8*b^2*Dxy;
  s13 = -15*a*p*Dy + 15*b*D1 + 6*b*Dxy;
  s14 = 5*a^2*Dy + 2*b^2*Dxy;
  s15 = 15*a*p*Dy - 6*b*Dxy;
  s16 = 10*b^2*Dx - 2*a^2*Dxy;
  s17 = 30*b*p^(-1)*Dx + 6*a*Dxy;
  s18 = 5*b^2*Dx + 2*a^2*Dxy;
  s19 = 15*b*p^(-1)*Dx - 6*a*Dxy;
  s20 = -60*p^(-2)*Dx + 30*p^2*Dy - 30*D1 - 84*Dxy;
  s21 = -30*p^(-2)*Dx - 30*p^2*Dy + 30*D1 + 84*Dxy;
  k1 = [s1 -s2 -s4 s7 0 -s8
    -s2 s3 s5 0 s9 s10
    -s4 s5 s6 s8 s10 s11
    s7 0 s8 s1 s2 s4
```

```
       0 s9 s10 s2 s3 s5
       -s8 s10 s11 s4 s5 s6];
    k2 = [s12 0 s13 s14 0 s15
        0 s16 -s17 0 s18 -s19
        s13 s17 s20 -s15 s19 s21
        s14 0 -s15 s12 0 -s13
        0 s18 -s19 0 s16 -s17
        s15 s19 s21 -s13 s17 s20];
    k3 = [s1 s2 -s4 s7 0 -s8
        s2 s3 -s5 0 s9 -s10
        -s4 -s5 s6 s8 -s10 s11
        s7 0 s8 s1 -s2 s4
        0 s9 -s10 -s2 s3 -s5
        -s8 -s10 s11 s4 -s5 s6];
%    k(:,:,a1) = [k1 k2; k2' k3]*(15*a*b)^(-1);
   K(:,:,a1) = [k1 k2; k2' k3]/(15*a*b);
%    K = k(:,:,a1);
   N1 = Connectivity(a1,1);
   N2 = Connectivity(a1,2);
   N3 = Connectivity(a1,3);
   N4 = Connectivity(a1,4);
   X = [3*N1-2 3*N1-1 3*N1 3*N2-2 3*N2-1 3*N2 3*N3-2
3*N3-1 3*N3 3*N4-2 3*N4-1 3*N4];
   Ks(X,X) = Ks(X,X)+ K(:,:,a1);
%    Ks(X,X) = Ks(X,X)+ K;
   disp(Ks);
end
disp('FINAL GLOBAL STIFFNESS MATRIX IS:');
disp(Ks);
Fqs = zeros(12,1,s);
Fq = zeros(3*f,1);
for c3 = 1:length(sq) % Number of Finite Elements
considered for UDL (sq = s or 0)
%    q = SpecificGravity*t; % for self weight or specify the
value of q
%    q = 6; % for example
   N1 = Connectivity(sq(c3),1);
   N2 = Connectivity(sq(c3),2);
   N3 = Connectivity(sq(c3),3);
   N4 = Connectivity(sq(c3),4);
   Mx1 = -a*b^2*q/24;
   My1 = a^2*b*q/24;
   Fw1 = a*b*q/4;
   Mx2 = a*b^2*q/24;
   My2 = a^2*b*q/24;
   Fw2 = a*b*q/4;
   Mx3 = -a*b^2*q/24;
   My3 = -a^2*b*q/24;
   Fw3 = a*b*q/4;
   Mx4 = a*b^2*q/24;
   My4 = -a^2*b*q/24;
   Fw4 = a*b*q/4;
   Nq = [Mx1 My1 Fw1 Mx2 My2 Fw2 Mx3 My3 Fw3 Mx4
My4 Fw4];
%    disp('Element Load Vector due to Distributed Load q:
');
%    disp(Nq');
```

```
   Fqs(:,:,c3) = Nq';
   X1 = [3*N1-2 3*N1-1 3*N1 3*N2-2 3*N2-1 3*N2 3*N3-2
3*N3-1 3*N3 3*N4-2 3*N4-1 3*N4];
   Fq(X1) = Fq(X1) + Fqs(:,:,c3);
end
% disp('Net Global Load Vector due to Distributed Load : ');
% disp(Fq);
% disp('Global Load Vector due to Nodal Loads : ');
FnL = zeros(3*f,1); % Nodal Loads
% nL = Number of Nodes containing Nodal Loads
for a4 = 1:nL
   Nn = NodalNumber(a4); % Nodal Number of Node
containing Nodal Loads
   Mx = NodalLoad(a4,1);
   My = NodalLoad(a4,2);
   Fw = NodalLoad(a4,3);
   FnL(3*Nn-2,1) = FnL(3*Nn-2,1) + Mx;
   FnL(3*Nn-1,1) = FnL(3*Nn-1,1) + My;
   FnL(3*Nn,1) = FnL(3*Nn,1) + Fw;
end
% disp('Net Global Load Vector due to Nodal Loads : ');
% disp(FnL);
Fs = Fq + FnL;
% disp('Net Global Load Vector = ');
% disp(Fs);
% disp('CALCULATION OF DISPLACEMENTS:');
% disp('Put zero boundary conditions for displacements :');
Ds = zeros(3*f,1);
BCs = zeros(3*f,1);
for c5=1:f;
   Ns = Nodes(c5); %input('Give nodal number : ');
   U = BoundaryCondition(c5,1); %input('Type [0] for no
Translation in the Global X Direction else Type [1] : ');
   V = BoundaryCondition(c5,2); %input('Type [0] for no
Translation in the Global Y Direction else Type [1] : ');
   W = BoundaryCondition(c5,3); %input('Type [0] for no
Translation in the Global Z Direction else Type [1] : ');
   BCs(3*Ns-2,1) = U;
   BCs(3*Ns-1,1) = V;
   BCs(3*Ns,1) = W;
end
Dr = find(BCs);
Kr = Ks(Dr,Dr);
disp('Reduced Structure Stiffness Matrix is:');
disp(Kr);
Fr = Fs(Dr,1);
% disp('Reduced Force Vector is:');
% disp(Fr);
% Ds(Dr,1) = Kr^(-1)*Fr;
Ds(Dr,1) = Kr\Fr;
% disp('DISPLACEMENT MATRIX IS OBTAINED AS :');
% disp(Ds);
% disp('CALCULATION OF MEMBER FORCES');
% % POST-PROCESSING
% % Calculation of STRAIN B and ELASTICITY MATRIX D
% % A = Inverse Matrix of C
A = [0 0 1 0 0 0 0 0 0 0 0 0
```

```
  0 1 0 0 0 0 0 0 0 0 0 0
  -1 0 0 0 0 0 0 0 0 0 0 0
  0 -2/a -3/a^2 0 0 0 0 -1/a 3/a^2 0 0 0
  1/a -1/b -1/(a*b) 0 1/b 1/(a*b) -1/a 0 1/(a*b) 0 0 -
1/(a*b)
  2/b 0 -3/b^2 1/b 0 3/b^2 0 0 0 0 0 0
  0 1/a^2 2/a^3 0 0 0 0 1/a^2 -2/a^3 0 0 0
  0 2/(a*b) 3/(a^2*b) 0 -2/(a*b) -3/(a^2*b) 0 1/(a*b) -
3/(a^2*b) 0 -1/(a*b) 3/(a^2*b)
  -2/(a*b) 0 3/(a*b^2) -1/(a*b) 0 -3/(a*b^2) 2/(a*b) 0 -
3/(a*b^2) 1/(a*b) 0 3/(a*b^2)
  -1/b^2 0 2/b^3 -1/b^2 0 -2/b^3 0 0 0 0 0 0
  0 -1/(a^2*b) -2/(a^3*b) 0 1/(a^2*b) 2/(a^3*b) 0 -
1/(a^2*b) 2/(a^3*b) 0 1/(a^2*b) -2/(a^3*b)
  1/(a*b^2) 0 -2/(a*b^3) 1/(a*b^2) 0 2/(a*b^3) -1/(a*b^2)
0 2/(a*b^3) -1/(a*b^2) 0 -2/(a*b^3)];
% disp('Inverse Matrix of C : ');
% disp(A);
syms x y
Wxy = [1 x y x^2 x*y y^2 x^3 x^2*y x*y^2 y^3 x^3*y x*y^3];
B1 = -[diff(Wxy,x,2); diff(Wxy,y,2); -2*diff(diff(Wxy,x),y)]*A;
D = [Dx D1 0; D1 Dy 0; 0 0 Dxy]; % D is Elasticity Matrix
% % CALCULATION OF ELEMENT FORCES IN LOCAL
COORDINATES
for Pannel = 1:Pannels
  for c6 = 1:Elements % Number of Finite Elements in a
Pannel for Post-Processing
    N11 = RespElemNumber(Pannel,c6);
    N1 = Connectivity(N11,1);
    N2 = Connectivity(N11,2);
    N3 = Connectivity(N11,3);
    N4 = Connectivity(N11,4);
    X2 = [3*N1-2 3*N1-1 3*N1 3*N2-2 3*N2-1 3*N2 3*N3-
2 3*N3-1 3*N3 3*N4-2 3*N4-1 3*N4];
    ElemDispl = Ds(X2,1);
    fprintf('Panel %d \n' , Pannel);
    fprintf('ELEMENT NUMBER %d \n' ,N11);
%       for c7 = 1:Ntp % Number of Interpoltion Points in the
Element
    x1 = ElemInterpolCoord(c6,1);
    y1 = ElemInterpolCoord(c6,2);
    B = subs(subs(B1,x,x1),y,y1); % B is the Strain Matrix
    Stresses = (D*B*Ds(X2,1))';
    NodalDisplacements = [ElemDispl(3*c6-2)
ElemDispl(3*c6-1) ElemDispl(3*c6)];
    fprintf('Stresses at Node %d \n' , c6);
    disp('Mx, My, Mxy : ');
    disp(Stresses);
    fprintf('Displacements at Node %d \n' , c6);
    disp('Rotation about X, Rotation about Y, Deflection :
');
    disp(NodalDisplacements);
  end
end
% Special Response
% Twisting and Bending Moments at Node 1 of First
Corners of Pannels
for c8 = 1:Pannels
  N22 = FirstCornerElemNumber(c8);
  N1 = Connectivity(N22,1);
  N2 = Connectivity(N22,2);
  N3 = Connectivity(N22,3);
  N4 = Connectivity(N22,4);
  X3 = [3*N1-2 3*N1-1 3*N1 3*N2-2 3*N2-1 3*N2 3*N3-2
3*N3-1 3*N3 3*N4-2 3*N4-1 3*N4];
  ElemDispl = Ds(X3,1);
  x1 = 0;
  y1 = 0;
  B = subs(subs(B1,x,x1),y,y1); % B is the Strain Matrix
  Stresses = (D*B*Ds(X3,1))';
  fprintf('ELEMENT NUMBER %d \n' ,N22);
  NodalDisplacements = [ElemDispl(1) ElemDispl(2)
ElemDispl(3)];
  fprintf('Stresses at Node 1 of Panel %d \n' , c8);
  disp('Mx, My, Mxy : ');
  disp(Stresses);
  fprintf('Displacements at Node 1 of Pannel %d \n' , c8);
  disp('Rotation about X, Rotation about Y, Deflection : ');
  disp(NodalDisplacements);
end
toc
```

Step 3. Run program for result. After all the essential information, click on the run menu to run the program for the result.

Step 4. Compare the result with the finite strip method and the exact method. After getting the results, compare the result with Timoshenko's exact method [40] and Cheung's finite strip method [4].

Step 5. Solve for a plate with different widths and lengths. Assume the width of a plate is four and the length is also four for each of the spans. Assume also the value of q in the *UDL* to be 10000 N/m^2, with thickness as 0.2; Young's modulus E=30×10^6 N/m^2, Poison's ratio v=0.2 and $D=Et^3/(12(1-v^2)$.

RESULTS AND DISCUSSION

Finite element analysis for thin plates continuous over three spans. This section of the research work has solved a thin rectangular plate continuous over three spans using the limited element software program developed by the authors. The scale is simply supported on all edges, including the intermediate lines.

Plate properties: S – Number of finite element; W – Width of plate; a=1; L – Length of plate = 3a = 1; E – Young's modulus = 30×10^6 N/m^2; V – Poison's ratio = 0.2; t – thickness = 0.2; q – Uniformly distributed load (UDL) = 10000 N/m^2.

$$D=\frac{Et^3}{12(1-v^2)}=\frac{30\times10^6\times0.2^3}{12(1-0.2^2)}=20833.33\ N/m^2$$

The analysis has been done for: a plate with width and length equals one; a plate with width and length equals four with the same thickness.

Results for a plate with width and length equal to one (Table 1). The results from this research were compared to [40, p. 231; 4, pp. 87–88].

Result for plate with width and length equals to four (Table 2). Consider a continuous plate with three equal spans of width = 4 and length = 4 using the same thickness of 0.2; q=10000 N/m^2; D=20833.33 N/m^2 ; a = 4.

Table 1 – Refined Methods – Finite element answer, and the answers also corresponds with that of finite strip method and exact method

m/n	Number of finite elements	Deflection of the middle of the panel 1, WA $(\frac{qa4}{D})$	Deflection of middle of panel 2, W	Longitudinal moment of first interior support, M_yB (qa^2)	Transverse moment of panel 2, M_xC	Longitudinal moment of panel 2, M_yC
5	48	-0.0719	0.0036	-6112	5152	6064
7	108	-0.0719	0.0036	-6112	5152	6064
9	192	-0.0719	0.0036	-6112	5152	6064
11	300	-0.0720	0.0036	-6112	5168	6096
13	432	-0.0719	0.0036	-6112	5152	6064
15	588	-0.0718	0.0036	-6112	5136	6048

Table 2 – Finite element method for the continuous plate

m/n	Number of finite elements	Deflection of the middle of the panel 1, WA $(\frac{qa4}{D})$	Deflection of middle of panel 2, W	Longitudinal moment of first interior support, M_yB (qa^2)	Transverse moment of panel 2, M_xC	Longitudinal moment of panel 2, M_yC
5	48	-0.0585	0.0029	-0.0382	0.0322	0.0379
7	108	-0.0585	0.0029	-0.0382	0.0322	0.0379
9	192	-0.0585	0.0029	-0.0382	0.0322	0.0379
11	300	-0.0586	0.0029	-0.0382	0.0323	0.0381
13	432	-0.0585	0.0029	-0.0382	0.0322	0.0379
15	588	-0.0584	0.0029	-0.0372	0.0317	0.0378
Finite strip method [4]		-0.0583	0.0029	-0.0372	0.0319	0.0373
Exact method [40]		-00583	0.0029	-0.0381	0.0317	0.0375

CONCLUSIONS

The results from MATLAB for continuous plates exhibit a high degree of correlation with the values of [40] and [4]. The analysis shows that the accuracy of results obtained for the plate in various finite elements had negligible differences.

Computerization of plate analysis by numerical methods (finite element method) presents the difficulty of computer memory management and ample execution time. It was observed that the execution time for various element software would increase as the number of finite elements. A new area of research has arisen recently, which would help tackle the problem of high execution time and efficient memory usage during program execution. Given the trouble of execution time, it is advised that a computer of higher memory capacity be used at least a quad-processor (4 GB RAM and above) for calculations to generate the desired solution.

REFERENCES

1. Belounar, L., & Guenfoud, M. (2005). A new rectangular finite element based on the strain approach for plate bending. *Thin-Walled Structures, 43*(1), 47–63. doi: 10.1016/j.tws.2004.08.003
2. Bernoulli, J. (1789). Essai theorique sur les vibrations de plaques elastiques rectangularies et libers. *Novi Commentari Acad Petropolit, 5*, 197–219
3. Cauchy, A. (2009). *Oeuvres complétes: Sur l'équilibre et le mouvement d'une plaque solide*. doi: 10.1017/CBO9780511702679.014
4. Cheung, M. S., Chidiac, S. E., & Li, W. (1996). *Finite Strip Analysis of Bridges*. London: CRC Press.
5. Clebsch, A., Flamant, A. A., & de Saint-Venant, A. C. (1883). *Théorie de l'élasticité des corps solides*. Paris: Dunod.
6. Clough, R. W. (1960). *The Finite Element Method in Plane Stress Analysis*. Retrieved from https://www.semanticscholar.org/paper/The-Finite-Element-Method-in-Plane-Stress-Analysis-Clough/035536cf1b0157b3cc7a6a19ed1b66638b388553
7. Cole, V. (2006, January 25). *A Critical History of Computer Graphics and Animation*. Retrieved from https://www.engadget.com/2006-01-25-a-critical-history-of-computer-graphics-and-animation.html
8. Cook, R. D. (2002). *Concepts and applications of finite element analysis*. New York: Wiley.
9. Courant, R. (1943). Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society, 49*, 1–23.
10. Dawe, D. J. (1984). *Matrix and finite element displacement analysis of structures*. Oxford: Claredon Press.
11. Düster, A., & Rank, E. (2001). The p-version of the finite element method compared to an adaptive h-version for the deformation theory of plasticity. *Computer Methods in Applied Mechanics and Engineering, 190*(15–17), 1925–1935. doi: 10.1016/s0045-7825(00)00215-2
12. Euler, L. (1776). *De motu vibratorio tympanorum*. Retrieved from https://scholarlycommons.pacific.edu/cgi/viewcontent.cgi?article=1301&context=euler-works
13. Germain, S. (1826). *Remarques sur la nature, les bornes et l'étendue de la question des surfaces élastiques et équation générale de ces surfaces*. Paris.
14. Han, J.-G., Ren, W.-X., & Huang, Y. (2007). A wavelet-based stochastic finite element method of thin plate bending. *Applied Mathematical Modelling, 31*(2), 181–193. doi: 10.1016/j.apm.2005.08.020
15. Hastings, J. K., Judes, M. A., & Brauer, J. R. (1985). Accuracy and Economy of Finite Element Magnetic Analysis. *33rd Annual National Relay Conference*.
16. Hrennikoff, A. (1941). Solution of Problems of Elasticity by the Framework Method. *Journal of Applied Mechanics, 8*(4), A169–A175. doi: 10.1115/1.4009129
17. Kelvin, W. T., & Tait, P. G. (2013). *Treatise on Natural Philosophy* (Vol. 1). Oxford: Clarendon Press.
18. Kirchhoff, G. (1985). Über das Gleichgewicht und die Bewegung einer elastischen Scheibe. *Journal Für Die Reine Und Angewandte Mathematik (Crelles Journal), 40*, 51–88. doi: 10.1515/crll.1850.40.51
19. Kreyszig, E. (2011). *Advanced Engineering Mathematics* (10th ed.). Retrieved from https://www.bau.edu.jo/UserPortal/UserProfile/PostsAttach/59003_3812_1.pdf
20. Krishnamoorthy, C. S. (2011). *Finite element analysis theory and programming* (2nd ed.). New Delhi: Tata McGraw Hill Education Private Limited.
21. Levy, M. (1877). Mémoire sur la théorie des plaques élastiques planes. *Journal de mathématiques pures et appliquées, 3*(3), 219–306.

22. Lim, G. T., & Reddy, J. N. (2003). On canonical bending relationships for plates. *International Journal of Solids and Structures, 40*(12), 3039–3067. doi: 10.1016/s0020-7683(03)00084-2

23. Mathew, J., Ma, L., Tan, A., Weijnen, M., & Lee, J. (Eds.). (2012). *Engineering Asset Management and Infrastructure Sustainability*. doi: 10.1007/978-0-85729-493-7

24. Melosh, R. J. (1963). Basis for derivation of matrices for the direct stiffness method. *AIAA Journal, 1*(7), 1631–1637. doi: 10.2514/3.1869

25. Nwagozie, I. L. (2008, January). *Finite Element Modelling Of Engineering Systems*. Retrieved from https://www.researchgate.net/publication/297739827_FINITE_ELEMENT_MODELLING_OF_ENGINEERING_SYSTEMS_With_Emphasis_in_Water_Resources

26. Poisson, S. D. (1828). *Mémoire sur l'équilibre et le mouvement des corps élastiques*. N. d.

27. Reddy, J. N. (2006). *Theory and Analysis of Elastic Plates and Shells*. doi: 10.1201/9780849384165

28. Reissner, E., & Stein, M. (1951). *Torsion and transverse bending of cantilever plates*. Retrieved from https://ntrs.nasa.gov/citations/19930090894

29. Rockey, K. C. (1974). *The Finite Element method: A Basic Introduction*. London: Collins professional and Technical Book.

30. Ross, C. (1996). *Finite Element Techniques in Structural Mechanics*. London: Woodhead Publishing.

31. Saibel, E., & Tadjbakhsh, I. (1960). Large deflections of circular plates under uniform and concentrated central loads. *Zeitschrift Für Angewandte Mathematik Und Physik ZAMP, 11*(6), 496–503. doi: 10.1007/bf01595401

32. Segerlind, L. (1985). *Applied Finite Element Analysis* (2nd ed.). New York: Wiley.

33. Semie, A. G. (2010, June). *Numerical Modelling of Thin Plates using the Finite Element Method*. Retrieved from http://etd.aau.edu.et/bitstream/handle/123456789/7199/Addisu%20Gezahegn.pdf?sequence=1&isAllowed=y

34. Shames, I. H., & Dym, C. L. (2017). *Energy and Finite Element Methods in Structural Mechanics*. doi: 10.1201/9780203757567

35. Shudhir, N. (2012). *Plate Bending Analaysis Using Finite Element Method*. Retrieved from http://ethesis.nitrkl.ac.in/3303/1/108ME015.pdf

36. Speare, P. R. S., & Kemp, K. O. (1977). A simplified reissner theory for plate bending. *International Journal of Solids and Structures, 13*(11), 1073–1079. doi: 10.1016/0020-7683(77)90077-4

37. Strang, G., & Fix, G. (2008). *An Analysis of the Finite Element Method* (2nd ed.). Cambridge: Wellesley-Cambridge Press.

38. Szabó, B., & Actis, R. (2011). *Simulation governance: New technical requirements for software tools in computational solid mechanics*. Retrieved from https://www3.nd.edu/~powers/vv.presentations/szabo.pdf

39. Thompson, E. G. (2004). *Introduction to the Finite Element Method: Theory, Programming and Applications*. New York: Wiley.

40. Timoshenko, S. P., & Woinowsky-Kerieger, S. (1959). *Theory of Plates and Shells* (2nd ed.). New York: McGraw-Hill Book Co.

41. Topping, B. H. V. (Ed.). (2003). *Developments in Structural Engineering*. doi: 10.1201/9781482298567

42. Turner, M. J., Clough, R. W., Martin, H. C., & Topp, L. J. (1956). Stiffness and Deflection Analysis of Complex Structures. *Journal of the Aeronautical Sciences, 23*(9), 805–823. doi: 10.2514/8.3664

43. Ventsel, E., & Krauthammer, T. (2001). *Thin Plates and Shells*. doi: 10.1201/9780203908723

44. Williams, M. S., & Todd. J. D. (2000). *Structures: Theory and analysis*. London: Red Globe Press.

45. Zienkiewicz, O. C., Taylor, R. L., & Zhu, J. Z. (2013). *The finite element method: its basis and fundamentals* (7th ed.). Oxford: Butterworth-Heinemann.