

DEVELOPMENT OF A QUASI-EXACT DYNAMIC FINITE ELEMENT (QDFE) METHOD FOR THE FREE VIBRATION ANALYSIS OF THIN RECTANGULAR MULTILAYERED PLATES

by:

Heenkenda Jayasinghe

Bachelor of Engineering, City University London (2011)

Master of Applied Science, Ryerson University (2013)

A dissertation

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the program of

Aerospace Engineering

Toronto, Ontario, Canada, 2019

©Heenkenda Jayasinghe 2019

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A DISSERTATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the dissertation, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

DEVELOPMENT OF A QUASI-EXACT DYNAMIC FINITE ELEMENT (QDFE) METHOD FOR THE FREE VIBRATION ANALYSIS OF THIN RECTANGULAR MULTILAYERED PLATES

Heenkenda Jayasinghe

Doctor of Philosophy, Aerospace Engineering, Ryerson University, Toronto (2019)

ABSTRACT

The Dynamic Finite Element (DFE) method is a well-established superconvergent semi-analytical method that has been used in the past to investigate the vibration behaviour of various beam-structures. Considered as a viable alternative to conventional FEM for preliminary stage modal analysis, the DFE method has consistently proven that it is capable of producing highly accurate results with a very coarse mesh; a feature that is attributed to the fact that the DFE method uses trigonometric, frequency-dependant shape functions that are based on the exact solution to the governing differential equation as opposed to the polynomial shape functions used in conventional FEM. In the past many researchers have contributed towards building a comprehensive library of DFE models for various line structural elements and configurations, which would serve as the building blocks that would help the DFE method evolve into a full-fledged, versatile tool like conventional FEM in the future. However, thus far a DFE formulation has not been developed for plate problems. Therefore, in this thesis an effort has been made for the first time to develop a DFE formulation for the realm of two-dimensional structural problems by formulating a Quasi-Exact Dynamic Finite Element (QDFE) solution to investigate the free vibration behaviour of thin single- and multi-layered, rectangular plates. As a starting point for this work, Hamiltonian mechanics and the Classical Plate Theory (CPT) are used to develop the governing differential equation for thin plates. Subsequently, a unique quasi-exact solution to the governing equation is sought by following a distinct procedure that, to the best of the author's knowledge, has never been presented before. Through this procedure, the characteristic equation is re-arranged as the sum of two beam-like expressions and then solved for by applying the quadratic formula. The resulting quasi-exact roots are then exploited to form the trigonometric basis functions, which in turn are used to derive the frequency-dependant shape

functions; the characteristic feature of the QDFE method. Once developed, the new QDFE technique is applied to determine the vibration behaviour of thin, isotropic, linearly elastic, rectangular, homogenous plates. Subsequently, it is also employed to formulate a Simplified Layerwise Quasi-Exact Dynamic Finite Element solution for the free vibration of thin, rectangular multilayered plates. In addition, the quasi-exact solution to the plate equation is also utilised to develop a Dynamic Coefficient Matrix (DCM) method to investigate the vibrational characteristics of thin, rectangular, homogeneous plates and thin, rectangular, multilayered plates. The Method of Homogenization is used as an alternative procedure to validate the results from the Simplified Layerwise Quasi-Exact Dynamic Finite Element method and the Simplified Layerwise Dynamic Coefficient Matrix method. The results from both the QDFE and DCM methods are, in general, verified for accuracy against the exact results existing in the open literature and those produced by two in-house developed conventional FEM codes and/or ANSYS® software.

ACKNOWLEDGEMENT

This thesis would not have been possible without the continuous support, guidance and encouragement of Prof. Seyed M. Hashemi, whose extensive knowledge and experience in the areas of Conventional and Dynamic Finite Elements, and Vibration was instrumental.

I would also like to thank Ryerson University for the numerous scholarships and awards bestowed upon me such as the Ryerson Graduate Scholarship (RGS), Ryerson Graduate Award (RGA), Aerospace Engineering Graduate Students Research Excellence Award (AGSREA) and Aerospace Engineering Graduate Students Research Support (AGSRS). My sincere thanks also go to the Government of Ontario for providing me with the Ontario Graduate Scholarship twice consecutively.

I would also like to thank my parents Gamini and Samadara for the tremendous love and support they have provided throughout this journey.

Special thanks go to my loving spouse Thilini, whose unwavering love, caring, understanding and patience has made this thesis a reality.

Last but not the least, my thanks also go to my siblings Amanda and Kusal for their constant encouragement.

Dedicated to my parents Gamini and Samadara and my loving spouse Thilini

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	x
LIST OF APPENDICES	xiii
NOMENCLATURE.....	xiv
1. INTRODUCTION.....	1
1.1 Overview	1
1.2 Governing Theories.....	2
1.3 Methods of Vibration Analysis	4
1.4 Research Motivations.....	14
1.5 Key Objectives and Methodology.....	14
1.6 Analysis Approach and Limitations.....	16
1.7 Thesis Organization	21
2. THEORETICAL BACKGROUND.....	23
2.1 Development of Governing Differential Equation.....	23
2.2 A Quasi-Exact Solution for the Thin Plate Equation	30
2.3 New Trigonometric, Frequency-Dependant Basis and Shape Functions.....	37
3. FREE VIBRATION OF A RECTANGULAR HOMOGENEOUS PLATE	44
3.1 Conventional Finite Element Method (FEM)	44
3.1.1 Finite Element Analysis with 12–DOF Rectangular Element	45
3.1.2 Finite Element Analysis with 16 – DOF Rectangular Element	54
3.2 Quasi-Exact Dynamic Finite Element Method (QDFE)	59
3.3 Dynamic Coefficient Matrix Method (DCM)	66
3.4 Numerical Results and Discussion.....	69

4.	FREE VIBRATION OF A RECTANGULAR MULTILAYER PLATE.....	83
4.1	Method of Homogenization	83
4.2	Simplistic Layerwise Conventional Finite Element Method	84
4.3	Simplistic Layerwise Quasi-Exact Dynamic Finite Element Method	88
4.4	Simplistic Layerwise Quasi-Exact Dynamic Coefficient Matrix Method	94
4.5	Numerical Results and Discussion.....	96
5.	VIBRATION BEHAVIOUR OF AN AIRCRAFT CABIN WINDOW BAY.....	102
5.1	Quasi-Exact Dynamic Finite Element Analysis of Cabin Window Bay Sections	102
5.2	Time Verification for the Quasi-Exact Dynamic Finite Element Method.....	116
6.	CONCLUDING REMARKS.....	119
6.1	Synopsis	119
6.2	Key Contributions	121
6.3	Impact of Research.....	122
6.4	Recommendations for Future Work.....	123
6.5	List of Publications	124
	APPENDICES.....	127
	Appendix A: Trigonometric, Frequency Dependant DFE Shape Functions	127
	Appendix B: QDFE Solution for a Simply Supported (S-S-S-S) Plate	136
	Appendix C: DCM Solution for a Simply Supported (S-S-S-S) Plate	175
	REFERENCES	181

LIST OF TABLES

Table 1: Trigonometric basis functions for thin plate.....	37
Table 2: Comparison of 12-DOF FEM and 16-DOF FEM results with ANSYS® simulation.....	72
Table 3: Natural frequencies for a homogeneous Steel plate with C-F-F-F B.C.....	77
Table 4: Natural frequencies for a homogeneous Steel plate with C-C-F-F B.C.	77
Table 5: Natural frequencies for a homogeneous Steel plate with C-F-C-F B.C.	77
Table 6: Natural frequencies for a homogeneous Steel plate with C-C-C-F B.C.....	78
Table 7: Natural frequencies for a homogeneous Steel plate with C-C-C-C B.C.	78
Table 8: Natural frequencies for a homogeneous Steel plate with C-C-C-S B.C.....	78
Table 9: Natural frequencies for a homogeneous Steel plate with C-C-S-S B.C.	79
Table 10: Natural frequencies for a homogeneous Steel plate with S-C-S-C B.C.	79
Table 11: Natural frequencies for a homogeneous Steel plate with S-F-S-F B.C.	79
Table 12: Natural frequencies for a homogeneous Steel plate with S-S-S-F B.C.	80
Table 13: Natural frequencies for a homogeneous Steel plate with S-S-S-SB.C.	80
Table 14: Comparison of 16-DOF layerwise FEM results with ANSYS® simulation.....	96
Table 15: Natural frequencies for a multilayered Al-St plate with C-F-F-FB.C.....	97
Table 16: Natural frequencies for a multilayered Al-St plate with C-C-F-F B.C.....	97
Table 17: Natural frequencies for a multilayered Al-St plate with C-F-C-F B.C.....	97
Table 18: Natural frequencies for a multilayered Al-St plate with C-C-C-F B.C.	98
Table 19: Natural frequencies for a multilayered Al-St plate with C-C-C-CB.C.....	98
Table 20: Natural frequencies for a multilayered Al-St plate with C-C-C-S B.C.	98
Table 21: Natural frequencies for a multilayered Al-St plate with C-C-S-S B.C.....	99
Table 22: Natural frequencies for a multilayered Al-St plate with S-C-S-CB.C.....	99
Table 23: Natural frequencies for a multilayered Al-St plate with S-F-S-F B.C.	99
Table 24: Natural frequencies for a multilayered Al-St plate with S-S-S-F B.C.	100
Table 25: Natural frequencies for a multilayered Al-St plate with S-S-S-S B.C.	100
Table 26: Natural frequency results for aircraft cabin window bay section – configuration 1 ..	105
Table 27: Natural frequency results for aircraft cabin window bay section – configuration 2 ..	106
Table 28: Natural frequency results for aircraft cabin window bay section – configuration 3 ..	108
Table 29: Natural frequency results for aircraft cabin window bay section – configuration 4 ..	109

Table 30: Natural frequency results for aircraft cabin window bay section – configuration 5 ..	111
Table 31: Natural frequency results for aircraft cabin window bay section – configuration 6 ..	114
Table 32: MATLAB [®] program flow chart and sample results for S-S-S-S plate using QDFE..	136
Table 33: MATLAB [®] program flow chart and sample results for S-S-S-S plate using DCM...	175

LIST OF FIGURES

Figure 1: (a) A thin plate of thickness, h , with infinite width and length. (b) An infinitesimal element of the thin plate subjected to bending and twisting moments, shear forces and distributed load per unit length [116].....	23
Figure 2: (a) An infinitesimal differential element with a lamina shown by the shaded region marked as a-b-c-d. (b) A cross section view of the differential element and lamina in bending. (c). A top view of the lamina subjected to shear loading. [116].....	26
Figure 3: Thin plate used for vibration analysis	44
Figure 4: 4-Node, 3-DOF per node rectangular plate element.....	45
Figure 5: Conforming four node rectangular element with 16 – DOF ensuring C^1 continuity	54
Figure 6: Convergence analysis for the 5 th natural frequency of a C-C-C-F plate	69
Figure 7: (a) ANSYS [®] SHELL 181 element used for meshing (b) Plate meshed in ANSYS [®]	70
Figure 8: Notation for the classical boundary conditions investigated.....	71
Figure 9: QDFE (left) and FEM (right) shape function for w_1 at node 1	73
Figure 10: QDFE (left) and FEM (right) shape function for θ_{x1} at node 1	73
Figure 11: QDFE (left) and FEM (right) shape function for θ_{y1} at node 1	74
Figure 12: QDFE (left) and FEM (right) shape function for θ_{xy1} at node 1.....	74
Figure 13: Shape function N_{1f} at ω_1 (left) and ω_5 (right).....	75
Figure 14: Shape function N_{2f} at ω_1 (left) and ω_5 (right).....	75
Figure 15: Shape function N_{3f} at ω_1 (left) and ω_5 (right).....	76
Figure 16: Shape function N_{4f} at ω_1 (left) and ω_5 (right).....	76
Figure 17: Two-layer Aluminium-Steel plate.....	83
Figure 18: (a) 16-DOF FEM mesh- 400 elems., (b) QDFE mesh- 8 elems. and (c) ANSYS [®] benchmark mesh-2877 elems. for cabin window bay Configuration 1	103
Figure 19: Convergence analyses for the benchmark data produced on ANSYS [®]	104
Figure 20: Convergence analyses for the in-house 16-DOF FEM method	104
Figure 21: (a) 16-DOF FEM mesh- 512 elems., (b) QDFE mesh- 8 elems. for Configuration 2.....	106
Figure 22: (a) 16-DOF FEM mesh- 864 elems., (b) QDFE mesh- 8 elems., for Configuration 3... ..	107

Figure 23: (a) 16-DOF FEM mesh- 320 elems., (b) QDFE mesh- 8 elems., for Configuration 4...	109
Figure 24: (a) 16-DOF FEM mesh- 448 elems., (b) QDFE mesh- 8 elems., for Configuration 5...	110
Figure 25: (a) 16-DOF FEM mesh- 608 elems., (b) QDFE mesh- 24 elems., for Configuration 6...	113
Figure 26: Element connectivity between thick and thin elements surrounding the cut out for configuration 6 of the cabin window bay section	114
Figure 27: CPU time to determine the first natural frequency (a) CPU time to determine the first five natural frequencies (b)	117
Figure 28: DFE (left) and FEM (right) shape function for w_2 at node 2	127
Figure 29: DFE (left) and FEM (right) shape function for θ_{x2} at node 2.....	127
Figure 30: DFE (left) and FEM (right) shape function for θ_{y2} at node 2.....	128
Figure 31: DFE (left) and FEM (right) shape function for θ_{xy2} at node 2.....	128
Figure 32: DFE (left) and FEM (right) shape function for w_3 at node 3	128
Figure 33: DFE (left) and FEM (right) shape function for θ_{x3} at node 3.....	129
Figure 34: DFE (left) and FEM (right) shape function for θ_{y3} at node 3.....	129
Figure 35: DFE (left) and FEM (right) shape function for θ_{xy3} at node 3.....	129
Figure 36: DFE (left) and FEM (right) shape function for w_4 at node 4	130
Figure 37: DFE (left) and FEM (right) shape function for θ_{x4} at node 4.....	130
Figure 38: DFE (left) and FEM (right) shape function for θ_{y4} at node 4.....	130
Figure 39: DFE (left) and FEM (right) shape function for θ_{xy4} at node 4.....	131
Figure 40: Shape function N_{5f} at ω_1 (left) and ω_5 (right)	132
Figure 41: Shape function N_{6f} at ω_1 (left) and ω_5 (right)	132
Figure 42: Shape function N_{7f} at ω_1 (left) and ω_5 (right)	132
Figure 43: Shape function N_{8f} at ω_1 (left) and ω_5 (right)	133
Figure 44: Shape function N_{9f} at ω_1 (left) and ω_5 (right)	133
Figure 45: Shape function N_{10f} at ω_1 (left) and ω_5 (right).....	133
Figure 46: Shape function N_{11f} at ω_1 (left) and ω_5 (right).....	134
Figure 47: Shape function N_{12f} at ω_1 (left) and ω_5 (right).....	134

Figure 48: Shape function N_{13f} at ω_1 (left) and ω_5 (right).....	134
Figure 49: Shape function N_{14f} at ω_1 (left) and ω_5 (right).....	135
Figure 50: Shape function N_{15f} at ω_1 (left) and ω_5 (right).....	135
Figure 51: Shape function N_{16f} at ω_1 (left) and ω_5 (right).....	135

LIST OF APPENDICES

Appendix A: Trigonometric, Frequency Dependant QDFE Shape Functions.....	127
Appendix B: QDFE Solution.....	136
Appendix C: DCM Solution.....	175

NOMENCLATURE

AR	Aspect ratio
b	Basis functions
D	Flexural rigidity of plate
E	Young's modulus
$\{E_{ij}\}$	Vector of unknown coefficients
G	Shear modulus
h	Thickness of plate
$[K]$	Global stiffness matrix
$[K_{DS}]$	Dynamic global stiffness matrix
$[K]_M$	Global stiffness matrix for multilayered plate
$[k]$	Element stiffness matrix
$[k_{DS}]$	Dynamic element stiffness matrix
k_1, k_2	Mass distributions constants
L	Length of the plate
M_x	Bending moment in x directions
M_y	Bending moment in y directions
M_{xy}, M_{yx}	Twisting moment in xy plane
$[M]$	Global mass matrix
$[M]_M$	Global mass matrix for multilayered plate

$[M_{DS}]$	Dynamic global mass matrix
$[m]$	Element mass matrix
$[m_{DS}]$	Dynamic element mass matrix
$\{N\}$	Shape functions
$N1, N2, N3$	Neutral axes
Q_x	Shear force in x direction
Q_y	Shear force in y direction
q	Distributed load
r_x	Radius of curvature in xz plane
r_y	Radius of curvature in yz plane
t	Time
U	Displacement component of a particle in x direction
V	Displacement component of a particle in y direction
V_{AL}	Volume fraction of Aluminium
V_{ST}	Volume fraction of Steel
W	Amplitude of vibration
\overline{W}_{EXT}	External virtual work
\overline{W}_f	Virtual work due to bending
\overline{W}_{INT}	Internal virtual work
W_n	Vector of the degrees of freedom

W'	Width of the plate
w	Bending displacement
x	Spatial coordinate in x direction
y	Spatial coordinate in y direction
$\alpha_x, \alpha_y, \beta_x, \beta_y$	Quasi-exact roots of the characteristic equation
δ_e	Element displacement
δW	Vector of the virtual degrees of freedom
ξ	Natural coordinates in x direction
$\varepsilon_x, \varepsilon_y$	Normal strains
η	Natural coordinates in y direction
θ_x	Rotation along the x-axis
θ_y	Rotation along the y-axis
θ_{xy}	Curvature of the xy plane
ρ	Mass density
ϕ_x	Rotation of lamina in x-direction
ϕ_y	Rotation of lamina in y-direction
σ_x, σ_y	Normal stresses
$\tau_{xy}, \tau_{yx}, \tau_{xz}, \tau_{yz}$	Shear stresses
ν	Poisson's ratio
ω	Frequency of vibration

γ_{xy}

Shear strain

1. INTRODUCTION

1.1 Overview

A plate can be considered as the two-dimensional extension of a beam in simple bending. Both beams and plates support transverse loads, perpendicular to their plane and through bending action. A plate is flat and while a beam has a single bending moment resistance, a plate resists bending about two axes and has a twisting moment.

Plates are versatile structural elements with diverse applications in thin-walled structures and many components of an aircraft such as the low curvature wing and fuselage skins, for example, are modeled as plates during the preliminary design stages. As the fuselage or wing skin envelopes the structural skeleton of the aircraft giving its streamlined shape, it is subjected to various loading conditions such as tension, compression and torsion since these skins aid in transmitting internal loads. Each of these loading conditions, individually or combined, affect the vibrational characteristics of the skins and as such structural elements are exposed to a large range of vibrational frequencies during the airframe's operational lifetime. Furthermore, aircraft skins are frequently in close proximity to, and get excited by, vibrating components such as engines. Therefore, it is of utmost importance to devise and develop solution techniques to study the vibrational characteristics, such as the natural frequencies and mode shapes of wing and fuselage skins, both swiftly and to an acceptable degree of accuracy during the preliminary design stages, before progressing to advanced stages of design. Using these results, the engineers and designers are able to alter and optimize the geometry of the system or the materials used, to gain a favourable outcome. They will also be able to determine the most suitable locations to add supports and areas that require additional reinforcements. Modal analysis will ensure that the natural frequencies of the structure are maintained within an accepted range.

Aerospace structural components modeled as plates can assume various shapes. They can also be classified as either thick or thin plates and as homogeneous, multilayered or composite by configuration. Unlike their thick counterparts, thin plates are characterized by thicknesses significantly smaller than the other dimensions and special theories have been developed over the past decades to study each of these cases separately. Solution methods have been developed,

whereby the researchers possess the liberty to study only the membrane, transverse/lateral, flexural or torsional vibrations individually or in combination. Furthermore, the effects of shear and warping could be included or neglected if needed. However, the reliability and accuracy of such modal analysis results depends on the method implemented. There are several exact, semi-analytical and numerical methods, developed and presented in the open literature that could be used to carry out the modal analysis of plate-type structures during various design stages.

All approaches mentioned above have their inherent advantages and disadvantages. Therefore, selecting the suitable method that yields fast but accurate solutions is crucial, especially during the early stages of design. In what follows, a brief general review of different plate theories developed over the past decades is presented. The next step will be to classify the studies based on their approach as exact/analytical, semi-analytical or numerical. A significant effort will be made to review studies encompassing as wide an array of plate shapes, sizes and configurations (homogeneous, multilayer, sandwiched, composite) as possible such that it would provide the reader an opportunity to appreciate the variation in the investigation methods used in relation to the geometry, configuration and boundary conditions of the problem. A preliminary discussion on the advantages and disadvantages of each exact, semi-analytical, and numerical method will also be made. Comments will be drawn and recommendations will be made as to which methods are more contemporarily relevant, effective and useful for the vibrational analyses of thin airframe components modeled as plates during the early stages of design, where the speed at which designers arrive at a reasonably accurate ballpark for natural frequencies and mode shapes of a system is as important, if not more important, than the accuracy of the results itself.

1.2 Governing Theories

Plate vibrational analysis dates back to the early 19th century when the free vibration of square plates subjected to free edge boundary conditions was investigated by Chladni [1]. Since this study, many researchers have investigated the vibration and stability characteristics, such as the natural frequencies, mode shapes and buckling loads for plates of various sizes, shapes, materials, boundary conditions and loading configurations.

Numerous investigations carried out by a large number of researchers from across the globe have led to various theories to study plate vibrations. Each theory developed is uniquely different from the others due to the utilization of various assumptions and as such each theory is only applicable to a certain type or a family of plate configurations. Although a detailed assessment of the advantages and disadvantages of each and all the existing plate theories is beyond the scope of this review, the discussion of the various exact/analytical, semi-analytical and numerical models and their applications would be incomplete without a suitable review of the host of governing theories available. Therefore, in what follows a brief description of different theories related to the dynamic analysis of flexible plates developed and presented to date, is presented.

Amongst the many types of plate theories, the thin plate (Kirchhoff-Love) theory, also known as classical plate theory (CPT), has been extensively investigated, where the effects of shear deformation and rotary inertia are neglected to simplify the formulation process (see, e.g., the series of reviews carried out by Leissa [2-7], Bert [8-13] and others [14-15]). It is important to mention here that Bert's [8-13] reviews were on thin composite and sandwiched plates. Timoshenko and Woinowsky-Kreiger [16] and Lekhnitskii [17] also used the Classical Plate Theory to study thin anisotropic plates. Furthermore, Vinson and Sierakowski [18] and Ashton and Whitney [19] exploited the Kirchhoff-Love theory to study laminated thin plates.

However, as the plate thickness increases, the effects of shear deformation and rotary inertia increases to the point that they cannot be ignored anymore. If Classical Plate Theory is used on thick plates, thereby, neglecting the rotary inertia and shear effects, it will overestimate the natural frequencies and buckling loads and as such the solutions will be unconservatively high. Therefore, for thicker plates Classical Plate Theory has been replaced with the Improved Plate Theory (IPT), commonly referred to as the Mindlin-Reissner, or thick plate theory. The latter was a product of the combined efforts of Reissner [20], who incorporated effects of transverse shear, and Mindlin [21], who included the effects of thickness dependent rotary inertia. The applicability of this theory to isotropic plates was investigated by Reissman [22]. Subsequently, Mindlin type plate theories were developed by researchers such as Bergan and Wang [23], Whitney and Pagano [24], as well as Yang et al. [25], where Yang et al. [25] developed a first order shear deformation theory (FSDT) upon extending the IPT. The First Order Shear

Deformation Theory was used by Yang et al. [25] to study laminates that are constructed using an arbitrary number of anisotropic layers. Since, the First Order Shear Deformation Theory does not satisfy the stress-free boundary conditions on the surface of a plate, in order to replace FSDT, Reddy [26] developed a higher order shear deformation theory (HSDT). Many other researchers such as Krishnamurthy [27], Nelson and Lorch [28], Lo et al. [29], Levinson [30], Murthy [31], and Doong et al. [32] have also introduced various higher order theories to study the vibrational characteristics of plates.

Therefore, starting from CPT, many theories such as FSDT and HSDT have been developed over the past decades by different researchers, in a quest to constantly improve on the drawbacks of the preceding theories. In the case of this research, however, CPT also known as Thin Plate Theory will be employed as most two dimensional airframe components such as aircraft wing and fuselage skins could be modelled as thin plates due to their extremely small thickness compared to the length and width.

1.3 Methods of Vibration Analysis

In combination with one or several of the above-mentioned theories, researchers have used many solution techniques to study the vibrational characteristics of thin and thick plates of various shapes. In general, these techniques fall into three main categories, namely, exact, or analytical, semi-analytical, and numerical methods.

The so-called analytical or ‘exact’ methods (within the limits of the theory) have been widely used by researchers over time and they involve determining the closed form solution of the governing differential equations. The Navier method [33] is one such analytical method, which has been used in the past to study plate vibrations of simply supported rectangular plates. The method utilises a double Fourier series and researchers such as Reddy and Phan [33], Senthilnathan et al. [34] and Burt and Chen [35] used the Navier method to study the vibration and buckling of isotropic, orthotropic, as well as laminated plates with simply supported boundary conditions. Kaplevastsky and Shestopal [36] also studied the flexure and buckling of multilayered plates simply supported along all four edges. Doong et al. [37] used the Navier method to investigate vibration and stability characteristics of pre-stressed laminated plates. Similar studies were also carried out by Bert [38] and Adali [39]. The Navier method can also be

extended to study plates simultaneously subjected to simply supported and slip shear boundary conditions. One main disadvantage of the Navier method, limiting its applicability severely, is that it can only be used for plates with simply supported boundary conditions. This means that the vibration behaviour of a fully fixed plate, cantilevered plate or a fully free plate cannot be studied using the Navier method. In most real configurations, e.g., thin aircraft structural components, many different boundary conditions occur, thus, rendering the Navier method unsuitable.

The Levy method [40] is another analytical solution technique that uses a single Fourier series, which can be applied to study the vibration of both thin and thick rectangular plates with two sides subjected to simply supported boundary conditions. Rectangular plates that are pinned on two opposing edges are, thus, termed Levy-type plates. The Levy method yields exact solutions for static deflections, free vibration and buckling loads, however, it is important to note that it does not take into account any bending-twisting coupling of the plates. Certain structural components such as, for example aircraft wing skins, simultaneously undergo substantial bending and torsional moments during flight which induces bending-twisting coupling of the skins. During the preliminary design stages, when the skins are modelled as simple plates, it is important to evaluate the effects of such bending-torsion coupling on the vibrational characteristics of the plate such as the natural frequencies and mode shapes. Another problem with the Levy method is that it is inapplicable to non-rectangular plate shapes, further limiting its application. Furthermore, most plate structures or thin sheets used in wing skins or fuselages are riveted along all four edges, commonly modeled as fully or partially clamped boundary conditions. Therefore, the requirement that two opposing edges should be simply supported is also another limiting factor, as it prevents its application to other boundary configurations such as fully free or clamped edges.

The Levy method can also be extended to plates which have one pair of opposite boundaries subjected to a combination of simply supported and slip-shear conditions, as a result of the similarities between the mathematical conditions of these two types of boundaries. Nevertheless, the fact that the Levy method allows the application of arbitrary boundary conditions along the remaining two edges has attracted some researchers such as Timoshenko and Gere [41] and

Bulson [42] who utilised the Levy method to investigate the vibration and buckling of thin isotropic plates and many others [43-45]. A generalized Levy type solution method was adopted by Khdeir [46] to examine the free vibration of antisymmetric angle-ply laminated rectangular plates with various edge conditions. The stability and vibration of unsymmetric cross-ply laminated plates were also investigated by the same author [47]. In a different study, the generalised Levy type procedure was again used to study the free vibration and buckling of antisymmetric, cross ply, rectangular laminates by Reddy and Khdeir [48]. The plates, in this study, were subjected to various boundary conditions as well.

The superposition technique, also called the Gorman method [49], is also a very powerful analytical method that has been used extensively by many researchers in the past to obtain highly accurate results for problems involving plate vibrations. It was developed and used by Gorman [49] to analyse the vibration behaviour of thin isotropic rectangular plates. In this method, the plate is divided into a number of subsystems, termed building blocks, which are under different boundary conditions compared to the global system, and are also subjected to a distributed force, moment, rotation or translation [49]. The steady state response of each subsystem is then superimposed. Unlike the other solution methods discussed earlier, this method is applicable to a variety of plates, including orthotropic, hybrid and laminated plates. The superposition technique also allows for the application of various classical and non-classical boundary conditions, as well as loading configurations, and it is readily applicable to thin plates, thick Mindlin plates, transverse shear deformable laminated plates, as well as open cylindrical shells. Gorman [50] also applied the method of superposition to study the free vibrational characteristics of a multitude of non-rectangular plates. Saliba [51-52] also used the superposition technique to study the free vibration of non-rectangular plates. Gorman [53], as well as Yu and Cleghorn [54-55], exploited the above method to investigate the free vibration of thin orthotropic plates. The stability and vibration of in-plane stressed orthotropic plates was also investigated by Yu and Cleghorn [56-57] using the same method. A buckling analysis of thin isotropic plates was carried out by Cleghorn and Yu [58] using the method of superposition. Yu and his coworkers also applied the superposition principle to deduce the natural frequencies and mode shapes of thick isotropic plates [59-63] and symmetrically laminated plates [64], where the latter study took also the effects of shear deformation into consideration.

Although it is confirmed by many researchers that the superposition method has an excellent convergence rate [65], the results available does not contain sufficient details on the boundedness of the method. According to the predictions of Illanko [66], whether this method yields the upper or lower bounds for the eigenvalues of a particular plate vibration problem depends on the type of boundary conditions applied on the system as well as the boundary conditions applied on the so called building blocks of the system. Illanko [66] states that if the boundary conditions of the building blocks are more flexible than the actual boundary conditions applied on the plate, lower bound results of the eigenvalues are yielded. Such a case can arise if the building blocks are under simply supported edge conditions and are subjected to driving forces or translations while at the same time the global system consists of clamped edges. However, a problem with the method of superposition is that for mixed boundary types, it has not been verified yet if the results yielded are an upper bound or a lower bound. Thus, this uncertainty may well be a problem when trying to estimate the error of the results.

The Dynamic Stiffness Method (DSM) is also another exact method that has been heavily exploited by researchers to study the vibration of a variety of plate configurations. Boscolo and Banerjee [67] used DSM to determine the vibration of plates using both classical plate theory and first order shear deformation theory. Later, they exploited the Dynamic Stiffness Method to perform exact in-plane free vibration analysis of plates and plate assemblies [68]. A Dynamic Stiffness formulation was also subsequently formed for exact Mindlin plates by Boscolo and Banerjee [69, 70]. Fazzolari and Banerjee [71] used the Dynamic Stiffness Method and higher order shear deformation theory to conduct a buckling analysis of plate assemblies, as well as free vibration analysis of composite plate assemblies [72]. In order to study the free vibration of laminated composite plates, a layerwise dynamic stiffness solution was formed by Boscolo and Banerjee [73]. Using higher-order elements, a free vibration analysis of composite plates was also carried out by Pagani et al. [74]. Later, the Dynamic Stiffness Matrix of a rectangular plate was formed by Banerjee and Papkov [75] for the general case, followed by the development of an exact spectral-dynamic stiffness method for free flexural vibration analysis of orthotropic composite plate assemblies by Liu and Banerjee [76, 77]. Using the novel spectral-dynamic stiffness method, the same authors [78] subsequently conducted an investigation into the free vibration of plates subjected to arbitrary boundary conditions.

However, all the above-mentioned exact methods have very limited applicability as they are based on some form of a simplifying assumption intended at making the derivation of the solutions to the plate governing equation less cumbersome and more tractable. Thus, they have lost their generality and are only useful to model very simple geometries, special cases of plates of certain geometries, and/or subjected to certain boundary conditions, etc. In addition, with every change made to the system configuration, the equations should be reformulated and it is difficult to use such analytical methods to model real life problems with variations in geometry and material properties. Thus, recourse will be frequently made to numerical methods when the closed form solution of a certain set of governing differential equations is intractable.

The Finite Element Method (FEM) is a highly powerful, popular and one of the most widely used numerical techniques that is endowed with the ability to divide any complex geometry to a number of smaller and much simpler geometries called elements, thus, representing the larger complex geometry as a collection of these elements. Therefore, rather than having to develop complicated (displacement) approximation functions for the entire domain, the Finite Element Method (FEM) allows the user to systematically derive and develop simpler approximate functions over each element. This often allows the approximate (field variable) functions to be a linear combination of algebraic polynomials. In order to determine the unknown polynomial coefficients, the governing differential equations are then satisfied over each element and element assembly is performed based on continuity of the solution and balance of internal fluxes. However, the bottom line is that the FEM is essentially a Rayleigh-Ritz or Galerkin method, the only difference being that it only requires locally admissible functions as opposed to the globally admissible functions needed for classical variational methods [79]. Once the approximation functions are derived and the element mass and stiffness matrices are found, the assembly of the element matrices followed by the application of the system boundary conditions, leads to a linear eigenvalue problem. The resulting eigenvalues and Eigenvectors represent the natural frequencies and mode shapes of the system, respectively. Included below, are some of many researchers who utilised the conventional FEM methods for the purpose of examining the vibration behaviour of plates.

Cawley and Adams [80] carried out a vibrational analysis of free laminated composites using FEM. Their study included square aluminum and Carbon Fibre Reinforced Polymer plates of varying ply angles. Reddy [81] also studied the free vibration of antisymmetric angle ply laminated plates using the FEM. Bert et al. [82] used the same method to examine the vibration characteristics of thick rectangular plates made of bi-modulus composite material. The FEM was also exploited by Reddy [83] to investigate the large amplitude bending vibration of rectangular plates consisting of cut-outs. Reddy and Kuppusamy [84] conducted a vibration analysis on rectangular laminated anisotropic plates. Corner supported thick composite plates were also examined using the FEM by Rao and Singh [85]. Using an eight-noded quadratic element, Chang and Chiang [86] studied the vibration of Mindlin plates of rectangular shape consisting of an interior cut-out. Shiau and Chang [87] also carried out a FEM modal analysis of a laminated plate, where the effects of transverse shear deformation was taken in to account. Similarly, the non-linear bending vibration of isotropic, orthotropic and cross-ply laminated square plates were studied by Ganapathi et al. [88]. The vibration characteristics of clamped thick circular plates were also examined by Cheung and Kwok [89] using a Finite Element method. Finally, for bi-modulus annular Mindlin plates, Chen and Chen [90] investigated the asymmetric vibration and dynamic stability using the Finite Element method.

Therefore, unlike analytical/exact solution methods, numerical methods such as the conventional Finite Element Method (FEM), are more advanced and are widely used for structural modal analysis. The FEM is very popular among researchers since it is convenient and adaptable to many complex systems including those consisting of material and geometric variations. Geometric variations are easily modelled as stepped, piecewise-uniform configurations, as opposed to analytical methods which, if not impossible, are very cumbersome to use for complex problems. However, the main shortcoming of the conventional FEM is that a very large number of elements are required to achieve an acceptable degree of accuracy especially for higher modes of vibration, thus, having to solve a large eigenvalue problem. Although the generality, distinguished strengths, and proven track record of the conventional Finite Elements Method (FEM) make it a perfect modeling means and numerical modal analysis tool for the advanced stages and detail design purposes, during the preliminary design stages it is important to arrive at an accurate ballpark of both lower and higher natural frequencies fast as the goal during the early

stages is to minimize the time spent on analysis and maximize the design phase. This disputes the need for an elaborate model and extremely fine mesh.

The boundary element method (BEM) is another numerical computational method that has been long used for the study of plate vibration behaviour. It is known to be less complex in terms of geometric data preparation compared to FEM, since only the boundaries of the problem is discretized in BEM before processing as opposed to conventional FEM where the entire domain is discretized in to finite elements to obtain the solution. In the Boundary Element Method, once the unknowns along the boundaries are determined the solution within the domain is computed as a post processing step. Thus, using the Boundary Element Method, a 3D problem could be modelled with a surface mesh and a 2D problem could be analysed using a line mesh, which greatly reduces time and resources spent during the pre-processing stages.

Early developments of the Boundary Element Method were limited to using the frequency sweep method or the determinant search method such as the studies presented by, Vivoli [91] and Vivoli and Fillipi [92], where an indirect formulation was used to develop the first boundary integral equation for the investigation of a plate vibration problem. In these studies, the solutions were given in terms of Hankel functions and the natural frequencies were determined by searching for frequencies that resulted in a zero determinant for the matrix eigenvalue problem. Using a direct formulation, Niwa et al. [93], and Wong and Hutchinson [94] also used the Boundary Element Method to analyze plate vibration behaviour. A mixed boundary integral-finite element approach to plate vibration problems was presented later by Bzine [95] later on. However, the first boundary-only algebraic eigenvalue problem was developed by Nardini and Brebbia [96]. Kanarachos and Provatidis [97] also later developed a boundary element solution using the indirect method to model acoustic problems. A new method called the Particular Integral Method (PIM) was presented by Ahmad and Banerjee [98] by following a different approach in formulating the generalized eigenvalue problem using the Boundary Element Method. Later Davies and Moslehy [99] used the Boundary Element Method to derive the vibrational frequencies and mode shapes of thin elastic plates.

Despite such continuous advancements, one major drawback of the Boundary Element Method (BEM) is its lack of versatility. More specifically, the Boundary Element Method is incapable of

efficiently handling problems with inhomogeneities and nonlinearities, common in aerospace environments. Conventional FEM, on the other hand, is very versatile and is more widely applicable.

Another class of methods, named semi-analytical, have also been in use for the purpose of studying plate vibrations. The Finite Strip, the Finite Layer, the Spline Strip and the Finite Prism methods are some of the well-known semi-analytical methods popular among users [89, 100-106]. The Finite Layer method was used by Cheung and Chakrabarti [100] to perform a free vibration analysis of thick, layered rectangular plates. In order to understand the vibration of thick laminated plates with curved boundaries, a circular sector element was developed by Cheung and Kwok [89, 101]. Square Mindlin plates subjected to various boundary conditions were investigated by Dawe [102] using the Finite Strip method. The Finite Strip method was once again used by Dawe and Roufaeil [103] to study rectangular Mindlin plates. Thick square plates subjected to simply supported and clamped end conditions were examined by Cheung and Chan [104] using the Finite Strip method. Furthermore, the same authors also studied thin and thick sectorial plate vibrations. Finally, Mizusawa [105] utilised the Finite Strip and Finite Prism methods to examine the vibration characteristics of thick annular sector plates. Mizusawa also utilised the Spline Strip method to examine plates with tapered thickness [106]. The above-mentioned semi-analytical methods, however, are not as powerful or versatile as the conventional FEM described previously and as such, are not very suitable for aerospace structural modal analysis applications.

Finally, the main goal of this review is to introduce the Dynamic Finite Element (DFE) technique, which is a relatively new semi-analytical method first introduced by Hashemi in 1990's [107]. Since its inception, the DFE method has been used in beam, beam-like, and blade dynamic, stability and vibration modelling and analysis. Hashemi and his co-workers have extensively studied the free vibration of isotropic, sandwich, composite and thin-walled beams subjected to various loading configurations, using the DFE method (see, for example, [107-113]). Although, reviewing studies on beam vibrations is not within the scope of this review, a quick reference to some of the many studies exploiting the DFE method for beam related problems is included here for the sake of completeness. Hashemi and Richard [108] presented a DFE

formulation to analyse the free vibration of bending– torsion coupled beams. The flexural axis of the beam studied by Hashemi et al. is not coincident with the inertial axis. Hashemi and Roach [109] also created a DFE formulation for the free vibration of an extension– torsion coupled composite beam. A Quasi–Exact Dynamic Finite Element formulation, for the free vibration analysis of a three–layered sandwich beam consisting of a thick, soft, low strength and density core as well as two face layers made of high strength material, was developed by Hashemi and Adique [110]. Borneman and Hashemi [111] developed a DFE for the free vibration analysis of a bending– torsion coupled laminated composite wing beam. The effects of shear deformation and rotary inertia were neglected in this study but were accounted for in another study by Hashemi and Pereira [112]. Furthermore, Hashemi and Richard [113] also conducted a vibration analysis on an axially loaded bending– torsion geometrically coupled beam using the DFEmethod.

Apart from the above, for additional clarity, a brief description of the DFEformulation process for beam problems is also included below to enhance the reader’s understanding on how this method will be developed for plate problems. To begin with, the Galerkin method of weighted residuals is also used in the DFE method to arrive at the weak integral form of the governing differential equations, as done in the conventional Finite Element Method. The weak integral form of the governing differential equations is then discretized to form the element integral equations, where the DFE formulation process deviates from the classical Finite Element Method. For the beam problems studied in the past, instead of using cubic polynomial and linear shape functions to approximate the flexural and axial/torsional displacements, respectively, the DFE method utilizes frequency-dependent trigonometric basis/shape functions presented in [107] through [113] to interpolate the elemental field variables. The trigonometric shape/interpolation functions have been obtained by taking the closed form solutions to the differential equations of motion governing the uncoupled vibrations of a uniform beam (exact within the limits of the theory), as the basis functions of approximation space. The nodal approximations of displacement are then stated in terms of the resulting frequency-dependent shape functions. This process leads to the element frequency-dependent dynamic stiffness matrices (unlike the frequency-independent element stiffness/mass matrices resulting from the conventional FEM). When these element matrices are assembled, and the system boundary conditions are enforced,

the frequency-dependent global dynamic stiffness matrix is obtained, giving rise to a non-linear eigenvalue problem, different from the linear eigenvalue problem of the conventional Finite Element Method. In order to find the natural frequencies of the system, the frequency domain should then be swept to search for particular frequencies that will make the determinant of the dynamic global stiffness matrix go to zero. The corresponding eigenvectors provide the mode shapes of the system.

From the existing beam vibration results [107-113], it is evident that the Dynamic Finite Element (DFE) formulation is a highly-convergent, hybrid, frequency-dependent, semi-analytical solution method which is more accurate than the conventional Finite Element Method and, unlike the analytical methods discussed above, is adaptable to many complex configurations. Some advantages of the DFE method, as evident from its previous applications, are highlighted below. Based on the results of many vibration analyses on various beam problems carried out by Hashemi and his coworkers [108-11, 114,115], it is clear that the DFE formulation is capable of converging to much more accurate results than the conventional FEM, using the same mesh size. In several cases, to achieve results similar (if not higher) in accuracy to those obtained from the conventional FEM, the DFE method needs eight times less elements than the conventional Finite Element Method. Thus, the user can reduce the modelling and computational overhead by implementing a much coarser mesh and obtain a solution faster without compromising solution accuracy. During the preliminary design stages of aerospace applications, speed at which results are achieved trumps the degree of accuracy of the results. Thus, the fact that the DFE method is a hybrid method, which combines the accuracy of exact methods and the versatility of numerical conventional FEM, while at the same time producing much faster, highly accurate results makes it a powerful and reliable tool whose advantages can be exploited during the preliminary design stages by designers. The fact that most of its formulation is quite similar to the conventional Finite Element Method is another advantage for researchers new to the DFE method.

Given the magnitude of aerospace structural components that can be modelled as plates, engineers and designers would be able to arrive at an acceptable ballpark for the vibrational characteristics, especially for multiple and/or higher modes, with much less effort and using an

extremely coarser mesh in comparison to conventional Finite Element Method, should a plate Dynamic Finite Element become available.

1.4 Research Motivations

Upon reviewing the advancement of methods used in the area of plate vibration analysis, it could be concluded that the existing solution methods all have certain advantages and disadvantages over each other. Although general in nature, the most well-established and widely used conventional FEM formulations need a large number of elements to capture the structures' higher natural frequencies and mode shapes with acceptable precision. On the other hand, the so-called analytical methods, although exact within the limits of the theory, are limited to special cases and simple configurations, and their derivations are often mathematically intensive and cumbersome. The present research was motivated by the lack of an intermediate method, combining the advantages of both FEM and other analytical methods. The hybrid Dynamic Finite Element (DFE) formulation, proven to be a super-convergent method in the vibration analysis of beams and beam-structures, may be the best solution to the current problem of plate vibration. A plate DFE model, once developed, can introduce the accuracy of exact methods to the ease of formulation provided by numerical methods, while yielding results much faster and to the best of the author's knowledge, such a frequency-dependent DFE formulation has never been developed for the vibration analysis of plates. Therefore, an effort is made in this thesis to fill this gap by applying the DFE concept to thin plate vibration analysis.

1.5 Key Objectives and Methodology

The primary objective of the present research was to develop a Quasi-Exact Dynamic Finite Element (QDFE) formulation for the free vibration analysis of thin plates using Classical Plate Theory (CPT) and 16-DOF thin rectangular elements. To achieve this goal, a systematic step-by-step approach, with the following key tasks, was followed:

- Task 1: A novel, quasi-exact solution to the differential equation of motion governing the free vibration of a rectangular plate should be derived, which would necessitate the governing differential equation to be re-written in a new, but equivalent form.

- Task 2: To formulate a special set of innovative dynamic (frequency-dependent) basis functions, unique to the QDFE method, using the quasi-exact solution. These basis functions are formed in such a way that they approach the conventional FEM basis functions (i.e., polynomials), when the natural frequency tends to zero; i.e., representing the state of non-vibration, or static configuration.
- Task 3: Derivation of a set of novel frequency-dependent, trigonometric shape functions for the thin rectangular plate element, using the above-mentioned basis functions, and following the standard procedure used in conventional FEM formulation.
 - o The elemental weak integral form of the governing differential equation of motion was then achieved by applying the Galerkin-type Weighted Residual Formulation and a set of necessary integrations by parts, followed by discretization of the structure (i.e., weak integral form of the governing differential equation of motion) using the thin plate elements.
- Task 4: To develop the element dynamic (frequency-dependent) mass and stiffness matrices, by introducing the field variables expressed in terms of the new dynamic interpolation functions back into the elemental weak integral form of the governing differential equation of motion.
- Task 5: To form the overall DFE model, by assembling, where applicable, the element matrices to formulate the eigenvalue problem of the system, whose solutions are the natural frequencies and corresponding modes of the structure. Applying various boundary conditions and obtaining the eigensolutions was also achieved through a script written in MATLAB[®] and this will yielded the vibrational characteristics of the plate structure under investigation.
- Task 6: To validate the developed QDFE through different numerical tests for single and multiple element meshes, and various boundary conditions, and to verify its convergence superiority, in comparison with conventional FEM-based models.

It is worth noting that prior to developing the QDFE solution for thin plates, conventional FEM solutions were formulated for the same plate problem using two different rectangular plate elements (12-DOF and 16-DOF). This helped in getting familiarized with the FEM formulation, and to use the FEM-based codes as a benchmark for the development of the QDFE formulation. Furthermore, after the QDFE method for thin plates was formed, the results for the natural frequencies were validated for accuracy by comparing them with the conventional FEM results, as well as the exact results available in the open literature. Various classical boundary conditions (B.C.) and geometries are considered. The correctness and accuracy of the conventional FEM results were first verified by comparing them to an ANSYS® simulation.

- Task 7: Once the QDFE method is developed for a homogeneous thin rectangular plate, it was extended to model the vibration behaviour of thin, rectangular, multilayered plates subjected to all types of boundary conditions. The results produced for multilayered plates are then compared with those available in the literature. They were also compared with results from conventional FEM and the Method of Homogenization.
- Task 8: Finally, to further demonstrate the capabilities of the QDFE method, it was applied to investigate the vibration behaviour of six aircraft cabin window bay sections. The QDFE results are then compared against the conventional 16-DOF FEM and ANSYS® results.

The secondary objective of this research was to use the unique quasi-exact solution derived for the plate governing equation to formulate a new Dynamic Coefficient Matrix (DCM) method based on a frequency dependant coefficient matrix, for both thin homogeneous and multilayered rectangular plates. This new DCM method differs from the Classical Method (CM) due to its dynamic nature. Furthermore, it will serve as an alternative analytical method that can be used to investigate the vibration behaviour of a general thin rectangular plate subjected to all types of boundary conditions. The results produced by the DCM solutions are then compared with QDFE and FEM results as well as exact results found in the open literature.

1.6 Analysis Approach and Limitations

In this section the analysis approach taken in this research is outlined along with a brief review of the limitations of the work. As a starting point to the analysis the derivation of the

plate governing equations will be briefly revisited. Once, the theoretical foundation is established the next step of the analytical process would be to derive the quasi-exact solution to the governing differential equation. This is achieved by re-arranging the characteristic equation as the sum of two beam-like expressions and then solving each expression by applying the quadratic formula. Once, the quasi-exact solution is constructed the terms of that solution will be used subsequently, to create the frequency-dependant basis functions unique to the QDFE method based on a 4-node 4-DOF per node rectangular plate element. Using these frequency-dependant basis functions, sixteen trigonometric shape functions will then be developed and subsequently the QDFE solution will be derived for the thin, rectangular, homogeneous and multilayered plates. The above mentioned quasi-exact solution will also be used to develop DCM models for thin, rectangular, homogeneous and multilayered plates. In parallel to this, two conventional FEM models, with 12-DOF and 16-DOF elements, will also be developed for thin, rectangular, homogeneous and multilayered plates for validation of the QDFE and DCM results. The results of these conventional FEM models will be first verified for accuracy using ANSYS[®] simulations. Apart from comparing with the conventional FEM results, the QDFE and DCM results will also be compared with exact data available in the open literature. As a last step, the QDFE method will be extended and applied to model the vibration behaviour of six aircraft cabin window bay sections; the results of which will be verified against in-house developed conventional FEM and ANSYS[®] results.

One of the main limitations of this study is that it disregards the effects of shear and rotary inertia as the quasi-exact solution that is sought is based on Classical Plate Theory. Therefore, the QDFE model presented cannot be used to model the vibration behaviour of thick plates and as such for thick plate vibration analysis, a separate QDFE solution needs to be developed. However, as this method was developed with aerospace applications in mind, this limitation does not largely affect the generality of the method as most, if not all, two dimensional structures encountered in the aerospace environment could be modelled as thin plates as their thickness is much smaller than the other two dimensions.

As the present study is purely a new solution method for an existing plate theory (i.e., CPT) and the corresponding governing differential equation, therefore, no experimental analyses will be

carried out for validation purposes. It is also worth noting that, to the author's best knowledge, there is a lack of experimental data to use for direct comparison/validation of the QDFE/DCM results for the homogeneous and multilayered plate studies as well as the QDFE results for the aircraft cabin window bay sections. However, in order to compensate for this lack of experimental data, the QDFE/DCM results for the homogeneous and multilayered plate studies are verified thoroughly for accuracy using an in-house developed, polynomial based, 16-DOF FEM code, whose accuracy has been validated using ANSYS® simulations and other existing data. Similarly, a comprehensive finite element analysis is done on ANSYS® for all cabin bay configurations using an extremely fine mesh that yields results, which have converged to the second decimal place, in order to substantiate their QDFE results. This compensates adequately for the lack of experimental data for comparison.

As also mentioned before, experimental validation would be required if the QDFE/DCM methods were used to study a brand new problem/theory governed by new governing differential equation(s) that has not been investigated before, to capture new behaviours, or characterize new parameters, etc. In such a case, experimental data are of utmost importance to determine if what is observed by applying the QDFE/DCM methods are in synchronization with what actually occurs in reality. However, the plate configurations investigated in this research are very common ones, which have been studied for decades and as such the existing analytical results are good enough for validation of the proposed element/solution methods. Since, the objective of this work is to introduce new, superconvergent methods to study the same vibration problems studied in the past, verifying the QDFE/DCM results against the exact data available in the open literature would suffice. Where exact data is unavailable, as in the case for the cabin window bay configurations, finite simulations based on ANSYS® and in-house codes are used for the comparison and validation purposes. Thus, although at first glance it may seem to be a limitation of this research, measures have been taken to ensure that the effect of not incorporating experimental validation is, if at all, marginal.

At the present stage the QDFE method is only applicable to rectangular plates. This is another limitation of this research and it arises due to the form of the governing equation that has been used for this study. The Kirchhoff-Love equation used to derive the quasi-exact solution here is

based on the general x-y coordinate system. To obtain the solutions for a circular plate for example, one should commence at the thin plate equation based on the cylindrical coordinate system, however pursuing a QDFE solution based on a circular element is not a priority and as such is not necessary for this thesis at this stage, because the main aim of this work at present is to apply the concept of Dynamic Finite Elements within the domain of plate vibration for the first time, with the goal of establishing the QDFE method as a viable alternative to conventional FEM for plate modal analysis. Therefore, a choice was made at the beginning to use the governing equation based on the x-y coordinate system, in order to facilitate the development of a four-noded, 4-DOF per node rectangular element later on. Nevertheless, developing a rectangular element meant that the study would be strictly limited to the vibration analysis of rectangular shaped 2D structures as it is not possible to accurately fit rectangular elements to study plates of other shapes, such as triangular/circular ones. Several factors contributed to this decision. Firstly, rectangular elements are of higher order compared to triangular elements for example and as such are more accurate, which is one reason why the rectangular shape was favoured. Secondly, most two-dimensional structures found in aerospace environments are rectangular in shape, therefore having a QDFE model based on a rectangular element would increase the scope of analysis as many different rectangular plate assemblies could be then modeled. Furthermore, although, two triangular elements would make up one rectangular element, building a QDFE model based on a triangular element means that more than double the number of elements (due to their lower accuracy) will be needed for an accurate analysis. This defies the whole purpose of building a QDFE model to study plate problems in the first place, as the key objective is to maximize solver efficiency by making the mesh coarser. Lastly, and most importantly, the approach taken to solve the thin plate governing equation also contributed greatly towards this decision. The distinctive procedure applied to obtain the quasi-exact solution required the thin plate equation to be decomposed to two beam-like expressions; one representing the x-direction and the other characterizing the y-direction of the plate. Thus, little choice is left but to use the thin plate equation based on the x-y coordinate system.

Another constraint of the QDFE method at its present stage of development is the restriction on the complexity of plate configurations that it could be applied to. As an example, the aircraft cabin window bay configurations included in this thesis have been greatly simplified. The small

curvature of the window bay section had to be ignored, since a QDFE model has not been developed for shell elements yet. The filleted radius at the edges of the window cut out was also omitted from the analysis, due to the unavailability of a QDFE solution based on a triangular element. The effect of having frames and stringers riveted laterally and longitudinally along the boundaries of the cut out was replaced with a fixed boundary condition due to the absence of a compatible beam element to create a beam-plate assembly. The added structural reinforcement around the cut out due to the presence of doublers were simulated by introducing QDFE elements with a higher thickness at the cut out perimeter, once again as a result of the unavailability of a QDFE beam model having compatible degrees of freedom (4 DOF per node) at the two nodes on each side. Thus far, the QDFE beam models developed in the past by other researchers have only 2 or at most 3 DOF's per node. Therefore, modelling a full and complete aircraft cabin bay window with all its complexities is out of scope of this study as it requires the development of other QDFE models for shell and beam elements as mentioned above.

The ultimate goal of developing the QDFE solution for a plate model is that, in the future, it can be used as a powerful alternative tool to quickly investigate the vibration characteristics of numerous aerospace applications that are modelled as thin plates to a very high degree of accuracy, during the preliminary design stages. However, in order to do this comprehensively, the Dynamic Finite Element method needs to be developed to a full-fledged, versatile tool like conventional FEM. This could only be achieved if a complete library of generic DFE models covering all types of structural elements and configurations are first developed as it was done in the case of conventional FEM during its early stages, many years back. Nevertheless, since the advent of this new method in the late nineties a number of researchers have contributed towards achieving that, by developing DFE models for various one-dimensional (line) structural elements (rods, beams, etc.) configurations. But, the DFE concept of pursuing an exact solution to the governing equation of the structure, using the terms of this general solution to form frequency-dependant basis functions that approach their polynomial counterparts when the frequency tends to zero and then exploiting these basis functions to derive the unique trigonometric basis functions was not applied to the case of plates until now. Therefore, the DFE concept described above, which originated two decades back, is developed for the vibration analysis of two-dimensional plate structural elements in this thesis. As such the primary objective of this

research is to extend the DFE method to 2D elements by developing, for the first time, a Quasi-Exact Dynamic Finite Element Model for thin homogeneous and multilayered plates, thereby adding to the library of generic DFE structural element models that have been created over the years by many a researcher.

1.7 Thesis Organization

In Chapter 2, the derivation of the thin plate governing differential equation will be briefly revisited using Hamiltonian mechanics and the Classical Plate Theory (CPT), for the sake of completeness. After the equations are reviewed, the novel and unique quasi-exact solution to the governing differential equation, sought using a distinctive procedure that has never been presented, will be introduced. Once the quasi-exact solution for the plate equation is derived, a set of new trigonometric, frequency-dependant basis functions characterizing the QDFE method will be presented. Subsequently, the QDFE shape functions formed using the terms of the quasi-exact solution to the governing differential equation, will be developed using the newly found basis functions.

In Chapter 3, the vibration behaviour of a thin, homogeneous, rectangular plate subjected to various boundary conditions will be investigated. Initially, the conventional FEM method will be developed wherein two separate elements will be used. The first formulation will be based on a 12-DOF thin rectangular plate element while the second FEM formulation will use a 16-DOF element. Next, in Chapter 4 the QDFE method will be applied to investigate the modal characteristics of the thin, homogeneous, rectangular plate using the trigonometric, frequency-dependant shape functions presented in Chapter 3. Subsequently, the DCM method, which is also based on the quasi-exact solution to the plate governing equation, will be presented in Chapter 4 to examine the same thin, homogeneous, rectangular plate problem. Later in Chapter 4, the surface plots for the new QDFE shape functions will be illustrated and compared with surface plots of the polynomial FEM shape functions. The numerical results of the 12-DOF FEM, 16-DOF FEM, QDFE and DCM methods will also be presented. A convergence study will be carried out on ANSYS® and the results of the ANSYS® simulation will be used to verify the accuracy of the 12-DOF FEM and 16-DOF FEM results. The QDFE and DCM results will in turn be compared with the verified FEM results as well as exact results found in the open

literature. To conclude, comments will be drawn in regards to the accuracy of the results from each method and an evaluation will be made with reference their convergent behaviour.

In Chapter 5, the vibration behaviour of a multi-layered, thin, rectangular plate will be investigated. Firstly, the Method of Homogenization (M of H) will be used to perform a modal analysis. Then the conventional FEM formulation presented in Chapter 4 will be extended to examine the multi-layered, thin, rectangular plate. Subsequently, the QDFE method for thin plates presented in Chapter 4 will also be extended to study the vibrational characteristics of the multilayered plate. The roots of the governing differential equation presented in Chapter 3, which were modified to suit a multilayered plate, will also be presented here. Later in Chapter 5, the DCM method introduced in Chapter 4 will be extended to investigate the multilayered plate problem. The numerical results for the Homogenization, FEM, QDFE and DCM methods will be presented prior to concluding this chapter. Once again, the FEM results will be compared with the results of an ANSYS[®] simulation and the QDFE and DCM results will be compared with the validated FEM results. Finally, as in Chapter 4, a discussion will be carried out and an assessment will be made on both the accuracy of the results and the rate of convergence of each method.

In Chapter 6 the QDFE method will be applied to study the natural frequencies of six aircraft cabin window bay sections in order to show case the advantage of utilizing an intermediate, superconvergent method during the early stages of design. The results produced by the QDFE method are also compared with the 16-DOF FEM and ANSYS[®] results here to further highlight this fact. Finally, a comparison will also be made between the CPU time elapsed for the QDFE method and the 16-DOF FEM method when solving for the systems natural frequencies, at the end of this Chapter.

Chapter 7 will be the final Chapter of this thesis. Here, concluding remarks will be drawn upon the objectives accomplished and contributions made to the scientific knowledge base, the limitations of the methods presented will be discussed, and comments will be made on the future direction of research progress.

2. THEORETICAL BACKGROUND

2.1 Development of Governing Differential Equation

This Chapter consists of a brief review of the thin plate governing equation formulation using Hamiltonian mechanics and the Classical Plate Theory (CPT) as shown in [116] for the sake of completeness and the formation of the quasi-exact solution to this equation leading to the trigonometric basis functions and the new frequency dependant shape functions.

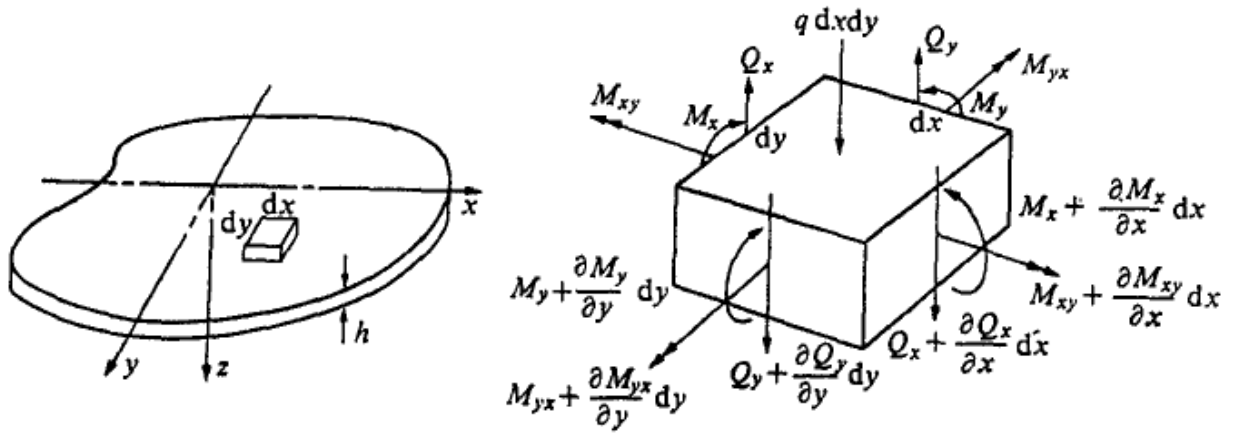


Figure 1: (a) A thin plate of thickness, h , with infinite width and length. (b) An infinitesimal element of the thin plate subjected to bending and twisting moments, shear forces and distributed load per unit length [116]

A plate is a two dimensional extension of a beam and like in a beam there are bending moments and transverse shear forces acting in this two dimensional elastic system as well. The kinematics of CPT is similar to that of Euler – Bernoulli beam theory. Consider a thin plate with a thickness h and infinite length and width as shown in Figure 1a. It is important to note the infinitesimal differential element $h \cdot dx \cdot dy$ on Figure 1a which has been enlarged and depicted as shown in Figure 1b. In Figure 1b, it is possible to see the various shear forces, bending moments, twisting moments and external loads that typically act on such a differential element. Here, M_x and M_y are the bending moments per unit length that occur as a result of the normal stresses σ_x and σ_y , respectively. Furthermore, the twisting moments per unit length M_{xy} and M_{yx} that arise as a result of the respective shearing stresses, T_{xy} and T_{yx} , as well as the shear forces per unit length Q_x and

Q_y that are formed due to the actions of the respective shearing stresses, T_{xz} and T_{yz} , are also illustrated in Figure 1b.

The next step is to balance the forces and moments about the differential plate element equilibrium. By balancing the forces in the x , y and z directions it can be seen that only the force equation of motion in the z -direction is non-trivial. Similarly, considering moments about each co-ordinate axis reveals that the moment equation of motion about the z -axis essentially goes to zero. Thus, we are left with the following three equations of motion:

$$-Q_x dy + \left(Q_x + \frac{\partial Q_x}{\partial x} dx \right) dy - Q_y dx + \left(Q_y + \frac{\partial Q_y}{\partial y} dy \right) dx + q dx dy = \rho h dx dy \frac{\partial^2 w}{\partial t^2} \quad (1)$$

$$\left(M_y + \frac{\partial M_y}{\partial y} dy \right) dx - M_y dx + M_{xy} dy - \left(M_{xy} + \frac{\partial M_{xy}}{\partial x} dx \right) dy - Q_y dx dy = 0 \quad (2)$$

$$\left(M_x + \frac{\partial M_x}{\partial x} dx \right) dy - M_x dy + \left(M_{yx} + \frac{\partial M_{yx}}{\partial y} dy \right) dx - M_{yx} dx - Q_x dy dx = 0 \quad (3)$$

In Eq. (1) w represents the lateral deflection of the mid plane of the thin plate along z -axis. It is also known as the flexural displacement of the plate and it is a function of the x and y spatial coordinates as well as time, i.e., $w(x,y,t)$. Also, ρ in Eq. (1) is the material density. It is also important to note here that in keeping with the assumptions of the Classical Plate Theory (CPT) used for thin plates the contributions due to rotary inertia are neglected in Eqs. (2) and (3). Furthermore, all higher order terms that arise as a result of the distributed load q are also not taken in to consideration when writing out the moment equations of motion. Thus, simplifying the terms in Eqs. (1), (2) and (3) results in the following expressions.

$$\frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y} + q = \rho h \frac{\partial^2 w}{\partial t^2} \quad (4)$$

$$\frac{\partial M_y}{\partial y} - \frac{\partial M_{xy}}{\partial x} - Q_y = 0 \quad (5)$$

$$Q_y = \frac{\partial M_y}{\partial y} - \frac{\partial M_{xy}}{\partial x} \quad (5-a)$$

$$\frac{\partial M_x}{\partial x} + \frac{\partial M_{yx}}{\partial y} - Q_x = 0 \quad (6)$$

$$Q_x = \frac{\partial M_x}{\partial x} + \frac{\partial M_{yx}}{\partial y} \quad (6-a)$$

Solving Eqs. (5) and (6) to obtain the definitions for the shear forces Q_y and Q_x gives Eqs. (5-a) and (6-a), respectively. Substituting Eqs. (5-a) and (6-a) in Eq. (4) will yield a single partial differential equation. This equation shown below, which is written in terms of the various moments, governs the lateral (bending) motion of the thin plate.

$$\frac{\partial^2 M_x}{\partial x^2} + \frac{\partial^2 M_{yx}}{\partial x \partial y} - \frac{\partial^2 M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_y}{\partial y^2} + q = \rho h \frac{\partial^2 w}{\partial t^2} \quad (7)$$

In order for free vibration to occur, there should be no distributed load q acting on the plate. Thus, this term is set to zero and Eq. (7) is re-written as follows.

$$\frac{\partial^2 M_x}{\partial x^2} + \frac{\partial^2 M_{yx}}{\partial x \partial y} - \frac{\partial^2 M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_y}{\partial y^2} = \rho h \frac{\partial^2 w}{\partial t^2} \quad (8)$$

The next step is to determine the relationship between the moments in Eq. (7) and the flexural displacement w . To this end, the kinematics of thin plate deformation is presented below.

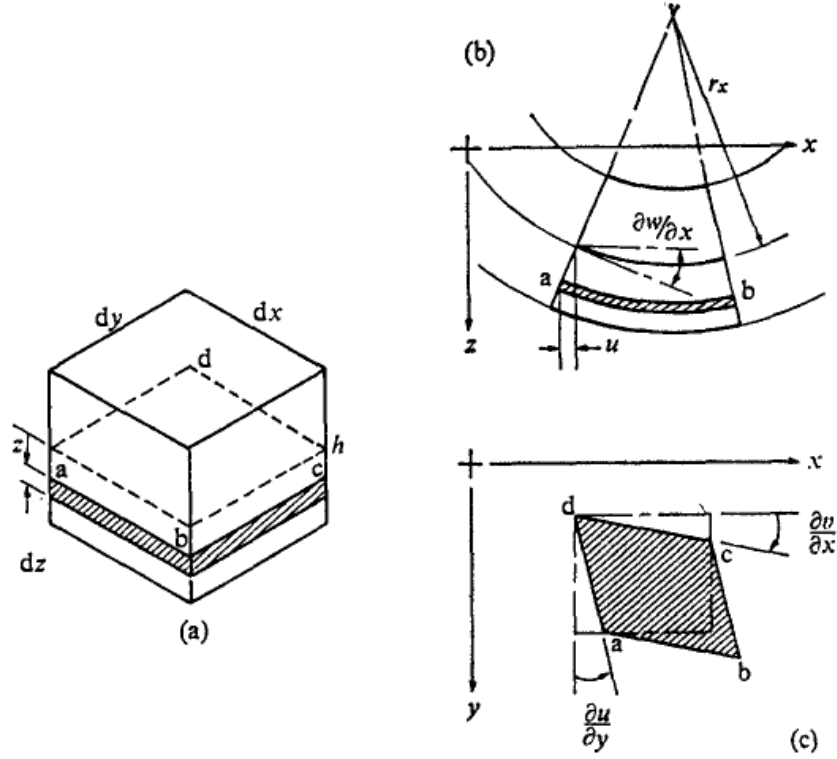


Figure 2: (a) An infinitesimal differential element with a lamina shown by the shaded region marked as $a-b-c-d$. (b) A cross section view of the differential element and lamina in bending. (c). A top view of the lamina subjected to shear loading. [116]

A cross sectional view of the deformation of the lamina marked as $a-b-c-d$ in Figure 2a is illustrated in Figure 2b when the infinitesimal beam element is in pure bending. As in Euler – Bernoulli beam theory, all plane sections are assumed to remain plain and perpendicular to the mid-plane even in Classical Plate Theory. A similar behaviour to that explained above is also seen in the $y-z$ plane. Thus, the normal strains in the lamina $a-b-c-d$ can be written as:

$$\varepsilon_x = \frac{z}{r_x} \quad (9)$$

$$\varepsilon_y = \frac{z}{r_y} \quad (10)$$

In Eqs. (9) and (10) r_x and r_y are the radii of curvature in the $x-z$ and $y-z$ planes, respectively, and z is the distance between the midplane and the lamina $a-b-c-d$. Assuming small deflections and

slopes, the curvatures r_x and r_y in eqs. (9) and (10) can be defined as $-\partial^2 x/\partial x^2$ and $-\partial^2 y/\partial y^2$, respectively. Thus, Eqs. (9) and (10) can be re-written as follows.

$$\varepsilon_x = -z \frac{\partial^2 x}{\partial x^2} \quad (11)$$

$$\varepsilon_y = -z \frac{\partial^2 y}{\partial y^2} \quad (12)$$

The next step is to define the shear strain. From Figure 2c it can be seen that the rotations of the sides are $\partial u/\partial y = \phi_x$ and $\partial v/\partial x = \phi_y$. Here, U and V are the displacement components of a particle in the x and y directions. Thus, the shear strain γ_{xy} can be defined as:

$$\gamma_{xy} = \frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} = \phi_x + \phi_y \quad (13)$$

However, from observing Figure 2b it is clear that the displacements U and V can be defined as $-z \cdot \partial w/\partial x$ and $-z \cdot \partial w/\partial y$, respectively. Substituting these expressions in Eq. (13) and re-writing the definition of shear strain gives:

$$\gamma_{xy} = -2z \frac{\partial^2 w}{\partial x \partial y} \quad (14)$$

Having derived all the definitions for the strains, the general Hooke's Law can be exploited now to arrive at the expressions for the stresses.

$$\sigma_x = \frac{E}{1-\nu^2} (\varepsilon_x + \nu \varepsilon_y) = -\frac{Ez}{1-\nu^2} \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right) \quad (15)$$

$$\sigma_y = \frac{E}{1-\nu^2} (\varepsilon_y + \nu \varepsilon_x) = -\frac{Ez}{1-\nu^2} \left(\frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right) \quad (16)$$

$$\tau_{xy} = G\gamma_{xy} = -2Gz \frac{\partial^2 w}{\partial x \partial y} \quad (17)$$

The bending and twisting moments on the element face defined as $h \cdot dy$ in Figure 2a can now be determined as follows.

$$M_x \cdot dy = \int_{-h/2}^{h/2} z \sigma_x dy dz \quad (18)$$

Eq. (18) can be re-arranged as:

$$M_x = \int_{-h/2}^{h/2} z \sigma_x dz \quad (19)$$

Substituting Eq. (15) in Eq. (19) and integrating across the thickness yields the definition for the bending moment about the x-axis M_x acting on the element face.

$$M_x = -D \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right) \quad (20)$$

In Eq. (20) D represents the plate modulus and it is defined as:

$$D = \frac{Eh^3}{12(1-\nu^2)} \quad (21)$$

The definition for the bending moment about the y-axis M_y acting on the face of the differential element can also be derived using the same procedure outlined above.

$$M_y = -D \left(\frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right) \quad (22)$$

Performing integration across the thickness of the element on the following expression yields the definition for the twisting moment M_{xy} .

$$M_{xy} = - \int_{-h/2}^{h/2} z \tau_{xy} dz \quad (23)$$

It should be noted here that a negative sign has been added in front of the integral in Eq. (23). The purpose of this is to ensure that the twisting moment caused by positive shear stress confirms with the sign convention used in Figure 2b. Thus, substituting the definition of shear stress given in Eq. (17) in to Eq. (23) and integrating gives the following expression.

$$M_{xy} = D(1-\nu)\frac{\partial^2 w}{\partial x \partial y} \quad (24)$$

In order to re-write the governing partial differential equation for thin plates in terms of the flexural displacement w , Eqs. (20), (22) and (24) as well as the expression $M_{xy} = -M_{yx}$ should be substituted in to Eq. (8). Thus, we obtain,

$$D\left(\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4}\right) + \rho h \frac{\partial^2 w}{\partial t^2} = 0 \quad (25)$$

The expression within parenthesis can also be re-arranged follows.

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right) = \nabla^2 \nabla^2 w \quad (26)$$

Finally, writing Eq. (25) in terms of the Laplacian of the Laplacian, also known as the biharmonic operator:

$$D\nabla^4 w(x, y, t) + \rho h \frac{\partial^2 w(x, y, t)}{\partial t^2} = 0 \quad (27)$$

As the governing differential equation for a thin plate has now been derived using Hamiltonian mechanics, the next step will be to pursue the development of a quasi-exact solution and the next section of this thesis will be dedicated to achieving this. With such a solution in hand, it becomes possible to study the free vibration behaviour of airframe components that can be modelled as thin rectangular plates. Before pursuing a quasi-exact solution to the governing equation, it is important to note here that it will not provide information regarding the effects of shear, rotary inertia and damping on the modal characteristics of the structure as the linear undamped free vibration equation is employed per Classical Plate Theory as the basis of the solution. However, this choice of theory can be reasonably justified, firstly, as most aircraft surfaces and skins can be classified as Kirchhoff plates which have extremely small thickness compared to their other dimensions and, secondly, because it is a relatively less complex starting point to develop a new Dynamic Finite Element formulation for plates for the first time. Upon developing a robust plate DFE model, the effects of shear, rotary inertia and damping can be later incorporated to the

solution as deemed fit. Thus, in the subsequent section the distinct quasi-exact solution to the thin plate governing equation will be developed and presented as a pre-requisite to develop the QDFE method.

2.2 A Quasi-Exact Solution for the Thin Plate Equation

In order to develop the Quasi-Exact Dynamic Finite Element (QDFE) and Dynamic Coefficient Matrix (DCM) methods, it is important to find the quasi-exact solutions for the plate governing partial differential equation given in Eq. (25). To this end, a new approach is taken here, which to the best of author's knowledge has not been done before. That is to decompose the plate equation into two separate beam-like expressions representing each spatial coordinate direction of the plate. For clarity, the main steps of this procedure are outlined below.

If the solution is assumed to take the following form, obtained through the separation of variables:

$$w(x, y, t) = W(x, y)e^{i\omega t}; \text{ where } W(x, y) = e^{Ax}e^{By} \quad (28)$$

Then, introducing expression (28) back into the governing differential equation (25), the corresponding characteristic equation can be written as:

$$A^4 + 2A^2B^2 + B^4 - \frac{\omega^2 \rho h}{D} = 0 \quad (29)$$

and then rearranged as:

$$\underbrace{\left(A^4 + A^2B^2 - k_1 \frac{\omega^2 \rho h}{D} \right)}_{*} + \underbrace{\left(B^4 + A^2B^2 - k_2 \frac{\omega^2 \rho h}{D} \right)}_{**} = 0 \quad (30)$$

where, k_1 and k_2 represent the mass distribution constants along the x- and y - directions, respectively. These constants were introduced to decompose the plate governing equation into the two beam-like expressions in Eq. (30) and such an approach has not been pursued before. Through careful observation, it can be seen that simply plugging numerical values in place of these constants will allow one to re-construct and re-write the plate governing equation into its

original form. The numerical values of k_1 and k_2 can be anything between 0 and 1 (i.e., $0 < k_1$ and $k_2 < 1$), however, the sum of the two mass distribution constants should be unity (i.e., $k_1 + k_2 = 1$). For example, k_1 and k_2 will both be equal to 0.5 for a square plate. They will assume other values for other rectangular plate shapes. The term (*) represents the x -direction and the term (**) is for the y -direction of the plate. In both expressions, A is the coordinate in the x – direction and B is the coordinate in the y -direction. The terms (*) and (**) are treated as two different (beam) equations for the purpose of determining roots. Furthermore, in expression (*), A can vary and B is held constant and for the term (**), A is held constant and B is allowed to vary. The quadratic formula was then applied on the expressions (*) and (**) separately as shown below. The mathematical manipulations are shown for one beam-like expression for brevity. The roots of the other beam-like term are obtained using an identical process.

Applying the quadratic formula on the expressions (*), treating B^2 , as constant and isolating for B^2 :

$$A^2 = \frac{-B^2 \pm \sqrt{B^4 + 4\left(\frac{k_1 \omega^2 \rho h}{D}\right)}}{2} \quad (30-a)$$

$$2A^2 + B^2 = \sqrt{B^4 + 4\left(\frac{k_1 \omega^2 \rho h}{D}\right)}$$

$$(2A^2 + B^2)^2 = B^4 + 4\left(\frac{k_1 \omega^2 \rho h}{D}\right)$$

$$4A^4 + 4A^2 B^2 + B^4 = B^4 + 4\left(\frac{k_1 \omega^2 \rho h}{D}\right)$$

$$A^2 + B^2 = \frac{\left(\frac{k_1 \omega^2 \rho h}{D}\right)}{A^2}$$

Therefore, for term (*) the following expression results:

$$B^2 = \frac{\left(\frac{k_1 \omega^2 \rho h}{D}\right)}{A^2} - A^2 = \frac{\left(\frac{k_1 \omega^2 \rho h}{D}\right) - A^4}{A^2} \quad (30-b)$$

Similarly, applying the quadratic formula for expression (**) while treating A^2 as a constant this time and simplifying using an identical process to isolate A^2 :

$$A^2 = \frac{\left(\frac{k_2 \omega^2 \rho h}{D}\right)}{B^2} - B^2 = \frac{\left(\frac{k_2 \omega^2 \rho h}{D}\right) - B^4}{B^2} \quad (30-c)$$

On their own, Eq. (30-b) will be the solution to expression (*) and Eq. (30-c) will be the exact solution to expression (**). Together they will form the exact solutions for the entire plate governing partial differential equation. Eq. (30-b) and Eq. (30-c) can then be solved simultaneously to determine the roots A_1, A_2, A_3 and A_4 for expression (*) and the roots B_1, B_2, B_3, B_4 for expression (**). The solution process for expression (*) is elaborated below.

By substituting Eq. (30-b) in Eq. (30-c), Eq. (30-d) is resulted.

$$A^2 = \frac{\left(\frac{k_2 \omega^2 \rho h}{D}\right)}{\left[\frac{\left(\frac{k_1 \omega^2 \rho h}{D}\right) - A^4}{A^2}\right]} - \left[\frac{\left(\frac{k_1 \omega^2 \rho h}{D}\right) - A^4}{A^2}\right] \quad (30-d)$$

Re-arranging and simplifying Eq. (30-d) as follows yields Eq. (30-e).

$$A^2 = \frac{A^2 \left(\frac{k_2 \omega^2 \rho h}{D}\right)}{\left[\left(\frac{k_1 \omega^2 \rho h}{D}\right) - A^4\right]} - \left[\frac{\left(\frac{k_1 \omega^2 \rho h}{D}\right) - A^4}{A^2}\right]$$

$$\begin{aligned}
A^2 &= \frac{A^4 \left(\frac{k_2 \omega^2 \rho h}{D} \right) - \left(\left(\frac{k_1 \omega^2 \rho h}{D} \right) - A^4 \right)^2}{A^2 \left[\left(\frac{k_1 \omega^2 \rho h}{D} \right) - A^4 \right]} \\
A^4 \left[\left(\frac{k_1 \omega^2 \rho h}{D} \right) - A^4 \right] &= A^4 \left(\frac{k_2 \omega^2 \rho h}{D} \right) - \left(\left(\frac{k_1 \omega^2 \rho h}{D} \right) - A^4 \right)^2 \\
A^4 \left[\left(\frac{k_1 \omega^2 \rho h}{D} \right) - A^4 \right] &= A^4 \left(\frac{k_2 \omega^2 \rho h}{D} \right) - \left(\left(\frac{k_1 \omega^2 \rho h}{D} \right)^2 - 2A^4 \left(\frac{k_1 \omega^2 \rho h}{D} \right) + A^4 \right) \\
A^4 \left(\frac{k_1 \omega^2 \rho h}{D} \right) - A^8 &= A^4 \left(\frac{k_2 \omega^2 \rho h}{D} \right) - \left(\frac{k_1 \omega^2 \rho h}{D} \right)^2 + 2A^4 \left(\frac{k_1 \omega^2 \rho h}{D} \right) - A^8 \quad (30-e) \\
A^4 \left(\frac{k_2 \omega^2 \rho h}{D} \right) - \left(\frac{k_1 \omega^2 \rho h}{D} \right)^2 + 2A^4 \left(\frac{k_1 \omega^2 \rho h}{D} \right) - A^4 \left(\frac{k_1 \omega^2 \rho h}{D} \right) &= 0 \\
A^4 \left(\frac{k_2 \omega^2 \rho h}{D} \right) - \left(\frac{k_1 \omega^2 \rho h}{D} \right)^2 + A^4 \left(\frac{k_1 \omega^2 \rho h}{D} \right) &= 0 \\
A^4 \left(\frac{\omega^2 \rho h}{D} \right) (k_1 + k_2) - \left(\frac{k_1 \omega^2 \rho h}{D} \right)^2 &= 0
\end{aligned}$$

Since, the sum of k_1 and k_2 equals unity, Eq. (30-e) can be re-written and simplified as outlined below to derive the roots for expression (*) of the characteristic equation, presented as Eq. (30-f).

$$A^4 = \frac{\left(\frac{k_1 \omega^2 \rho h}{D} \right)^2}{\left(\frac{\omega^2 \rho h}{D} \right)} = \frac{k_1^2 \omega^4 \rho^2 h^2}{D^2} \frac{D}{\left(\frac{\omega^2 \rho h}{D} \right)}$$

$$A^4 = \frac{k_1^2 \omega^2 \rho h}{D}$$

$$A^2 = \sqrt{\frac{k_1^2 \omega^2 \rho h}{D}}$$

$$A^2 = \pm k_1 \omega \sqrt{\frac{\rho h}{D}}$$

$$A = \sqrt{\pm k_1 \omega \sqrt{\frac{\rho h}{D}}} \quad (30-f)$$

Similarly, substituting Eq. (30-c) in Eq. (30-b) gives the following roots for expression (**):

$$B = \sqrt{\pm k_2 \omega \sqrt{\frac{\rho h}{D}}} \quad (30-g)$$

Thus, from Eq. (30-f) it can be seen that there will be four roots A_i ($i = 1, 2, 3, 4$), for the expression (*) as defined in Eqs. (31) and (32) of which two are real and two are imaginary.

$$A_1 = -A_3 = \sqrt{k_1 \omega \sqrt{\frac{\rho h}{D}}} = \beta_x \quad (31)$$

$$A_2 = -A_4 = i \sqrt{k_1 \omega \sqrt{\frac{\rho h}{D}}} = \alpha_x \quad (32)$$

Similarly, from Eq. (30-g) it can be seen that there are four roots B_i ($i = 1, 2, 3, 4$), for the expression (**) as defined in Eqs. (33) and (34) of which two are real and two are imaginary.

$$B_1 = -B_3 = \sqrt{k_2 \omega \sqrt{\frac{\rho h}{D}}} = \beta_y \quad (33)$$

$$B_2 = -B_4 = i \sqrt{k_2 \omega \sqrt{\frac{\rho h}{D}}} = \alpha_y \quad (34)$$

It is important to note here that the roots shown in Eqs. (31) through (34) not only satisfy their individual expressions separately but together any real-real or imaginary-imaginary combination (A_i and B_i) of these roots also satisfy Eq. (30) as a whole. Thus, each real-real and imaginary-imaginary pair of roots (A_i and B_i) is an exact solution to the plate governing equation. There are 8 such pairs of exact solutions and they are A_1 and B_1 , A_1 and B_3 , A_3 and B_1 , A_3 and B_3 , A_2 and B_2 , A_2 and B_4 , A_4 and B_2 , A_4 and B_4 . However, if a real-imaginary combination of roots (A_i and B_i) is

substituted in to the characteristic equation, it can be seen that such a pair does not fully satisfy the entire characteristics equation in general, although on their own each of these roots satisfy their respective beam-like expressions (*) and (**). There are 8 such real-imaginary combinations that can be made from the solutions presented in Eqs. (31) through (34) and these pairs (A_i and B_i) are not exact solution to the plate governing equation. These roots are A_1 and B_2 , A_1 and B_4 , A_3 and B_2 , A_3 and B_4 , A_2 and B_1 , A_2 and B_3 , A_4 and B_1 and A_4 and B_3 . Thus, out of the 16 combinations of roots that can be developed, 8 satisfy the governing equation fully, but the other 8 fails to do so and as such the solution becomes a quasi-exact solution to the plate governing equation.

As the solution to the plate equation was assumed to take the form shown in Eq. (28) the following expressions were constructed using the roots shown in Eqs. (31) to (34).

$$e^{Ax} = C_1 e^{-i\alpha_x} + C_2 e^{i\alpha_x} + C_3 e^{-\beta_x} + C_4 e^{\beta_x} \quad (35)$$

$$e^{By} = D_1 e^{-i\alpha_y} + D_2 e^{i\alpha_y} + D_3 e^{-\alpha_y} + D_4 e^{\alpha_y} \quad (36)$$

Combining the expressions in Eqs. (35) and (36) with the trigonometric identities for, $\sin(x) = (e^{ix} - e^{-ix})/2i$, $\cos(x) = (e^{ix} + e^{-ix})/2$, $\sinh(x) = (e^x - e^{-x})/2$ and $\cosh(x) = (e^x + e^{-x})/2$ yields Eqs. (37) and (38).

$$e^{Ax} = C_1 \sin(\alpha_x x) + C_2 \cos(\alpha_x x) + C_3 \sinh(\beta_x x) + C_4 \cosh(\beta_x x) \quad (37)$$

$$e^{By} = D_1 \sin(\alpha_y y) + D_2 \cos(\alpha_y y) + D_3 \sinh(\beta_y y) + D_4 \cosh(\beta_y y) \quad (38)$$

where, C_1 to C_4 and D_1 to D_4 are unknown coefficients. Since, the solution is assumed to take the form defined by Eq. (28) the final 16-term quasi-exact solution for a thin plate can be derived by multiplying Eqs. (37) and Eq. (38) as follows.

$$\begin{aligned}
W(x, y) = & E_{11} \sin(\alpha_x x) \sin(\alpha_y y) + E_{12} \sin(\alpha_x x) \cos(\alpha_y y) + E_{13} \sin(\alpha_x x) \sinh(\beta_y y) \\
& + E_{14} \sin(\alpha_x x) \cosh(\beta_y y) + E_{21} \cos(\alpha_x x) \sin(\alpha_y y) + E_{22} \cos(\alpha_x x) \cos(\alpha_y y) \\
& + E_{23} \cos(\alpha_x x) \sinh(\beta_y y) + E_{24} \cos(\alpha_x x) \cosh(\beta_y y) + E_{31} \sinh(\beta_x x) \sin(\alpha_y y) \\
& + E_{32} \sinh(\beta_x x) \cos(\alpha_y y) + E_{33} \sinh(\beta_x x) \sinh(\beta_y y) + E_{34} \sinh(\beta_x x) \cosh(\beta_y y) \\
& + E_{41} \cosh(\beta_x x) \sin(\alpha_y y) + E_{42} \cosh(\beta_x x) \cos(\alpha_y y) + E_{43} \cosh(\beta_x x) \sinh(\beta_y y) \\
& + E_{44} \cosh(\beta_x x) \cosh(\beta_y y)
\end{aligned} \tag{39}$$

where, E_{ij} in Eq. (39) are the new unknown coefficients defined as follows.

$$E_{ij} = C_i \cdot D_j \tag{40}$$

2.3 New Trigonometric, Frequency-Dependant Basis and Shape Functions

The new trigonometric, frequency dependant basis functions of the approximation space are shown in Table 1. These basis functions are designed as linear combinations of the solutions to the plate governing equation. Furthermore, they have been developed such that when the natural frequency ω and subsequently the roots, α_x , α_y , β_x and β_y of the characteristic equation tends to zero, they change to the basis functions of a standard thin plate element in classical FEM which are based on an incomplete quintic polynomial.

Table 1: Trigonometric basis functions for thin plate

Sym-bol	FEM	Trigonometric
b ₁	1	$\cos(\alpha_x \xi) \cos(\alpha_y \eta)$
b ₂	x	$(\cos(\alpha_y \eta) \sin(\alpha_x \xi)) / \alpha_x$
b ₃	y	$(\cos(\alpha_x \xi) \sin(\alpha_y \eta)) / \alpha_y$
b ₄	x ²	$\frac{\cosh(\beta_x \xi) \cos(\alpha_y \eta) - \cos(\alpha_x \xi) \cos(\alpha_y \eta)}{\alpha_x^2 + \beta_x^2}$
b ₅	xy	$(\sin(\alpha_x \xi) \sin(\alpha_y \eta)) / \alpha_x \alpha_y$
b ₆	y ²	$\frac{\cosh(\beta_y \eta) \cos(\alpha_x \xi) - \cos(\alpha_y \eta) \cos(\alpha_x \xi)}{\alpha_y^2 + \beta_y^2}$
b ₇	x ³	$\frac{\sinh(\beta_x \xi) \cos(\alpha_y \eta) - \sin(\alpha_x \xi) \cos(\alpha_y \eta)}{\alpha_x^3 + \beta_x^3}$
b ₈	x ² y	$\frac{\cosh(\beta_x \xi) \sin(\alpha_y \eta) - \cos(\alpha_x \xi) \sin(\alpha_y \eta)}{\alpha_y \alpha_x^2 + \alpha_y \beta_x^2}$

Table 1 continued: Trigonometric basis functions for thin plate

Sym -bol	FEM	Trigonometric
b_9	xy^2	$\frac{\cosh(\beta_y \eta) \sin(\alpha_x \xi) - \cos(\alpha_y \eta) \sin(\alpha_x \xi)}{\alpha_x \alpha_y^2 + \alpha_x \beta_y^2}$
b_{10}	y^3	$\frac{\sinh(\beta_y \eta) \cos(\alpha_x \xi) - \sin(\alpha_y \eta) \cos(\alpha_x \xi)}{\alpha_y^3 + \beta_y^3}$
b_{11}	$x^3 y$	$\frac{\sinh(\beta_x \xi) \sin(\alpha_y \eta) - \sin(\alpha_x \xi) \sin(\alpha_y \eta)}{\alpha_y \alpha_x^3 + \alpha_y \beta_x^3}$
b_{12}	$x^2 y^2$	$\frac{\cosh(\beta_x \xi) \cosh(\beta_y \eta) - \cosh(\beta_x \xi) \cos(\alpha_y \eta) - \cos(\alpha_x \xi) \cosh(\beta_y \eta) + \cos(\alpha_x \xi) \cos(\alpha_y \eta)}{\alpha_x^2 \alpha_y^2 + \alpha_x^2 \beta_y^2 + \beta_x^2 \alpha_y^2 + \beta_x^2 \beta_y^2}$
b_{13}	xy^3	$\frac{\sinh(\beta_y \eta) \sin(\alpha_x \xi) - \sin(\alpha_y \eta) \sin(\alpha_x \xi)}{\alpha_x \alpha_y^3 + \alpha_x \beta_y^3}$
b_{14}	$x^3 y^2$	$\frac{\sinh(\beta_x \xi) \cosh(\beta_y \eta) - \sinh(\beta_x \xi) \cos(\alpha_y \eta) - \sin(\alpha_x \xi) \cosh(\beta_y \eta) + \sin(\alpha_x \xi) \cos(\alpha_y \eta)}{\alpha_x^3 \alpha_y^2 + \alpha_x^3 \beta_y^2 + \beta_x^3 \alpha_y^2 + \beta_x^3 \beta_y^2}$
b_{15}	$x^2 y^3$	$\frac{\cosh(\beta_x \xi) \sinh(\beta_y \eta) - \cosh(\beta_x \xi) \sin(\alpha_y \eta) - \cos(\alpha_x \xi) \sinh(\beta_y \eta) + \cos(\alpha_x \xi) \sin(\alpha_y \eta)}{\alpha_x^2 \alpha_y^3 + \alpha_x^2 \beta_y^3 + \beta_x^2 \alpha_y^3 + \beta_x^2 \beta_y^3}$
b_{16}	$x^3 y^3$	$\frac{\sinh(\beta_x \xi) \sinh(\beta_y \eta) - \sinh(\beta_x \xi) \sin(\alpha_y \eta) - \sin(\alpha_x \xi) \sinh(\beta_y \eta) + \sin(\alpha_x \xi) \sin(\alpha_y \eta)}{\alpha_x^3 \alpha_y^3 + \alpha_x^3 \beta_y^3 + \beta_x^3 \alpha_y^3 + \beta_x^3 \beta_y^3}$

The roots β_x , α_x , β_y and α_y were defined previously and marked as Eqs. (33), (34), (35) and (36), respectively. The expansion terms in Table 1 can be more concisely written as follows.

$$\langle P(\omega) \rangle_f = \langle b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6 \ b_7 \ b_8 \ b_9 \ b_{10} \ b_{11} \ b_{12} \ b_{13} \ b_{14} \ b_{15} \ b_{16} \rangle \quad (41)$$

Thus, the non-nodal approximation of the solution function W and the test function δW written in terms of generalised parameters will take the following form.

$$W = \langle P(\xi, \eta) \rangle_f * \{a\} \quad \delta W = \langle P(\xi, \eta) \rangle_f * \{\delta a\} \quad (42)$$

Replacing the generalized parameters, $\langle a \rangle$ and $\langle \delta a \rangle$ in Eq. (42) with the nodal variables, $\langle w_1 \theta_{x1} \theta_{y1} \theta_{xy1} \dots \dots w_4 \theta_{x4} \theta_{y4} \theta_{xy4} \rangle$ and $\langle \delta w_1 \delta \theta_{x1} \delta \theta_{y1} \delta \theta_{xy1} \dots \dots \delta w_4 \delta \theta_{x4} \delta \theta_{y4} \delta \theta_{xy4} \rangle$ re-writing Eq. (42) will result in Eq. (43).

$$\{W_n\} = [P_n]_f \{a\} \quad \{\delta W_n\} = [P_n]_f \{\delta a\} \quad (43)$$

The matrix, $[P_n]_f$ is defined as,

$$\begin{bmatrix} b_1(0,0) & b_2(0,0) & b_3(0,0) & b_4(0,0) & & & b_{13}(0,0) & b_{14}(0,0) & b_{15}(0,0) & b_{16}(0,0) \\ b_{1x}(0,0) & b_{2x}(0,0) & b_{3x}(0,0) & b_{4x}(0,0) & \text{---} & \text{---} & b_{13x}(0,0) & b_{14x}(0,0) & b_{15x}(0,0) & b_{16x}(0,0) \\ b_{1y}(0,0) & b_{2y}(0,0) & b_{3y}(0,0) & b_{4y}(0,0) & \text{---} & \text{---} & b_{13y}(0,0) & b_{14y}(0,0) & b_{15y}(0,0) & b_{16y}(0,0) \\ b_{1xy}(0,0) & b_{2xy}(0,0) & b_{3xy}(0,0) & b_{4xy}(0,0) & & & b_{13xy}(0,0) & b_{14xy}(0,0) & b_{15xy}(0,0) & b_{16xy}(0,0) \\ & | & | & & & & & | & | & \\ & | & | & & & & & | & | & \\ b_1(0,1) & b_2(0,1) & b_3(0,1) & b_4(0,1) & & & b_{13}(0,1) & b_{14}(0,1) & b_{15}(0,1) & b_{16}(0,1) \\ b_{1x}(0,1) & b_{2x}(0,1) & b_{3x}(0,1) & b_{4x}(0,1) & \text{---} & \text{---} & b_{13x}(0,1) & b_{14x}(0,1) & b_{15x}(0,1) & b_{16x}(0,1) \\ b_{1y}(0,1) & b_{2y}(0,1) & b_{3y}(0,1) & b_{4y}(0,1) & \text{---} & \text{---} & b_{13y}(0,1) & b_{14y}(0,1) & b_{15y}(0,1) & b_{16y}(0,1) \\ b_{1xy}(0,1) & b_{2xy}(0,1) & b_{3xy}(0,1) & b_{4xy}(0,1) & & & b_{13xy}(0,1) & b_{14xy}(0,1) & b_{15xy}(0,1) & b_{16xy}(0,1) \end{bmatrix} \quad (44)$$

Thus, Eq. (43) and the $[P_n]_f$ matrix can be combined in the following manner to construct nodal approximations for flexural displacement, $W(\xi, \eta)$.

$$W(\xi, \eta) = \langle P(\xi, \eta) \rangle_f [P_n]_f^{-1} \{w_n\} = \langle N(\xi, \eta) \rangle_f \{w_n\} \quad (45)$$

In Eq. (45), $\langle N(\xi, \eta) \rangle_f$, is the frequency dependent trigonometric shape functions for flexure which can also be re-written as,

$$\{W(\xi, \eta)\} = \langle N(\xi, \eta) \rangle \{w_n\} \quad (46)$$

where,

$$\{w_n\} = \langle w_1, \theta_{1x}, \theta_{1y}, \theta_{1xy}, \dots, w_4, \theta_{4x}, \theta_{4y}, \theta_{4xy} \rangle^T \quad (47)$$

and,

$$\langle N(\xi, \eta) \rangle = \langle N_{1f}, N_{2f}, N_{3f}, N_{4f}, N_{5f}, N_{6f}, N_{7f}, N_{8f}, N_{9f}, N_{10f}, N_{11f}, N_{12f}, N_{13f}, N_{14f}, N_{15f}, N_{16f} \rangle \quad (48)$$

The definitions of the new frequency-dependent trigonometric shape functions for flexure, N_I through N_{16} , are explicitly presented in expression (49) through (64).

$$\begin{aligned} N_{1f} = & \{ [\cos(\alpha_y \eta) + \cosh(\beta_y \eta) - \cos(\alpha_y) \cosh(\beta_y (\eta - 1)) - \cosh(\beta_y) \cos(\alpha_y (\eta - 1)) + \\ & (\alpha_y \sin(\alpha_y) \sinh(\beta_y (\eta - 1))) / \beta_y - (\beta_y \sinh(\beta_y) \sin(\alpha_y (\eta - 1))) / \alpha_y] [\cos(\alpha_x \xi) + \cosh(\beta_x \xi) \\ & - \cos(\alpha_x) \cosh(\beta_x (\xi - 1)) - \cosh(\beta_x) \cos(\alpha_x (\xi - 1)) + (\alpha_x \sin(\alpha_x) \sinh(\beta_x (\xi - 1))) / \beta_x - \\ & (\beta_x \sinh(\beta_x) \sin(\alpha_x (\xi - 1))) / \alpha_x] \} / \{ [2 \cos(\alpha_x) \cosh(\beta_x) + (\sin(\alpha_x) \sinh(\beta_x) (\alpha_x^2 - \beta_x^2)) / \\ & (\alpha_x \beta_x) - 2] [2 \cos(\alpha_y) \cosh(\beta_y) + (\sin(\alpha_y) \sinh(\beta_y) (\alpha_y^2 - \beta_y^2)) / (\alpha_y \beta_y) - 2] \} \end{aligned} \quad (49)$$

$$\begin{aligned} N_{2f} = & - \{ [\beta_x (\cosh(\beta_x) \sin(\alpha_x (\xi - 1)) - \sin(\alpha_x \xi) + \sin(\alpha_x) \cosh(\beta_x (\xi - 1))) + \\ & \alpha_x (\cos(\alpha_x) \sinh(\beta_x (\xi - 1)) - \sinh(\beta_x \xi) + \sinh(\beta_x) \cos(\alpha_x (\xi - 1)))] [\cos(\alpha_y \eta) + \cosh(\beta_y \eta) \\ & - \cos(\alpha_y) \cosh(\beta_y (\eta - 1)) - \cosh(\beta_y) \cos(\alpha_y (\eta - 1)) + (\alpha_y \sin(\alpha_y) \sinh(\beta_y (\eta - 1))) / \beta_y - \\ & (\beta_y \sinh(\beta_y) \sin(\alpha_y (\eta - 1))) / \alpha_y] \} / \{ \alpha_x \beta_x [2 \cos(\alpha_x) \cosh(\beta_x) + (\sin(\alpha_x) \sinh(\beta_x) (\alpha_x^2 - \beta_x^2)) / \\ & (\alpha_x \beta_x) - 2] [2 \cos(\alpha_y) \cosh(\beta_y) + (\sin(\alpha_y) \sinh(\beta_y) (\alpha_y^2 - \beta_y^2)) / (\alpha_y \beta_y) - 2] \} \end{aligned} \quad (50)$$

$$\begin{aligned} N_{3f} = & - \{ [\beta_y (\cosh(\beta_y) \sin(\alpha_y (\eta - 1)) - \sin(\alpha_y \eta) + \sin(\alpha_y) \cosh(\beta_y (\eta - 1))) + \\ & \alpha_y (\cos(\alpha_y) \sinh(\beta_y (\eta - 1)) - \sinh(\beta_y \eta) + \sinh(\beta_y) \cos(\alpha_y (\eta - 1)))] [\cos(\alpha_x \xi) + \cosh(\beta_x \xi) \\ & - \cos(\alpha_x) \cosh(\beta_x (\xi - 1)) - \cosh(\beta_x) \cos(\alpha_x (\xi - 1)) + (\alpha_x \sin(\alpha_x) \sinh(\beta_x (\xi - 1))) / \beta_x - \\ & (\beta_x \sinh(\beta_x) \sin(\alpha_x (\xi - 1))) / \alpha_x] \} / \{ \alpha_y \beta_y [2 \cos(\alpha_x) \cosh(\beta_x) + (\sin(\alpha_x) \sinh(\beta_x) (\alpha_x^2 - \beta_x^2)) / \\ & (\alpha_x \beta_x) - 2] [2 \cos(\alpha_y) \cosh(\beta_y) + (\sin(\alpha_y) \sinh(\beta_y) (\alpha_y^2 - \beta_y^2)) / (\alpha_y \beta_y) - 2] \} \end{aligned} \quad (51)$$

$$\begin{aligned} N_{4f} = & \{ [\beta_y (\cosh(\beta_y) \sin(\alpha_y (\eta - 1)) - \sin(\alpha_y \eta) + \sin(\alpha_y) \cosh(\beta_y (\eta - 1))) + \\ & \alpha_y (\cos(\alpha_y) \sinh(\beta_y (\eta - 1)) - \sinh(\beta_y \eta) + \sinh(\beta_y) \cos(\alpha_y (\eta - 1)))] [\beta_x (\cosh(\beta_x) \sin(\alpha_x (\xi - 1)) \\ & - \sin(\alpha_x \xi) + \sin(\alpha_x) \cosh(\beta_x (\xi - 1))) + \alpha_x (\cos(\alpha_x) \sinh(\beta_x (\xi - 1)) - \sinh(\beta_x \xi) + \sinh(\beta_x) \cdot \\ & \cos(\alpha_x (\xi - 1)))] \} / \{ \alpha_x \alpha_y \beta_x \beta_y [2 \cos(\alpha_x) \cosh(\beta_x) + (\sin(\alpha_x) \sinh(\beta_x) (\alpha_x^2 - \beta_x^2)) / \\ & (\alpha_x \beta_x) - 2] [2 \cos(\alpha_y) \cosh(\beta_y) + (\sin(\alpha_y) \sinh(\beta_y) (\alpha_y^2 - \beta_y^2)) / (\alpha_y \beta_y) - 2] \} \end{aligned} \quad (52)$$

$$\begin{aligned} N_{5f} = & \{ [\cos(\alpha_y \eta) + \cosh(\beta_y \eta) - \cos(\alpha_y) \cosh(\beta_y (\eta - 1)) - \cosh(\beta_y) \cos(\alpha_y (\eta - 1)) + \\ & (\alpha_y \sin(\alpha_y) \sinh(\beta_y (\eta - 1))) / \beta_y - (\beta_y \sinh(\beta_y) \sin(\alpha_y (\eta - 1))) / \alpha_y] [\cos(\alpha_x (\xi - 1)) + \\ & \cosh(\beta_x (\xi - 1)) - \cos(\alpha_x \xi) \cosh(\beta_x) - \cosh(\beta_x \xi) \cos(\alpha_x) - (\alpha_x \sinh(\beta_x \xi) \sin(\alpha_x)) / \beta_x + \\ & (\beta_x \sin(\alpha_x \xi) \sinh(\beta_x)) / \alpha_x] \} / \{ [2 \cos(\alpha_x) \cosh(\beta_x) + (\sin(\alpha_x) \sinh(\beta_x) (\alpha_x^2 - \beta_x^2)) / \\ & (\alpha_x \beta_x) - 2] [2 \cos(\alpha_y) \cosh(\beta_y) + (\sin(\alpha_y) \sinh(\beta_y) (\alpha_y^2 - \beta_y^2)) / (\alpha_y \beta_y) - 2] \} \end{aligned} \quad (53)$$

$$N_{6f} = \{[\beta_x(\sin(\alpha_x(\xi-1)) + \cosh(\beta_x\xi)\sin(\alpha_x) - \sin(\alpha_x\xi)\cosh(\beta_x)) + \alpha_x(\sinh(\beta_x(\xi-1)) + \cos(\alpha_x\xi)\sinh(\beta_x) - \sinh(\beta_x\xi)\cos(\alpha_x))][\cos(\alpha_y\eta) + \cosh(\beta_y\eta) - \cos(\alpha_y)\cosh(\beta_y(\eta-1)) - \cosh(\beta_y)\cos(\alpha_y(\eta-1)) + (\alpha_y\sin(\alpha_y)\sinh(\beta_y(\eta-1)))/\beta_y - (\beta_y\sinh(\beta_y)\sin(\alpha_y(\eta-1)))/\alpha_y] \} / \{ \alpha_x\beta_x[2\cos(\alpha_x)\cosh(\beta_x) + (\sin(\alpha_x)\sinh(\beta_x)(\alpha_x^2 - \beta_x^2))/(\alpha_x\beta_x) - 2][2\cos(\alpha_y)\cosh(\beta_y) + (\sin(\alpha_y)\sinh(\beta_y)(\alpha_y^2 - \beta_y^2))/(\alpha_y\beta_y) - 2] \} \quad (54)$$

$$N_{7f} = -\{[\beta_y(\cosh(\beta_y)\sin(\alpha_y(\eta-1)) - \sin(\alpha_y\eta) + \sin(\alpha_y)\cosh(\beta_y(\eta-1))) + \alpha_y(\cos(\alpha_y)\sinh(\beta_y(\eta-1)) - \sinh(\beta_y\eta) + \sinh(\beta_y)\cos(\alpha_y(\eta-1))][\cos(\alpha_x(\xi-1)) + \cosh(\beta_x(\xi-1)) - \cos(\alpha_x\xi)\cosh(\beta_x) - \cosh(\beta_x\xi)\cos(\alpha_x) - (\alpha_x\sinh(\beta_x\xi)\sin(\alpha_x))/\beta_x + (\beta_x\sin(\alpha_x\xi)\sinh(\beta_x))/\alpha_x] \} / \{ \alpha_y\beta_y[2\cos(\alpha_x)\cosh(\beta_x) + (\sin(\alpha_x)\sinh(\beta_x)(\alpha_x^2 - \beta_x^2))/(\alpha_x\beta_x) - 2][2\cos(\alpha_y)\cosh(\beta_y) + (\sin(\alpha_y)\sinh(\beta_y)(\alpha_y^2 - \beta_y^2))/(\alpha_y\beta_y) - 2] \} \quad (55)$$

$$N_{8f} = -\{[\beta_y(\cosh(\beta_y)\sin(\alpha_y(\eta-1)) - \sin(\alpha_y\eta) + \sin(\alpha_y)\cosh(\beta_y(\eta-1))) + \alpha_y(\cos(\alpha_y)\sinh(\beta_y(\eta-1)) - \sinh(\beta_y\eta) + \sinh(\beta_y)\cos(\alpha_y(\eta-1))][\beta_x(\sin(\alpha_x(\xi-1)) + \cosh(\beta_x\xi)\sin(\alpha_x) - \sin(\alpha_x\xi)\cosh(\beta_x)) + \alpha_x(\sinh(\beta_x(\xi-1)) + \cos(\alpha_x\xi)\sinh(\beta_x) - \sinh(\beta_x\xi)\cos(\alpha_x)) \} / \{ \alpha_x\alpha_y\beta_x\beta_y[2\cos(\alpha_x)\cosh(\beta_x) + (\sin(\alpha_x)\sinh(\beta_x)(\alpha_x^2 - \beta_x^2))/(\alpha_x\beta_x) - 2][2\cos(\alpha_y)\cosh(\beta_y) + (\sin(\alpha_y)\sinh(\beta_y)(\alpha_y^2 - \beta_y^2))/(\alpha_y\beta_y) - 2] \} \quad (56)$$

$$N_{9f} = \{[\cos(\alpha_y(\eta-1)) + \cosh(\beta_y(\eta-1)) - \cos(\alpha_y\eta)\cosh(\beta_y) - \cosh(\beta_y\eta)\cos(\alpha_y) - (\alpha_y\sinh(\beta_y\eta)\sin(\alpha_y))/\beta_y + (\beta_y\sin(\alpha_y\eta)\sinh(\beta_y))/\alpha_y][\cos(\alpha_x(\xi-1)) + \cosh(\beta_x(\xi-1)) - \cos(\alpha_x\xi)\cosh(\beta_x) - \cosh(\beta_x\xi)\cos(\alpha_x) - (\alpha_x\sinh(\beta_x\xi)\sin(\alpha_x))/\beta_x + (\beta_x\sin(\alpha_x\xi)\sinh(\beta_x))/\alpha_x] \} / \{ [2\cos(\alpha_x)\cosh(\beta_x) + (\sin(\alpha_x)\sinh(\beta_x)(\alpha_x^2 - \beta_x^2))/(\alpha_x\beta_x) - 2][2\cos(\alpha_y)\cosh(\beta_y) + (\sin(\alpha_y)\sinh(\beta_y)(\alpha_y^2 - \beta_y^2))/(\alpha_y\beta_y) - 2] \} \quad (57)$$

$$N_{10f} = \{[\beta_x(\sin(\alpha_x(\xi-1)) + \cosh(\beta_x\xi)\sin(\alpha_x) - \sin(\alpha_x\xi)\cosh(\beta_x)) + \alpha_x(\sinh(\beta_x(\xi-1)) + \cos(\alpha_x\xi)\sinh(\beta_x) - \sinh(\beta_x\xi)\cos(\alpha_x))][\cos(\alpha_y(\eta-1)) + \cosh(\beta_y(\eta-1)) - \cos(\alpha_y\eta)\cosh(\beta_y) - \cosh(\beta_y\eta)\cos(\alpha_y) - (\alpha_y\sinh(\beta_y\eta)\sin(\alpha_y))/\beta_y + (\beta_y\sin(\alpha_y\eta)\sinh(\beta_y))/\alpha_y] \} / \{ \alpha_x\beta_x[2\cos(\alpha_x)\cosh(\beta_x) + (\sin(\alpha_x)\sinh(\beta_x)(\alpha_x^2 - \beta_x^2))/(\alpha_x\beta_x) - 2][2\cos(\alpha_y)\cosh(\beta_y) + (\sin(\alpha_y)\sinh(\beta_y)(\alpha_y^2 - \beta_y^2))/(\alpha_y\beta_y) - 2] \} \quad (58)$$

$$N_{11f} = \{[\beta_y(\sin(\alpha_y(\eta-1)) + \cosh(\beta_y\eta)\sin(\alpha_y) - \sin(\alpha_y\eta)\cosh(\beta_y)) + \alpha_y(\sinh(\beta_y(\eta-1)) + \cos(\alpha_y\eta)\sinh(\beta_y) - \sinh(\beta_y\eta)\cos(\alpha_y))][\cos(\alpha_x(\xi-1)) + \cosh(\beta_x(\xi-1)) - \cos(\alpha_x\xi)\cosh(\beta_x) - \cosh(\beta_x\xi)\cos(\alpha_x) - (\alpha_x\sinh(\beta_x\xi)\sin(\alpha_x))/\beta_x + (\beta_x\sin(\alpha_x\xi)\sinh(\beta_x))/\alpha_x] \} / \{ \alpha_y\beta_y[2\cos(\alpha_x)\cosh(\beta_x) + (\sin(\alpha_x)\sinh(\beta_x)(\alpha_x^2 - \beta_x^2))/(\alpha_x\beta_x) - 2][2\cos(\alpha_y)\cosh(\beta_y) + (\sin(\alpha_y)\sinh(\beta_y)(\alpha_y^2 - \beta_y^2))/(\alpha_y\beta_y) - 2] \} \quad (59)$$

$$\begin{aligned}
N_{12f} = & \{[\beta_y (\sin(\alpha_y (\eta - 1)) + \cosh(\beta_y \eta) \sin(\alpha_y) - \sin(\alpha_y \eta) \cosh(\beta_y)) + \\
& \alpha_y (\sinh(\beta_y (\eta - 1)) + \cos(\alpha_y \eta) \sinh(\beta_y) - \sinh(\beta_y \eta) \cos(\alpha_y))] [\beta_x (\sin(\alpha_x (\xi - 1)) \\
& + \cosh(\beta_x \xi) \sin(\alpha_x) - \sin(\alpha_x \xi) \cosh(\beta_x)) + \alpha_x (\sinh(\beta_x (\xi - 1)) + \cos(\alpha_x \xi) \sinh(\beta_x) - \sinh(\beta_x \xi) \cdot \\
& \cos(\alpha_x))] \} / \{ \alpha_x \alpha_y \beta_x \beta_y [2 \cos(\alpha_x) \cosh(\beta_x) + (\sin(\alpha_x) \sinh(\beta_x) (\alpha_x^2 - \beta_x^2)) / \\
& (\alpha_x \beta_x) - 2] [2 \cos(\alpha_y) \cosh(\beta_y) + (\sin(\alpha_y) \sinh(\beta_y) (\alpha_y^2 - \beta_y^2)) / (\alpha_y \beta_y) - 2] \}
\end{aligned} \tag{60}$$

$$\begin{aligned}
N_{13f} = & \{[\cos(\alpha_x \xi) + \cosh(\beta_x \xi) - \cos(\alpha_x) \cosh(\beta_x (\xi - 1)) - \cosh(\beta_x) \cos(\alpha_x (\xi - 1)) + \\
& (\alpha_x \sin(\alpha_x) \sinh(\beta_x (\xi - 1))) / \beta_x - (\beta_x \sinh(\beta_x) \sin(\alpha_x (\xi - 1))) / \alpha_x] [\cos(\alpha_y (\eta - 1)) + \cosh(\beta_y (\eta - 1)) \\
& - \cos(\alpha_y \eta) \cosh(\beta_y) - \cosh(\beta_y \eta) \cos(\alpha_y) - (\alpha_y \sinh(\beta_y \eta) \sin(\alpha_y)) / \beta_y + \\
& (\beta_y \sin(\alpha_y \eta) \sinh(\beta_y)) / \alpha_y] \} / \{ [2 \cos(\alpha_x) \cosh(\beta_x) + (\sin(\alpha_x) \sinh(\beta_x) (\alpha_x^2 - \beta_x^2)) / \\
& (\alpha_x \beta_x) - 2] [2 \cos(\alpha_y) \cosh(\beta_y) + (\sin(\alpha_y) \sinh(\beta_y) (\alpha_y^2 - \beta_y^2)) / (\alpha_y \beta_y) - 2] \}
\end{aligned} \tag{61}$$

$$\begin{aligned}
N_{14f} = & -\{[\beta_x (\cosh(\beta_x) \sin(\alpha_x (\xi - 1)) - \sin(\alpha_x \xi) + \sin(\alpha_x) \cosh(\beta_x (\xi - 1))) + \\
& \alpha_x (\cos(\alpha_x) \sinh(\beta_x (\xi - 1)) - \sinh(\beta_x \xi) + \sinh(\beta_x) \cos(\alpha_x (\xi - 1)))] [\cos(\alpha_y (\eta - 1)) + \cosh(\beta_y (\eta - 1)) \\
& - \cos(\alpha_y \eta) \cosh(\beta_y) - \cosh(\beta_y \eta) \cos(\alpha_y) - (\alpha_y \sinh(\beta_y \eta) \sin(\alpha_y)) / \beta_y + \\
& (\beta_y \sin(\alpha_y \eta) \sinh(\beta_y)) / \alpha_y] \} / \{ \alpha_x \beta_x [2 \cos(\alpha_x) \cosh(\beta_x) + (\sin(\alpha_x) \sinh(\beta_x) (\alpha_x^2 - \beta_x^2)) / \\
& (\alpha_x \beta_x) - 2] [2 \cos(\alpha_y) \cosh(\beta_y) + (\sin(\alpha_y) \sinh(\beta_y) (\alpha_y^2 - \beta_y^2)) / (\alpha_y \beta_y) - 2] \}
\end{aligned} \tag{62}$$

$$\begin{aligned}
N_{15f} = & \{[\beta_y (\sin(\alpha_y (\eta - 1)) + \cosh(\beta_y \eta) \sin(\alpha_y) - \sin(\alpha_y \eta) \cosh(\beta_y)) + \\
& \alpha_y (\sinh(\beta_y (\eta - 1)) + \cos(\alpha_y \eta) \sinh(\beta_y) - \sinh(\beta_y \eta) \cos(\alpha_y))] [\cos(\alpha_x \xi) + \cosh(\beta_x \xi) \\
& - \cos(\alpha_x) \cosh(\beta_x (\xi - 1)) - \cosh(\beta_x) \cos(\alpha_x (\xi - 1)) + (\alpha_x \sin(\alpha_x) \sinh(\beta_x (\xi - 1))) / \beta_x - \\
& (\beta_x \sinh(\beta_x) \sin(\alpha_x (\xi - 1))) / \alpha_x] \} / \{ \alpha_y \beta_y [2 \cos(\alpha_x) \cosh(\beta_x) + (\sin(\alpha_x) \sinh(\beta_x) (\alpha_x^2 - \beta_x^2)) / \\
& (\alpha_x \beta_x) - 2] [2 \cos(\alpha_y) \cosh(\beta_y) + (\sin(\alpha_y) \sinh(\beta_y) (\alpha_y^2 - \beta_y^2)) / (\alpha_y \beta_y) - 2] \}
\end{aligned} \tag{63}$$

$$\begin{aligned}
N_{16f} = & -\{[\beta_x (\cosh(\beta_x) \sin(\alpha_x (\xi - 1)) - \sin(\alpha_x \xi) + \sin(\alpha_x) \cosh(\beta_x (\xi - 1))) + \\
& \alpha_x (\cos(\alpha_x) \sinh(\beta_x (\xi - 1)) - \sinh(\beta_x \xi) + \sinh(\beta_x) \cos(\alpha_x (\xi - 1)))] [\beta_y (\sin(\alpha_y (\eta - 1)) \\
& + \cosh(\beta_y \eta) \sin(\alpha_y) - \sin(\alpha_y \eta) \cosh(\beta_y)) + \alpha_y (\sinh(\beta_y (\eta - 1)) + \cos(\alpha_y \eta) \sinh(\beta_y) - \sinh(\beta_y \eta) \cdot \\
& \cos(\alpha_y))] \} / \{ \alpha_x \alpha_y \beta_x \beta_y [2 \cos(\alpha_x) \cosh(\beta_x) + (\sin(\alpha_x) \sinh(\beta_x) (\alpha_x^2 - \beta_x^2)) / \\
& (\alpha_x \beta_x) - 2] [2 \cos(\alpha_y) \cosh(\beta_y) + (\sin(\alpha_y) \sinh(\beta_y) (\alpha_y^2 - \beta_y^2)) / (\alpha_y \beta_y) - 2] \}
\end{aligned} \tag{64}$$

The aforementioned new trigonometric DFE shape functions are plotted against the polynomial FEM shape functions for comparison, and will be presented and further discussed in the next Chapter (Figures 8 through 15), and in Figures 20 through 31 in Appendix A. Although dependent on the frequency ω elsewhere (see Figures 32 to 43 in Appendix A), the trigonometric shape functions for bending are independent of the frequency at the element boundaries and as such they take the following values at each node. At node 1, $N_{1f} = 1$ and $N_{2f} = N_{3f} = N_{4f} = 0$. At the

second node, $N_{5f} = 1$ and $N_{6f} = N_{7f} = N_{8f} = 0$. Similarly at node 3, $N_{9f} = 1$ and $N_{10f} = N_{11f} = N_{12f} = 0$. Finally, at node 4, $N_{13f} = 1$ and $N_{14f} = N_{15f} = N_{16f} = 0$.

Another important feature to note regarding the dependency of these shape function on the natural frequency through the roots α_x , α_y , β_x and β_y , is that they can also be applied to study the static behaviour of thin rectangular plates when the natural frequency is set to zero as these trigonometric shape functions then become identical to the polynomial shape functions used in conventional FEM as mentioned in the beginning of this section.

Thusfar in this thesis the plate governing differential equation was derived starting from Hamiltonian mechanics and the Classical Plate Theory (CPT) and a quasi-exact solution to the plate governing equation was sorted. Building up on this quasi-exact solution, new trigonometric basis functions and frequency dependant shape functions were also presented for a 4 node, 4-DOF per node rectangular plate element. In the next Chapter this new solution will be used to develop a Quasi-Exact Dynamic Finite Element Method (QDFE) for a thin rectangular plate. But before forming the QDFE solution two conventional FEM formulations using 12-DOF and 16-DOF elements will be developed to validate the QDFE results. Furthermore, a Dynamic Coefficient Matrix Method (DCM) will also be presented followed by the graphical illustrations of the QDFE shape functions and the numerical results for the 12-DOF FEM, 16-DOF FEM, QDFE and DCM methods.

3. FREE VIBRATION OF A RECTANGULAR HOMOGENEOUS PLATE

The formulation of the QDFE method, which is essentially a finite element method in itself, starts at the discretized weak integral form of the governing equation after the boundary terms have vanished. In the conventional FEM method, this is where element discretization begins and the polynomial shape functions are applied to form the element stiffness and mass matrices. Thus, for this reason and for the purpose of validating the results produced by the QDFE method, two conventional FEM formulations will be developed using 12-DOF and 16-DOF rectangular plate elements in this chapter. Then starting from the discretized weak integral form of the conventional 16-DOF FEM method, a QDFE solution will be developed using the trigonometric shape functions presented in the previous chapter. Using the same solution to the governing differential equation presented in Section 3.2, a DCM method will also be presented for a 4 node, 4 DOF per node rectangular plate element. The surface plots for the QDFE shape functions will be plotted against the surface plots for the polynomial based shape functions used in conventional FEM in order to give the reader an opportunity to compare their identical nature and finally, the numerical results for all the methods outline above will also be discussed.

3.1 Conventional Finite Element Method (FEM)

Consider a linearly elastic, homogeneous, isotropic, thin plate as shown in Figure 3 having length, L , width, W , and thickness, h . The thickness h is assumed to be much smaller compared to the other characteristic dimensions as well as the wavelength. Furthermore, during vibration the rotary inertia and shear effects are neglected and only small deflections are assumed; i.e. linear vibration.

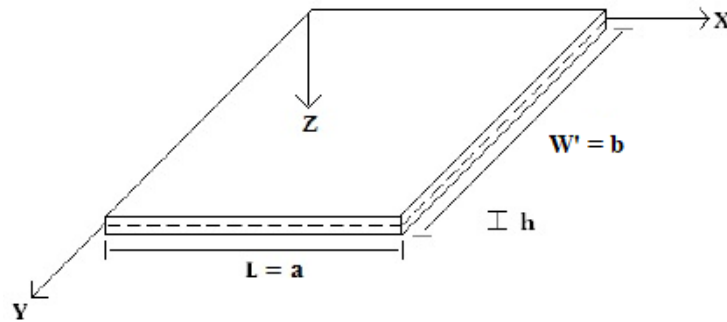


Figure 3: Thin plate used for vibration analysis

In what follows, the conventional Finite Element solution is formulated using two different thin rectangular plate elements. The first element is a 12–DOF element and the second element is a 16–DOF element. The 12–DOF element has 3–DOF per node, where the first DOF is the flexural (lateral) displacement w , along z -axis. The other two DOF's, in this case, are the rotations around x -axis, φ_x , and y -axis, φ_y , respectively. The 16–DOF element has 4–DOF per node, where again the first three DOF's of each node are the same as those of the 12–DOF element, with an additional degree of freedom per node, that is the curvature of the x - y plane, φ_{xy} . Both FEM formulations are included below as references, and to pave the road for the QDFE formulation. Both 12- and 16-DOF FEM elements were implemented in MATLAB[®] codes, where the elements are then assembled and the applicable boundary conditions are enforced to generate the linear Eigenproblem of the system. These codes are used to generate benchmark natural frequency/mode data for comparison and validation purposes, where such data are not available in the open literature.

3.1.1 Finite Element Analysis with 12–DOF Rectangular Element

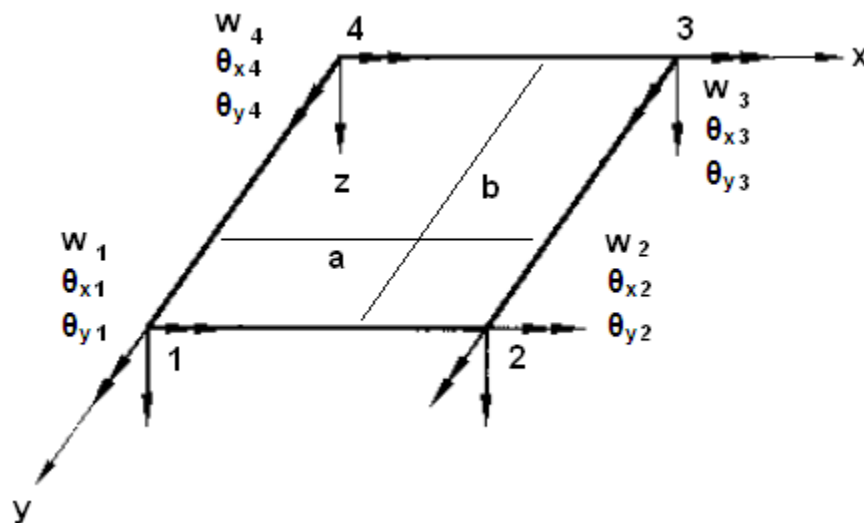


Figure 4: 4-Node, 3–DOF per node rectangular plate element

Figure 4 is an illustration of the 12-DOF rectangular element used in the development of the first plate element. In order to eliminate time dependency in Eq. (25), free, linear, harmonic vibration is considered and the following transformation is used.

$$w = We^{i\omega t} \quad (65)$$

where, W is the transverse displacement amplitude. Substituting (65) into Eq. (25), the latter can then be re-written in the following form, i.e., in terms of frequency, without the time dependency.

$$D \left(\frac{\partial^4 W}{\partial x^4} + 2 \frac{\partial^4 W}{\partial x^2 \partial y^2} + \frac{\partial^4 W}{\partial y^4} \right) - \omega^2 \rho h W = 0 \quad (66)$$

where, ω is the natural frequency. The definitions of the bending moments given in Eqs. (20), (22) and (24) can be written in the matrix form as follows.

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = -D \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu) \end{bmatrix} \begin{Bmatrix} \frac{\partial^2 W}{\partial x^2} \\ \frac{\partial^2 W}{\partial y^2} \\ \frac{\partial^2 W}{\partial x \partial y} \end{Bmatrix} \quad (67)$$

As will be further discussed later, the FEM formulation heavily depends on the application of the weighted residual method. At that point, it will be easier to spot the boundary terms and obtain the weak integral form when the governing partial differential equation is written in terms of bending and twisting moments. Therefore, the moment definitions in Eq. (67) are back-substituted into Eq. (66) to obtain the following form:

$$\left(\frac{\partial^2 M_x}{\partial x^2} + 2 \frac{\partial^2 M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_y}{\partial y^2} \right) - \omega^2 \rho h W = 0 \quad (68)$$

The Galerkin method of weighted residuals [(118)108] is then used to develop the integral form of Eq. (68). This is done by multiplying Eq. (68) by the weighting function, δW , and integrating over the area of the plate, which gives:

$$\bar{W}_f = \iint_A \left(\frac{\partial^2 M_x}{\partial x^2} + \frac{\partial^2 M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_y}{\partial y^2} - \omega^2 \rho h W \right) \delta W = 0 \quad (69)$$

Using Green's theorem [117], once on each of the derivative terms, results in the following expression:

$$\begin{aligned} & - \iint_A \left(\frac{\partial M_x}{\partial x} \frac{\partial(\delta W)}{\partial x} + \frac{\partial M_{xy}}{\partial y} \frac{\partial(\delta W)}{\partial x} + \frac{\partial M_{xy}}{\partial x} \frac{\partial(\delta W)}{\partial y} + \frac{\partial M_y}{\partial y} \frac{\partial(\delta W)}{\partial y} \right) dA - \\ & \iint_A \omega^2 \rho h W (\delta W) dA + \int_S \left(\frac{\partial M_x}{\partial x} + \frac{\partial M_{xy}}{\partial y} \right) n_x \delta W dS + \int_S \left(\frac{\partial M_y}{\partial y} + \frac{\partial M_{xy}}{\partial x} \right) n_y \delta W dS \end{aligned} \quad (70)$$

where, n_x and n_y are the components of the outward unit normal vector. Noting that the terms in parentheses in the boundary integrals are the definitions of the shear forces and replacing these with Q_x and Q_y gives:

$$\begin{aligned} & - \iint_A \left(\frac{\partial M_x}{\partial x} \frac{\partial(\delta W)}{\partial x} + \frac{\partial M_{xy}}{\partial y} \frac{\partial(\delta W)}{\partial x} + \frac{\partial M_{xy}}{\partial x} \frac{\partial(\delta W)}{\partial y} + \frac{\partial M_y}{\partial y} \frac{\partial(\delta W)}{\partial y} \right) dA - \\ & \iint_A \omega^2 \rho h W (\delta W) dA + \int_S Q_x n_x \delta W dS + \int_S Q_y n_y \delta W dS \end{aligned} \quad (71)$$

The application of the Green's theorem once more results in the weak integral form [117] of Eq. (69), written as:

$$\begin{aligned} & \iint_A \left(M_x \frac{\partial^2(\delta W)}{\partial x^2} + 2M_{xy} \frac{\partial^2(\delta W)}{\partial x \partial y} + M_y \frac{\partial^2(\delta W)}{\partial y^2} \right) dA - \iint_A \omega^2 \rho h W (\delta W) dA + \\ & \int_S (Q_x n_x + Q_y n_y) \delta W dS - \int_S (M_x n_x + M_{xy} n_y) \frac{\partial(\delta W)}{\partial x} dS - \int_S (M_{xy} n_x + M_y n_y) \frac{\partial(\delta W)}{\partial y} dS = 0 \end{aligned} \quad (72)$$

In Eq. (72), which is fully equivalent to Eq. (69), the boundary integral terms vanish and thus the following terms will remain.

$$\iint_A \left(M_x \frac{\partial^2(\delta W)}{\partial x^2} + 2M_{xy} \frac{\partial^2(\delta W)}{\partial x \partial y} + M_y \frac{\partial^2(\delta W)}{\partial y^2} \right) dA - \iint_A \omega^2 \rho h W (\delta W) dA = 0 \quad (73)$$

Back substituting the moment definitions shown in Eq. (67) in Eq. (73) in order to write the equation in terms of the flexural displacement, w , then results in:

$$\begin{aligned}
 & \begin{array}{ccc} \text{Term 1} & \text{Term 2} & \text{Term 3} \end{array} \\
 & \iint_A \left[\overbrace{D \frac{\partial^2 W}{\partial x^2} \left(\frac{\partial^2 (\delta W)}{\partial x^2} \right)} + \overbrace{D \nu \frac{\partial^2 W}{\partial y^2} \left(\frac{\partial^2 (\delta W)}{\partial x^2} \right)} + \overbrace{2D(1-\nu) \frac{\partial^2 W}{\partial x \partial y} \left(\frac{\partial^2 (\delta W)}{\partial x \partial y} \right)} + \right. \\
 & \left. \underbrace{D \nu \frac{\partial^2 W}{\partial x^2} \left(\frac{\partial^2 (\delta W)}{\partial y^2} \right)} + \underbrace{D \frac{\partial^2 W}{\partial y^2} \left(\frac{\partial^2 (\delta W)}{\partial y^2} \right)} + \underbrace{\omega^2 \rho h W \delta W} \right] dA = 0 \\
 & \begin{array}{ccc} \text{Term 4} & \text{Term 5} & \text{Term 6} \end{array}
 \end{aligned} \tag{74}$$

The expression in Eq. (74) also satisfies the principle of virtual work, written as:

$$\bar{W} = \bar{W}_{INT} - \bar{W}_{EXT} = 0 \tag{75}$$

where,

$$\bar{W}_{EXT} = 0 \tag{76}$$

for free vibrations and thus,

$$\bar{W}_{INT} = \bar{W}_f \tag{77}$$

The total virtual work, internal virtual work and external virtual work components are denoted by \bar{W} , \bar{W}_{INT} and \bar{W}_{EXT} , respectively.

The next step is to formulate the finite element solution. The discretization is performed using the 4-node, 3-DOF freedom per node element, as shown in Figure 4, such that,

$$\bar{W} = \bar{W}_{INT} = \sum_{k=1}^{\#elements} \bar{W}^k = \sum_{k=1}^{\#elements} \bar{W}_f^k \tag{78}$$

The shape functions used to develop the classical Finite Element formulation are developed based on the following 12-term polynomial non-nodal displacement function,

$$w(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 + \alpha_7 x^3 + \alpha_8 x^2 y + \alpha_9 xy^2 + \alpha_{10} y^3 + \alpha_{11} x^3 y + \alpha_{12} xy^3 \quad (79)$$

Thus, the above polynomial representing the deflection field over the finite element depicted in Figure 4 (designated by subscript e) can be written as:

$$w_e = [L]\{\alpha\} \quad (80)$$

where,

$$[L] = [1, x, y, x^2, xy, y^2, x^3, x^2 y, xy^2, y^3, x^3 y, xy^3] \quad (81)$$

and,

$$\{\alpha\} = \{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{10}, \alpha_{11}, \alpha_{12}\}^T \quad (82)$$

Introducing the element displacement matrix,

$$\{\delta_e\} = [\delta_1, \delta_2, \delta_3, \delta_4]^T \quad (83)$$

where, δ_i ($i = 1, 2, 3, 4$) are the nodal displacements such that the nodal displacement of the first node can be written as,

$$\{\delta_1\} = [w_1, (\partial w / \partial x)_1, (\partial w / \partial y)_1]^T = [w_1, \theta_{x1}, \theta_{y1}]^T \quad (84)$$

and the nodal displacements of the remaining nodes take an identical form. If the unknown coefficients α_m ($m = 1, 2, 3, \dots, 12$) are expressed via the nodal displacements above, then the following relationship can be written.

$$\{\delta_e\} = [C]\{\alpha\} \quad (85)$$

where, the matrix $[C]$ is a 12×12 matrix whose constituent elements depend on the x - and y -coordinates of the nodes of the finite element illustrated in Figure 4. The above relation can be then rearranged to determine the unknown coefficients, $\{\alpha\}$, as follows:

$$\{\alpha\}=[C]^{-1}\{\delta_e\} \quad (86)$$

Substituting Eq. (86) in Eq. (80) results in the nodal displacement function, written as:

$$w_e=[L][C]^{-1}\{\delta_e\} \quad (87)$$

where,

$$[N]=[L][C]^{-1} \quad (88)$$

are the cubic shape functions used to develop the finite element formulation. Given below are the 12 shape functions that result from the above mathematical manipulations (see also [117]).

For node 1:

$$N_1=\left[1-3\left(\frac{x}{a}\right)^2+2\left(\frac{x}{a}\right)^3\right]\left[1-3\left(\frac{y}{b}\right)^2+2\left(\frac{y}{b}\right)^3\right] \quad (88-a)$$

$$N_2=\left[x\left(\frac{x}{a}-1\right)^2\right]\left[1-3\left(\frac{y}{b}\right)^2+2\left(\frac{y}{b}\right)^3\right] \quad (88-b)$$

$$N_3=\left[1-3\left(\frac{x}{a}\right)^2+2\left(\frac{x}{a}\right)^3\right]\left[y\left(\frac{y}{b}-1\right)^2\right] \quad (88-c)$$

For node 2:

$$N_4=\left[3\left(\frac{x}{a}\right)^2-2\left(\frac{x}{a}\right)^3\right]\left[1-3\left(\frac{y}{b}\right)^2+2\left(\frac{y}{b}\right)^3\right] \quad (88-d)$$

$$N_5=\left[x\left(\frac{x}{a}\right)^2-\left(\frac{x^2}{a}\right)\right]\left[1-3\left(\frac{y}{b}\right)^2+2\left(\frac{y}{b}\right)^3\right] \quad (88-e)$$

$$N_6=\left[3\left(\frac{x}{a}\right)^2-2\left(\frac{x}{a}\right)^3\right]\left[y\left(\frac{y}{b}-1\right)^2\right] \quad (88-f)$$

For node 3:

$$N_7 = \left[3\left(\frac{x}{a}\right)^2 - 2\left(\frac{x}{a}\right)^3 \right] \left[3\left(\frac{y}{b}\right)^2 - 2\left(\frac{y}{b}\right)^3 \right] \quad (88-g)$$

$$N_8 = \left[x\left(\frac{x}{a}\right)^2 - \left(\frac{x^2}{a}\right) \right] \left[3\left(\frac{y}{b}\right)^2 - 2\left(\frac{y}{b}\right)^3 \right] \quad (88-h)$$

$$N_9 = \left[3\left(\frac{x}{a}\right)^2 - 2\left(\frac{x}{a}\right)^3 \right] \left[y\left(\frac{y}{b}\right)^2 - \left(\frac{y^2}{b}\right) \right] \quad (88-i)$$

For node 4:

$$N_{10} = \left[1 - 3\left(\frac{x}{a}\right)^2 + 2\left(\frac{x}{a}\right)^3 \right] \left[3\left(\frac{y}{b}\right)^2 - 2\left(\frac{y}{b}\right)^3 \right] \quad (88-j)$$

$$N_{11} = \left[x\left(\frac{x}{a} - 1\right)^2 \right] \left[3\left(\frac{y}{b}\right)^2 - 2\left(\frac{y}{b}\right)^3 \right] \quad (88-k)$$

$$N_{12} = \left[1 - 3\left(\frac{x}{a}\right)^2 + 2\left(\frac{x}{a}\right)^3 \right] \left[y\left(\frac{y}{b}\right)^2 - \left(\frac{y^2}{b}\right) \right] \quad (88-l)$$

Introducing the element displacements expressed (interpolated) using the above shape functions (88-a through 88-l) and the nodal displacements (see Eq. (87)), into expression (74), leads to the element stiffness and mass matrices. The element matrices, when assembled within the FEM code written in MATLAB[®], will result in the system's linear Eigenvalue problem shown in Eq. (89).

$$\begin{aligned} \langle \delta W_n \rangle (K_{12} - \omega^2 M_{12}) \{ W_n \} &= 0 \\ \det(K_{12} - \omega^2 M_{12}) &= 0 \end{aligned} \quad (89)$$

Here, the subscript *12* indicates that all ensuing matrices are unique to the 12-DOF FEM formulation. $[K]_{12}$ is the global stiffness matrix and $[M]_{12}$ is the global mass matrix. Boundary conditions are also enforced within the MATLAB[®] code. The five matrices (90) to (94) shown below ($[k_{ie}]_{12}$, $i=1-5$), are all sub-matrices that, when summed up, result in the final element

stiffness matrix, $[k_e]_{12}$, marked as (95). The right-hand side subscript e denotes that they are element stiffness matrices. The subscript on the left-hand side refers to the term in Eq. (35) which gives rise to each matrix. For example, matrix, $[k_{1e}]$, is an element stiffness matrix that results from the first term of Eq. (35).

$$[k_{1e}]_{12} = D \begin{bmatrix} N_{1xx} \cdot N_{1xx} & \text{---} & N_{1xx} \cdot N_{12xx} \\ | & & | \\ N_{12xx} \cdot N_{1xx} & \text{---} & N_{12xx} \cdot N_{12xx} \end{bmatrix} \quad (90)$$

$$[k_{2e}]_{12} = D\nu \begin{bmatrix} N_{1xx} \cdot N_{1yy} & \text{---} & N_{1xx} \cdot N_{12yy} \\ | & & | \\ N_{12xx} \cdot N_{1yy} & \text{---} & N_{12xx} \cdot N_{12yy} \end{bmatrix} \quad (91)$$

$$[k_{3e}]_{12} = 2D(1-\nu) \begin{bmatrix} N_{1xy} \cdot N_{1xy} & \text{---} & N_{1xy} \cdot N_{12xy} \\ | & & | \\ N_{12xy} \cdot N_{1xy} & \text{---} & N_{12xy} \cdot N_{12xy} \end{bmatrix} \quad (92)$$

$$[k_{4e}]_{12} = D\nu \begin{bmatrix} N_{1yy} \cdot N_{1xx} & \text{---} & N_{1yy} \cdot N_{12xx} \\ | & & | \\ N_{12yy} \cdot N_{1xx} & \text{---} & N_{12yy} \cdot N_{12xx} \end{bmatrix} \quad (93)$$

$$[k_{5e}]_{12} = D \begin{bmatrix} N_{1yy} \cdot N_{1yy} & \text{---} & N_{1yy} \cdot N_{12yy} \\ | & & | \\ N_{12yy} \cdot N_{1yy} & \text{---} & N_{12yy} \cdot N_{12yy} \end{bmatrix} \quad (94)$$

Thus, the final element stiffness matrix is:

$$[k_e]_{12} = [k_{1e}]_{12} + [k_{2e}]_{12} + [k_{3e}]_{12} + [k_{4e}]_{12} + [k_{5e}]_{12} \quad (95)$$

Matrix (96) is the element mass matrix.

$$[m]_{12} = -\rho h \omega^2 \begin{bmatrix} N_1 \cdot N_1 & \text{---} & N_1 \cdot N_{12} \\ | & & | \\ N_{12} \cdot N_1 & \text{---} & N_{12} \cdot N_{12} \end{bmatrix} \quad (96)$$

A 4-node, 3-DOF per node rectangular element satisfies C^0 continuity [117], and it is quite suitable for problems that require only continuity of the field variable w at the element boundaries. For problems requiring C^0 continuity, it is customary to use the nodal values of w to be the DOF of the element. Since w varies along an element edge, inter-element continuity is achieved when there are sufficient nodes and hence nodal values of w along the edges of the element that will allow for the determination of the variation. For example, if w displays a quadratic variation across a certain element boundary, then inter-element continuity of the field variable can only be achieved if there are three nodal values of w along that edge. To this end, there should be three nodes along the boundary [117]. Thus, by continuing to add nodes along element edges and through introducing extra DOF's to an element it is possible to easily satisfy the C^0 continuity requirement in an infinite number of ways. Generally, using such higher-order elements rapidly increases the accuracy of convergence as opposed to using lower-order elements [117]. While, this does not imply that one should always make higher-order elements a first choice in FEM analysis, one main drawback of the 4-node, 3-DOF per node (12 DOF in total) C^0 -continuous element, developed and presented in this section is that it only provides continuity of the dependant field variable and not its first derivatives.

C^1 continuity is an extremely important requirement for plate bending analysis. What C^1 continuity means is that inter-element continuity is achieved not only for the field variable w , but also for its normal derivative $\partial w / \partial n$. To preserve C^1 continuity one must ensure that w and $\partial w / \partial n$ are uniquely specified along the element boundaries by the degree of freedom assigned to the nodes along a particular boundary. In physical terms, the inability of the 12-DOF element to satisfy C^1 continuity means that the normal slope is discontinuous between any two adjacent elements because the normal slope has not been uniquely specified by the common nodal DOF's shared by the elements. Thus, such an element becomes a non-confirming or incompatible element [117].

For plate bending problems exploiting the Classical Plate Theory (CPT), the field variable is the flexural displacement of the midplane and it is essential that the displacement and its derivative are continuous across the element boundaries. Since, the functional for plate bending consists of second order derivatives, this is not only a physical requirement but also a mathematical requirement that will ensure convergence as element size is reduced. Thus, the failure of the 12–DOF element to provide C^1 continuity calls for the need for a higher–order element and to this end a 4-node, 4–DOF per node element is used to re-formulate the conventional Finite Element solution.

3.1.2 Finite Element Analysis with 16 – DOF Rectangular Element

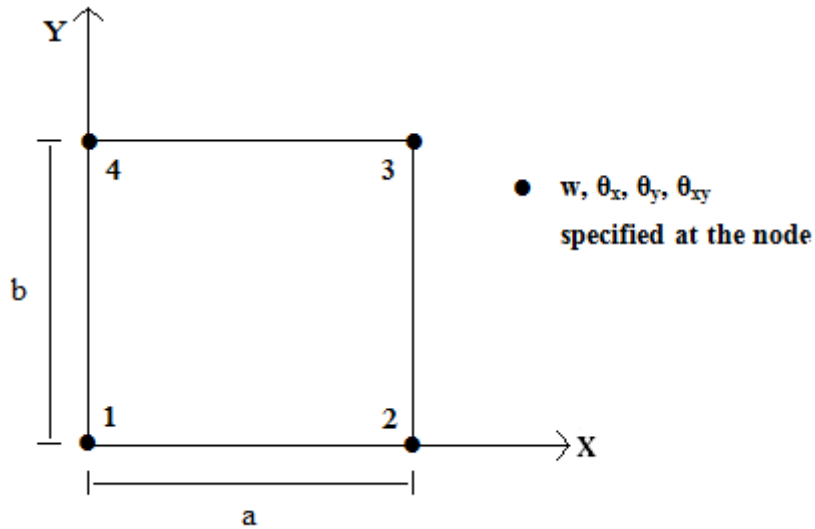


Figure 5: Conforming four node rectangular element with 16 – DOF ensuring C^1 continuity

Figure 5 depicts the 16 – DOF element studied. For this FEM formulation an incomplete quintic polynomial was assumed excluding the x^4 , x^4y , xy^4 and y^4 terms but incorporating the x^3y^3 term[117]. The resulting 16 unknown coefficients were found using the same procedure outlined in the previous section. With an additional fourth DOF, $\partial^2 w / \partial x \cdot \partial y$, at each node, this element is fully compatible and satisfies C^1 continuity. That is, it ensures continuity of the normal derivatives across adjacent element boundaries.

The formulation of the conventional FEM solution using the 16–DOF element starts from the point where the polynomial approximation function is defined. All mathematical manipulations

before and after this stage are exactly the same as for 12-DOF formulation and as such those steps are not repeated here for brevity. All that changes after this step are the definitions of the polynomial vector, $[L]$, the unknown coefficient vector, $\{\alpha\}$, and the nodal displacement vector, $\{\delta_i\}$. The 16-term function used to develop the classical Finite Element formulation is shown in Eq. (97)[117].

$$w(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 + \alpha_7 x^3 + \alpha_8 x^2 y + \alpha_9 xy^2 + \alpha_{10} y^3 + \alpha_{11} x^3 y + \alpha_{12} x^2 y^2 + \alpha_{13} xy^3 + \alpha_{14} x^3 y^2 + \alpha_{15} x^2 y^3 + \alpha_{16} x^3 y^3 \quad (97)$$

The polynomial vector, $[L]$, becomes:

$$[L] = [1, x, y, x^2, xy, y^2, x^3, x^2 y, xy^2, y^3, x^3 y, x^2 y^2, xy^3, x^3 y^2, x^2 y^3, x^3 y^3] \quad (98)$$

The unknown coefficient vector will now be:

$$[L] = [\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{14}, \alpha_{15}, \alpha_{16}] \quad (99)$$

The nodal displacement vector will now have four DOF's per node instead of three as in the previous case. For example, the nodal displacements for the first node will be:

$$\{\delta_1\} = [w_1, (\partial w / \partial x)_1, (\partial w / \partial y)_1, (\partial^2 w / \partial x \partial y)_1]^T = [w_1, \theta_{x1}, \theta_{y1}, \theta_{xy1}]^T \quad (100)$$

The $[C]$ matrix for this FEM formulation will have dimensions of 16 x 16. Performing the exact same mathematical manipulations as shown in Eq. (41) and Eqs. (46 – 49) will yield the 16 shape functions shown in Eqs. (101-a) to (101-p).

$$\langle N(x, y) \rangle = \langle N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8, N_9, N_{10}, N_{11}, N_{12}, N_{13}, N_{14}, N_{15}, N_{16} \rangle \quad (101)$$

where, N_I to N_{16} are expressed as follows.

For node 1:

$$N_1 = \frac{1}{a^3 b^3} \left[(a^3 + 2x^3 - 3ax^2)(b^3 + 2y^3 - 3by^2) \right] \quad (101-a)$$

$$N_2 = \frac{1}{a^3 b^3} \left[ax(x-a)^2(b^3 + 2y^3 - 3by^2) \right] \quad (101-b)$$

$$N_3 = \frac{1}{a^3 b^3} \left[b(a^3 + 2x^3 - 3ax^2)y(y-b)^2 \right] \quad (101-c)$$

$$N_4 = \frac{1}{a^3 b^3} \left[abxy(x-a)^2(y-b)^2 \right] \quad (101-d)$$

For node 2:

$$N_5 = \frac{1}{a^3 b^3} \left[(3ax^2 - 2x^3)(b^3 + 2y^3 - 3by^2) \right] \quad (101-e)$$

$$N_6 = \frac{1}{a^3 b^3} \left[a(x^3 - ax^2)(b^3 + 2y^3 - 3by^2) \right] \quad (101-f)$$

$$N_7 = \frac{1}{a^3 b^3} \left[b(3ax^2 - 2x^3)y(y-b)^2 \right] \quad (101-g)$$

$$N_8 = \frac{1}{a^3 b^3} \left[abxy(x^2 - ax)(y-b)^2 \right] \quad (101-h)$$

For node 3:

$$N_9 = \frac{1}{a^3 b^3} \left[(3ax^2 - 2x^3)(3by^2 - 2y^3) \right] \quad (101-i)$$

$$N_{10} = \frac{1}{a^3 b^3} \left[a(x^3 - ax^2)(3by^2 - 2y^3) \right] \quad (101-j)$$

$$N_{11} = \frac{1}{a^3 b^3} \left[b(3ax^2 - 2x^3)(y^3 - by^2) \right] \quad (101-k)$$

$$N_{12} = \frac{1}{a^3 b^3} \left[abxy(x^2 - ax)(y^2 - by) \right] \quad (101-l)$$

For node 4:

$$N_{13} = \frac{1}{a^3 b^3} \left[(a^3 + 2x^3 - 3ax^2)(3by^2 - 2y^3) \right] \quad (101-m)$$

$$N_{14} = \frac{1}{a^3 b^3} \left[ax(x-a)^2(3by^2 - 2y^3) \right] \quad (101-n)$$

$$N_{15} = \frac{1}{a^3 b^3} \left[b(a^3 + 2x^3 - 3ax^2)(y^3 - by^2) \right] \quad (101-o)$$

$$N_{16} = \frac{1}{a^3 b^3} \left[abxy(x-a)^2(y^2 - by) \right] \quad (101-p)$$

Once again, discretization leads to element and mass stiffness matrices, however, the element matrices for the 16 – DOF formulation are 16 x 16 in dimension. The final element stiffness matrix, $[k_e]_{16}$, marked as (102), is comprised of five sub-matrices, $[k_{ie}]_{16}$, which are numbered (103) to (107). The notations used for these matrices are similar to those used before.

$$[k_e]_{16} = [k_{1e}]_{16} + [k_{2e}]_{16} + [k_{3e}]_{16} + [k_{4e}]_{16} + [k_{5e}]_{16} \quad (102)$$

where,

$$[k_{1e}]_{16} = D \begin{bmatrix} N_{1xx} \cdot N_{1xx} & \text{---} & N_{1xx} \cdot N_{16xx} \\ | & & | \\ N_{16xx} \cdot N_{1xx} & \text{---} & N_{16xx} \cdot N_{16xx} \end{bmatrix} \quad (103)$$

$$[k_{2e}]_{16} = D\nu \begin{bmatrix} N_{1xx} \cdot N_{1yy} & \text{---} & N_{1xx} \cdot N_{16yy} \\ | & & | \\ N_{16xx} \cdot N_{1yy} & \text{---} & N_{16xx} \cdot N_{16yy} \end{bmatrix} \quad (104)$$

$$[k_{3e}]_{16} = 2D(1-\nu) \begin{bmatrix} N_{1xy} \cdot N_{1xy} & \text{---} & N_{1xy} \cdot N_{16xy} \\ | & & | \\ N_{16xy} \cdot N_{1xy} & \text{---} & N_{16xy} \cdot N_{16xy} \end{bmatrix} \quad (105)$$

$$[k_{4e}]_{16} = D\nu \begin{bmatrix} N_{1yy} \cdot N_{1xx} & \text{---} & N_{1yy} \cdot N_{16xx} \\ | & & | \\ N_{16yy} \cdot N_{1xx} & \text{---} & N_{16yy} \cdot N_{16xx} \end{bmatrix} \quad (106)$$

$$[k_{5e}]_{16} = D \begin{bmatrix} N_{1yy} \cdot N_{1yy} & \text{---} & N_{1yy} \cdot N_{16yy} \\ | & & | \\ N_{16yy} \cdot N_{1yy} & \text{---} & N_{16yy} \cdot N_{16yy} \end{bmatrix} \quad (107)$$

Matrix (108) is the element mass matrix.

$$[m]_{16} = -\rho h \omega^2 \begin{bmatrix} N_1 \cdot N_1 & \text{---} & N_1 \cdot N_{16} \\ | & & | \\ N_{16} \cdot N_1 & \text{---} & N_{16} \cdot N_{16} \end{bmatrix} \quad (108)$$

As with the 12 – DOF formulation, another MATLAB[®] code was written to assemble the element stiffness and mass matrices which gave rise to the global element, $[K]_{16}$ and global mass, $[M]_{16}$ matrices, respectively. Applying various boundary conditions, an Eigenvalue problem similar to that described in Eq. (89) was formed and solved to obtain the Eigensolutions for the plate system.

3.2 Quasi-Exact Dynamic Finite Element Method (QDFE)

In this section a Quasi-Exact Dynamic Finite Element (QDFE) formulation is developed for the homogeneous, isotropic, linearly elastic, thin, rectangular Steel plate depicted in Figure 3. The new trigonometric, frequency dependant shape functions (Section 3.3) based on the quasi-exact solutions to the plate governing equation (Section 3.2) will be used to develop the QDFE method. As mentioned previously, the QDFE method is a hybrid and intermediate method that combines the accuracy of the exact methods as well as the adaptability of the conventional FEM method to obtain a better finite element model. Therefore, the starting point of the QDFE formulation will be the discretized weak form of the equation from the conventional FEM shown in Eq. (109), after the boundary terms vanished.

$$\begin{aligned} \bar{W}_f^k(x, y) = & \int_{y_j}^{y_{j+1}} \int_{x_j}^{x_{j+1}} \left[\underbrace{D \frac{\partial^2 W}{\partial x^2} \left(\frac{\partial^2 (\delta W)}{\partial x^2} \right)}_{\text{Term 1}} + \underbrace{D \nu \frac{\partial^2 W}{\partial y^2} \left(\frac{\partial^2 (\delta W)}{\partial x^2} \right)}_{\text{Term 2}} + \underbrace{2D(1-\nu) \frac{\partial^2 W}{\partial x \partial y} \left(\frac{\partial^2 (\delta W)}{\partial x \partial y} \right)}_{\text{Term 3}} + \right. \\ & \left. \underbrace{D \nu \frac{\partial^2 W}{\partial x^2} \left(\frac{\partial^2 (\delta W)}{\partial y^2} \right)}_{\text{Term 4}} + \underbrace{D \frac{\partial^2 W}{\partial y^2} \left(\frac{\partial^2 (\delta W)}{\partial y^2} \right)}_{\text{Term 5}} + \underbrace{\omega^2 \rho h W \delta W}_{\text{Term 6}} \right] dx dy \end{aligned} \quad (109)$$

The green's theorem is applied, and integration by parts are carried out on each term in Eq. (109), except the mass Term 6. Terms 1 and 4 are integrated twice with respect to x , while Terms 2 and 5 are integrated twice with respect to y . Before integrating Term 3, it is re-written as follows and a special set of integration by part will be performed.

$$\begin{aligned} \iint_A D(1-\nu) \frac{\partial^2 w}{\partial x \partial y} \frac{\partial^2 (\delta w)}{\partial x \partial y} dA + \iint_A D(1-\nu) \frac{\partial^2 w}{\partial x \partial y} \frac{\partial^2 (\delta w)}{\partial x \partial y} dA \end{aligned} \quad (109\text{-a})$$

(a)
(b)

Two sets of integration by parts will be applied to Term (a) of Eq. (109-a), firstly with respect to x and secondly with respect to y . Similarly, for Term (b) of Eq. (109-a) another two sets of

integration by parts will be carried out, however, this time it will first be with respect to y and then with respect to x . Completing all the above-mentioned integration by parts leads to the following equation.

$$\begin{aligned}
& \left[\int_S D \frac{\partial w}{\partial x} \frac{\partial^2(\delta w)}{\partial x^2} n_x dS + \int_S D \nu \frac{\partial w}{\partial x} \frac{\partial^2(\delta w)}{\partial y^2} n_x dS + \int_S D(1-\nu) \frac{\partial w}{\partial x} \frac{\partial^2(\delta w)}{\partial x \partial y} n_y dS \right] \\
& \left[+ \int_S D \nu \frac{\partial w}{\partial y} \frac{\partial^2(\delta w)}{\partial x^2} n_y dS + \int_S D \frac{\partial w}{\partial y} \frac{\partial^2(\delta w)}{\partial y^2} n_y dS + \int_S D(1-\nu) \frac{\partial w}{\partial y} \frac{\partial^2(\delta w)}{\partial x \partial y} n_x dS \right] + \\
& \left[- \int_S D w \frac{\partial^3(\delta w)}{\partial x^3} n_x dS - \int_S D w \frac{\partial^3(\delta w)}{\partial x \partial y^2} n_x dS \right] + \left[- \int_S D w \frac{\partial^3(\delta w)}{\partial y^3} n_y dS - \int_S D w \frac{\partial^3(\delta w)}{\partial x^2 \partial y} n_y dS \right] \quad (110) \\
& + \iint_A D w \frac{\partial^4(\delta w)}{\partial x^4} dA + \iint_A 2D(1-\nu) w \frac{\partial^4(\delta w)}{\partial x^2 \partial y^2} dA + \iint_A D \nu w \frac{\partial^4(\delta w)}{\partial x^2 \partial y^2} dA \\
& + \iint_A D \nu w \frac{\partial^4(\delta w)}{\partial x^2 \partial y^2} dA + \iint_A D w \frac{\partial^4(\delta w)}{\partial y^4} dA - \iint_A \omega^2 \rho h w \delta w dA
\end{aligned}$$

The terms underlined in color are the boundary terms which are described as a product of the virtual loads and the real displacements. Underlined in red, blue and purple are the expressions for the moments M_x , M_y and M_{xy} , respectively. Furthermore, the term underlined in green represents the shear force Q_x and the expression underlined in yellow represents the shear force Q_y . Lastly, underlined in orange is the governing partial differential equation written in terms of virtual displacements. Eq. (110) can also be re-written as follows using the shear force definitions given in Eqs. (5-a) and (6-a) and the expressions for bending moment outlined in Eqs. (20), (22) and (24). The same color scheme above was used to illustrate how each expression was condensed.

$$\begin{aligned}
& - \int_S (M_x n_x + M_{xy} n_y) \frac{\partial w}{\partial x} dS - \int_S (M_y n_y + M_{xy} n_x) \frac{\partial w}{\partial y} dS + \int_S (Q_x n_x + Q_y n_y) w dS \\
& + \left[\iint_A D \frac{\partial^4(\delta w)}{\partial x^4} dA + \iint_A 2D \frac{\partial^4(\delta w)}{\partial x^2 \partial y^2} dA + \iint_A D \frac{\partial^4(\delta w)}{\partial y^4} dA - \iint_A \omega^2 \rho h \delta w \right] w dA \quad (111)
\end{aligned}$$

As mentioned previously, the expression underlined in orange above is the governing partial differential equation written in terms of virtual lateral displacement δw . Introducing $\xi=x/a$ and $\eta=y/b$ into expression (110) leads to its non-dimensionalized form, written as:

$$\begin{aligned}
& \left[\int_{\eta=0}^{\eta=1} \left[\frac{Db}{a^3} \frac{\partial w}{\partial \xi} \frac{\partial^2(\delta w)}{\partial \xi^2} \right]_{\xi=0}^{\xi=1} d\eta + \int_{\eta=0}^{\eta=1} \left[\frac{D\nu}{ab} \frac{\partial w}{\partial \xi} \frac{\partial^2(\delta w)}{\partial \eta^2} \right]_{\xi=0}^{\xi=1} d\eta + \int_{\xi=0}^{\xi=1} \left[\frac{D(1-\nu)}{ab} \frac{\partial w}{\partial \xi} \frac{\partial^2(\delta w)}{\partial \xi \partial \eta} \right]_{\eta=0}^{\eta=1} d\xi \right] \\
& + \left[\int_{\xi=0}^{\xi=1} \left[\frac{D\nu}{ab} \frac{\partial w}{\partial \eta} \frac{\partial^2(\delta w)}{\partial \xi^2} \right]_{\eta=0}^{\eta=1} d\xi + \int_{\xi=0}^{\xi=1} \left[\frac{Da}{b^3} \frac{\partial w}{\partial \eta} \frac{\partial^2(\delta w)}{\partial \eta^2} \right]_{\eta=0}^{\eta=1} d\xi + \int_{\eta=0}^{\eta=1} \left[\frac{D(1-\nu)}{ab} \frac{\partial w}{\partial \eta} \frac{\partial^2(\delta w)}{\partial \xi \partial \eta} \right]_{\xi=0}^{\xi=1} d\eta \right] \\
& + \left[- \int_{\eta=0}^{\eta=1} \left[\frac{Db}{a^3} w \frac{\partial^3(\delta w)}{\partial \xi^3} \right]_{\xi=0}^{\xi=1} d\eta - \int_{\eta=0}^{\eta=1} \left[\frac{D}{ab} w \frac{\partial^3(\delta w)}{\partial \xi \partial \eta^2} \right]_{\xi=0}^{\xi=1} d\eta \right] \\
& + \left[- \int_{\xi=0}^{\xi=1} \left[\frac{Da}{b^3} w \frac{\partial^3(\delta w)}{\partial \eta^3} \right]_{\eta=0}^{\eta=1} d\xi - \int_{\xi=0}^{\xi=1} \left[\frac{D}{ab} w \frac{\partial^3(\delta w)}{\partial \xi^2 \partial \eta} \right]_{\eta=0}^{\eta=1} d\xi \right] \\
& + \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \frac{Db}{a^3} w \frac{\partial^4(\delta w)}{\partial \xi^4} d\xi d\eta + \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \frac{2D}{ab} w \frac{\partial^4(\delta w)}{\partial \xi^2 \partial \eta^2} d\xi d\eta + \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \frac{Da}{b^3} w \frac{\partial^4(\delta w)}{\partial \eta^4} d\xi d\eta \\
& - \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} ab \omega^2 \rho h w \delta w d\xi d\eta
\end{aligned} \tag{112}$$

Thus, exploiting Eq. (112) and the trigonometric shape functions defined in Eqs. (49) to (64) results in the dynamic element stiffness matrix $[k_{DS}]$, which consists of ten sub-matrices. The two sub-matrices resulting from the bending moment M_x are as follows.

$$[k_{Mx1}] = \left(\frac{Db}{a^3} \right) \int_{\eta=0}^{\eta=1} \begin{bmatrix} \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_1}{\partial \xi^2} & \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \xi^2} \\ \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_1}{\partial \xi^2} & \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \xi^2} \end{bmatrix}_{\xi=0}^{\xi=1} d\eta \tag{113}$$

$$[k_{Mx2}] = \left(\frac{D\nu}{ab} \right) \int_{\eta=0}^{\eta=1} \begin{bmatrix} \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_1}{\partial \eta^2} & \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \eta^2} \\ \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_1}{\partial \eta^2} & \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \eta^2} \end{bmatrix}_{\xi=0}^{\xi=1} d\eta \tag{114}$$

where,

$$[k_{Mx}] = [k_{Mx1}] + [k_{Mx2}] \quad (115)$$

The bending moment M_y also contributes two sub – matrices towards the element dynamic stiffness matrix $[k_{DS}]$, written as:

$$[k_{My1}] = \left(\frac{D\nu}{ab} \right) \int_{\xi=0}^{\xi=1} \begin{bmatrix} \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_1}{\partial \xi^2} & \text{---} & \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \xi^2} \\ | & & | \\ \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_1}{\partial \xi^2} & \text{---} & \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \xi^2} \end{bmatrix}_{\eta=0}^{\eta=1} d\xi \quad (116)$$

$$[k_{My2}] = \left(\frac{Da}{b^3} \right) \int_{\xi=0}^{\xi=1} \begin{bmatrix} \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_1}{\partial \eta^2} & \text{---} & \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \eta^2} \\ | & & | \\ \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_1}{\partial \eta^2} & \text{---} & \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \eta^2} \end{bmatrix}_{\eta=0}^{\eta=1} d\xi \quad (117)$$

where,

$$[k_{My}] = [k_{My1}] + [k_{My2}] \quad (118)$$

The following matrices also appear due to actions of the twisting moment, M_{xy} .

$$[k_{Mxy1}] = \left(\frac{D(1-\nu)}{ab} \right) \int_{\xi=0}^{\xi=1} \begin{bmatrix} \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_1}{\partial \xi \partial \eta} & \text{---} & \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \xi \partial \eta} \\ | & & | \\ \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_1}{\partial \xi \partial \eta} & \text{---} & \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \xi \partial \eta} \end{bmatrix}_{\eta=0}^{\eta=1} d\xi \quad (119)$$

$$[k_{Mxy2}] = \left(\frac{D(1-\nu)}{ab} \right) \int_{\eta=0}^{\eta=1} \begin{bmatrix} \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_1}{\partial \xi \partial \eta} & \text{---} & \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \xi \partial \eta} \\ | & & | \\ \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_1}{\partial \xi \partial \eta} & \text{---} & \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \xi \partial \eta} \end{bmatrix}_{\xi=0}^{\xi=1} d\eta \quad (120)$$

where,

$$[k_{M_{xy}}] = [k_{M_{xy1}}] + [k_{M_{xy2}}] \quad (121)$$

Numbered (122) and (123) are the two sub – matrices that is resulted due to the shear force, Q_x .

$$[k_{Q_{x1}}] = - \left(\frac{Db}{a^3} \right) \int_{\eta=0}^{\eta=1} \begin{bmatrix} N_1 \frac{\partial^3 N_1}{\partial \xi^3} & \text{---} & N_1 \frac{\partial^3 N_{16}}{\partial \xi^3} \\ | & & | \\ N_{16} \frac{\partial^3 N_1}{\partial \xi^3} & \text{---} & N_{16} \frac{\partial^3 N_{16}}{\partial \xi^3} \end{bmatrix}_{\xi=0}^{\xi=1} d\eta \quad (122)$$

$$[k_{Q_{x2}}] = - \left(\frac{D}{ab} \right) \int_{\eta=0}^{\eta=1} \begin{bmatrix} N_1 \frac{\partial^3 N_1}{\partial \xi \partial \eta^2} & \text{---} & N_1 \frac{\partial^3 N_{16}}{\partial \xi \partial \eta^2} \\ | & & | \\ N_{16} \frac{\partial^3 N_1}{\partial \xi \partial \eta^2} & \text{---} & N_{16} \frac{\partial^3 N_{16}}{\partial \xi \partial \eta^2} \end{bmatrix}_{\xi=0}^{\xi=1} d\eta \quad (123)$$

where,

$$[k_{Q_x}] = [k_{Q_{x1}}] + [k_{Q_{x2}}] \quad (124)$$

The shear force Q_y contributes the following sub – matrices towards the element stiffness matrix, $[k_{DS}]$.

$$[k_{Q_{y1}}] = - \left(\frac{Da}{b^3} \right) \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 \frac{\partial^3 N_1}{\partial \eta^3} & \text{---} & N_1 \frac{\partial^3 N_{16}}{\partial \eta^3} \\ | & & | \\ N_{16} \frac{\partial^3 N_1}{\partial \eta^3} & \text{---} & N_{16} \frac{\partial^3 N_{16}}{\partial \eta^3} \end{bmatrix}_{\eta=0}^{\eta=1} d\xi \quad (125)$$

$$[k_{Q_{y2}}] = - \left(\frac{D}{ab} \right) \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 \frac{\partial^3 N_1}{\partial \xi^2 \partial \eta} & \text{---} & N_1 \frac{\partial^3 N_{16}}{\partial \xi^2 \partial \eta} \\ | & & | \\ N_{16} \frac{\partial^3 N_1}{\partial \xi^2 \partial \eta} & \text{---} & N_{16} \frac{\partial^3 N_{16}}{\partial \xi^2 \partial \eta} \end{bmatrix}_{\eta=0}^{\eta=1} d\xi \quad (126)$$

where,

$$[k_{Q_y}] = [k_{Q_{y1}}] + [k_{Q_{y2}}] \quad (127)$$

The element stiffness matrices arising from the expression underlined in orange in Eq. (112) are marked as (128), (129) and (130).

$$[k_{G1}] = \left(\frac{Db}{a^3} \right) \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 \frac{\partial^4 N_1}{\partial \xi^4} & \text{---} & N_1 \frac{\partial^4 N_{16}}{\partial \xi^4} \\ | & & | \\ N_{16} \frac{\partial^4 N_1}{\partial \xi^4} & \text{---} & N_{16} \frac{\partial^4 N_{16}}{\partial \xi^4} \end{bmatrix} d\xi d\eta \quad (128)$$

$$[k_{G2}] = \left(\frac{2D}{ab} \right) \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 \frac{\partial^4 N_1}{\partial \xi^2 \partial \eta^2} & \text{---} & N_1 \frac{\partial^4 N_{16}}{\partial \xi^2 \partial \eta^2} \\ | & & | \\ N_{16} \frac{\partial^4 N_1}{\partial \xi^2 \partial \eta^2} & \text{---} & N_{16} \frac{\partial^4 N_{16}}{\partial \xi^2 \partial \eta^2} \end{bmatrix} d\xi d\eta \quad (129)$$

$$[k_{G3}] = \left(\frac{Da}{b^3} \right) \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 \frac{\partial^4 N_1}{\partial \eta^4} & \text{---} & N_1 \frac{\partial^4 N_{16}}{\partial \eta^4} \\ | & & | \\ N_{16} \frac{\partial^4 N_1}{\partial \eta^4} & \text{---} & N_{16} \frac{\partial^4 N_{16}}{\partial \eta^4} \end{bmatrix} d\xi d\eta \quad (130)$$

The element mass matrix arising from the governing differential equation written in terms of virtual work is shown in Eq. (131).

$$[m_{DS}] = ab\omega^2 \rho h \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 N_1 & \text{---} & N_1 N_{16} \\ | & & | \\ N_{16} N_1 & \text{---} & N_{16} N_{16} \end{bmatrix} d\xi d\eta \quad (131)$$

The final element dynamic stiffness matrix $[k_{DS}]$ is determined by assembling these ten sub – matrices, leading to the discretized (elemental) expression of the virtual work, as shown in Eq. (132).

$$[k_{DS}] = \{ [k_{M_x}] + [k_{M_y}] + [k_{M_{xy}}] + [k_{Q_x}] + [k_{Q_y}] + [k_{G1}] + [k_{G2}] + [k_{G3}] \} \quad (132)$$

Where necessary and applicable, the global dynamic stiffness matrix, $[K_{DS}]$, and the global mass matrix, $[K_M]$, are then obtained by assembling all the element stiffness and mass matrices and enforcing the system boundary conditions. This process is performed using a program written in MATLAB[®] software as shown in Appendix B, which results in the Eigenvalue problem expressed as Eq. (133).

$$\begin{aligned} \langle \delta W_n \rangle (K_{DS} - \omega^2 M_{DS}) \{W_n\} &= 0 \\ \det(K_{DS} - \omega^2 M_{DS}) &= 0 \end{aligned} \quad (133)$$

Various classical boundary conditions were investigated. The natural frequencies of the system are the values of ω that yield a zero determinant for the Eigenvalue problem in Eq. (133), evaluated using a determinant search. This is done by sweeping the frequency domain to find particular values of ω that produce a zero determinant. The Eigenvectors corresponding to these natural frequencies provide the mode shapes of the system.

3.3 Dynamic Coefficient Matrix Method (DCM)

In this section, a new Dynamic Coefficient Matrix Method (DCM) based on the quasi-exact solutions to the plate governing equation will be formed. This analytical method, previously used by some researchers to model the free vibration behaviour of various structural elements, can be considered as a special case of Dynamic Stiffness Method (DSM), developed by Banerjee and his co-workers [67-78]. However, to the best of authors knowledge, this method has never been attempted before for the free vibration of thin plates, and using the novel semi-exact solution to the governing equation, introduced earlier in this report.

The starting point of the DCM method is the quasi-exact general solution to the differential equation, governing the free vibration of thin rectangular plates, developed and presented for the first time as Eq. (39), in Section 3.2. Thus, using this quasi-exact general solution the non-nodal flexural displacement, $W(x, y)$, anywhere in the plate can be written in the matrix form as:

$$W(x, y) = \begin{Bmatrix} T_1 & T_2 & \cdot & \cdot & T_{15} & T_{16} \end{Bmatrix} \begin{Bmatrix} E_{11} \\ E_{12} \\ \cdot \\ \cdot \\ E_{43} \\ E_{44} \end{Bmatrix} ; \quad \text{or} \quad W(x, y) = \langle T \rangle_{1 \times 16} \{E\}_{16 \times 1} \quad (134)$$

where, the row vector $\langle T \rangle$ is the solution vector which contains the roots to the plate governing differential equation and the column vector $\{E\}$ is the vector of unknown coefficients. The slope along the x direction can then be written as follows.

$$\theta_x(x, y) = \begin{Bmatrix} T_{1x} & T_{2x} & \cdot & \cdot & T_{15x} & T_{16x} \end{Bmatrix} \{E\}_{16 \times 1} = \langle T_{ix} \rangle_{1 \times 16} \{E\}_{16 \times 1} \quad (135)$$

In Eq. (135) the row vector $\langle T_{ix} \rangle$ is determined by differentiating the solution vector $\langle T \rangle$ with respect to x . Similarly, the slope along the y direction can be expressed as:

$$\theta_y(x, y) = \begin{Bmatrix} T_{1y} & T_{2y} & \cdot & \cdot & T_{15y} & T_{16y} \end{Bmatrix} \{E\}_{16 \times 1} = \langle T_{iy} \rangle_{1 \times 16} \{E\}_{16 \times 1} \quad (136)$$

where, the row vector $\langle T_{iy} \rangle$ is obtained by taking the derivatives of the roots T_1 to T_{16} contained within the with solution vector, with respect to y . The curvature of the plate, $\theta_{xy}(x, y)$, can also be represented as follows.

$$\theta_{xy}(x, y) = \langle T_{1,xy} \quad T_{2,xy} \quad \cdot \quad \cdot \quad T_{15,xy} \quad T_{16,xy} \rangle \{E\}_{16 \times 1} = \langle T_{ixy} \rangle_{1 \times 16} \{E\}_{16 \times 1} \quad (137)$$

where, the row vector, $\langle T_{ixy} \rangle$, is determined by obtaining the derivatives of the solution vector with respect x and y both.

The end conditions for the displacements, at the element extremities, are:

$$\begin{aligned} x=0, y=0, \quad W &= W_1; \quad \theta_x = \theta_{x1}; \quad \theta_y = \theta_{y1}; \quad \theta_{xy} = \theta_{xy1} \\ x=a, y=0, \quad W &= W_2; \quad \theta_x = \theta_{x2}; \quad \theta_y = \theta_{y2}; \quad \theta_{xy} = \theta_{xy2} \\ x=a, y=b, \quad W &= W_3; \quad \theta_x = \theta_{x3}; \quad \theta_y = \theta_{y3}; \quad \theta_{xy} = \theta_{xy3} \\ x=0, y=b, \quad W &= W_4; \quad \theta_x = \theta_{x4}; \quad \theta_y = \theta_{y4}; \quad \theta_{xy} = \theta_{xy4} \end{aligned} \quad (138)$$

By applying the boundary conditions for displacements, i.e. substituting Eq. (138) in to Eqs. (134) to (137) the following matrix relationship is obtained.

$$\begin{Bmatrix} W_1 \\ \theta_{x1} \\ \theta_{y1} \\ \theta_{xy1} \\ W_2 \\ \theta_{x2} \\ \theta_{y2} \\ \theta_{xy2} \\ W_3 \\ \theta_{x3} \\ \theta_{y3} \\ \theta_{xy3} \\ W_4 \\ \theta_{x4} \\ \theta_{y4} \\ \theta_{xy4} \end{Bmatrix} = \begin{bmatrix} \langle T_i \rangle_{x=0,y=0} \\ \langle T_{ix} \rangle_{x=0,y=0} \\ \langle T_{iy} \rangle_{x=0,y=0} \\ \langle T_{ixy} \rangle_{x=0,y=0} \\ \langle T_i \rangle_{x=a,y=0} \\ \langle T_{ix} \rangle_{x=a,y=0} \\ \langle T_{iy} \rangle_{x=a,y=0} \\ \langle T_{ixy} \rangle_{x=a,y=0} \\ \langle T_i \rangle_{x=a,y=b} \\ \langle T_{ix} \rangle_{x=a,y=b} \\ \langle T_{iy} \rangle_{x=a,y=b} \\ \langle T_{ixy} \rangle_{x=a,y=b} \\ \langle T_i \rangle_{x=0,y=b} \\ \langle T_{ix} \rangle_{x=0,y=b} \\ \langle T_{iy} \rangle_{x=0,y=b} \\ \langle T_{ixy} \rangle_{x=0,y=b} \end{bmatrix} \begin{Bmatrix} E_{11} \\ E_{12} \\ E_{13} \\ E_{14} \\ E_{21} \\ E_{22} \\ E_{23} \\ E_{24} \\ E_{31} \\ E_{32} \\ E_{33} \\ E_{34} \\ E_{41} \\ E_{42} \\ E_{43} \\ E_{44} \end{Bmatrix} \quad (139)$$

The expression in Eq. (139) can be simplified and re-written as:

$$\{W_n\} = [K_D(\omega)]_{16 \times 16} \{E\}_{16 \times 1} \quad (140)$$

where, $[K_D(\omega)]$ is the 16x16 Dynamic Coefficient Matrix of the system. The Dynamic Coefficient Matrix in Eq. (140) consists of the essential requirements to compute the natural frequencies for a thin rectangular plate subjected to any boundary condition. In order to obtain the system's natural frequencies using the DCM method, boundary conditions are applied on the Dynamic Coefficient Matrix and a determinant sweep was conducted by sweeping the frequency domain in search of frequencies at which the determinant of the Dynamic Coefficient Matrix will be zero; i.e., $|K_D(\omega)| = 0$.

3.4 Numerical Results and Discussion

Consider a thin, homogeneous, isotropic, linearly elastic, rectangular plate made of Steel ($E = 200 \times 10^9$ MPa and $\rho = 7800$ kg/m³) having length $L = 0.6$ m, width $W = 0.4$ m and thickness $h = 0.004$ m. In order to validate the results from the 12-DOF and 16-DOF conventional FEM codes, a modal analysis was carried out on the commercial software ANSYS[®] using the same number of elements used in the conventional FEM formulations. The number of elements required was decided after carrying out a convergence analysis using the 12-DOF FEM-based MATLAB[®] code, to determine the number of elements required to drive down the error of the 5th natural frequency of the Cantilevered (C-C-C-F) plate to less than 10 percent when compared with the exact results taken from Reference [118]. The same number of elements was then used for the 16-DOF FEM formulation and the ANSYS[®] simulation for comparison.

The 12-DOF FEM formulation was used as the basis of the convergence analysis as its results displayed a higher percentage of error compared to the ANSYS[®] and 16-DOF FEM results. Similarly, the 5th natural frequency and the C-C-C-F boundary condition was used for the convergence study as the percent error was comparatively higher for those cases too. The results of the convergence analysis are shown in Figure 6. The convergence graphs for the ANSYS[®] model as well as the 16-DOF FEM solution are also included in Figure 6 for comparison.

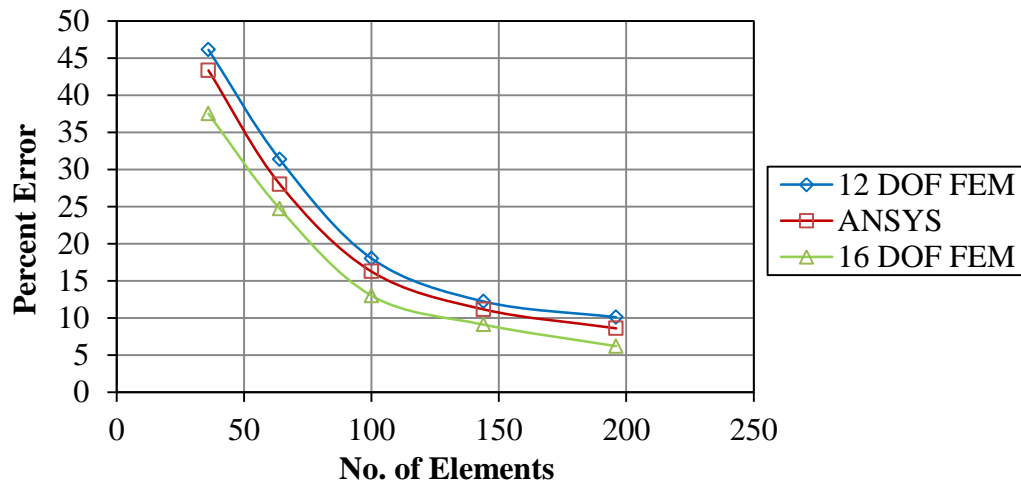


Figure 6: Convergence analysis for the 5th natural frequency of a C-C-C-F plate

A percent error of 10 percent is taken as the baseline. Therefore, Figure 6 shows that at least 196 elements are required for the 12-DOF FEM formulation, which is the weakest of the three models in terms of convergence rate, to produce results with a percent error less than 10 percent. Thus, 196 elements were used for the 12-DOF FEM model. Despite having higher rates of convergence, 196 elements were also used for the 16-DOF FEM formulation as well as the ANSYS® simulation for ease of comparison. The 3D, 4 node, SHELL 181 element was used for meshing in ANSYS®. This element has 6 DOFs per node and these are the three translations and three rotations in the x -, y - and z -axes. Figure 7a is a sketch of the ANSYS® SHELL 181 element and Figure 7b is an illustration of the plate structure mesh created in ANSYS®.

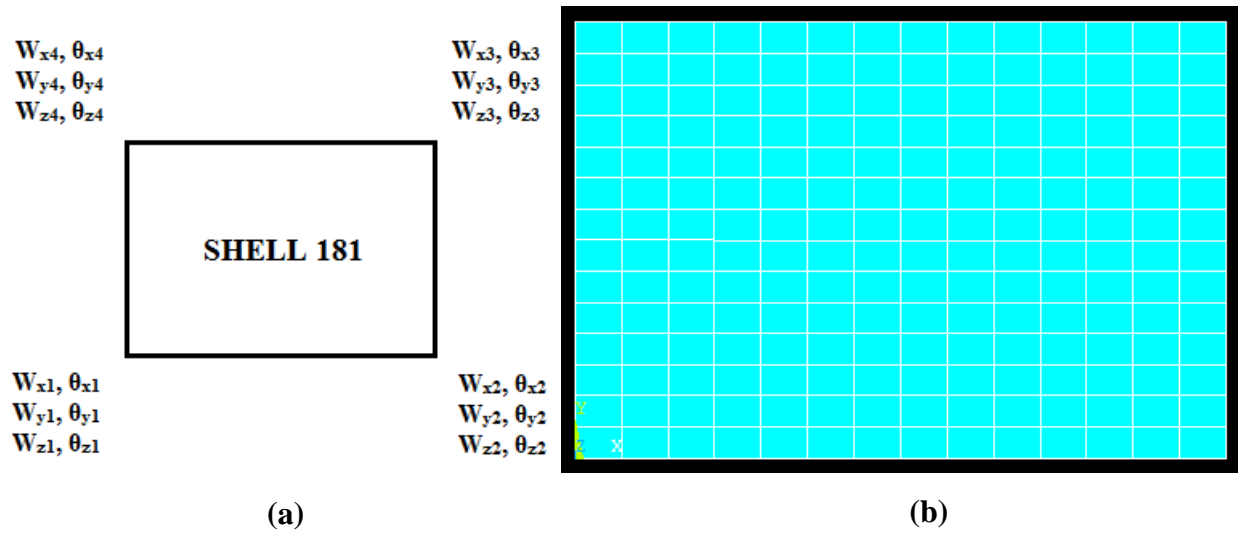


Figure 7: (a) ANSYS® SHELL 181 element used for meshing (b) Plate meshed in ANSYS®

Before beginning, some descriptive comments regarding the notation for the boundary conditions, which apply to the entire thesis, should be made here. The symbolism C-F-F-F for example, will identify in clockwise motion, a rectangular plate with its left boundary clamped and the bottom, right and top boundaries free. Figure 8 below shows the notation used for all the classical boundary conditions investigated in this study.

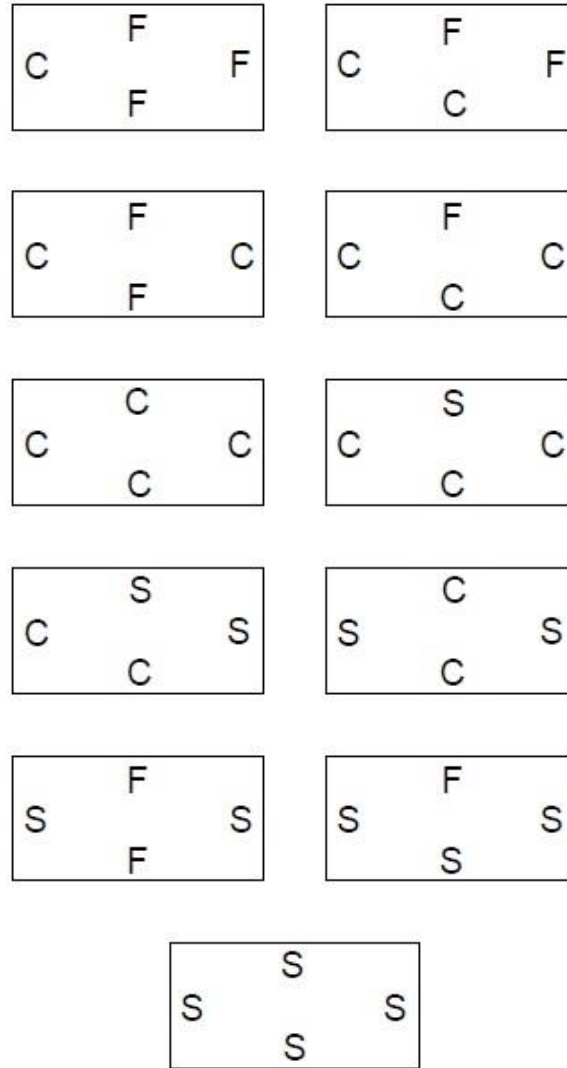


Figure 8: Notation for the classical boundary conditions investigated

Prior to capturing any results using the FEM codes, they were validated against ANSYS[®] results. Table 2 shows the 12-DOF FEM, 16-DOF FEM and ANSYS[®] results for a cantilevered plate with one edge clamped and the other three edges free (C-F-F-F).

Table 2: Comparison of 12-DOF FEM and 16-DOF FEM results with ANSYS® simulation

Natural Frequencies (Hz) for C-F-F-F plate						
Exact Natural Freqs. [118]	16-DOF FEM	% Error 16-DOF FEM vs. Exact	ANSYS	% Error ANSYS vs. Exact	12-DOF FEM	% Error 12-DOF FEM vs. Exact
	# Elem: 196		# Elem: 196		# Elem: 196	
9.36	9.36	0.00	9.36	0.00	9.37	0.11
31.51	31.59	0.25	31.60	0.29	31.61	0.32
58.15	58.71	0.96	58.89	1.27	59.01	1.48
106.29	107.20	0.86	107.43	1.07	107.58	1.21
144.92	146.04	0.77	146.35	0.99	146.56	1.13

As can be seen from Table 2, the ANSYS® frequency results match well with the exact results and the results from the 12-DOF and 16-DOF FEM formulations. As far as the ANSYS® simulation is concerned, the highest error of 1.27 percent is found for the third natural frequency. The highest error also occurs at the third natural frequency for both the 16-DOF FEM and 12-DOF FEM formulations, and the errors are 0.96 percent and 1.48 percent, respectively. Thus, in terms of accuracy, the ANSYS® results falls in between the 16-DOF FEM and 12-DOF FEM formulations. In other words, although the same number of elements are used the results produced by the 16-DOF FEM formulation is more accurate than the ANSYS® simulation while those obtained from the 12-DOF FEM formulation are less accurate. The higher accuracy of the 16-DOF FEM formulation compared to the ANSYS® simulation could be associated with the fact that the 3D, 4-node, SHELL 181 element used in ANSYS® takes into consideration shear and warping effects that are not accounted for in the 16-DOF FEM code. The lower accuracy of the 12-DOF FEM formulation compared to the ANSYS® simulation and the 16-DOF FEM formulation could be attributed to the lesser degrees of freedom associated with that element. However, what is important to note here is that both the 12-DOF FEM and 16-DOF FEM formulations are accurate and their respective FEM codes produce acceptable results that can be used, in turn, for the verification of the results from the QDFE and DCM formulations.

Before presenting the numerical results for the QDFE method and comparing those with the FEM results, the new trigonometric, frequency-dependant shape functions unique to the QDFE method are plotted for an extremely low, almost zero frequency and compared against the FEM

shape functions to give the reader an opportunity to compare their similarity graphically. Thus, the sixteen new QDFE shape functions for node 1 of the 16-DOF thin plate element are presented adjacent to their corresponding FEM (incomplete quintic) polynomial shape functions [119] in Figures 9 to 12. The QDFE shape functions for nodes 2, 3 and 4 are compared with the FEM shape functions in Figures 26 through 37 in Appendix A.

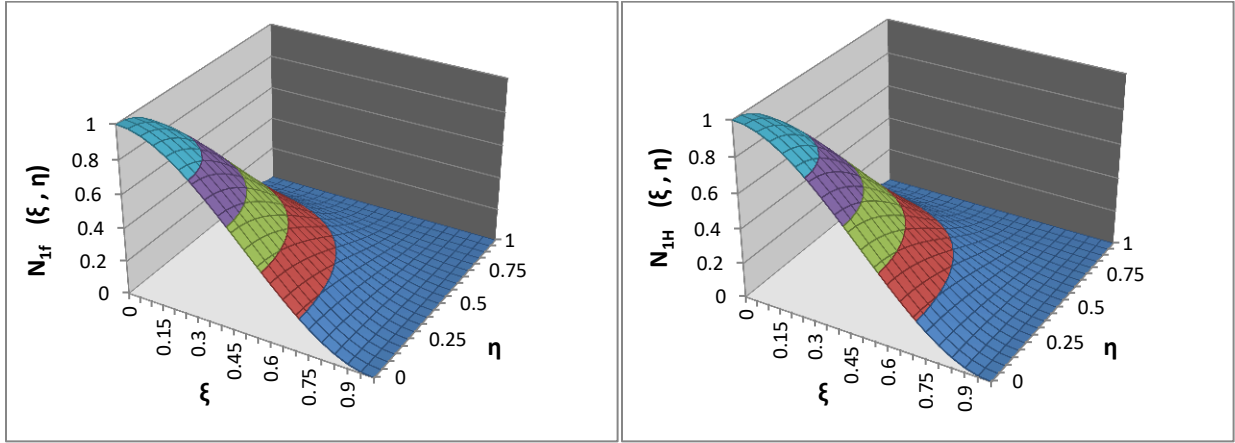


Figure 9: QDFE (left) and FEM (right) shape function for w_1 at node 1

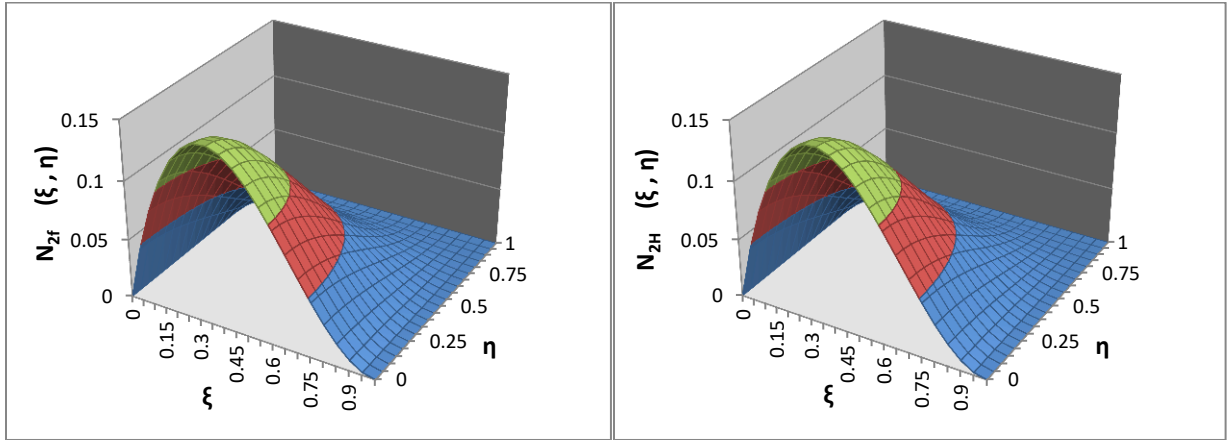


Figure 10: QDFE (left) and FEM (right) shape function for θ_{x1} at node 1

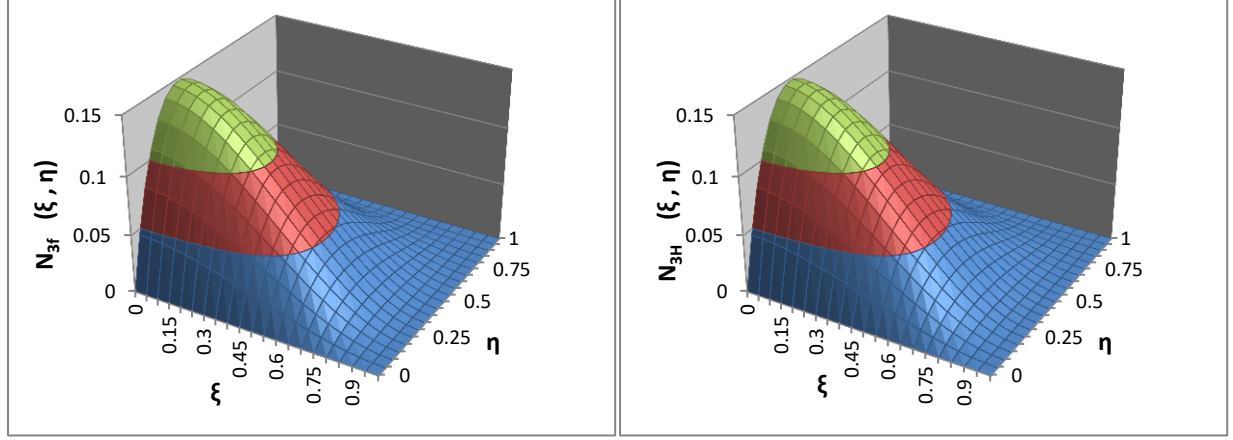


Figure 11: QDFE (left) and FEM (right) shape function for θ_{y1} at node 1

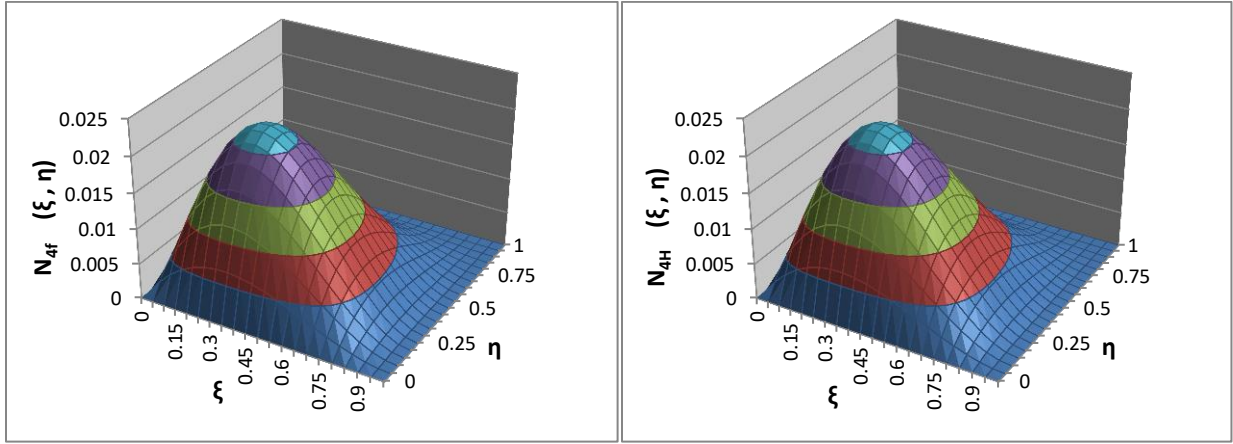


Figure 12: QDFE (left) and FEM (right) shape function for θ_{xy1} at node 1

As expected, for both the QDFE and FEM shape functions the flexural displacement w_1 is 1 at node 1, where the natural coordinates ξ and η are zero. The slope, $\theta_{x1}(\partial w/\partial \xi)$ is also 1 at node 1 for both shape functions N_{2f} and N_{2H} . Similarly, gradients of the shape functions N_{3f} and N_{3H} which approximate the slope, $\theta_{y1}(\partial w/\partial \eta)$ assumes a value of 1 at node 1, too. Furthermore, the slope θ_{xy1} which can also be expressed as $\partial^2 w/\partial \xi \partial \eta$ becomes 1 at node 1 for both shape function types. Thus, it is evident from Figures 9 through 12 and Figures 26 through 37 in Appendix A that the new QDFE shape functions are identical to the conventional FEM polynomial shape functions [144] in how the displacements and slopes are approximated.

As mentioned in the Section 3.3, the new QDFE shape functions are also frequency-dependent. That is, they oscillate and change with varying frequency ω . In order to illustrate their dependency on frequency, in Figures 13 to 16 the QDFE shape functions, N_{1f} , N_{2f} , N_{3f} and N_{4f} are plotted at two different frequencies ω_1 (82.06 Hz) and ω_5 (206.67 Hz) for a thin, homogeneous, square plate of $E = 200 \times 10^9$ MPa, $\rho = 7800$ kg/m³, $L = 0.5$ m, $W = 0.5$ m and $h = 0.004$ m. The two frequencies ω_1 and ω_5 are the second and fifth natural frequencies of the square plate. The frequency dependant nature of the other 12 shape functions for nodes 2, 3 and 4 of the rectangular element is illustrated in Figures 38 through 49 in Appendix A.

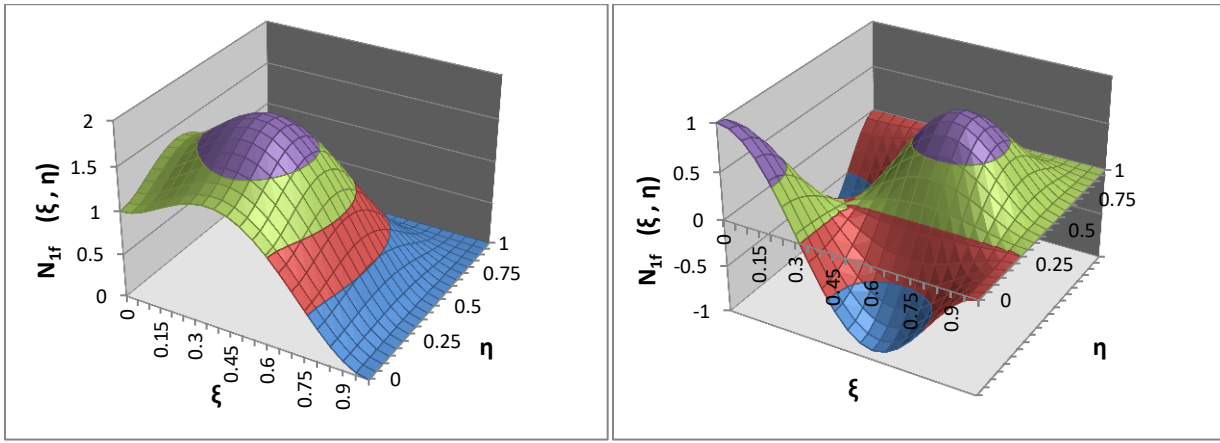


Figure 13: Shape function N_{1f} at ω_1 (left) and ω_5 (right)

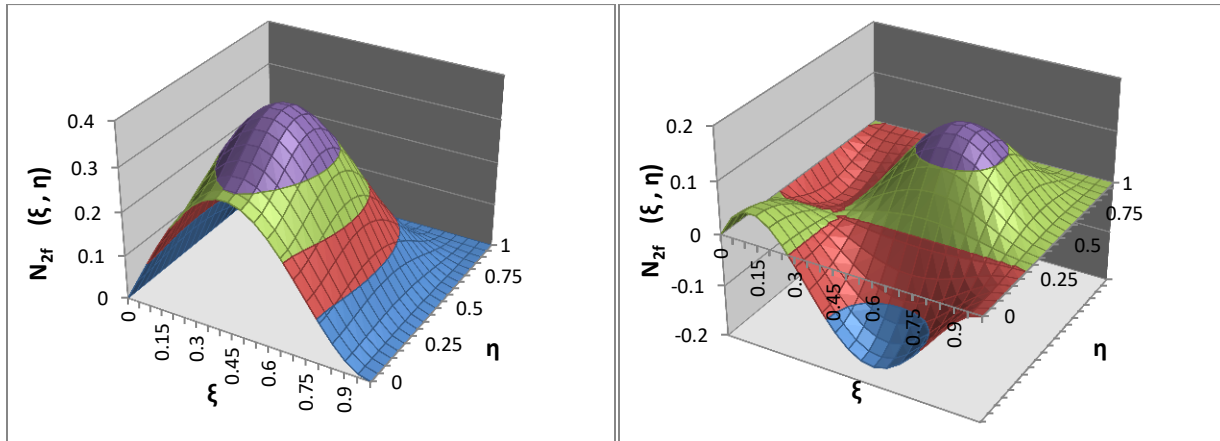


Figure 14: Shape function N_{2f} at ω_1 (left) and ω_5 (right)

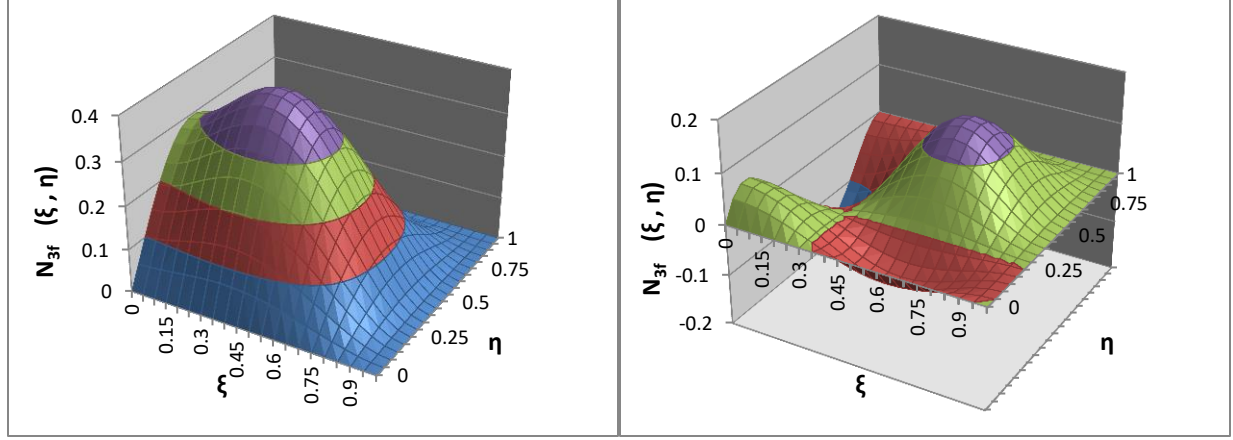


Figure 15: Shape function N_{3f} at ω_1 (left) and ω_5 (right)

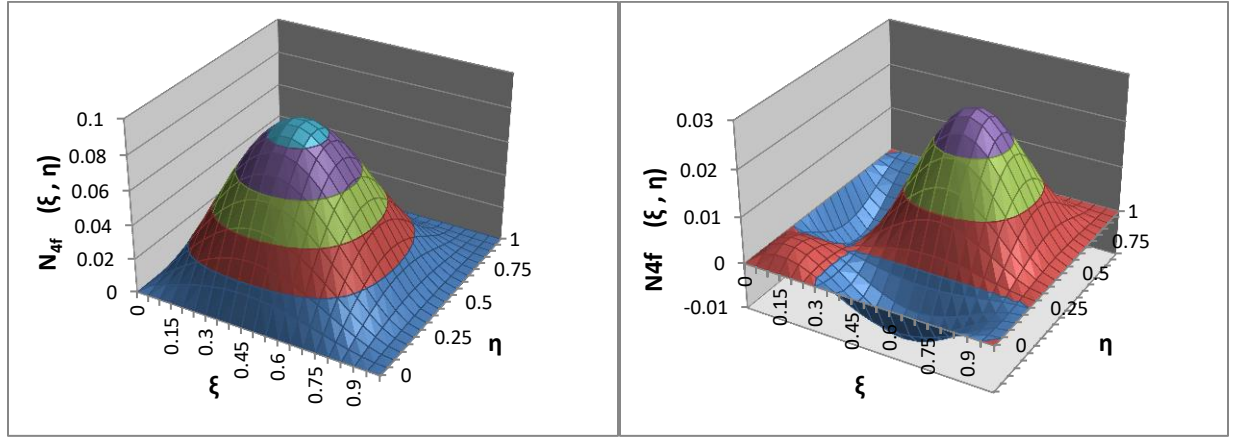


Figure 16: Shape function N_{4f} at ω_1 (left) and ω_5 (right)

The natural frequencies produced by the QDFE and DCM methods for eleven different boundary condition types are shown in Tables 3 through 13, and are compared with the results from the 12-DOF FEM and 16-DOF FEM formulations, as well exact results found in the open literature [118].

Table 3: Natural frequencies for a homogeneous Steel plate with C-F-F-F B.C.

Mode Number	Natural Frequencies (Hz) for C-F-F-F plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	12-DOF FEM	% Diff. QDFE vs. 12-DOF FEM
			# Elem: 1	# Elem: 196		# Elem: 196	
1	9.36	9.36	9.36	9.36	0.00	9.37	0.11
2	31.51	31.51	31.51	31.59	0.25	31.61	0.32
3	58.15	58.15	58.15	58.71	0.96	59.01	1.48
4	106.29	106.29	106.29	107.20	0.86	107.58	1.21
5	144.92	144.92	144.92	146.04	0.77	146.56	1.13

Table 4: Natural frequencies for a homogeneous Steel plate with C-C-F-F B.C.

Mode Number	Natural Frequencies (Hz) for C-C-F-F Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	12-DOF FEM	% Diff. QDFE vs. 12-DOF FEM
			# Elem: 1	# Elem: 196		# Elem: 196	
1	30.25	30.25	30.25	30.30	0.17	30.31	0.20
2	80.57	80.57	80.57	81.17	0.74	81.44	1.08
3	141.76	141.76	141.76	143.07	0.92	143.72	1.38
4	183.35	183.35	183.35	187.64	2.34	189.83	3.53
5	207.65	207.65	207.65	209.72	1.00	210.72	1.48

Table 5: Natural frequencies for a homogeneous Steel plate with C-F-C-F B.C.

Mode Number	Natural Frequencies (Hz) for C-F-C-F Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	12-DOF FEM	% Diff. QDFE vs. 12-DOF FEM
			# Elem: 1	# Elem: 196		# Elem: 196	
1	59.83	59.83	59.83	60.59	1.27	60.98	1.92
2	83.23	83.23	83.23	83.89	0.79	84.18	1.14
3	165.07	165.07	165.07	170.69	3.40	173.73	5.25
4	191.58	191.58	191.58	192.79	0.63	193.29	0.89
5	199.80	199.80	199.80	204.96	2.58	207.66	3.93

Table 6: Natural frequencies for a homogeneous Steel plate with C-C-C-F B.C.

Mode Number	Natural Frequencies (Hz) for C-C-C-F Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	12-DOF FEM	% Diff. QDFE vs. 12-DOF FEM
			# Elem: 1	# Elem: 196		# Elem: 196	
1	72.07	72.07	72.07	72.72	0.90	73.05	1.36
2	178.17	178.17	178.17	179.93	0.99	180.28	1.18
3	178.42	178.42	178.42	183.76	2.99	186.63	4.60
4	288.34	288.34	288.34	293.24	1.70	295.73	2.56
5	337.80	337.80	337.80	358.76	6.20	370.59	9.71

Table 7: Natural frequencies for a homogeneous Steel plate with C-C-C-C B.C.

Mode Number	Natural Frequencies (Hz) for C-C-C-C Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	12-DOF FEM	% Diff. QDFE vs. 12-DOF FEM
			# Elem: 1	# Elem: 196		# Elem: 196	
1	164.52	164.52	164.52	166.02	0.91	166.81	1.39
2	254.00	254.00	254.00	258.58	1.80	261.06	2.78
3	402.53	402.53	402.53	414.12	2.88	420.43	4.45
4	405.06	405.06	405.06	423.75	4.61	434.35	7.23
5	485.62	485.62	485.62	497.29	2.40	503.57	3.70

Table 8: Natural frequencies for a homogeneous Steel plate with C-C-C-S B.C.

Mode Number	Natural Frequencies (Hz) for C-C-C-S Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	12-DOF FEM	% Diff. QDFE vs. 12-DOF FEM
			# Elem: 1	# Elem: 196		# Elem: 196	
1	130.34	130.34	130.34	131.27	0.71	131.73	1.07
2	231.30	231.30	231.30	236.06	2.06	238.58	3.15
3	335.40	335.40	335.40	342.73	2.19	346.63	3.35
4	389.44	389.44	389.44	408.88	4.99	419.83	7.80
5	427.90	427.90	427.90	436.58	2.03	441.10	3.08

Table 9: Natural frequencies for a homogeneous Steel plate with C-C-S-S B.C.

Mode Number	Natural Frequencies (Hz) for C-C-S-S Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	12-DOF FEM	% Diff. QDFE vs. 12-DOF FEM
			# Elem: 1	# Elem: 196		# Elem: 196	
1	121.45	121.45	121.45	122.28	0.68	122.66	1.00
2	207.00	207.00	207.00	210.22	1.56	211.84	2.34
3	330.95	330.95	330.95	338.36	2.24	342.28	3.42
4	349.98	349.98	349.98	364.02	4.01	371.70	6.21
5	412.25	412.25	412.25	420.33	1.96	424.43	2.95

Table 10: Natural frequencies for a homogeneous Steel plate with S-C-S-C B.C.

Mode Number	Natural Frequencies (Hz) for S-C-S-C Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	12-DOF FEM	% Diff. QDFE vs. 12-DOF FEM
			# Elem: 1	# Elem: 196		# Elem: 196	
1	152.52	152.52	152.52	153.97	0.95	154.71	1.44
2	213.68	213.68	213.68	215.97	1.07	217.11	1.61
3	333.22	333.22	333.22	342.67	2.84	347.74	4.36
4	395.68	395.68	395.68	407.47	2.98	413.85	4.59
5	459.86	459.86	459.86	470.89	2.40	476.71	3.66

Table 11: Natural frequencies for a homogeneous Steel plate with S-F-S-F B.C.

Mode Number	Natural Frequencies (Hz) for S-F-S-F Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	12-DOF FEM	% Diff. QDFE vs. 12-DOF FEM
			# Elem: 1	# Elem: 196		# Elem: 196	
1	25.89	25.89	25.89	26.02	0.50	26.09	0.77
2	58.30	58.30	58.30	58.66	0.62	58.74	0.75
3	104.88	104.88	104.88	106.96	1.98	108.06	3.03
4	148.13	148.13	148.13	150.41	1.54	151.44	2.23
5	177.72	177.72	177.72	178.94	0.69	179.38	0.93

Table 12: Natural frequencies for a homogeneous Steel plate with S-S-S-F B.C.

Mode Number	Natural Frequencies (Hz) for S-S-S-F Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	12-DOF FEM	% Diff. QDFE vs. 12-DOF FEM
			# Elem: 1	# Elem: 196		# Elem: 196	
1	37.04	37.04	37.04	37.25	0.57	37.32	0.76
2	117.89	117.89	117.89	120.00	1.79	121.06	2.69
3	129.29	129.29	129.29	130.12	0.64	130.42	0.87
4	219.89	219.89	219.89	222.66	1.26	223.81	1.78
5	250.88	250.88	250.88	261.48	4.23	267.20	6.51

Table 13: Natural frequencies for a homogeneous Steel plate with S-S-S-SB.C.

Mode Number	Natural Frequencies (Hz) for S-S-S-S Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	12-DOF FEM	% Diff. QDFE vs. 12-DOF FEM
			# Elem: 1	# Elem: 196		# Elem: 196	
1	86.68	86.68	86.68	87.17	0.57	87.33	0.75
2	166.63	166.63	166.63	168.92	1.37	169.94	1.99
3	266.99	266.99	266.99	271.56	1.71	273.87	2.58
4	300.14	300.14	300.14	310.54	3.47	315.99	5.28
5	346.49	346.49	346.49	352.12	1.62	354.74	2.38

As can be seen from Tables 3 through 13, not only that the results obtained for the natural frequencies using both the DCM and QDFE methods converged to within two decimal places of the exact data reported in Reference [118], but also a single-element model, in this case, produces exact results for all the first five natural frequencies and all boundary conditions. They also agree very well with the results produced by both conventional FEM formulations, which deviate slightly from the exact data. In contrast, both conventional FEM formulations require 196 elements to produce results with a percent difference less than 10 percent. The percent difference of the conventional FEM results increase as the mode number increases and even with 196 elements the results for the 3rd, 4th and 5th natural frequencies at times consist of percent differences between 5 and 10 percent. Thus, in order to produce near-exact results for just the first five natural frequencies, the conventional FEM method will require hundreds more or if not thousands of elements. The number of elements required to produce results to an acceptable

degree of precision will increase even more for modes above the 5th natural frequency. The higher the number of elements needed, the finer the mesh will be and that is directly proportional to the convergence time and computational resources consumed.

On the contrary, the QDFE method consumes lesser time and computational resources compared to the conventional FEM method as it uses an extremely coarse mesh; only a single element, in the presented case. Its ability to converge to the exact results with a very low number of elements is attributed to its usage of shape functions derived from the enriched dynamic basis functions of approximation space, formed based on the semi-exact solution to the plate governing equation. It is important to note here that even with such a coarse (one-element) mesh the QDFE method produces exact results for all natural frequencies and if a similar mesh density is used for the conventional FEM method, the errors will be significant. For example, the convergence analysis in Figure 6 shows that with even as many as 36 elements, the 12-DOF FEM formulation results for the 5th natural frequency of the C-C-C-F plate contains errors as high as 46 percent! Thus, the data presented here further highlight the difference in convergence rates between the QDFE method and the conventional FEM methods for higher modes, thus, casting light on the super-convergence of the QDFE method.

It is known that the operating frequencies of real-life aircraft wing and fuselage skins, and those commonly used for an acoustic modelling and analysis, are usually very high. Thus, the conventional FEM method would be extremely inefficient for conducting a preliminary modal analysis of these upper range natural frequencies as it will require thousands of elements to arrive at an accurate ballpark of the exact result. During the preliminary design stages the speed at which one arrives at the results is as important as the accuracy, since, the main goal during this design phase is to reduce the time spent on analysis. Thus, designers can increase the efficiency of their analysis by exploiting the capabilities of the QDFE method for the preliminary modal and structural dynamics modelling and analysis purposes and then employing the conventional FEM during the advanced design stages where a more detailed and elaborated and detailed analysis of the structures are carried out.

As mentioned earlier, the DCM results in Tables 3 through 13 matches with the exact data to two decimal places, and they are also in excellent agreement with the 16-DOF and 12-DOF FEM

results. Furthermore, it can be observed that the DCM, which is an analytical method, is as capable as the QDFE in producing exact results for all boundary conditions. Unlike most exact methods available, which are limited to simple structural configurations and special boundary conditions as a result of simplifying assumptions being made, the DCM method presented here is a powerful tool that can be used to study the vibration behaviour of square or rectangular plates of any dimension subjected to any type of boundary condition as its solution does not incorporate any simplifying assumptions. However, when compared with the QDFE, the DCM method is less flexible, as the corresponding formulations must be redone, starting from scratch, when the system properties, or boundary conditions change. In addition, The DCM method is only applicable to simple square or rectangular plate geometries, whereas the QDFE elements can be assembled to model complex geometries, and the application of the boundary conditions is done in a way similar to conventional FEM.

So far in this Chapter the trigonometric shape functions have been compared with the polynomial shape functions graphically and the QDFE, DCM, 12-DOF FEM and 16-DOF FEM formulations have been developed for the thin rectangular plate problem. Having compared the accuracy and the rates of convergence for the various methods developed and having established the fact that the DFE method requires very much less elements compared to the conventional FEM method, the next step will be to extend the QDFE method to model the vibration behaviour of thin rectangular multilayered plates. Thus, in the next Chapter a Simplified Layerwise Quasi-Exact Dynamic Finite Element method and a Simplified Layerwise Dynamic Coefficient Matrix method will be presented. In order to validate and compare the layerwise QDFE and layerwise DCM results the same plate problem will be modelled using the Method of Homogenization and the Simplified Layerwise Conventional Finite Element method.

4. FREE VIBRATION OF A RECTANGULAR MULTILAYER PLATE

Many aircraft components can be modelled as thin, rectangular multilayered plates and as such the QDFE and DCM methods presented previously will be extended to investigate the natural frequencies of a thin, rectangular two layer plate in this Chapter. But prior to developing the layerwise QDFE and DCM solutions the Method of Homogenization and the layerwise Conventional Finite Element method will be presented for comparison purposes.

4.1 Method of Homogenization

In this section, a two-layered, homogeneous, linearly elastic, isotropic thin Aluminium-Steel plate shown in Figure 17 is considered. The multilayer plate has edge lengths, $L = 0.6\text{m}$ and $W = 0.4\text{m}$, resulting in an aspect ratio of 1.5. The total thickness h of the plate is 0.004m . The Aluminium layer of the plate has a thickness of 0.001m and the Steel layer has a thickness of 0.003m . Each layer has its own neutral axis. The neutral axis of the Aluminium layer is $N1$ and the neutral axis of the Steel layer is $N2$. The neutral axis for the entire multilayered plate, $N3$, is determined by employing the Parallel Axis Theorem.

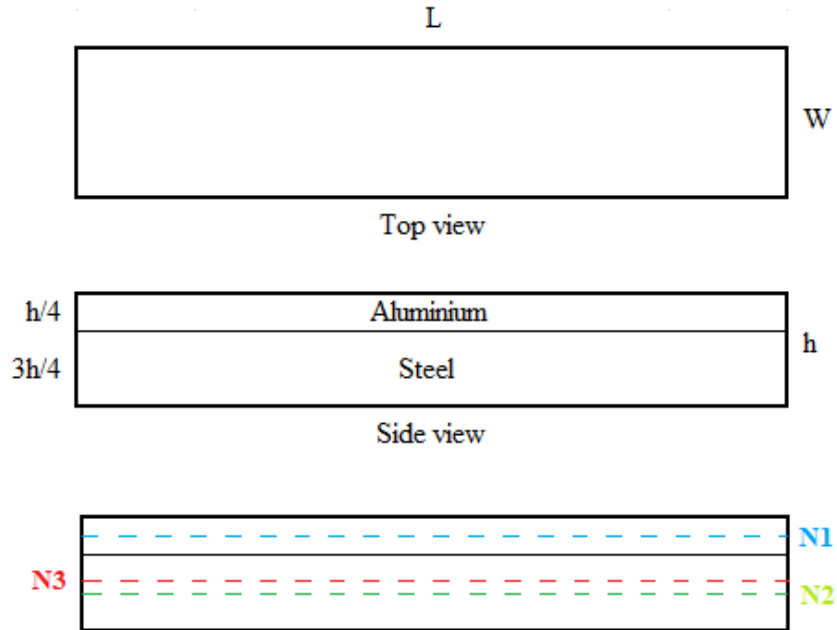


Figure 17: Two-layer Aluminium-Steel plate

Classical Lamination Theory (CLT) [117] is exploited to homogenize the material properties and determine the equivalent of a single layer plate. After finding the equivalent properties it is possible to treat the problem as a uniform plate. The equivalent properties are determined using Eqs. (141) to (144).

$$V_{ST} = 1 - V_{AL} \quad (141)$$

$$\rho_{EQ} = \rho_{ST} V_{ST} + \rho_{AL} V_{AL} \quad (142)$$

$$E_{EQ} = E_{ST} V_{ST} + E_{AL} V_{AL} \quad (143)$$

$$\nu_{EQ} = \nu_{ST} V_{ST} + \nu_{AL} V_{AL} \quad (144)$$

where, V is the volume fraction and indices AL and ST represent properties of the Aluminium and Steel layers, respectively.

4.2 Simplistic Layerwise Conventional Finite Element Method

A simplistic layerwise conventional FEM formulation will be presented in this section for the plate example shown in Figure 7. As for the single-layer homogeneous plate studied in Section 4.0, Classical Plate Theory, also known as Kirchhoff Plate Theory will be used to develop the layerwise FEM solution for the Aluminum-Steel multilayer plate. The layerwise Conventional FEM process, briefly explained here, also starts at the governing partial differential equation. However, as we now have two layers of materials in the plate there will be two equations for each layer as shown in Eqs. (145) and (146).

$$\left(\frac{\partial^2 M_{1x}}{\partial x^2} + 2 \frac{\partial^2 M_{1xy}}{\partial x \partial y} + \frac{\partial^2 M_{1y}}{\partial y^2} \right) - \omega^2 \rho_1 h_1 W = 0 \quad (145)$$

$$\left(\frac{\partial^2 M_{2x}}{\partial x^2} + 2 \frac{\partial^2 M_{2xy}}{\partial x \partial y} + \frac{\partial^2 M_{2y}}{\partial y^2} \right) - \omega^2 \rho_2 h_2 W = 0 \quad (146)$$

where, indices 1 and 2 represent the properties of layer 1 and layer 2, respectively. Upon enforcing interlayer linear continuity for lateral displacement and slope through the thickness of the plate, the summation of the two governing Eqs. (145) and (146) describes the dynamics of the entire plate resulting in the following equation.

$$\left(\frac{\partial^2 (M_{1x} + M_{2x})}{\partial x^2} + 2 \frac{\partial^2 (M_{1xy} + M_{2xy})}{\partial x \partial y} + \frac{\partial^2 (M_{1y} + M_{2y})}{\partial y^2} \right) - \omega^2 (\rho_1 h_1 + \rho_2 h_2) W = 0 \quad (147)$$

In order to develop the integral form of Eq. (147) it is multiplied with a weighting function, δW , and integrated over the area of the plate, which yields Eq. (148).

$$\bar{W}_f = \iint_A \left(\frac{\partial^2 (M_{1x} + M_{2x})}{\partial x^2} + \frac{\partial^2 (M_{1xy} + M_{2xy})}{\partial x \partial y} + \frac{\partial^2 (M_{1xy} + M_{2xy})}{\partial x \partial y} + \frac{\partial^2 (M_{1y} + M_{2y})}{\partial y^2} - \omega^2 (\rho_1 h_1 + \rho_2 h_2) W \right) \delta W = 0 \quad (148)$$

Next, the Green's theorem is applied once on each term of Eq. (148) except the mass term. This will result in the following equation. By observing Eq. (149) it can be seen that the line integral terms within the parenthesis are the definitions of the x and y direction shear forces.

$$\begin{aligned} - \iint_A \left(\frac{\partial (M_{1x} + M_{2x})}{\partial x} \frac{\partial (\delta W)}{\partial x} + \frac{\partial (M_{1xy} + M_{2xy})}{\partial y} \frac{\partial (\delta W)}{\partial x} + \frac{\partial (M_{1xy} + M_{2xy})}{\partial x} \frac{\partial (\delta W)}{\partial y} + \frac{\partial (M_{1y} + M_{2y})}{\partial y} \frac{\partial (\delta W)}{\partial y} \right) dA - \\ \iint_A \omega^2 (\rho_1 h_1 + \rho_2 h_2) W (\delta W) dA + \int_S \left(\frac{\partial (M_{1x} + M_{2x})}{\partial x} + \frac{\partial (M_{1xy} + M_{2xy})}{\partial y} \right) n_x \delta W dS + \\ \int_S \left(\frac{\partial (M_{1y} + M_{2y})}{\partial y} + \frac{\partial (M_{1xy} + M_{2xy})}{\partial x} \right) n_y \delta W dS \end{aligned} \quad (149)$$

Eq. (149) can be re-written as shown below upon substituting the respective symbols Q_x and Q_y in place of the x and y direction shear force definitions.

$$\begin{aligned} - \iint_A \left(\frac{\partial (M_{1x} + M_{2x})}{\partial x} \frac{\partial (\delta W)}{\partial x} + \frac{\partial (M_{1xy} + M_{2xy})}{\partial y} \frac{\partial (\delta W)}{\partial x} + \frac{\partial (M_{1xy} + M_{2xy})}{\partial x} \frac{\partial (\delta W)}{\partial y} + \frac{\partial (M_{1y} + M_{2y})}{\partial y} \frac{\partial (\delta W)}{\partial y} \right) dA - \\ \iint_A \omega^2 (\rho_1 h_1 + \rho_2 h_2) W (\delta W) dA + \int_S Q_x n_x \delta W dS + \int_S Q_y n_y \delta W dS \end{aligned} \quad (150)$$

Applying the Green's theorem once more on each term of Eq. (150) except the mass term and the line integral terms, will result in the weak integral form in Eq. (151).

$$\begin{aligned}
& \iint_A \left((M_{1x} + M_{2x}) \frac{\partial^2(\delta W)}{\partial x^2} + 2(M_{1xy} + M_{2xy}) \frac{\partial^2(\delta W)}{\partial x \partial y} + (M_{1y} + M_{2y}) \frac{\partial^2(\delta W)}{\partial y^2} \right) dA - \\
& \iint_A \omega^2 (\rho_1 h_1 + \rho_2 h_2) W (\delta W) dA + \int_S (Q_x n_x + Q_y n_y) \delta W dS - \int_S \left[(M_{1x} + M_{2x}) n_x + (M_{1xy} + M_{2xy}) n_y \right] \frac{\partial(\delta W)}{\partial x} dS - \\
& \int_S \left[(M_{1xy} + M_{2xy}) n_x + (M_{1y} + M_{2y}) n_y \right] \frac{\partial(\delta W)}{\partial y} dS = 0
\end{aligned} \quad (151)$$

Once the boundary terms (line integrals) in Eq. (151) vanish, the following terms will remain.

$$\begin{aligned}
& \iint_A \left((M_{1x} + M_{2x}) \frac{\partial^2(\delta W)}{\partial x^2} + 2(M_{1xy} + M_{2xy}) \frac{\partial^2(\delta W)}{\partial x \partial y} + (M_{1y} + M_{2y}) \frac{\partial^2(\delta W)}{\partial y^2} \right) dA - \\
& \iint_A \omega^2 (\rho_1 h_1 + \rho_2 h_2) W (\delta W) dA = 0
\end{aligned} \quad (152)$$

Rewriting Eq. (152) upon substituting the moment definitions shown in Eqs. (4) and (5) will yield Eq. (153) which is the discretized weak form of the governing equation.

$$\begin{aligned}
& \iint_A \left[\left(D_1 \frac{\partial^2 W}{\partial x^2} + D_2 \frac{\partial^2 W}{\partial x^2} \right) \left(\frac{\partial^2(\delta W)}{\partial x^2} \right) + \left(D_1 \nu_1 \frac{\partial^2 W}{\partial y^2} + D_2 \nu_2 \frac{\partial^2 W}{\partial y^2} \right) \left(\frac{\partial^2(\delta W)}{\partial x^2} \right) + \right. \\
& \quad \left. 2 \left(D_1 (1 - \nu_1) \frac{\partial^2 W}{\partial x \partial y} + D_2 (1 - \nu_2) \frac{\partial^2 W}{\partial x \partial y} \right) \left(\frac{\partial^2(\delta W)}{\partial x \partial y} \right) + \right. \\
& \quad \left. \left(D_1 \nu_1 \frac{\partial^2 W}{\partial x^2} + D_2 \nu_2 \frac{\partial^2 W}{\partial x^2} \right) \left(\frac{\partial^2(\delta W)}{\partial y^2} \right) + \left(D_1 \frac{\partial^2 W}{\partial y^2} + D_2 \frac{\partial^2 W}{\partial y^2} \right) \left(\frac{\partial^2(\delta W)}{\partial y^2} \right) + \omega^2 (\rho_1 h_1 + \rho_2 h_2) W \delta W \right] dA = 0
\end{aligned} \quad (153)$$

Eq. (153) also satisfies the principal of virtual work shown in Eqs. (36) to (38). In order to form the layerwise FEM solution, the system now has to be discretized. This will be done as described in Eq. (39) using the 4-node, 4-DOF freedom per node element shown in Figure 5. Exploiting Eq. (153) and the polynomial shape functions defined in Eqs. (101-a) through (101-p), the element matrices for the layerwise conventional FEM formulation is developed. The final 16x16 element stiffness matrix shown in Eq. (154) comprises of the 5 sub matrices defined in Eqs. (154-a) through (154-e). The 16x16 element mass matrix is shown in Eq. (155). In Eqs. (154) through (155), the subscript M denotes that the respective matrix is for the multilayer Aluminium-Steel plate.

$$[k_e]_M = [k_{1e}]_M + [k_{2e}]_M + [k_{3e}]_M + [k_{4e}]_M + [k_{5e}]_M \quad (154)$$

where,

$$[k_{1e}]_M = (D_1 + D_2) \begin{bmatrix} N_{1xx} \cdot N_{1xx} & \text{---} & N_{1xx} \cdot N_{16xx} \\ | & & | \\ N_{16xx} \cdot N_{1xx} & \text{---} & N_{16xx} \cdot N_{16xx} \end{bmatrix} \quad (154\text{-a})$$

$$[k_{2e}]_M = (D_1 \nu_1 + D_2 \nu_2) \begin{bmatrix} N_{1xx} \cdot N_{1yy} & \text{---} & N_{1xx} \cdot N_{16yy} \\ | & & | \\ N_{16xx} \cdot N_{1yy} & \text{---} & N_{16xx} \cdot N_{16yy} \end{bmatrix} \quad (154\text{-b})$$

$$[k_{3e}]_M = 2(D_1(1-\nu_1) + D_2(1-\nu_2)) \begin{bmatrix} N_{1xy} \cdot N_{1xy} & \text{---} & N_{1xy} \cdot N_{16xy} \\ | & & | \\ N_{16xy} \cdot N_{1xy} & \text{---} & N_{16xy} \cdot N_{16xy} \end{bmatrix} \quad (154\text{-c})$$

$$[k_{4e}]_M = (D_1 \nu_1 + D_2 \nu_2) \begin{bmatrix} N_{1yy} \cdot N_{1xx} & \text{---} & N_{1yy} \cdot N_{16xx} \\ | & & | \\ N_{16yy} \cdot N_{1xx} & \text{---} & N_{16yy} \cdot N_{16xx} \end{bmatrix} \quad (154\text{-d})$$

$$[k_{5e}]_M = (D_1 + D_2) \begin{bmatrix} N_{1yy} \cdot N_{1yy} & \text{---} & N_{1yy} \cdot N_{16yy} \\ | & & | \\ N_{16yy} \cdot N_{1yy} & \text{---} & N_{16yy} \cdot N_{16yy} \end{bmatrix} \quad (154\text{-e})$$

$$[m]_M = -\rho h \omega^2 \begin{bmatrix} N_1 \cdot N_1 & \text{---} & N_1 \cdot N_{16} \\ | & & | \\ N_{16} \cdot N_1 & \text{---} & N_{16} \cdot N_{16} \end{bmatrix} \quad (155)$$

The element stiffness and mass matrices are assembled using a special layerwise FEM program written on MATLAB[®] and the assembly process yields the global stiffness, $[K]_M$, and global

mass, $[M]_M$, matrices. The eigenvalue problem shown in Eq. (156) will be then formed and solved within the layerwise FEM program, the solutions of which will be the natural frequencies and mode shapes of the multilayer Aluminium-Steel plate. Various boundary conditions will also be enforced prior to solving.

$$\begin{aligned} \langle \delta W_n \rangle ([K]_M - \omega^2 [M]_M) \{W_n\} &= 0 \\ \det([K]_M - \omega^2 [M]_M) &= 0 \end{aligned} \quad (156)$$

4.3 Simplistic Layerwise Quasi-Exact Dynamic Finite Element Method

The Dynamic Finite Element method presented previously for the thin homogeneous plate will be extended to the two-layer Aluminium-Steel plate here. In order to form the QDFE solution for the two-layer Aluminium-Steel plate the roots of the governing differential equation should be modified to incorporate the effect of having two layers of material with contrasting properties. Thus, the modified roots, defined as Eqs. (157) to (160) are included below.

$$A_1 = -A_3 = \sqrt{k_1 \omega \sqrt{\frac{\rho_1 h_1 + \rho_2 h_2}{D_1 + D_2}}} = \beta_x \quad (157)$$

$$A_2 = -A_4 = i \sqrt{k_1 \omega \sqrt{\frac{\rho_1 h_1 + \rho_2 h_2}{D_1 + D_2}}} = \alpha_x \quad (158)$$

$$B_1 = -B_3 = \sqrt{k_2 \omega \sqrt{\frac{\rho_1 h_1 + \rho_2 h_2}{D_1 + D_2}}} = \beta_y \quad (159)$$

$$B_2 = -B_4 = i \sqrt{k_2 \omega \sqrt{\frac{\rho_1 h_1 + \rho_2 h_2}{D_1 + D_2}}} = \alpha_y \quad (160)$$

The starting point of the QDFE formulation is Eq. (153) from Section 5.2, which is the discretized weak form of the governing equation, after the boundary terms vanished.

$$\begin{aligned} \bar{W}_f^k(x, y) = \int_{y_j}^{y_{j+1}} \int_{x_j}^{x_{j+1}} & \left[(D_1 + D_2) \frac{\partial^2 W}{\partial x^2} \left(\frac{\partial^2 (\delta W)}{\partial x^2} \right) + (D_1 \nu_1 + D_2 \nu_2) \frac{\partial^2 W}{\partial y^2} \left(\frac{\partial^2 (\delta W)}{\partial x^2} \right) + 2(D_1(1 - \nu_1) + D_2(1 - \nu_2)) \frac{\partial^2 W}{\partial x \partial y} \left(\frac{\partial^2 (\delta W)}{\partial x \partial y} \right) + \right. \\ & \left. (D_1 \nu_1 + D_2 \nu_2) \frac{\partial^2 W}{\partial x^2} \left(\frac{\partial^2 (\delta W)}{\partial y^2} \right) + (D_1 + D_2) \frac{\partial^2 W}{\partial y^2} \left(\frac{\partial^2 (\delta W)}{\partial y^2} \right) + \omega^2 (\rho_1 h_1 + \rho_2 h_2) W \delta W \right] dx dy \end{aligned} \quad (161)$$

Integration by parts is carried out on each term in Eq. (161), except the mass term. Terms 1 and 4 are integrated twice with respect to x , while terms 2 and 5 are integrated twice with respect to y . Before integrating term 3, it is re-written as follows and a special set of integration by part is performed.

$$\iint_A (D_1(1-\nu_1) + D_2(1-\nu_2)) \frac{\partial^2 w}{\partial x \partial y} \frac{\partial^2 (\delta w)}{\partial x \partial y} dA + \iint_A (D_1(1-\nu_1) + D_2(1-\nu_2)) \frac{\partial^2 w}{\partial x \partial y} \frac{\partial^2 (\delta w)}{\partial x \partial y} dA \quad (161-a)$$

Two sets of integration by parts were applied to the LHS Term of Eq. (161-a), firstly with respect to x and secondly with respect to y . Similarly, for the RHS Term of Eq. (161-a), another two sets of integration by parts were carried out, however, this time it was performed first with respect to y and then with respect to x . Completing all the above-mentioned integration by parts leads to Eq. (162) shown below.

$$\begin{aligned} & \left[\int_S (D_1 + D_2) \frac{\partial w}{\partial x} \frac{\partial^2 (\delta w)}{\partial x^2} n_x dS + \int_S (D_1 \nu_1 + D_2 \nu_2) \frac{\partial w}{\partial x} \frac{\partial^2 (\delta w)}{\partial y^2} n_x dS + \int_S (D_1(1-\nu_1) + D_2(1-\nu_2)) \frac{\partial w}{\partial x} \frac{\partial^2 (\delta w)}{\partial x \partial y} n_y dS \right] \\ & + \left[\int_S (D_1 \nu_1 + D_2 \nu_2) \frac{\partial w}{\partial y} \frac{\partial^2 (\delta w)}{\partial x^2} n_y dS + \int_S (D_1 + D_2) \frac{\partial w}{\partial y} \frac{\partial^2 (\delta w)}{\partial y^2} n_y dS + \int_S (D_1(1-\nu_1) + D_2(1-\nu_2)) \frac{\partial w}{\partial y} \frac{\partial^2 (\delta w)}{\partial x \partial y} n_x dS \right] + \\ & \left[- \int_S (D_1 + D_2) w \frac{\partial^3 (\delta w)}{\partial x^3} n_x dS - \int_S (D_1 + D_2) w \frac{\partial^3 (\delta w)}{\partial x \partial y^2} n_x dS \right] + \left[- \int_S (D_1 + D_2) w \frac{\partial^3 (\delta w)}{\partial y^3} n_y dS - \int_S (D_1 + D_2) w \frac{\partial^3 (\delta w)}{\partial x^2 \partial y} n_y dS \right] \quad (162) \\ & + \iint_A (D_1 + D_2) w \frac{\partial^4 (\delta w)}{\partial x^4} dA + \iint_A 2(D_1(1-\nu_1) + D_2(1-\nu_2)) w \frac{\partial^4 (\delta w)}{\partial x^2 \partial y^2} dA + \iint_A (D_1 \nu_1 + D_2 \nu_2) w \frac{\partial^4 (\delta w)}{\partial x^2 \partial y^2} dA \\ & + \iint_A (D_1 \nu_1 + D_2 \nu_2) w \frac{\partial^4 (\delta w)}{\partial x^2 \partial y^2} dA + \iint_A (D_1 + D_2) w \frac{\partial^4 (\delta w)}{\partial y^4} dA - \iint_A \omega^2 (\rho_1 h_1 + \rho_2 h_2) w \delta w dA \end{aligned}$$

In Eq. (162) the bending moments M_x and M_y are underlined in red and blue, respectively, and the twisting moment and M_{xy} is underlined in purple colour. Similarly, the shear force Q_x is underlined in green and the shear force Q_y is underline in yellow. The area integral expression underlined in orange is the thin plate governing equation expressed in terms virtual displacements. Using the bending moment definitions in Eqs. (20), (22) and (24) and the shear force definitions in Eqs. (5-a) and (6-a), Eq. (158) is condensed and re-written as follows. The same color scheme was used to illustrate how each expression was reduced.

$$\begin{aligned} & - \int_S (M_x n_x + M_{xy} n_y) \frac{\partial w}{\partial x} dS - \int_S (M_y n_y + M_{xy} n_x) \frac{\partial w}{\partial y} dS + \int_S (Q_x n_x + Q_y n_y) w dS \\ & + \left[\iint_A (D_1 + D_2) \frac{\partial^4 (\delta w)}{\partial x^4} dA + \iint_A 2(D_1 + D_2) \frac{\partial^4 (\delta w)}{\partial x^2 \partial y^2} dA + \iint_A (D_1 + D_2) \frac{\partial^4 (\delta w)}{\partial y^4} dA - \iint_A \omega^2 (\rho_1 h_1 + \rho_2 h_2) \delta w w dA \right] \quad (163) \end{aligned}$$

Non-dimensionalizing Eq. (163) by substituting $\xi=x/a$ and $\eta=y/b$ results in Eq. (164).

$$\begin{aligned}
& \left[\int_{\eta=0}^{\eta=1} \left[\frac{(D_1 + D_2)b}{a^3} \frac{\partial w}{\partial \xi} \frac{\partial^2(\delta w)}{\partial \xi^2} \right]_{\xi=0}^{\xi=1} d\eta + \int_{\eta=0}^{\eta=1} \left[\frac{(D_1\nu_1 + D_2\nu_2)}{ab} \frac{\partial w}{\partial \xi} \frac{\partial^2(\delta w)}{\partial \eta^2} \right]_{\xi=0}^{\xi=1} d\eta + \int_{\xi=0}^{\xi=1} \left[\frac{(D_1(1-\nu_1) + D_2(1-\nu_2))}{ab} \frac{\partial w}{\partial \xi} \frac{\partial^2(\delta w)}{\partial \xi \partial \eta} \right]_{\eta=0}^{\eta=1} d\xi \right] \\
& + \left[\int_{\xi=0}^{\xi=1} \left[\frac{(D_1\nu_1 + D_2\nu_2)}{ab} \frac{\partial w}{\partial \eta} \frac{\partial^2(\delta w)}{\partial \xi^2} \right]_{\eta=0}^{\eta=1} d\xi + \int_{\xi=0}^{\xi=1} \left[\frac{(D_1 + D_2)a}{b^3} \frac{\partial w}{\partial \eta} \frac{\partial^2(\delta w)}{\partial \eta^2} \right]_{\eta=0}^{\eta=1} d\xi + \int_{\eta=0}^{\eta=1} \left[\frac{(D_1(1-\nu_1) + D_2(1-\nu_2))}{ab} \frac{\partial w}{\partial \eta} \frac{\partial^2(\delta w)}{\partial \xi \partial \eta} \right]_{\xi=0}^{\xi=1} d\eta \right] \\
& + \left[- \int_{\eta=0}^{\eta=1} \left[\frac{(D_1 + D_2)b}{a^3} w \frac{\partial^3(\delta w)}{\partial \xi^3} \right]_{\xi=0}^{\xi=1} d\eta - \int_{\eta=0}^{\eta=1} \left[\frac{(D_1 + D_2)}{ab} w \frac{\partial^3(\delta w)}{\partial \xi \partial \eta^2} \right]_{\xi=0}^{\xi=1} d\eta \right] \\
& + \left[- \int_{\xi=0}^{\xi=1} \left[\frac{(D_1 + D_2)a}{b^3} w \frac{\partial^3(\delta w)}{\partial \eta^3} \right]_{\eta=0}^{\eta=1} d\xi - \int_{\xi=0}^{\xi=1} \left[\frac{(D_1 + D_2)}{ab} w \frac{\partial^3(\delta w)}{\partial \xi^2 \partial \eta} \right]_{\eta=0}^{\eta=1} d\xi \right] \\
& + \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \frac{(D_1 + D_2)b}{a^3} w \frac{\partial^4(\delta w)}{\partial \xi^4} d\xi d\eta + \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \frac{2(D_1 + D_2)}{ab} w \frac{\partial^4(\delta w)}{\partial \xi^2 \partial \eta^2} d\xi d\eta + \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \frac{(D_1 + D_2)a}{b^3} w \frac{\partial^4(\delta w)}{\partial \eta^4} d\xi d\eta \\
& - \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} ab\omega^2(\rho_1 h_1 + \rho_2 h_2) w \delta w d\xi d\eta
\end{aligned} \tag{164}$$

Using Eq. (164) and the new trigonometric shape functions in Eqs. (49) to (64), the element stiffness matrix $[k_{DS}]_M$ for the Aluminium-Steel two-layer plate is obtained. This element stiffness matrix is comprised of thirteen sub matrices resulting due to the bending moments M_x and M_y , twisting moment M_{xy} , shear forces Q_x and Q_y as well as the area integral terms underlined in orange except for the mass term. The bending moment M_x contributes the two sub matrices (165) and (166) shown below. The subscript M denotes that the matrices are for the two-layer Aluminium-Steel plate.

$$[k_{Mx1}]_M = \left(\frac{(D_1 + D_2)b}{a^3} \right) \int_{\eta=0}^{\eta=1} \left[\begin{array}{cc} \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_1}{\partial \xi^2} & \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \xi^2} \\ \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_1}{\partial \xi^2} & \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \xi^2} \end{array} \right]_{\xi=0}^{\xi=1} d\eta \tag{165}$$

$$[k_{Mx2}]_M = \left(\frac{(D_1\nu_1 + D_2\nu_2)}{ab} \right) \int_{\eta=0}^{\eta=1} \left[\begin{array}{cc} \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_1}{\partial \eta^2} & \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \eta^2} \\ \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_1}{\partial \eta^2} & \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \eta^2} \end{array} \right]_{\xi=0}^{\xi=1} d\eta \tag{166}$$

where,

$$[k_{Mx}]_M = [k_{Mx1}]_M + [k_{Mx2}]_M \tag{167}$$

Similarly, two sub matrices are contributed towards the final element stiffness matrix due to the actions of the bending moment M_y and these can be written as:

$$[k_{My1}]_M = \left(\frac{(D_1 \nu_1 + D_2 \nu_2)}{ab} \right) \int_{\xi=0}^{\xi=1} \begin{bmatrix} \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_1}{\partial \xi^2} & \text{---} & \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \xi^2} \\ \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_1}{\partial \xi^2} & \text{---} & \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \xi^2} \end{bmatrix}_{\eta=0}^{\eta=1} d\xi \quad (168)$$

$$[k_{My2}]_M = \left(\frac{(D_1 + D_2)a}{b^3} \right) \int_{\xi=0}^{\xi=1} \begin{bmatrix} \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_1}{\partial \eta^2} & \text{---} & \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \eta^2} \\ \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_1}{\partial \eta^2} & \text{---} & \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \eta^2} \end{bmatrix}_{\eta=0}^{\eta=1} d\xi \quad (169)$$

where,

$$[k_{My}]_M = [k_{My1}]_M + [k_{My2}]_M \quad (170)$$

Two sub matrices also arise as a result of the twisting moment M_{xy} .

$$[k_{Mxy1}]_M = \left(\frac{(D_1(1-\nu_1) + D_2(1-\nu_2))}{ab} \right) \int_{\xi=0}^{\xi=1} \begin{bmatrix} \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_1}{\partial \xi \partial \eta} & \text{---} & \frac{\partial N_1}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \xi \partial \eta} \\ \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_1}{\partial \xi \partial \eta} & \text{---} & \frac{\partial N_{16}}{\partial \xi} \frac{\partial^2 N_{16}}{\partial \xi \partial \eta} \end{bmatrix}_{\eta=0}^{\eta=1} d\xi \quad (171)$$

$$[k_{Mxy2}]_M = \left(\frac{(D_1(1-\nu_1) + D_2(1-\nu_2))}{ab} \right) \int_{\eta=0}^{\eta=1} \begin{bmatrix} \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_1}{\partial \xi \partial \eta} & \text{---} & \frac{\partial N_1}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \xi \partial \eta} \\ \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_1}{\partial \xi \partial \eta} & \text{---} & \frac{\partial N_{16}}{\partial \eta} \frac{\partial^2 N_{16}}{\partial \xi \partial \eta} \end{bmatrix}_{\xi=0}^{\xi=1} d\eta \quad (172)$$

where,

$$[k_{Mxy}]_M = [k_{Mxy1}]_M + [k_{Mxy2}]_M \quad (173)$$

The x direction shear force Q_x also contributes two sub matrices (174) and (175) shown below.

$$[k_{Qx1}]_M = -\left(\frac{(D_1 + D_2)b}{a^3}\right) \int_{\eta=0}^{\eta=1} \begin{bmatrix} N_1 \frac{\partial^3 N_1}{\partial \xi^3} & \text{---} & N_1 \frac{\partial^3 N_{16}}{\partial \xi^3} \\ | & & | \\ N_{16} \frac{\partial^3 N_1}{\partial \xi^3} & \text{---} & N_{16} \frac{\partial^3 N_{16}}{\partial \xi^3} \end{bmatrix}_{\xi=0}^{\xi=1} d\eta \quad (174)$$

$$[k_{Qx2}]_M = -\left(\frac{(D_1 + D_2)}{ab}\right) \int_{\eta=0}^{\eta=1} \begin{bmatrix} N_1 \frac{\partial^3 N_1}{\partial \xi \partial \eta^2} & \text{---} & N_1 \frac{\partial^3 N_{16}}{\partial \xi \partial \eta^2} \\ | & & | \\ N_{16} \frac{\partial^3 N_1}{\partial \xi \partial \eta^2} & \text{---} & N_{16} \frac{\partial^3 N_{16}}{\partial \xi \partial \eta^2} \end{bmatrix}_{\xi=0}^{\xi=1} d\eta \quad (175)$$

where,

$$[k_{Qx}]_M = [k_{Qx1}]_M + [k_{Qx2}]_M \quad (176)$$

Similarly, two more sub matrices result due to the actions of the y direction shear force Q_y .

$$[k_{Qy1}]_M = -\left(\frac{(D_1 + D_2)a}{b^3}\right) \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 \frac{\partial^3 N_1}{\partial \eta^3} & \text{---} & N_1 \frac{\partial^3 N_{16}}{\partial \eta^3} \\ | & & | \\ N_{16} \frac{\partial^3 N_1}{\partial \eta^3} & \text{---} & N_{16} \frac{\partial^3 N_{16}}{\partial \eta^3} \end{bmatrix}_{\eta=0}^{\eta=1} d\xi \quad (177)$$

$$[k_{Qy2}]_M = -\left(\frac{(D_1 + D_2)}{ab}\right) \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 \frac{\partial^3 N_1}{\partial \xi^2 \partial \eta} & \text{---} & N_1 \frac{\partial^3 N_{16}}{\partial \xi^2 \partial \eta} \\ | & & | \\ N_{16} \frac{\partial^3 N_1}{\partial \xi^2 \partial \eta} & \text{---} & N_{16} \frac{\partial^3 N_{16}}{\partial \xi^2 \partial \eta} \end{bmatrix}_{\eta=0}^{\eta=1} d\xi \quad (178)$$

where,

$$[k_{Qy}]_M = [k_{Qy1}]_M + [k_{Qy2}]_M \quad (179)$$

The first three integral terms underlined in orange in Eq. (164) contribute the following matrices towards the final elements stiffness matrix.

$$[k_{G1}]_M = \left(\frac{(D_1 + D_2)b}{a^3} \right) \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 \frac{\partial^4 N_1}{\partial \xi^4} & \text{---} & N_1 \frac{\partial^4 N_{16}}{\partial \xi^4} \\ | & & | \\ N_{16} \frac{\partial^4 N_1}{\partial \xi^4} & \text{---} & N_{16} \frac{\partial^4 N_{16}}{\partial \xi^4} \end{bmatrix} d\xi d\eta \quad (180)$$

$$[k_{G2}]_M = \left(\frac{2(D_1 + D_2)}{ab} \right) \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 \frac{\partial^4 N_1}{\partial \xi^2 \partial \eta^2} & \text{---} & N_1 \frac{\partial^4 N_{16}}{\partial \xi^2 \partial \eta^2} \\ | & & | \\ N_{16} \frac{\partial^4 N_1}{\partial \xi^2 \partial \eta^2} & \text{---} & N_{16} \frac{\partial^4 N_{16}}{\partial \xi^2 \partial \eta^2} \end{bmatrix} d\xi d\eta \quad (181)$$

$$[k_{G3}]_M = \left(\frac{(D_1 + D_2)a}{b^3} \right) \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 \frac{\partial^4 N_1}{\partial \eta^4} & \text{---} & N_1 \frac{\partial^4 N_{16}}{\partial \eta^4} \\ | & & | \\ N_{16} \frac{\partial^4 N_1}{\partial \eta^4} & \text{---} & N_{16} \frac{\partial^4 N_{16}}{\partial \eta^4} \end{bmatrix} d\xi d\eta \quad (182)$$

The fourth area integral term underlined in orange in Eq. (164) results in the element mass matrix (183) below.

$$[m_{DS}]_M = -ab\omega^2 \rho h \int_{\eta=0}^{\eta=1} \int_{\xi=0}^{\xi=1} \begin{bmatrix} N_1 N_1 & \text{---} & N_1 N_{16} \\ | & & | \\ N_{16} N_1 & \text{---} & N_{16} N_{16} \end{bmatrix} d\xi d\eta \quad (183)$$

Adding Eqs. (167), (170), (173), (176), (179), (180), (181) and (182) together yields the final element dynamic stiffness matrix $[k_{DS}]_M$. Thus, the discretized (elemental) expression of the virtual work can be written as follows.

$$[k_{DS}]_M = \{ [k_{Mx}]_M + [k_{My}]_M + [k_{Mxy}]_M + [k_{Qx}]_M + [k_{Qy}]_M + [k_{G1}]_M + [k_{G2}]_M + [k_{G3}]_M \} \quad (184)$$

Assembling the element stiffness matrices within a layerwise QDFE program written in MATLAB[®] results in the non-linear eigenvalue problem shown in Eq. (185) of which, $[K_{DS}(\omega)]_M$, is the global dynamic stiffness matrix. The boundary conditions are also enforced using the MATLAB[®] code.

$$\begin{aligned} \langle \delta W_n \rangle (K_{DS} - \omega^2 M_{DS}) \{W_n\} &= 0 \\ \det(K_{DS} - \omega^2 M_{DS}) &= 0 \end{aligned} \quad (185)$$

Solving this Eigenvalue problem will give the eigenvalues and eigenvectors of the system, which correspond to the natural frequencies and mode shapes of the two-layer Aluminium-Steel plate.

4.4 Simplistic Layerwise Quasi-Exact Dynamic Coefficient Matrix Method

The simplistic layerwise Dynamic Coefficient Matrix method is an extension of the DCM Method shown in Section 4.3 to the two-layer Aluminium-Steel plate. It is formed by incorporating the modified roots for the multilayer plate shown in Eqs. (157) to (160) in to the general solution shown in Eq. (39).

The general solution shown in Eq. (39) can be expressed in the matrix form as follows in order to describe the displacement, $W(x, y)_M$, at any point within the plate. The subscript M denotes the ‘multilayer plate’.

$$W(x, y)_M = \langle T_1 \quad T_2 \quad \cdot \quad \cdot \quad T_{15} \quad T_{16} \rangle_M \begin{Bmatrix} E_{11} \\ E_{12} \\ \cdot \\ \cdot \\ E_{43} \\ E_{44} \end{Bmatrix}_M ; \quad \text{or} \quad W(x, y)_M = \langle T \rangle_M \{E\}_M \quad (186)$$

Differentiating Eq. (186) once in terms of x results in the matrix expression for the slope along the x direction, $\theta_x(x, y)_M$, shown in Eq. (187).

$$\theta_x(x, y)_M = \langle T_{1x} \quad T_{2x} \quad \cdot \quad \cdot \quad T_{15x} \quad T_{16x} \rangle_M \{E\}_M = \langle T_{ix} \rangle_M \{E\}_M \quad (187)$$

where, $i = 1, \dots, 16$. Similarly, taking the first derivative of Eq. (186) in terms of y results in the slope along the y direction, $\theta_y(x, y)_M$, which can be expressed as:

$$\theta_y(x, y)_M = \langle T_{1y} \quad T_{2y} \quad \cdot \quad \cdot \quad T_{15y} \quad T_{16y} \rangle_M \{E\}_M = \langle T_{iy} \rangle_M \{E\}_M \quad (188)$$

where, $i = 1, \dots, 16$. Finally, the curvature of the plate, $\theta_{xy}(x, y)_M$, shown in Eq. (189) can be determined by differentiating Eq. (186) first with respect to x and then with respect to y .

$$\theta_{xy}(x, y)_M = \langle T_{1xy} \quad T_{2xy} \quad \cdot \quad \cdot \quad T_{15xy} \quad T_{16xy} \rangle_M \{E\}_M = \langle T_{ixy} \rangle_M \{E\}_M \quad (189)$$

where, $i = 1, \dots, 16$. The boundary conditions for the displacements are shown in Eq. (138) in Section 4.3. The next step is to substitute these boundary conditions in Eqs. (186) to (189) and this process will yield the matrix relationship shown in Eq. (190).

$$\left\{ \begin{matrix} W_1 \\ \theta_{x1} \\ \theta_{y1} \\ \theta_{xy1} \\ W_2 \\ \theta_{x2} \\ \theta_{y2} \\ \theta_{xy2} \\ W_3 \\ \theta_{x3} \\ \theta_{y3} \\ \theta_{xy3} \\ W_4 \\ \theta_{x4} \\ \theta_{y4} \\ \theta_{xy4} \end{matrix} \right\}_M = \left[\begin{matrix} \langle T_i \rangle_{x=0,y=0} \\ \langle T_{ix} \rangle_{x=0,y=0} \\ \langle T_{iy} \rangle_{x=0,y=0} \\ \langle T_{ixy} \rangle_{x=0,y=0} \\ \langle T_i \rangle_{x=a,y=0} \\ \langle T_{ix} \rangle_{x=a,y=0} \\ \langle T_{iy} \rangle_{x=a,y=0} \\ \langle T_{ixy} \rangle_{x=a,y=0} \\ \langle T_i \rangle_{x=a,y=b} \\ \langle T_{ix} \rangle_{x=a,y=b} \\ \langle T_{iy} \rangle_{x=a,y=b} \\ \langle T_{ixy} \rangle_{x=a,y=b} \\ \langle T_i \rangle_{x=0,y=b} \\ \langle T_{ix} \rangle_{x=0,y=b} \\ \langle T_{iy} \rangle_{x=0,y=b} \\ \langle T_{ixy} \rangle_{x=0,y=b} \end{matrix} \right]_M \left\{ \begin{matrix} E_{11} \\ E_{12} \\ E_{13} \\ E_{14} \\ E_{21} \\ E_{22} \\ E_{23} \\ E_{24} \\ E_{31} \\ E_{32} \\ E_{33} \\ E_{34} \\ E_{41} \\ E_{42} \\ E_{43} \\ E_{44} \end{matrix} \right\}_M \quad (190)$$

Simplifying the matrix relationship shown in Eq. (190) and re-writing it in a condensed form gives the following expression.

$$\{W_n\} = [K_D(\omega)]_M \{E\}_M \quad (191)$$

where, $[K_D(\omega)]_M$, is the 16 x 16 Dynamic Coefficient Matrix (DCM) of the system. Boundary conditions are applied on the Dynamic Coefficient Matrix in Eq. (191) within a special DCM

program written using the MATLAB[®] software. The quasi-exact natural frequencies for a thin, rectangular, two-layer plate of any size, subjected to all boundary condition types can now be determined by conducting a determinant sweep, whereby the frequency domain is swept until an array of frequencies corresponding to a zero determinant of the Dynamic Coefficient Matrix is returned.

4.5 Numerical Results and Discussion

This section includes the modal analysis results for the multilayered Aluminium-Steel plate introduced in Section 5. As shown in Table 14, an ANSYS[®] simulation was used once again to validate the results produced by the 16-DOF layerwise FEM code written in MATLAB[®]. The results generated through the layerwise DCM, layerwise QDFE and Homogenization methods were then verified for accuracy using the 16-DOF layerwise FEM results. Based on the findings of the convergence analysis shown in Figure 6 of Section 4.4, 196 elements were used for both the ANSYS[®] simulation and the 16-DOF layerwise FEM analysis.

Table 14: Comparison of 16-DOF layerwise FEM results with ANSYS[®] simulation

Mode Number	Natural Frequencies (Hz) of C-F-F-F plate				
	Exact[118]	16-DOF FEM	% Error 16-DOF FEM vs. Exact	ANSYS [®]	% Error ANSYS vs. Exact
		# Elem: 196		# Elem: 196	
1	9.36	9.37	0.11	9.37	0.11
2	31.45	31.55	0.32	31.56	0.35
3	58.17	58.74	0.98	58.92	1.29
4	106.16	107.11	0.89	107.33	1.10
5	144.92	146.07	0.79	146.39	1.01

Results from the conventional FEM formulation in Table 14 is in excellent agreement with the exact [143] and ANSYS[®] results. Thus, it shows that the conventional FEM code produces accurate results, which can be used to validate the modal analysis results obtained from the DCM, QDFE and Homogenization formulations. Therefore, in Tables 15 through 25 the conventional FEM, DCM, QDFE and Method of Homogenization (M. of H.) results are presented and compared with each other.

Table 15: Natural frequencies for a multilayered Al-St plate with C-F-F-FB.C.

Mode Number	Natural Frequencies (Hz) for C-F-F-F plates						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	M. of H.	% Diff. QDFE vs. M. of H.
			# Elem: 1	# Elem: 196		# Elem: 196	
1	9.36	9.36	9.36	9.37	0.11	9.37	0.11
2	31.45	31.45	31.45	31.55	0.32	31.55	0.32
3	58.17	58.17	58.17	58.74	0.98	58.82	1.12
4	106.16	106.16	106.16	107.11	0.89	107.21	0.99
5	144.92	144.92	144.92	146.07	0.79	146.21	0.89

Table 16: Natural frequencies for a multilayered Al-St plate with C-C-F-F B.C.

Mode Number	Natural Frequencies (Hz) for C-C-F-F Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	M. of H.	% Diff. QDFE vs. M. of H.
			# Elem: 1	# Elem: 196		# Elem: 196	
1	30.25	30.25	30.25	30.30	0.17	30.30	0.17
2	80.53	80.53	80.53	81.15	0.77	81.23	0.87
3	141.84	141.84	141.84	143.18	0.94	143.36	1.07
4	183.38	183.38	183.38	187.74	2.38	188.32	2.69
5	207.68	207.68	207.68	209.82	1.03	210.08	1.16

Table 17: Natural frequencies for a multilayered Al-St plate with C-F-C-F B.C.

Mode Number	Natural Frequencies (Hz) for C-F-C-F Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	M. of H.	% Diff. QDFE vs. M. of H.
			# Elem: 1	# Elem: 196		# Elem: 196	
1	59.89	59.89	59.89	60.65	1.27	60.76	1.45
2	83.19	83.19	83.19	83.87	0.82	83.94	0.90
3	165.18	165.18	165.18	170.87	3.44	171.68	3.94
4	191.65	191.65	191.65	192.90	0.65	193.04	0.73
5	199.86	199.86	199.86	205.01	2.58	205.72	2.93

Table 18: Natural frequencies for a multilayered Al-St plate with C-C-C-F B.C.

Mode Number	Natural Frequencies (Hz) for C-C-C-F Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	M. of H.	% Diff. QDFE vs. M. of H.
			# Elem: 1	# Elem: 196		# Elem: 196	
1	72.09	72.09	72.09	72.76	0.93	72.84	1.04
2	178.30	178.30	178.30	179.77	0.82	179.95	0.93
3	178.50	178.50	178.50	183.91	3.03	184.67	3.46
4	288.49	288.49	288.49	293.46	1.72	294.12	1.95
5	337.94	337.94	337.94	359.13	6.27	362.23	7.19

Table 19: Natural frequencies for a multilayered Al-St plate with C-C-C-CB.C.

Mode Number	Natural Frequencies (Hz) for C-C-C-C Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	M. of H.	% Diff. QDFE vs. M. of H.
			# Elem: 1	# Elem: 196		# Elem: 196	
1	164.75	164.75	164.75	166.26	0.92	166.47	1.04
2	254.34	254.34	254.34	258.96	1.82	259.62	2.08
3	403.03	403.03	403.03	414.73	2.90	416.40	3.32
4	405.51	405.51	405.51	424.37	4.65	427.15	5.34
5	486.24	486.24	486.24	498.02	2.42	499.69	2.77

Table 20: Natural frequencies for a multilayered Al-St plate with C-C-C-S B.C.

Mode Number	Natural Frequencies (Hz) for C-C-C-S Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	M. of H.	% Diff. QDFE vs. M. of H.
			# Elem: 1	# Elem: 196		# Elem: 196	
1	130.52	130.52	130.52	131.47	0.73	131.59	0.82
2	231.58	231.58	231.58	236.41	2.09	237.08	2.37
3	335.83	335.83	335.83	343.24	2.21	344.28	2.52
4	389.83	389.83	389.83	409.49	5.04	412.35	5.78
5	428.42	428.42	428.42	437.23	2.06	438.43	2.34

Table 21: Natural frequencies for a multilayered Al-St plate with C-C-S-S B.C.

Mode Number	Natural Frequencies (Hz) for C-C-S-S Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	M. of H.	% Diff. QDFE vs. M. of H.
			# Elem: 1	# Elem: 196		# Elem: 196	
1	121.60	121.60	121.60	122.46	0.71	122.56	0.79
2	207.24	207.24	207.24	210.54	1.59	210.97	1.80
3	331.36	331.36	331.36	338.86	2.26	339.90	2.58
4	350.33	350.33	350.33	364.56	4.06	366.58	4.64
5	412.72	412.72	412.72	420.95	1.99	422.04	2.26

Table 22: Natural frequencies for a multilayered Al-St plate with S-C-S-CB.C.

Mode Number	Natural Frequencies (Hz) for S-C-S-C Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	M. of H.	% Diff. QDFE vs. M. of H.
			# Elem: 1	# Elem: 196		# Elem: 196	
1	152.72	152.72	152.72	154.20	0.97	154.39	1.09
2	213.94	213.94	213.94	216.29	1.10	216.60	1.24
3	333.59	333.59	333.59	343.18	2.87	344.52	3.28
4	396.15	396.15	396.15	408.07	3.01	409.76	3.44
5	460.39	460.39	460.39	471.58	2.43	473.13	2.77

Table 23: Natural frequencies for a multilayered Al-St plate with S-F-S-F B.C.

Mode Number	Natural Frequencies (Hz) for S-F-S-F Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	M. of H.	% Diff. QDFE vs. M. of H.
			# Elem: 1	# Elem: 196		# Elem: 196	
1	25.90	25.90	25.90	26.03	0.50	26.05	0.58
2	58.17	58.17	58.17	58.58	0.70	58.60	0.74
3	104.93	104.93	104.93	107.04	2.01	107.34	2.30
4	147.97	147.97	147.97	150.35	1.61	150.63	1.80
5	177.74	177.74	177.74	179.04	0.73	179.16	0.80

Table 24: Natural frequencies for a multilayered Al-St plate with S-S-S-F B.C.

Mode Number	Natural Frequencies (Hz) for S-S-S-F Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	M. of H.	% Diff. QDFE vs. M. of H.
			# Elem: 1	# Elem: 196		# Elem: 196	
1	37.00	37.00	37.00	37.24	0.65	37.25	0.68
2	117.90	117.90	117.90	120.07	1.84	120.35	2.08
3	129.33	129.33	129.33	130.23	0.70	130.31	0.76
4	219.92	219.92	219.92	222.80	1.31	223.11	1.45
5	250.97	250.97	250.97	261.71	4.28	263.22	4.88

Table 25: Natural frequencies for a multilayered Al-St plate with S-S-S-S B.C.

Mode Number	Natural Frequencies (Hz) for S-S-S-S Plate						
	Exact[118]	DCM	QDFE	16-DOF FEM	% Diff. QDFE vs. 16-DOF FEM	M. of H.	% Diff. QDFE vs. M. of H.
			# Elem: 1	# Elem: 196		# Elem: 196	
1	86.77	86.77	86.77	87.30	0.61	87.34	0.66
2	166.79	166.79	166.79	169.17	1.43	169.44	1.59
3	267.31	267.31	267.31	271.96	1.74	272.58	1.97
4	300.42	300.42	300.42	311.00	3.52	312.44	4.00
5	346.83	346.83	346.83	352.64	1.68	353.34	1.88

Once again, the numerical results in Tables 15 through 25 show that the layerwise QDFE results match well with the layerwise conventional FEM results. The QDFE results are also in good agreement with the results generated using the Method of Homogenization. However, the conventional FEM results are more accurate than the results produced by the Method of Homogenization. This is because the latter method uses the rule of mixtures to treat the two-layer Aluminium-Steel plate as a single layer plate homogenized with equivalent properties as opposed to the former method, which uses exact properties to form a layerwise solution procedure.

Similar to the homogeneous plate problem and as shown in Tables 5 through 25, the QDFE method produces exact results to two decimal places with just one element, for all 5 natural frequencies and all boundary conditions of the multilayered Aluminium-Steel plate. However, it

can be seen that the percent difference for the conventional FEM method and the Method of Homogenization increases for the higher modes even though 196 elements are used. So, for very high natural frequencies the layerwise FEM method will require thousands of elements to produce results within an accurate ballpark of the exact result. This proves that the QDFE method is a highly accurate super-convergent solution procedure. Thus, the layerwise QDFE method can be used as a strong alternative to replace the layerwise conventional FEM method for preliminary modal analysis of aircraft structures made of low curvature, layered sheets that are modelled as thin rectangular multilayered plates.

On the other hand, the layerwise DCM method also produces exact results to two decimal places for all 5 modes of the multilayered Aluminium-Steel plate, when any kind of boundary condition is imposed. The layerwise DCM results also match very well with the layerwise conventional FEM and Method of Homogenization results. As mentioned before, most analytical methods currently available are based on simplifying assumptions which limit their applicability to simple geometries and special boundary conditions. In contrast, the DCM solution does not benefit from any such simplification as it is built upon a very general quasi-exact solution to the plate governing equation derived using a distinctive procedure that has not been presented before. Thus, the availability of a layerwise DCM plate model means that researchers are now able to derive the quasi-exact results of 2D layered structures of low curvature and any edge dimension subjected to all types of boundary conditions.

Thus far in this thesis the QDFE and DCM methods have been applied to model the vibration behaviour of thin, rectangular homogeneous and multilayered plates and the numerical tests have consistently shown that the QDFE method is capable of producing exact results for the natural frequencies with one element due to the very rich nature of the quasi-exact solution as opposed to the conventional FEM method. Therefore, to further extend the QDFE method it will be applied to study the vibration characteristics of two aircraft cabin window bay sections in the next Chapter and the results produced will be compared with FEM and ANSYS[®] results.

5. VIBRATION BEHAVIOUR OF AN AIRCRAFT CABIN WINDOW BAY

5.1 Quasi-Exact Dynamic Finite Element Analysis of Cabin Window Bay Sections

In the previous Chapters the Quasi-Exact Dynamic Finite Element (QDFE) method was used to model simple, thin rectangular plates. The results stands proof to the super-convergent nature of the method, thus, to further highlight its suitability to effectively replace conventional FEM during preliminary design analysis, the QDFE method is applied to investigate the vibration behaviour of a cabin window bay section of an aircraft fuselage. Six different structural configurations made of Aluminium ($E = 69 \times 10^9$ Pa, $\rho = 2700$ kg/m³, $\nu = 0.32$) will be considered. Due to the unavailability of exact results, the ANSYS[®] software was used to generate a benchmark set of data for the first five natural frequencies of each configuration, using extremely fine meshing. The number of elements for each mesh was based on a convergence analysis aimed at deriving results that are exact to the second decimal place. The QDFE results generated for each configuration are compared with the benchmark data from ANSYS[®] as well as the results from the in-house developed 16-DOF FEM method. The number of elements used for the 16-DOF FEM method was also decided by carrying out a convergence analysis to reduce the percent difference below two percent, when compared with the benchmark data. The reinforcements around the window cut out are not accounted for in the first five configurations, however, in Configuration 6 the effect of having doublers to reinforce around the central cut out is studied. The two frames riveted along the vertical edges and the two stringers riveted along the horizontal edges of the window bay are modelled using a fully clamped boundary condition. Included below are the results for the first five natural frequencies of each cabin window bay configuration.

For Configuration 1 a rectangular window of length $L = 1.8$ m, width $W = 0.9$ m and thickness $t = 0.001$ m, having a rectangular central cut out of 0.3 m x 0.6 m (hatched area) is considered. Since, the mass distribution constants $k_1 = 0.333$ and $k_2 = 0.667$, the aspect ratio, AR , of the plate is 2 (k_2/k_1). The same AR is applied to the elements of the QDFE mesh. Shown below in Figure 18 are the in-house 16-DOF FEM mesh with 400 elements (a), QDFE mesh with 8 equal sized rectangular elements (b), and the ANSYS[®] benchmark mesh with 2877 elements (c), for the

aircraft cabin window bay Configuration 1. The ANSYS® meshes were created using the same SHELL 181 element used previously.

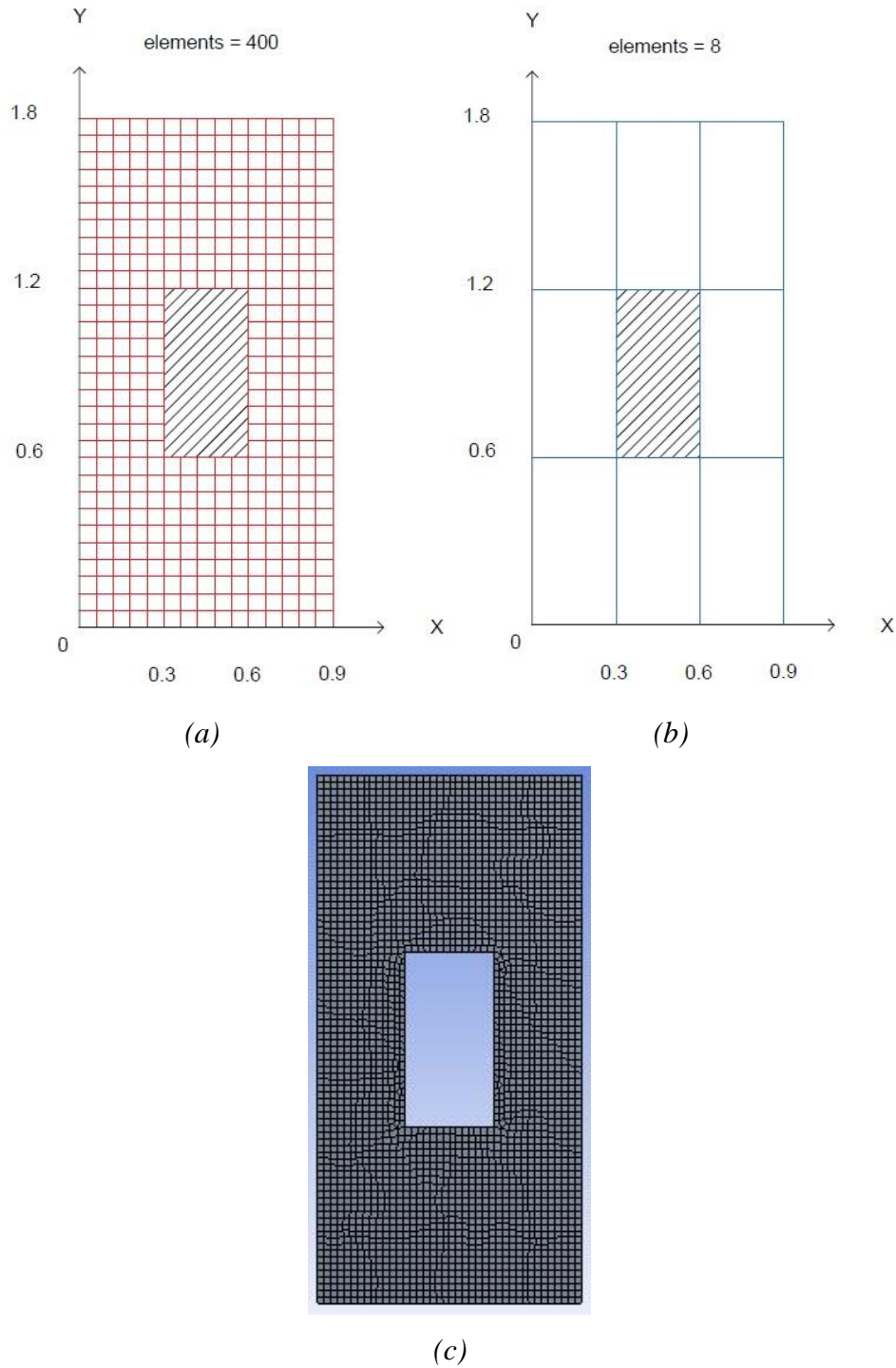


Figure 18: (a) 16-DOF FEM mesh- 400 elems., (b) QDFE mesh- 8 elems. and (c) ANSYS® benchmark mesh-2877 elems. for cabin window bay Configuration 1

Similar ANSYS® meshes were also created for the other five configurations but they are not presented for brevity. The results of the convergence analyses carried out to determine the minimum number of elements required for the benchmark data produced on ANSYS® and the results produced by the in-house 16-DOF FEM method for Configuration 1 are graphically illustrated in Figures 19 and 20 below, respectively. For both cases the 5th natural frequency of the cabin window bay section was used as it is the most unstable frequency. Similar convergence analyses were carried out to determine the minimum number of elements required for the other five configurations as well, however, their respective results are not included here for brevity.

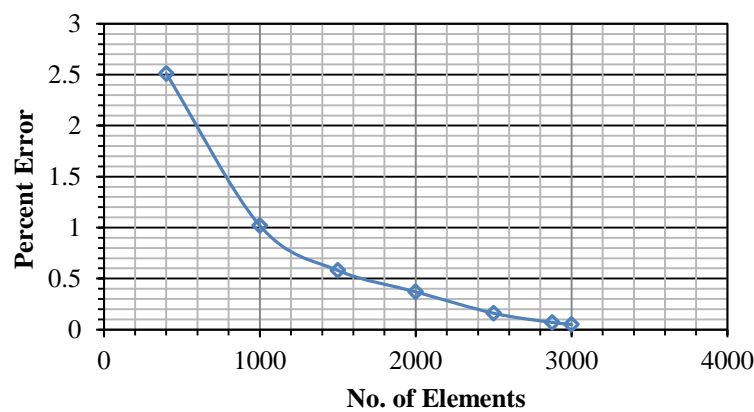


Figure 19: Convergence analyses for the benchmark data produced on ANSYS®

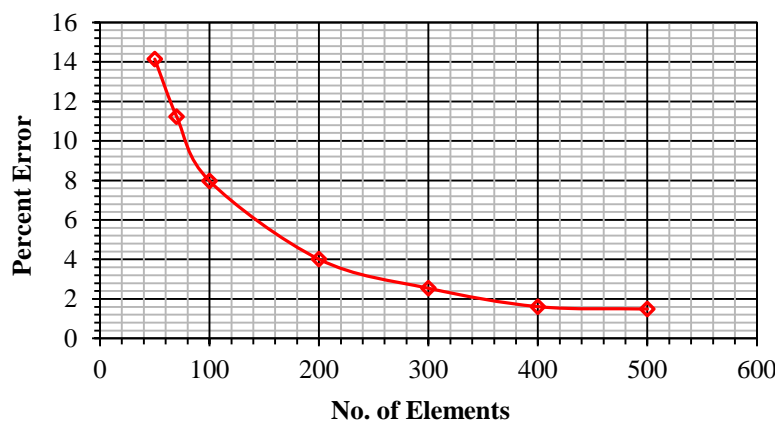


Figure 20: Convergence analyses for the in-house 16-DOF FEM method

Included in Table 26 below are the results for the first five natural frequencies of window bay configurations 1.

Table 26: Natural frequency results for aircraft cabin window bay section – configuration 1

Mode Number	Natural Frequencies (Hz)			
	Benchmark (ANSYS®: 2877 elems)	QDFE (8 elems)	16-DOF FEM (400 elems)	Percent Difference (%) 16-DOF FEM vs. QDFE
1	8.67	8.67	8.72	0.58
2	9.62	9.62	9.67	0.52
3	14.04	14.04	14.17	0.92
4	15.14	15.14	15.21	0.46
5	19.67	19.67	19.99	1.61

As can be seen from the results in Tables 26 above, the QDFE method is capable of converging to the benchmark solution produced on ANSYS® with just 8 elements. That the QDFE method is capable of yielding the same result for the first five natural frequencies that required the ANSYS® simulation 2877 elements (approximately 360 times more elements than QDFE) is attributed to the fact that the new trigonometric shape functions are based on the quasi-exact solution to the governing differential equation. The above finding stands proof to the richness of the quasi-exact solution. Furthermore, compared to the QDFE method, the in-house developed 16-DOF FEM method requires 400 elements to produce results that are within a 2 percent difference. Despite employing 400 elements the 16-DOF FEM method still does not produce the benchmark results and the percent difference increases with the mode number. In order to arrive even at this level of accuracy the 16-DOF FEM method requires 50 times more elements than the QDFE method and that is still only for the first five natural frequencies. As a solution is sought for higher natural frequencies, which become ever more unstable and chaotic, the conventional FEM method will need hundreds if not thousands more elements to retain the same degree of precision. Therefore, it is clear that the QDFE method could be advantageously used to capture the higher natural frequencies that are more commonly present in aerospace environments. Therefore, in summary the above results show that the QDFE method is a superconvergent method, which is capable of producing near-exact results with a very small number of elements whose advantage becomes more pronounced as the mode number increases.

For aircraft cabin window bay configuration 2, consider a square section with length $L = 1.5\text{m}$, width $W = 1.5\text{m}$ and thickness $t = 0.001\text{m}$ having a square shaped central cut out of $0.5\text{m} \times 0.5\text{m}$. Since, the mass distribution constants $k_1 = 0.5$ and $k_2 = 0.5$, the aspect ratio, AR , of the

plate is 1 (k_2/k_1). The same AR is applied to the elements of the QDFE mesh. The 16-DOF FEM mesh uses 512 elements and the QDFE mesh is made of 8 equal sized square elements. Figure 21 below shows the two meshes for cabin window bay configuration 2.

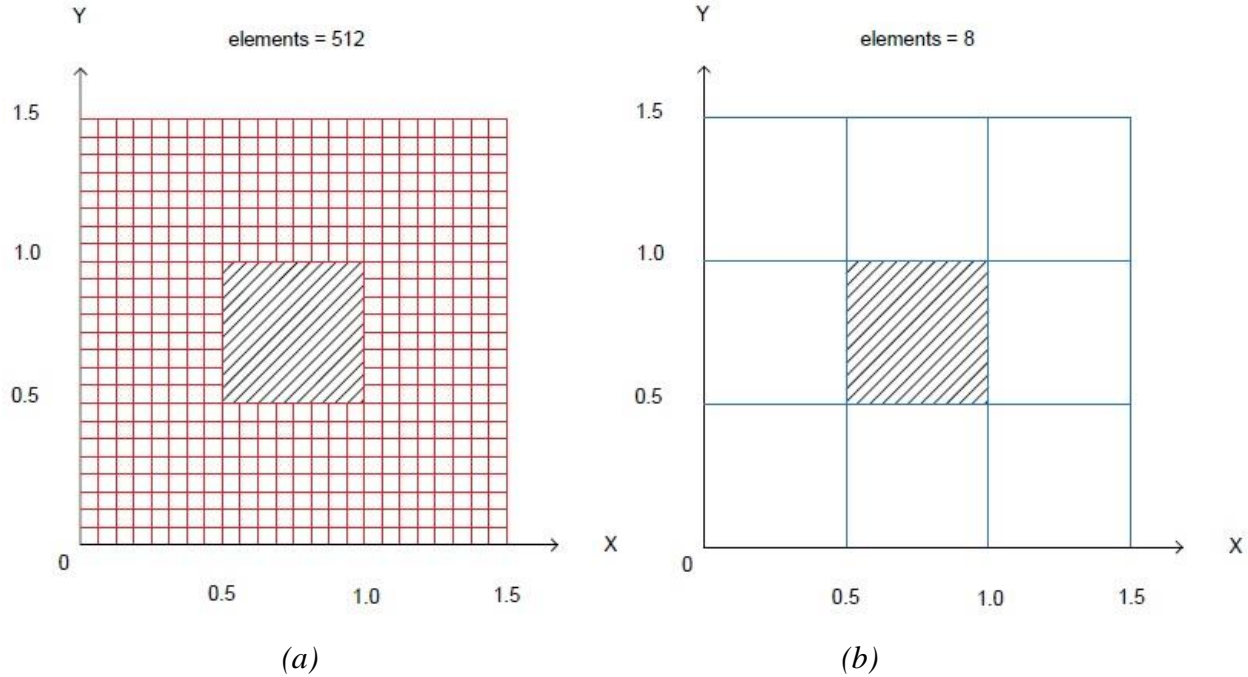


Figure 21: (a) 16-DOF FEM mesh- 512 elems., (b) QDFE mesh- 8 elems. for Configuration 2

Table 27 below shows the results for the first five natural frequencies of cabin window bay configuration 2.

Table 27: Natural frequency results for aircraft cabin window bay section – configuration 2

Mode Number	Natural Frequencies (Hz)			
	Benchmark (ANSYS®: 3926 elems)	QDFE (8 elems)	16-DOF FEM (512 elems)	Percent Difference (%) 16-DOF FEM vs. QDFE
1	4.67	4.67	4.67	0.00
2	7.03	7.03	7.07	0.57
3	7.03	7.03	7.07	0.57
4	10.85	10.85	10.91	0.55
5	12.48	12.48	12.68	1.59

As seen before the results in Table 27 show that the QDFE method is capable of reproducing the benchmark results, based on a 3926 element ANSYS® model, with just 8 elements. On the contrary, the in-house developed 16-DOF FEM method requires at least 512 elements to generate

results that are within a 2 percent difference. Thus, it's evident that the 16-DOF FEM method requires 64 times more elements than the QDFE method and the latter also uses approximately 490 times less elements than the benchmark ANSYS[®] solution. Also once again, by observing how the percent difference increases with the mode number, it could be deduced that many more elements would be required for the 16-DOF FEM method, if it is to retain the same level of accuracy when results for higher natural frequencies are pursued.

For aircraft cabin window bay configuration 3 a rectangular section having length $L = 2.0\text{m}$, width $W = 1.75\text{m}$ and thickness $t = 0.001\text{m}$ is considered. The plate has a rectangular cut out of length 0.5m and width 0.25m at the centre. The mass distribution constants $k_1 = 0.467$ and $k_2 = 0.533$ for the plate, therefore, its aspect ratio, AR , is 1.143 (k_2/k_1). Illustrated in Figure 22 below are the meshes generated by the in-house developed 16-DOF FEM method (a) and QDFE method (b). The 16-DOF FEM mesh and QDFE mesh has 864 and 8 elements, respectively. What is important to note here is that there are 3 different types of elements used for the QDFE mesh. The QDFE mesh is made up of four $0.75\text{m} \times 0.75\text{m}$ square elements, two $0.75\text{m} \times 0.5\text{m}$ rectangular elements and two $0.75\text{m} \times 0.25\text{m}$ rectangular elements.

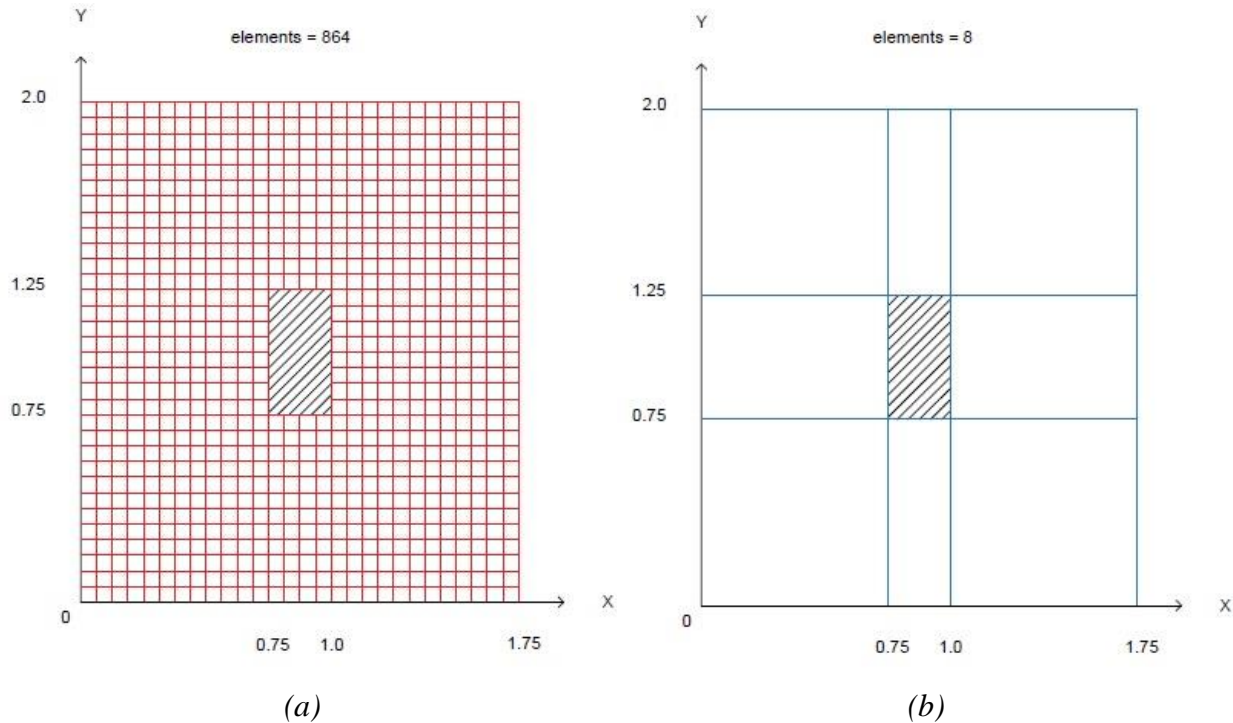


Figure 22: (a) 16-DOF FEM mesh- 864 elems., (b) QDFE mesh- 8 elems., for Configuration 3

Table 28 below contains the results for the first five natural frequencies of cabin bay window configuration 3.

Table 28: Natural frequency results for aircraft cabin window bay section – configuration 3

Mode Number	Natural Frequencies (Hz)			
	Benchmark (ANSYS®: 6618 elems)	QDFE (8 elems)	16-DOF FEM (864 elems)	Percent Difference (%) 16-DOF FEM vs. QDFE
1	2.55	2.55	2.55	0.00
2	4.67	4.67	4.68	0.21
3	5.14	5.14	5.16	0.39
4	7.41	7.41	7.45	0.54
5	8.36	8.36	8.43	0.83

Once again, it can be seen that with just 8 elements the QDFE method produces the same results as the benchmark data derived from the 6618 element ANSYS® model. In comparison to this the 16-DOF FEM method requires 864 elements to produce results that are within a 2 percent difference compared to the QDFE method. Therefore, the QDFE method requires 108 times less elements than the 16-DOF FEM method and over 827 times less elements compared to the benchmark ANSYS® simulation. This proves that the QDFE method is a super-convergent procedure that requires a very coarse mesh. This greatly reduces the number of element matrices to be computed, thus, resulting in a much smaller eigenvalue problem to be solved which drastically reduces the computational time. A more detailed discussion on the time savings associated with the QDFE method will be included later in this section.

Configuration 4 of the aircraft cabin window bay section has a rectangular shape. It's length $L = 1.5\text{m}$, width $W = 1\text{m}$ and thickness $t = 0.001\text{m}$. It consists of a square shaped central cut out having dimensions $0.5\text{m} \times 0.5\text{m}$. For configuration 4, the mass distribution constants $k_1 = 0.4$ and $k_2 = 0.6$, thus, the aspect ratio, AR , is $1.5 (k_2/k_1)$. The 16-DOF FEM (a) and QDFE (b) meshes for configuration 4 can be seen in Figure 23 below. The QDFE mesh for configuration 4 also consists of two different types of elements. Two of the elements are $0.5\text{m} \times 0.5\text{m}$ square shaped elements and the other six elements are $0.5\text{m} \times 0.25\text{m}$ rectangular shaped elements.

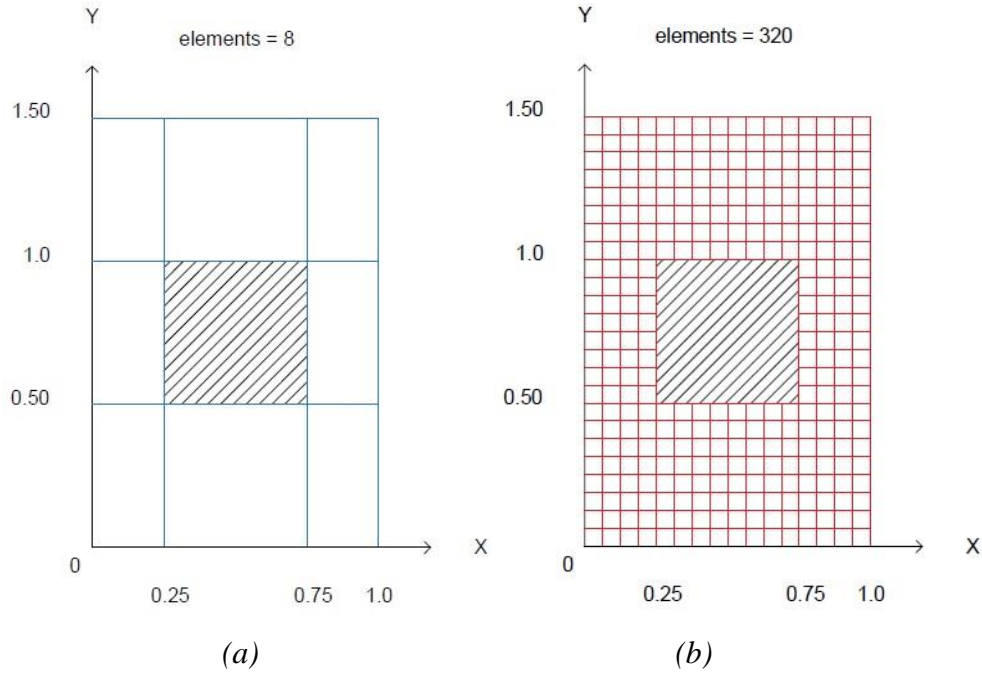


Figure 23: (a) 16-DOF FEM mesh- 320 elems., (b) QDFE mesh- 8 elems., for Configuration 4

The results for the first five natural frequencies of cabin window bay configuration 4 are shown in Table 29 below.

Table 29: Natural frequency results for aircraft cabin window bay section – configuration 4

Mode Number	Natural Frequencies (Hz)			
	Benchmark (ANSYS®: 2518 elems)	QDFE (8 elems)	16-DOF FEM (320 elems)	Percent Difference (%) 16-DOF FEM vs. QDFE
1	8.69	8.69	8.72	0.34
2	9.20	9.20	9.23	0.33
3	16.17	16.17	16.29	0.74
4	18.50	18.50	18.68	0.97
5	20.30	20.30	20.51	1.03

The results are consistent for every configuration that has been investigated thus far and the data found in Table 29 above further confirms the super-convergent behaviour of the QDFE method. It shows that the 16-DOF FEM method requires 320 elements to yield results that are within a two percent difference compared to the QDFE method, which only requires 8 elements once again. This is 40 fold reduction in the number of elements used compared to the 16-DOF FEM method. As mentioned before the benchmark ANSYS® formulation is based on a SHELL 181, four-noded, rectangular element with six degrees of freedom per node, where the degrees of

freedom are the three translations and three rotations in the x, y and z spatial directions. The QDFE method is based on a four-noded, four DOF per node element and generally, the higher the degrees of freedom per node of the elements the more accurately it is capable of predicting the behaviour of the structure. Despite this, the QDFE method is capable of reproducing the benchmark data, which is based on 2518 SHELL 181 element model, with just 8 elements. This is over 314 times less elements than what was required by the ANSYS® simulation.

Window bay configuration 5 has length $L = 1.5\text{m}$, width $W = 1.25\text{m}$ and thickness $t = 0.001\text{m}$. It also has a central window cut out of length 0.5m and width 0.25m . The mass distribution constants for configuration 5 are, $k_1 = 0.455$ and $k_2 = 0.545$. Thus, the aspect ratio, AR , of the window bay section is 1.2 (k_2/k_1). Figure 24 below shows the 16-DOF FEM (a) and QDFE (b) meshes for window bay configuration 5. As can be seen, two different types of elements are used for the QDFE mesh. Among the eight elements used, two are $0.5\text{m} \times 0.25\text{m}$ rectangular elements and six are $0.5\text{m} \times 0.5\text{m}$ square shaped elements.

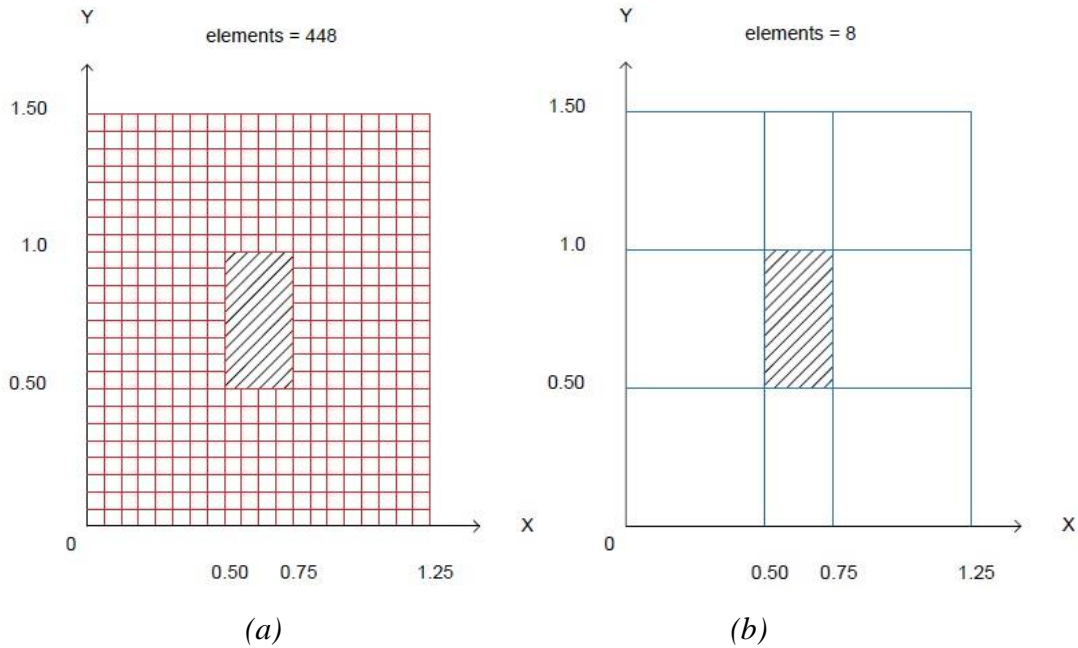


Figure 24: (a) 16-DOF FEM mesh- 448 elems., (b) QDFE mesh- 8 elems., for Configuration 5

The modal analysis results for cabin window bay configuration 5 are included in Table 30 below.

Table 30: Natural frequency results for aircraft cabin window bay section – configuration 5

Mode Number	Natural Frequencies (Hz)			
	Benchmark (ANSYS®: 3364 elems)	QDFE (8 elems)	16-DOF FEM (448 elems)	Percent Difference (%) 16-DOF FEM vs. QDFE
1	5.06	5.06	5.06	0.00
2	8.42	8.42	8.45	0.36
3	8.77	8.77	8.85	0.91
4	13.52	13.52	13.64	0.88
5	14.92	14.92	15.13	1.40

The vibration analysis results for configuration 5 also shows that the QDFE method is capable of converging to near-exact results with a very coarse mesh as it only required 8 elements to reproduce the benchmark data. This is 56 times less elements than the 16-DOF FEM method and over 420 times less elements than the ANSYS® simulation.

Both the ANSYS® simulation and the 16-DOF FEM results are based on the conventional Finite Element Method, which is known to require a very fine mesh even for the lower natural frequencies shown here. Therefore, for much higher natural frequencies tens of thousands of elements would be required by conventional FEM. Although, widely used due to its versatile nature, the average run time for a conventional FEM based modal analysis of a fairly common industrial scale aerospace application with all its intricacies, such as a full fuselage or entire wing box, is no less than a few days. So it is very common to see designers run an FEM analysis and wait for a number of days for the solution to converge, thus, wasting precious time and resources that could be utilized elsewhere during the early design stages. This is because there are hundreds of thousands of elements in such large real life Finite Element models and more often than not the engineers are interested in determining the higher natural frequencies that could be encountered during real-life operation.

In addition to solver time, there is also a lot of time involved in generating such complicated meshes in the first place, thus, further contributing to the time lost. The higher the number of elements used the more mathematical complexity and computation that is involved and there is a direct correlation between the complexity of the mesh and the total run time for the solution. This inefficiency is largely due to the strictly numerical nature of conventional FEM. On the other hand, the QDFE method is a semi-analytical method, which combines the accuracy of the

exact methods to the versatility of conventional FEM. As the shape functions used in the QDFE method are formed using the terms of the quasi-exact solution to the governing equation, it is inherently more accurate than a strictly numerical tool and as such it is capable of precisely predicting the natural frequencies of the structure to an extremely accurate ballpark of the exact value, with a limited number of elements (i.e. coarser mesh). This advantage provided by the QDFE method grows exponentially as one seeks a solution for higher mode numbers. As conventional FEM continuously requires more and more elements to determine the higher modes to a reasonably acceptable degree of precision the QDFE method would require only a few more elements to yield much more accurate (near-exact) results, thus, not only improving the reliability of the data by improving its accuracy, but also saving a lot of run time and time spent building the mesh as the QDFE mesh is comparatively several magnitudes simpler. Therefore, one could only imagine the potential of such a powerful tool, once it is developed to a full-fledged, versatile method such as conventional FEM. In order to achieve that, firstly a comprehensive library of the QDFE models for the basic structural elements should be developed, as it was done for conventional FEM during its inception a few decades back, and this research is a step in that direction.

In Configuration 6, a rectangular aircraft cabin window bay section with length $L = 2.5\text{m}$, width $W = 1\text{m}$ and thickness $t = 0.001\text{m}$ is considered. It also has a $0.5\text{m} \times 0.25\text{m}$ rectangular cut out at the centre. However, unlike other configurations, the effect of having doublers around the perimeter of the cut out for additional structural reinforcement is also incorporated in to Configuration 6. This is achieved by increasing the thickness of the plate around the cut out to 0.005m . Figure 25 below illustrates the 16-DOF FEM mesh (a) and QDFE mesh (b) for cabin window bay Configuration 6. The area in yellow colour, around the cut out (hatched region) is the area where the thickness has been increased in order to simulate the effect of having doublers attached for structural reinforcement. The mass distribution constants for cabin window bay Configuration 6 are, $k_1 = 0.286$ and $k_2 = 0.714$. Therefore, the aspect ratio, AR , of this window bay section is $2.5(k_2/k_1)$. For Configuration 6, the 16-DOF FEM mesh used 608 elements and the QDFE mesh required 24 elements, out of which 20 elements are rectangular shaped. They are comprised of six $0.875\text{m} \times 0.25\text{m}$ elements, four $0.875\text{m} \times 0.125\text{m}$ elements, four $0.25\text{m} \times 0.125\text{m}$ elements and two $0.5\text{m} \times 0.125\text{m}$ elements. The other four elements are $0.125\text{m} \times$

0.125msquare shaped elements. The eight elements shaded in yellow colour have a higher thickness (0.005m) compared to the other elements (0.001m). This introduces an extra degree of rigidity and stiffness to the structure in the perimetric region of the cut out as would have been in the case where doublers were present for added reinforcement.

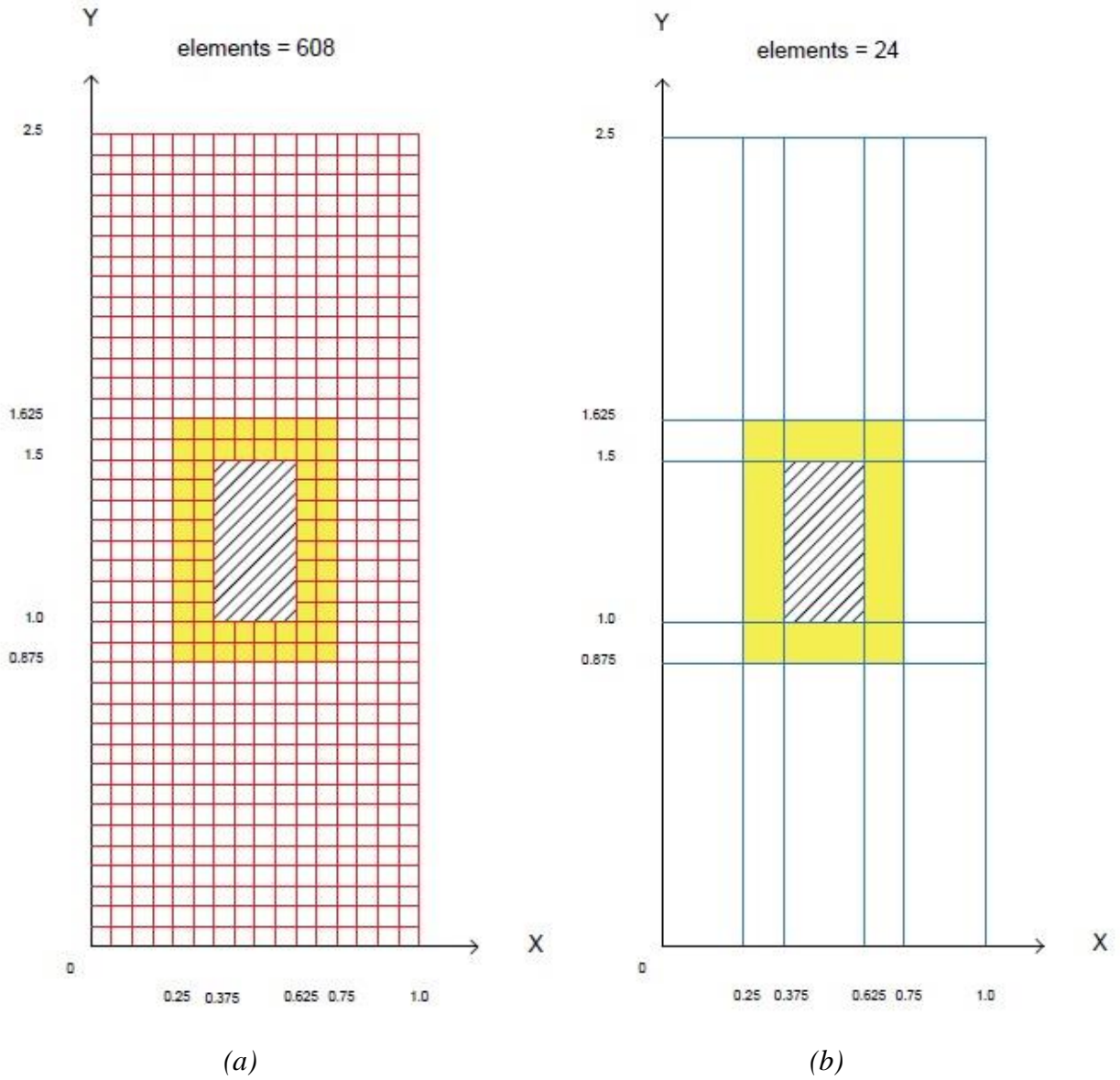


Figure 25: (a) 16-DOF FEM mesh- 608 elems., (b) QDFE mesh- 24 elems., for Configuration 6

Figure 26 below depicts how the thick elements are connected to the thin elements along their centrelines in configuration 6 of the cabin window bay section where there is additional reinforcement to model the presence of doublers around the perimeter of the cutout.

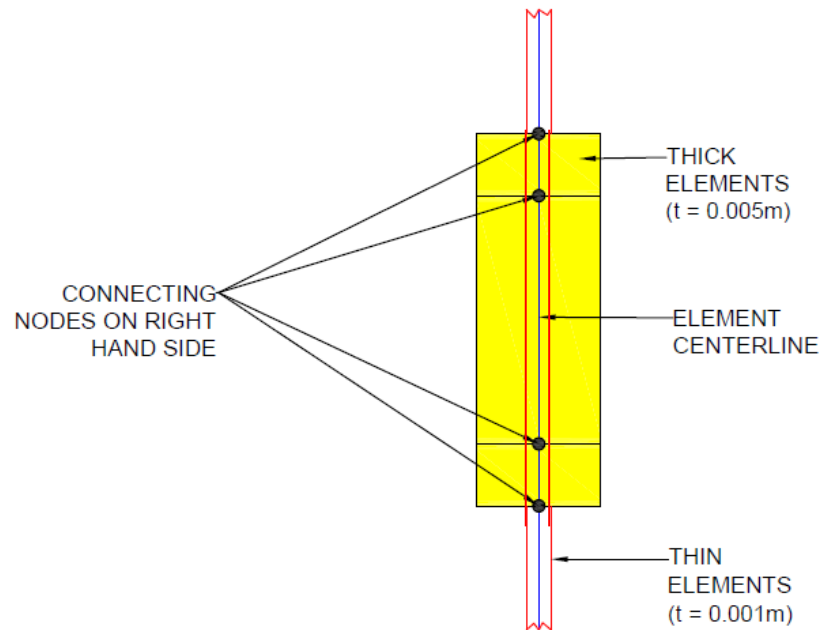


Figure 26: Element connectivity between thick and thin elements surrounding the cut out for configuration 6 of the cabin window bay section

Table 31 below includes the modal analysis results for the first five natural frequencies of aircraft cabin window bay Configuration 6.

Table 31: Natural frequency results for aircraft cabin window bay section – configuration 6

Mode Number	Natural Frequencies (Hz)			
	Benchmark (ANSYS®: 4565 elems)	QDFE (24 elems)	16-DOF FEM (608 elems)	Percent Difference (%) 16-DOF FEM vs. QDFE
1	6.91	6.91	6.91	0.00
2	7.58	7.58	7.60	0.26
3	10.47	10.47	10.56	0.86
4	12.04	12.04	12.17	1.07
5	15.26	15.26	15.49	1.50

For Configuration 6 with the reinforcement around the perimeter of the cut out similar results are seen. The benchmark data were generated using a 4565 element ANSYS[®] model and used for comparing the QDFE and 16-DOF FEM results. In this case the QDFE method required 24 elements to produce the exact same result as the ANSYS[®] simulation. Comparatively, the 16-DOF FEM method required 608 elements to arrive at a solution within a two percent difference. Therefore, once again it can be seen that the QDFE method requires over 25 times less elements than the 16-DOF FEM method and over 190 times less elements compared to the benchmark solution from ANSYS[®].

Lastly, it is important to briefly address the absence of experimental data to further verify the QDFE results. The long term goal of this entire research is to contribute towards the development of a new, versatile, semi-analytical tool for the modal analysis of airframe components that would be a more computationally efficient alternative to conventional FEM dsuitable for the preliminary design stages. However, in order to achieve that objective in the distant future one needs to develop the Dynamic Finite Element models for the most basic structural elements such as beams, plates and shells for all the different boundary condition types. Not only that, these basic element models should then also be extended to incorporate the effects of some common situations, including but not limited to, having multiple layers of different materials, being made of advanced composites and consisting of delamination. Many researchers in the past have contributed their share to achieve this monumental task and in doing so have covered various beam related structural configurations, thereby by gradually furnishing a comprehensive library of Dynamic Finite Element models. However, this work is the first time that the Dynamic Finite Element concept has been extended to the two dimensional elements such as thin plates. Therefore, what is crucial for the reader to understand here is that the purpose of this work is to revolutionize the solution process. The sole objective is to create a more efficient semi-analytical tool that, eventually can, replace conventional FEM computationally. But the problem has not changed. More specifically, the problem is not a brand new one that has not been studied before. What is captured through the application of the QDFE method is not something that was not caught in the past. The problems that are under investigation here have been studied for decades with the use of conventional FEM as well as many other exact, semi-analytical and numerical techniques and the results have been well published in the open

literature. What is new here is the solution technique, which is capable of producing much more reliable results with considerably reduced mesh complexity and solver time. Therefore, a validation of the QDFE method through experimental investigation would add very little. For the purpose of authenticating the results, a comparison with exact data found in the open literature or in the absence of such analytical data, the results from a Finite Element simulation on a commonly used software such ANSYS® should be sufficient.

5.2 Time Verification for the Quasi-Exact Dynamic Finite Element Method

In this section the CPU (Central Processing Unit) time taken to solve the eigenvalue problem for the 16-DOF FEM and QDFE methods are compared with each other. As an example, Configuration 6 of the aircraft cabin window bay section presented above is considered for time verification purposes. On an ASUS TP550L computer station in this case, two series of analyses was performed. Firstly, the CPU time elapsed to solve the eigenvalue problem for the first natural frequency of aircraft cabin window bay configuration 6 was recorded for both the 16-DOF FEM method and the QDFE method. Subsequently, the time taken to determine the first five natural frequencies was also monitored for both methods. These results are illustrated in Figure 27 below. In each case the tests were repeated five times and an average CPU time was calculated. For the QDFE method the frequency domain was swept for frequencies that render the determinant of the system to be zero (i.e. natural frequencies), using a MATLAB® code with an embedded root finding algorithm [120].

As mentioned before, the mesh for the 16-DOF FEM method has 608 elements and the QDFE mesh consists of 24 elements. Also, it is important to note here that the 16-DOF FEM method, even with 608 elements, only yields results to within a two percent difference of the benchmark data.

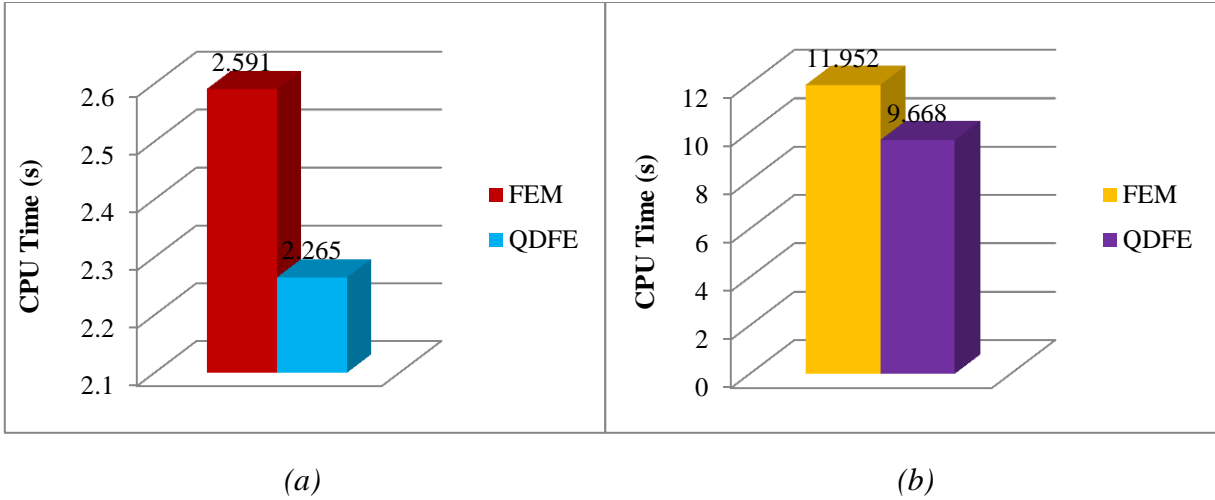


Figure 27: CPU time to determine the first natural frequency (a) CPU time to determine the first five natural frequencies (b)

The first analysis shows that to arrive at the first natural frequency the 16-DOF FEM method requires 2.591s and the QDFE method takes 2.265s. Although, the difference looks insignificant here, it is still a 14.4% percent reduction in the time elapsed, and even more impressively that is to produce results that are similar to the benchmark data to two decimal places. To arrive at this result, the benchmark solution, based on a 4565 element ANSYS® model took 10.460s.

The second test showed that the 16-DOF FEM method required 11.952s to arrive at the solution for the first five natural frequencies, with an overall percent difference less than two percent. In contrast the QDFE method took 9.668s to converge to the benchmark solution for the first five natural frequencies to two decimal places. This is a 2.284s advantage and in this case the QDFE method is 23.6% faster than the 16-DOF FEM method. The ANSYS® solution with 4565 elements took 51.403s to produce the benchmark results for the first five natural frequencies. Therefore, the reduction in solver time is clearly evident as the mode number increases.

The solver time saving would increase exponentially when the problem is extrapolated to a much larger and more intricate structure such as an aircraft fuselage or a wing box and the higher natural frequencies are pursued. Thus, conclusively this confirms once more, that when higher natural frequencies and improved reliability of the results are of interest, the QDFE method can be advantageously used.

This proves that the QDFE method is a superconvergent procedure which is more suitable for preliminary stage modal analysis where time and resources are as important as the accuracy of the result itself. Thus, the availability of a generic QDFE model for thin plates now leaves engineers with a powerful tool with which to determine the vibration behaviour of simplified structures such as the aircraft cabin window configurations investigated above. However, in order to apply the Dynamic Finite Element concept to much more complex structures and to perform a rather detailed analysis on them as can be done using conventional FEM requires further development. More specifically, it requires the generic DFE models for structural elements such as shells and stiffened shell configurations to be complete.

6. CONCLUDING REMARKS

6.1 Synopsis

In this study, a Quasi Exact Dynamic Finite Element (QDFE) method and a Dynamic Coefficient Matrix (DCM) method was developed for thin rectangular plates. Prior to forming the QDFE and DCM solutions, a quasi-exact solution to the governing partial differential equation was sought using an unique approach never presented before, whereby the characteristic equation was re-arranged as the sum of two beam-like expressions representing each spatial direction of the plate and each beam-like expression was separately solved by applying the quadratic formula to obtain sixteen roots. Out of the sixteen roots, eight satisfied the characteristic equation fully and the other eight did not, thus, giving rise to a quasi-exact solution. Based on a 4 node, 4-DOF per node rectangular element and using the terms of the general expression for the quasi-exact solution, sixteen frequency-dependant basis functions characteristic to the Dynamic Finite Element procedure were formed. These new basis functions were subsequently exploited to create sixteen new trigonometric shape functions. Using these shape functions the QDFE thin plate model was later developed and numerous tests were carried out using the new QDFE method. At the beginning the QDFE method was applied to investigate the vibration behaviour of two different rectangular plates subjected to eleven classical boundary conditions. One of the two plates was an intact, homogeneous, plate while the other one was a hybrid, two layered plate. The results from these modal analyses showed that the QDFE method is a superconvergent, semi-analytical method that has the potential to fill the void between the numerical and analytical methods that are already available for use. The capabilities of the QDFE method can be best exploited during the preliminary design stages of plate-type structures and structural components, where the goal is to reduce the time spent by creating a coarse model for modal analyses and minimize computational overhead without compromising the accuracy of results. The new trigonometric, frequency-dependant shape functions, unique to the QDFE method, were capable of producing exact results to two decimal places using just one element for the first five natural frequencies. In comparison, for both homogeneous and multilayered plates the 12-DOFFEM and 16-DOF FEM methods, which use polynomial based shape functions, required at least 196 elements to arrive at results that are within a 10 percent difference, suggesting that for much higher mode numbers the conventional FEM methods would require

tens of thousands of elements. This very high efficiency of the QDFE shape functions is attributed to the fact that they are developed using expressions from the quasi- exact solution to the governing differential equation. However, the QDFE shape functions, exhibit behaviour identical to the polynomial shape functions at each node and they are designed in such a way that they approach their FEM counterparts as the natural frequency tends to zero (static case). Furthermore, despite their differences, the formulation process for the QDFE method is very similar to that of the FEM method and this is an extra advantage as most researchers are very familiar with the conventional FEM formulation.

The findings from the cabin window analyses were also consistent with the homogeneous and multilayered plate studies described above. Six different cabin window bay configurations were investigated and the results produced by the QDFE method were compared with the 16-DOF FEM data and an ANSYS[®] simulation based on a very fine mesh. For the first five configurations the QDFE method produced results that are identical to the benchmark data to two decimal places with just 8 elements. On the contrary the 16-DOF FEM method required between 312 and 864 elements, depending on the cabin window bay configuration. Thus, the number of elements required by the QDFE method for different configurations varied between 39 to 108 times less than what was used by the 16-DOF FEM method. For the sixth configuration, a cabin window bay with thicker elements surrounding the cut out region was studied to simulate the presence of doublers for further structural reinforcement and even for this case the QDFE method required over 25 times less elements. Therefore, all the studies conducted consistently proved that the QDFE method is capable of yielding very accurate results with a much coarser mesh for the first five natural frequencies compared to the conventional FEM method.

A time verification was also carried out to quantify the savings made on computational time as a result of the coarser mesh that the QDFE method uses. It was found here that, even for just one natural frequency the QDFE method is capable of consuming 14.4% less computational time than the 16-DOF FEM method in addition to the higher accuracy that it provides. The same analysis also showed that a time saving of 23.6% is achieved if the first five natural frequencies are sought using the QDFE method as opposed to using the 16-DOF FEM method. This means that the time savings made when the QDFE method is used increases exponentially as the higher

natural frequencies are sought and this further proves that when higher mode numbers and more accuracy is of interest the QDFE method has an advantage over conventional FEM.

In parallel to the QDFE solution, a Dynamic Coefficient Matrix (DCM) method was also developed to study the vibration behaviour of the homogeneous and multilayered plate problems subjected to eleven different boundary conditions, using the distinct quasi-exact solution that is not based on any simplifying assumption. The results showed that the DCM method yields exact results to two decimal places for all the plate problems analysed, just like the QDFE method. Although, somewhat limited in its application due to its strictly analytical nature, the main advantage of the DCM method is that it is a very general method, which can be readily applied to determine the exact modal characteristics of a wide range of thin, rectangular, homogeneous and multilayered plates of any dimension or aspect ratio, subjected to all types of boundary conditions, in contrast to the other exact methods that have been tried in the past, which are only applicable to simple and special cases such as square plates or plates with two opposing edges simply supported.

6.2 Key Contributions

The key contributions of this thesis have been summarized and enumerated below.

1. Development of a unique quasi-exact solution to the thin plates' governing partial differential equation based on Classical Plate Theory (CPT) using a distinctive procedure that has not been followed before, whereby the plate governing equation is treated as the sum of two beam-like expressions.
2. Formation of frequency-dependant, trigonometric basis functions for a 4-node, 4-DOF per node, thin rectangular plate element using the newly found quasi-exact solutions such that the trigonometric basis functions approach the polynomial basis functions used in conventional FEM when the natural frequency tends to zero.
3. Derivation of 16 trigonometric, frequency-dependant shape functions for a 4-node, 4-DOF per node, thin rectangular plate element that approximates the flexural displacements, slopes and curvatures identically to the polynomial shape functions of conventional FEM.

4. Formulation of a Quasi-Exact Dynamic Finite Element (QDFE) method as a highly accurate, superconvergent, semi-analytical procedure that can be confidently and accurately used to perform vibration analysis of two-dimensional (2D) airframe components that are modelled as thin, rectangular, homogeneous plates during the preliminary design stages using a very coarse mesh. The objective here is to develop a powerful alternative method of analysis that is ideal for use during the early stages of design where accuracy and speed is both important.
5. Development of a new Dynamic Coefficient Matrix (DCM) method as an alternative, more general, quasi-exact procedure, that does not incorporate any simplifying assumptions, and in so doing is not limited to simple and special cases. This will leave researchers with a flexible quasi-exact tool that can be exploited to determine the natural frequencies of thin, rectangular, homogeneous plates of any size, subjected to any type of boundary condition.
6. Extension of the Quasi-Exact Dynamic Finite Element (QDFE) solution to investigate the vibration behaviour of thin, two-layer, rectangular plates of any size, subjected to any type of boundary condition.
7. Extension of the Dynamic Coefficient Matrix (DCM) method to study the modal characteristics of thin, two-layer, rectangular plates having varying edge lengths and edge conditions.

6.3 Impact of Research

The new plate QDFE model can be advantageously used for dynamic analysis and vibration analysis of plates, plate assemblages and plate-type structures with low curvature. Furthermore, the plate and beam DFE models can be combined in the future to model complex aerospace structures, such as stiffened plates. The DCM plate model leaves engineers with a powerful quasi-exact method that can be exploited to investigate the vibration characteristics of structural components that are modelled as thin, rectangular, homogeneous or layered plates. Most importantly, engineers are now made available a quasi-exact solution procedure that is not based on any simplifying assumptions and that has its generality intact. Therefore, the contributions of this research are expected to make a major impact across structural vibration communities. As a result of the increased reliability in free vibration analysis results, the accuracy of response

analysis, aeroelastic research work, and structural health monitoring will increase. Increased accuracy will also improve structural optimization, resulting in more efficient structural designs with improved vibration, response, and aeroelastic features.

6.4 Recommendations for Future Work

The future research will first focus on the extension of the QDFE model to investigate the vibration behaviour of thick plates using first order and higher order shear deformation theories. In addition to this, research is also underway to use non-rectangular, arbitrary shaped four-noded elements to develop a novel Dynamic Finite Element solution for thin plates. The next step would be to utilize the capabilities of the QDFE method to determine the vibration behaviour of composite structures, which are both intact and delaminated. Combining the DFE beam models formulated in the past with the QDFE plate model presented here the modal behaviour of stiffened plates or plate assemblies can also be studied. Furthermore, exploiting the high accuracy of the QDFE method, laminates embedded with piezoelectric layers acting as sensors could also be modeled upon further development.

6.5 List of Publications

Journal Paper Publications

1. **Supun Jayasinghe**, and Seyed M. Hashemi “A Symbolic Dynamic Finite Element Formulation for Multilayered Thin Rectangular Plates,” For submission to Computers & Structures (*Special Issue*).
 2. **Supun Jayasinghe** and Seyed M. Hashemi “A Quasi-Exact Dynamic Finite Element Method for the Free Vibration Analysis of Thin Rectangular Plates,” Journal of Sound and Vibrations (JSV) (Under Review; Submitted; February 11, 2018, Ref #: JSV-S-18-00483).
 3. **Supun Jayasinghe** and Seyed M. Hashemi, “New Frequency-Dependent Trigonometric Interpolation Functions for the Dynamic Finite Element Analysis of Thin Rectangular Plates”, Shock & Vibration Journal, vol. 2018, Article ID 6980536, 16 pages, 2018. <https://doi.org/10.1155/2018/6980536>.
 4. **Supun Jayasinghe** and Seyed M. Hashemi, “A Dynamic Coefficient Matrix Method for the Free Vibration of Thin Rectangular Plates”, Shock & Vibration Journal, vol. 2018, Article ID 1071830, 8 pages, 2018. <https://doi.org/10.1155/2018/1071830/>.
 5. M.T.T Kashani, **S. Jayasinghe**, and S.M. Hashemi " Dynamic Finite Element Analysis of Bending-Torsion Coupled Beams Subjected to Axial Load and End Moment,” Shock & Vibration, vol. 2015, Article ID 471270, 12 pages, 2015. doi:[10.1155/2015/471270](https://doi.org/10.1155/2015/471270). [https://www.hindawi.com/journals/sv/2015/471270/\(Masters’ research but related to DFE\)](https://www.hindawi.com/journals/sv/2015/471270/(Masters%20research%20but%20related%20to%20DFE)).
 6. M.T.T. Kashani, **S. Jayasinghe**, and S.M. Hashemi "On the Flexural–Torsional Vibration and Stability of Beams Subjected to Axial Load and End Moment," Journal of Shock & Vibration, Vol. 2014 (2014), Article ID 153532, 11 pages. <http://dx.doi.org/10.1155/2014/153532> (Masters’ research but related to DFE).
-

Conference Papers

1. **Supun Jayasinghe** and Seyed M. Hashemi, “A Symbolic Dynamic Finite Element Formulation for Multilayered Thin Rectangular Plates”, Presented at the 10th International Conference on Engineering Computational Technology (ECT 2018), September 4-6, 2018, Sitges, Barcelona, Spain.
2. **Supun Jayasinghe** and Seyed M. Hashemi, “Dynamic Coefficient Matrix Method for Multilayered Thin Plate Vibration – A Symbolic Formulation”, Presented at the 24th International Congress on Sound and Vibration (ICSV 25), July 8-12, 2018, Hiroshima, Japan.
3. Seyed M. Hashemi and **Supun Jayasinghe**, “A Quasi-Exact Dynamic Finite Element Model for the Vibration of a Fully Clamped Thin Plate”, Presented at MechAero 2017, May 11, 2017, Tehran, Iran.
4. **Supun Jayasinghe** and Seyed M. Hashemi, “An Exact Dynamic Coefficient Matrix (DCM) Method for the Free Vibration Analysis of Thin Rectangular Plates”, Presented at the 6th International Conference on Acoustics & Vibration (ISAV-2016), Tehran, Iran, December 7-8, 2016 (*Poster Presentation*)
5. **Supun Jayasinghe** and Seyed M. Hashemi, “A Dynamic Exact Coefficient Method for the Free Vibration Analysis of Thin Rectangular Plates”, Presented at the 6th International Conference of Acoustics and Vibration, Tehran, Iran, December 7-8, 2016.
6. **Supun Jayasinghe** and Seyed M. Hashemi, “A Symbolic Dynamic Finite Element Formulation for Free Vibration Analysis of Thin Rectangular Plates”, Presented at the 6th International Conference of Acoustics and Vibration, Tehran, Iran, December 7-8, 2016.
7. **Supun Jayasinghe** and Seyed M. Hashemi, “Frequency Dependant Trigonometric Shape Functions for Thin Rectangular Plate Elements”, Presented at the 23rd International Congress on Sound and Vibration, ICSV 23, Athens, Greece, July 10-14, 2016.
8. **S. Jayasinghe**, M.T. Kashani, and S.M. Hashemi, “Coupled Flexural–Torsional Free Vibration Analysis of Pre-Loaded Beams – A Dynamic Finite Element Method”, Presented at the 3rd International Conference of Acoustics and Vibration, ISAV 2013, Tehran, Iran, December 25 – 26, 2013. (*Master’s research but related to DFE*)

9. M.T.T. Kashani, **S. Jayasinghe**, S.M. Hashemi, “Stability Analysis of Beams Subjected to Axial Load and End Moment- A Dynamic Finite Element”, Presented at the 3rd International Conference of Acoustics and Vibration, ISAV 2013, Tehran, Iran, December 25 – 26, 2013. (*Master’s research but related to DFE*)
 10. M.T.T. Kashani, **S. Jayasinghe** and S.M. Hashemi, “On the Flexural – Torsional Vibration and Stability of Layered Beams Subjected to Axial Load and End Moment”, Presented at the 20th International Congress of Sound and Vibration, ICSV 20, Bangkok, Thailand, July 7 – 11, 2013. (*Master’s research but related to DFE*)
 11. **S. Jayasinghe**, M.T.T. Kashani, S.M. Hashemi, “The Effects of Axial Load and End Moment on the Flexural – Torsional Vibration and Stability Characteristics of Beams”, Presented at the Canadian Aeronautics and Space Institute (CASI) AERO 13 Conference, Toronto, Canada, April 30 – May 2, 2013. (*Master’s research but related to DFE*)
-

APPENDICES

Appendix A: Trigonometric, Frequency Dependant DFE Shape Functions

Comparison of DFE and FEM Shape Functions

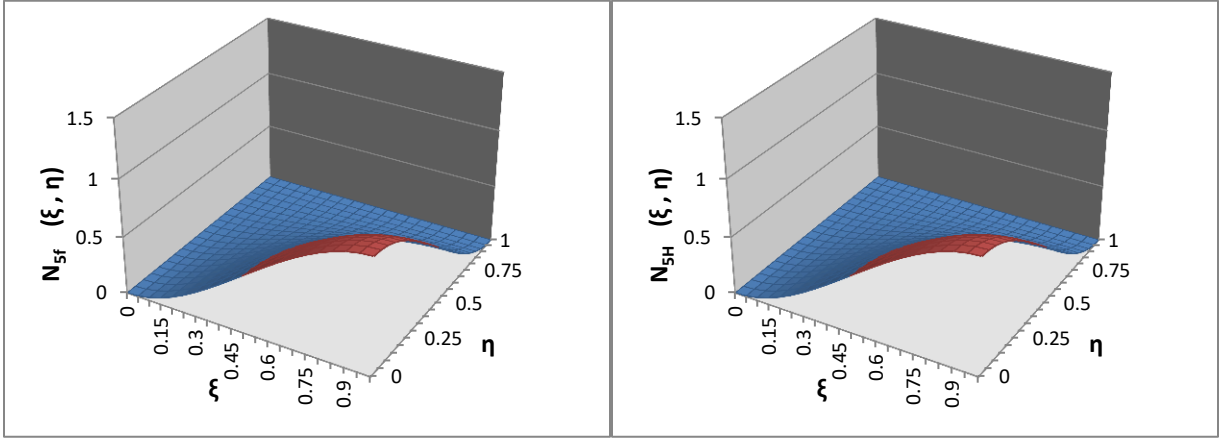


Figure 28: DFE (left) and FEM (right) shape function for w_2 at node 2

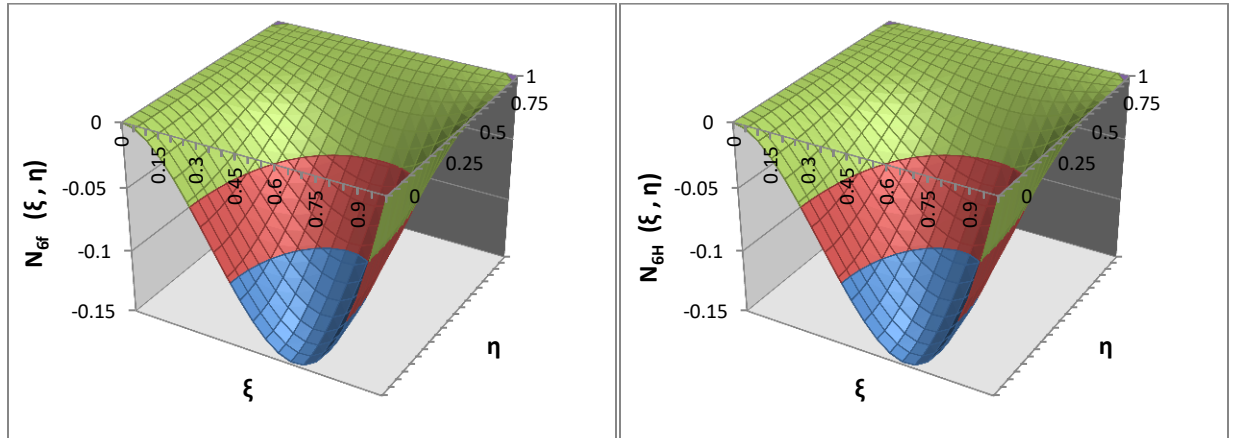


Figure 29: DFE (left) and FEM (right) shape function for θ_{x2} at node 2

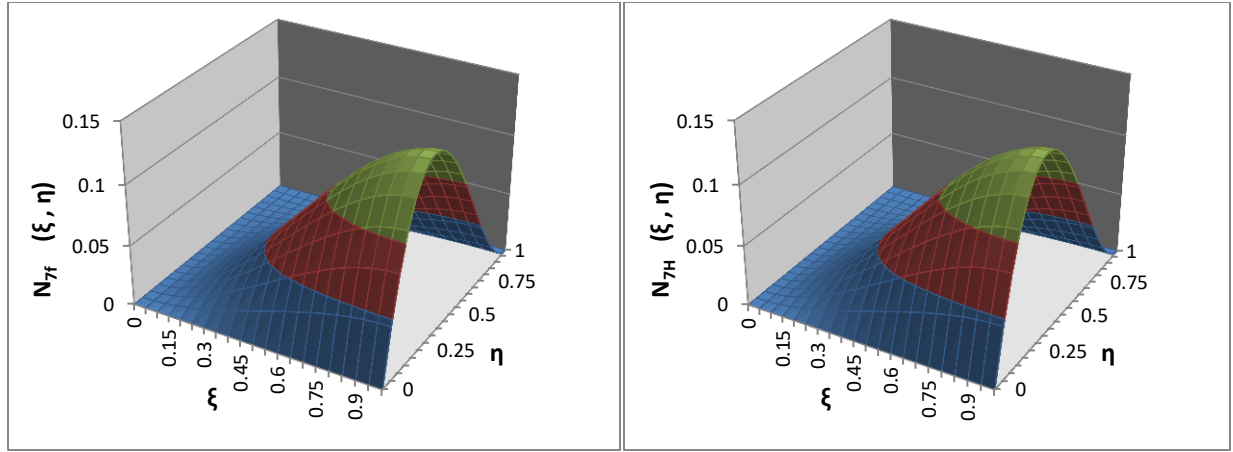


Figure 30: DFE (left) and FEM (right) shape function for θ_{y2} at node 2

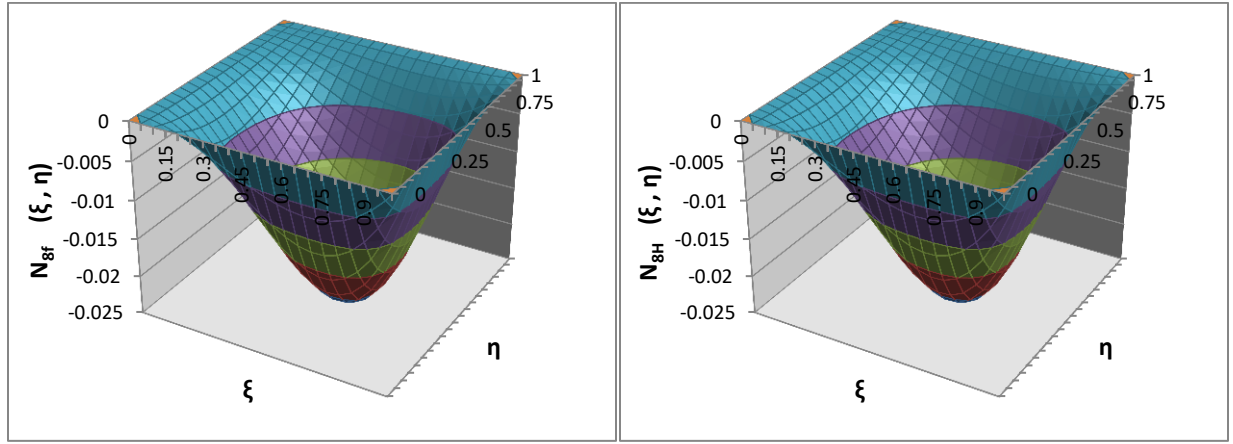


Figure 31: DFE (left) and FEM (right) shape function for θ_{xy2} at node 2

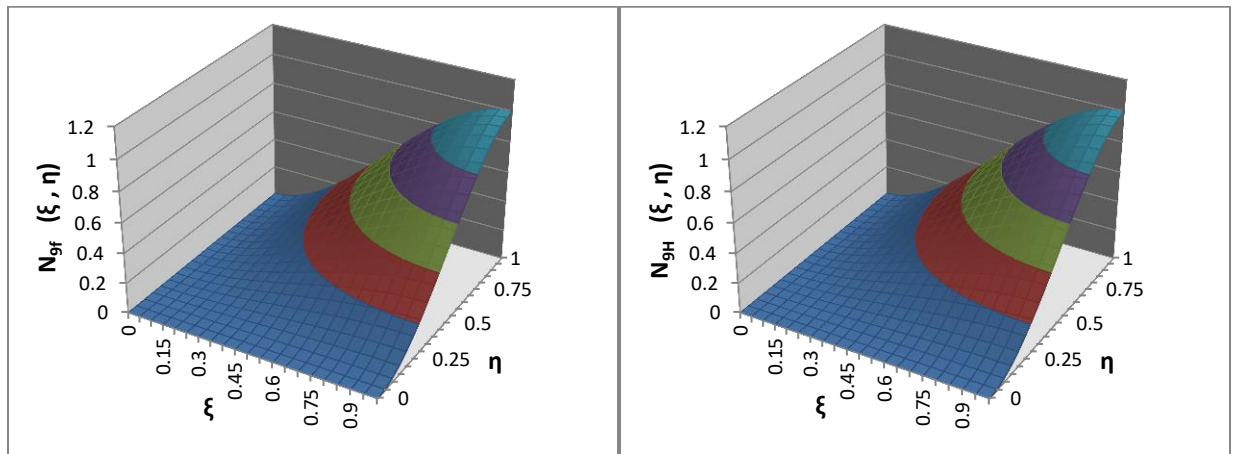


Figure 32: DFE (left) and FEM (right) shape function for w_3 at node 3

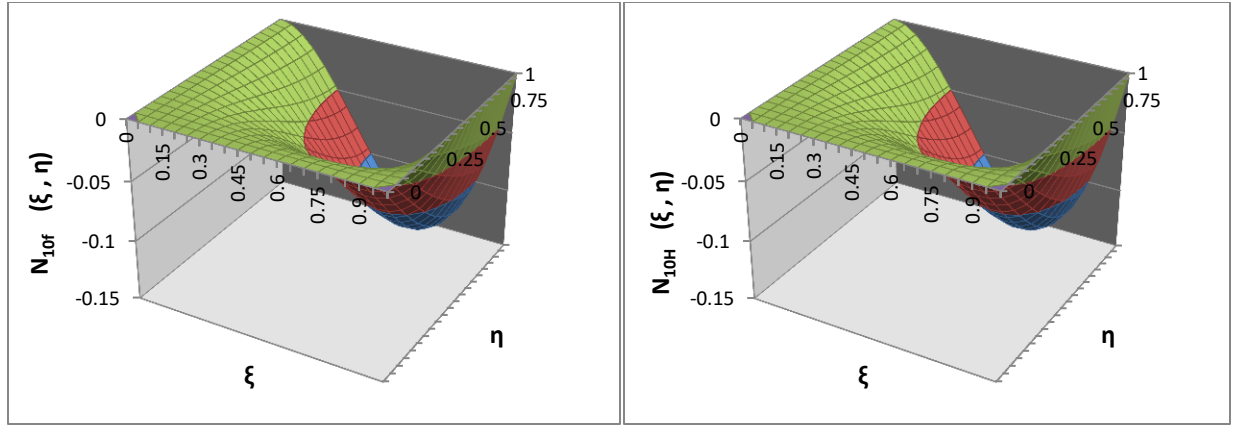


Figure 33: DFE (left) and FEM (right) shape function for θ_{x3} at node 3

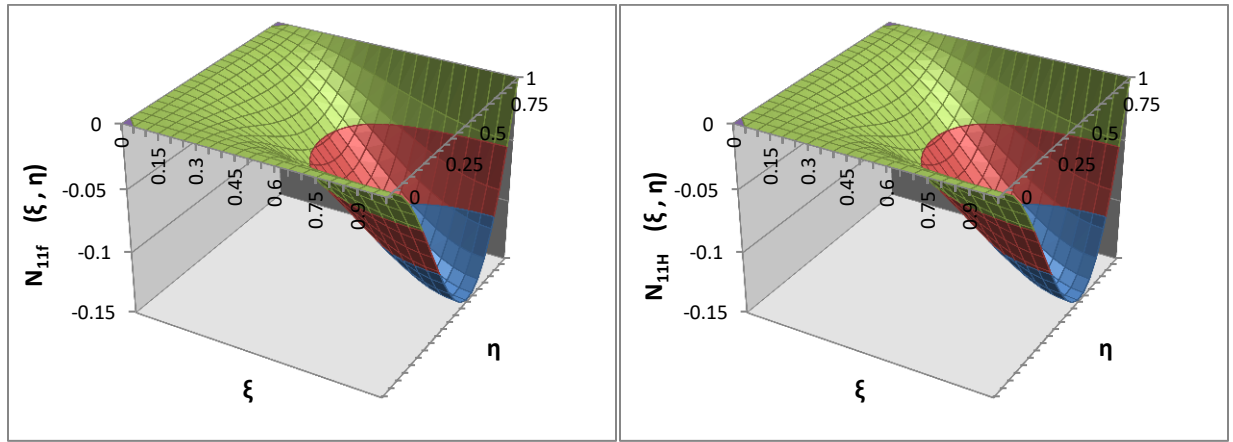


Figure 34: DFE (left) and FEM (right) shape function for θ_{y3} at node 3

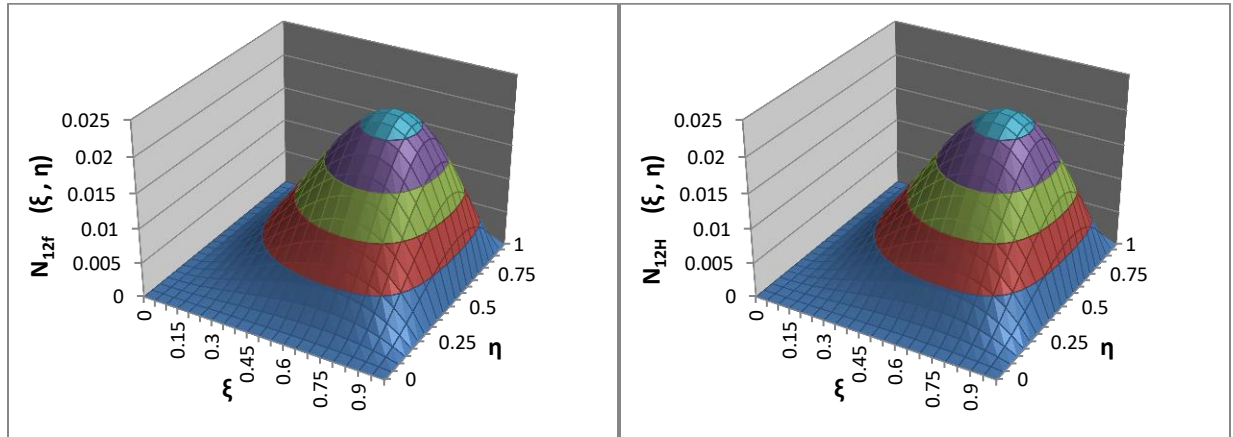


Figure 35: DFE (left) and FEM (right) shape function for θ_{xy3} at node 3

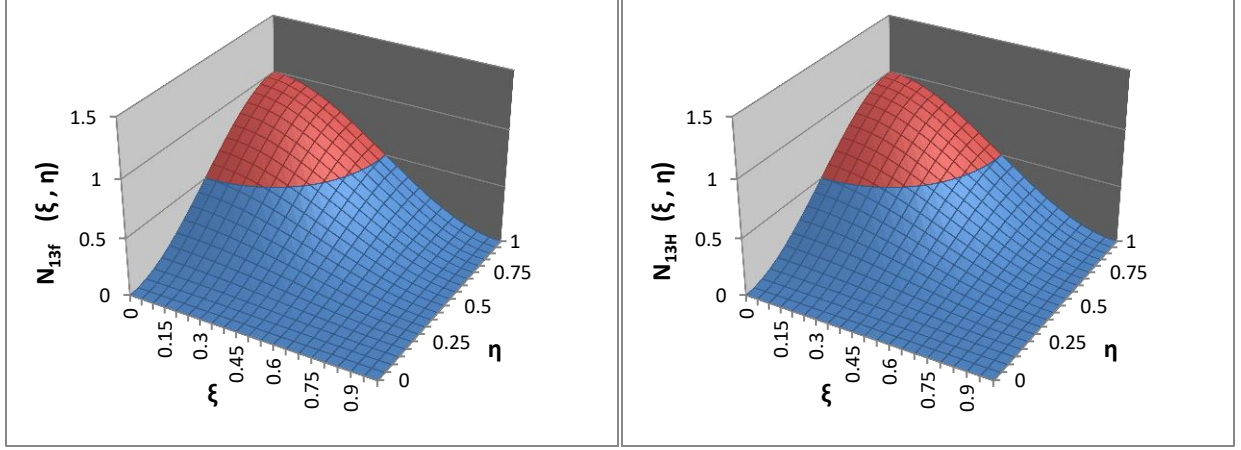


Figure 36: DFE (left) and FEM (right) shape function for w_4 at node 4

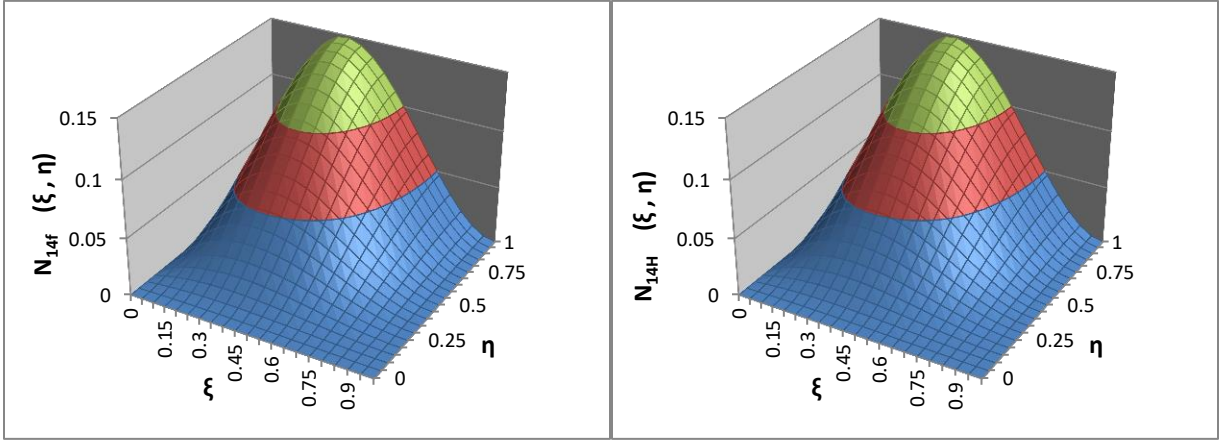


Figure 37: DFE (left) and FEM (right) shape function for θ_{x4} at node 4

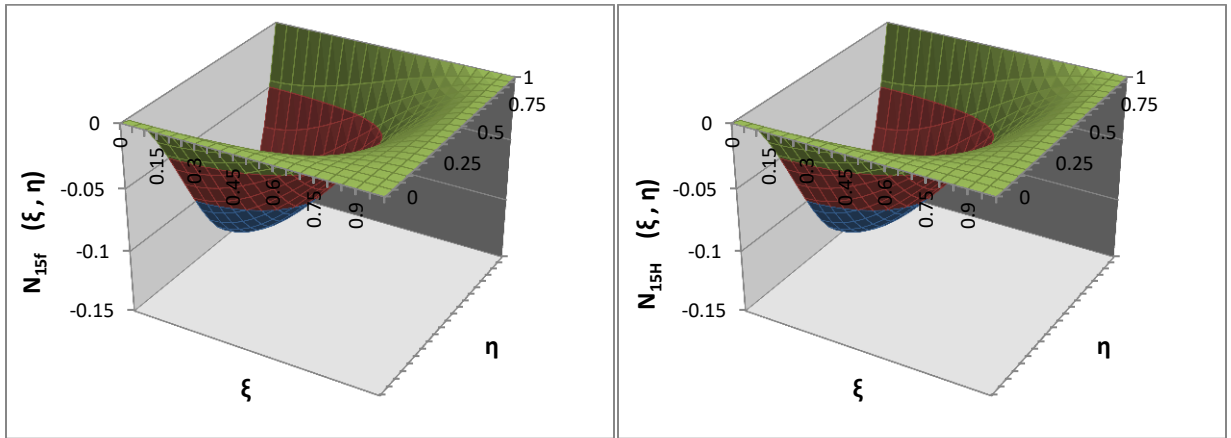


Figure 38: DFE (left) and FEM (right) shape function for θ_{y4} at node 4

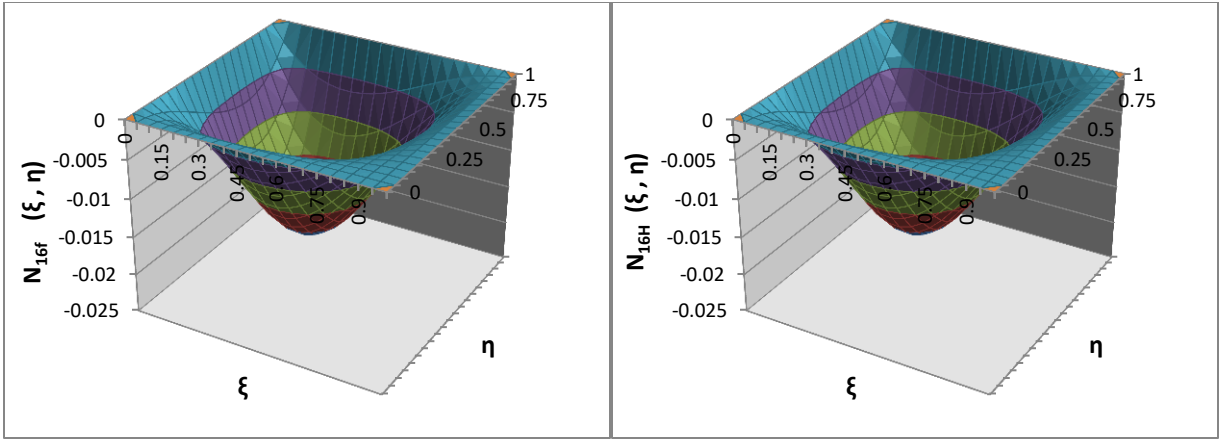


Figure 39: DFE (left) and FEM (right) shape function for θ_{xy4} at node 4

Frequency dependency of the DFE shape functions

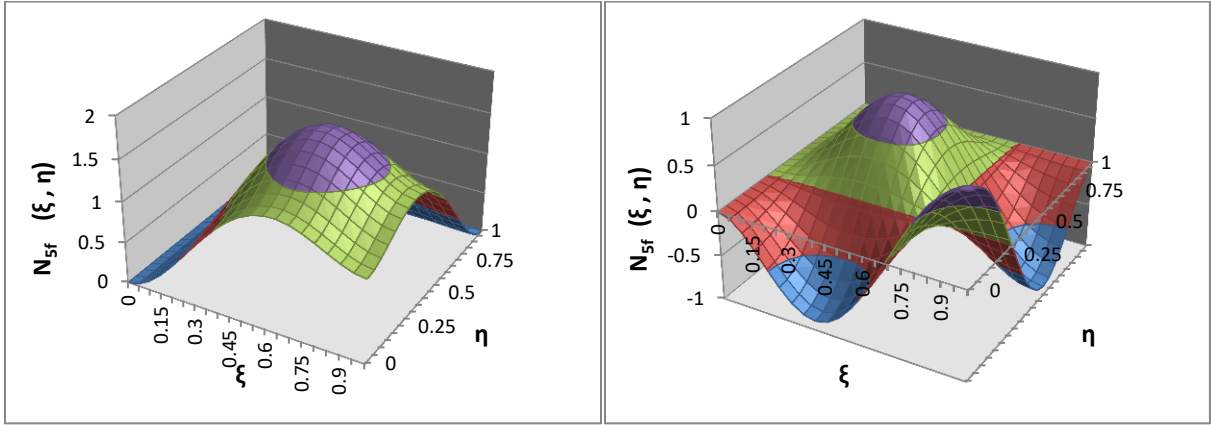


Figure 40: Shape function N_{5f} at ω_1 (left) and ω_5 (right)

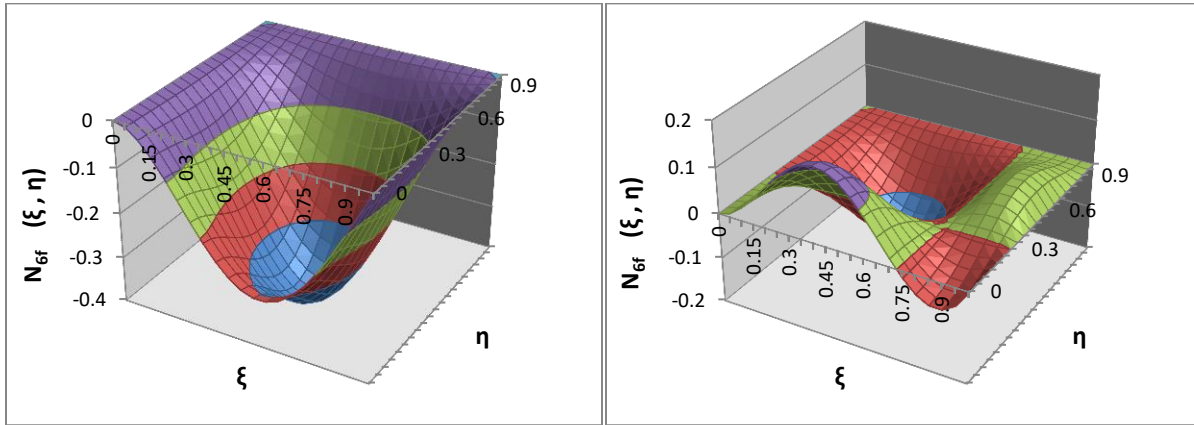


Figure 41: Shape function N_{6f} at ω_1 (left) and ω_5 (right)

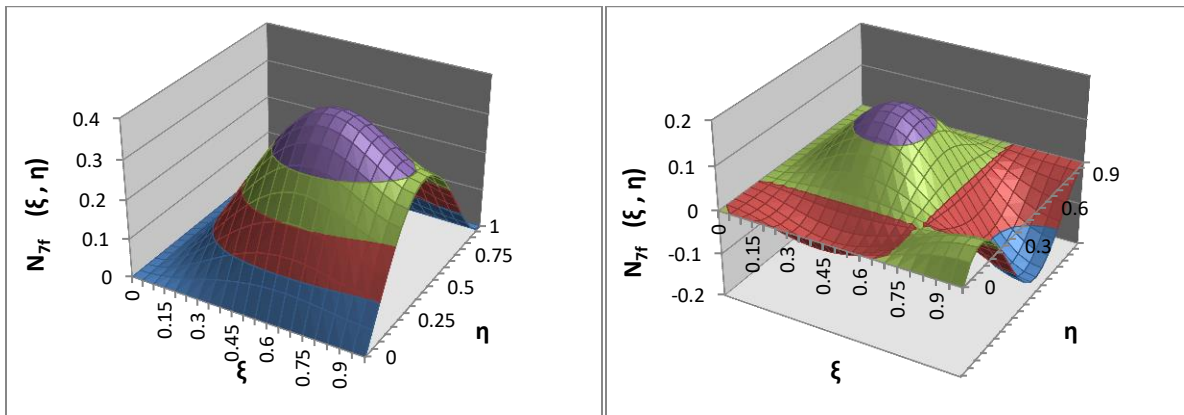


Figure 42: Shape function N_{7f} at ω_1 (left) and ω_5 (right)

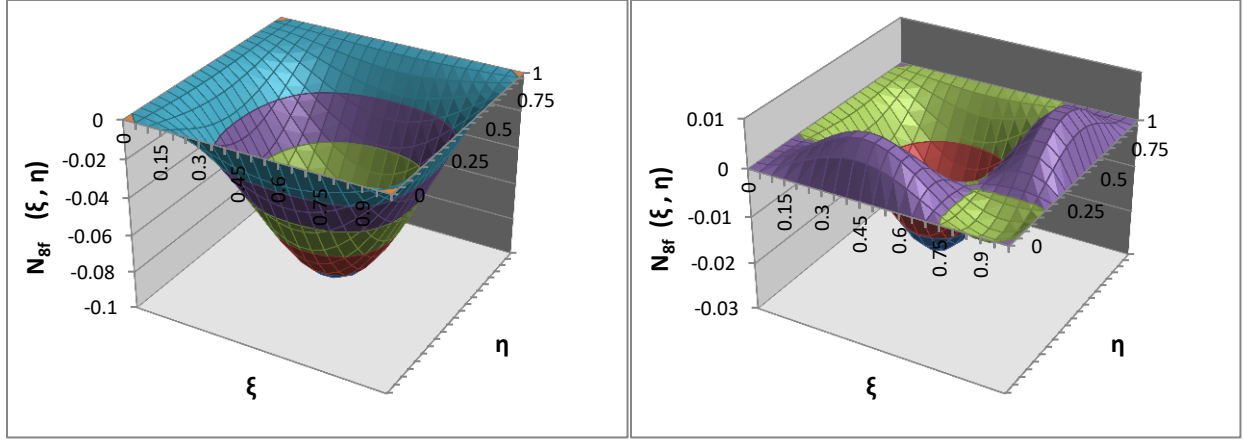


Figure 43: Shape function N_{8f} at ω_1 (left) and ω_5 (right)

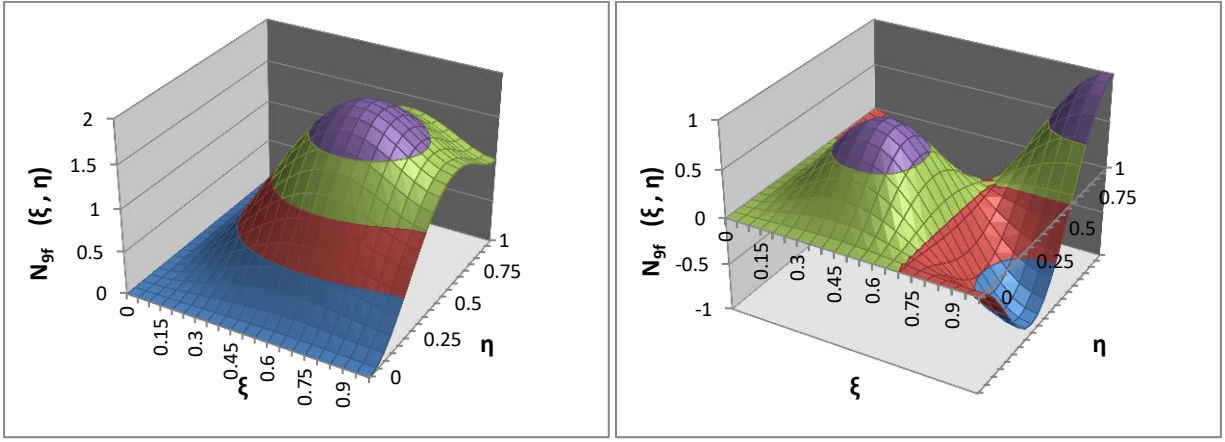


Figure 44: Shape function N_{9f} at ω_1 (left) and ω_5 (right)

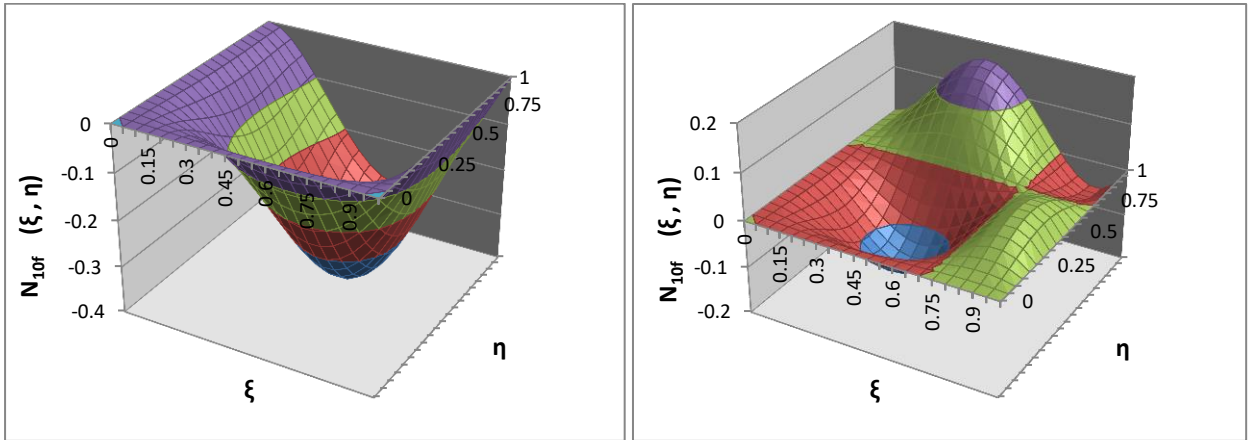


Figure 45: Shape function N_{10f} at ω_1 (left) and ω_5 (right)

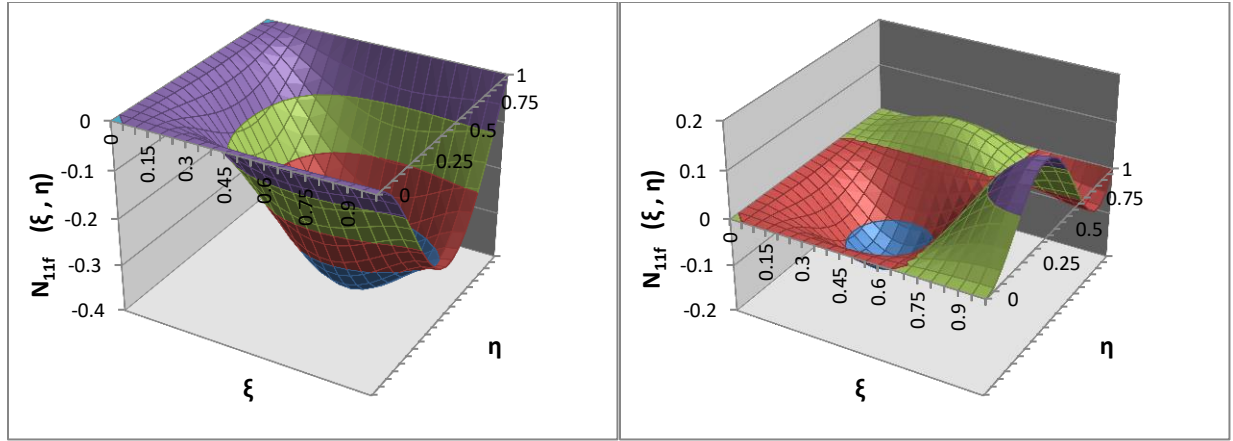


Figure 46: Shape function N_{11f} at ω_1 (left) and ω_5 (right)

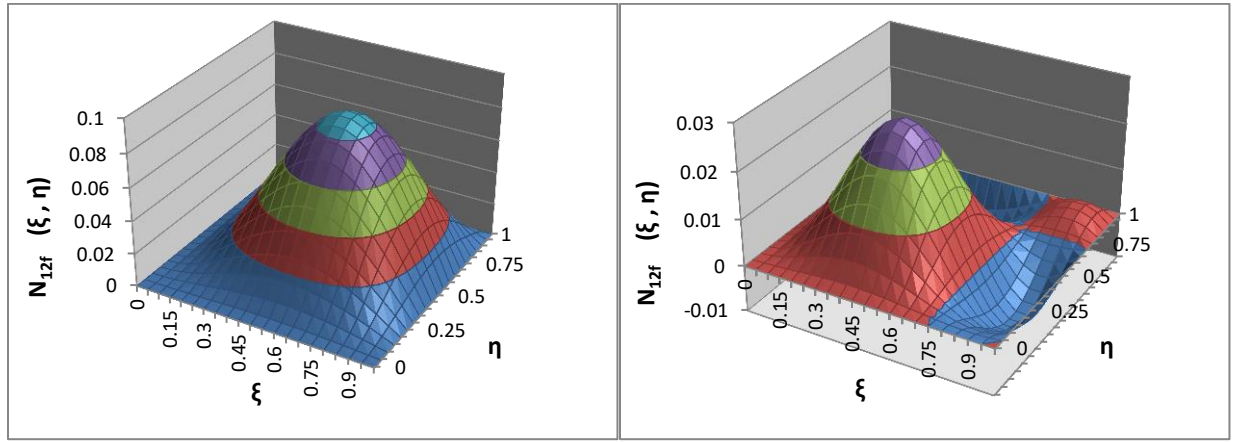


Figure 47: Shape function N_{12f} at ω_1 (left) and ω_5 (right)

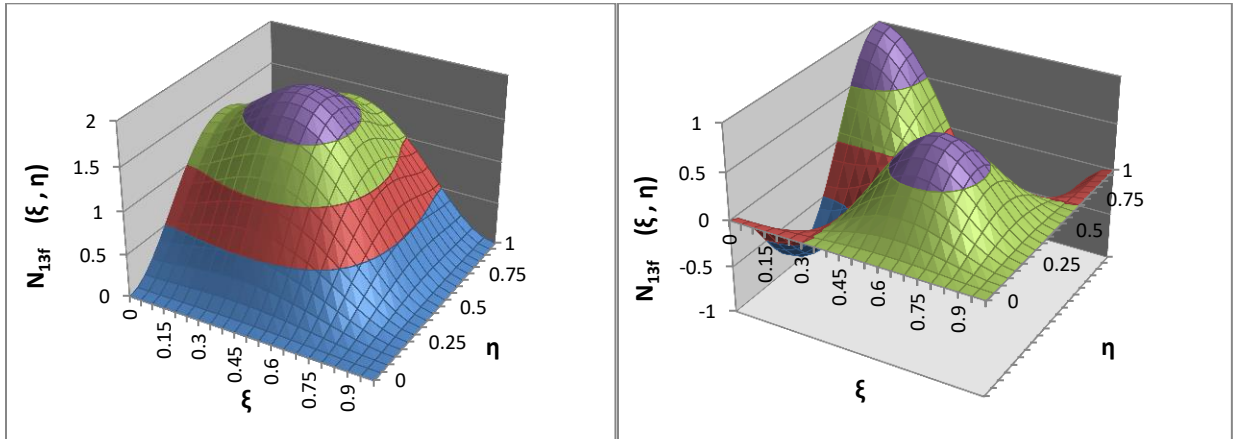


Figure 48: Shape function N_{13f} at ω_1 (left) and ω_5 (right)

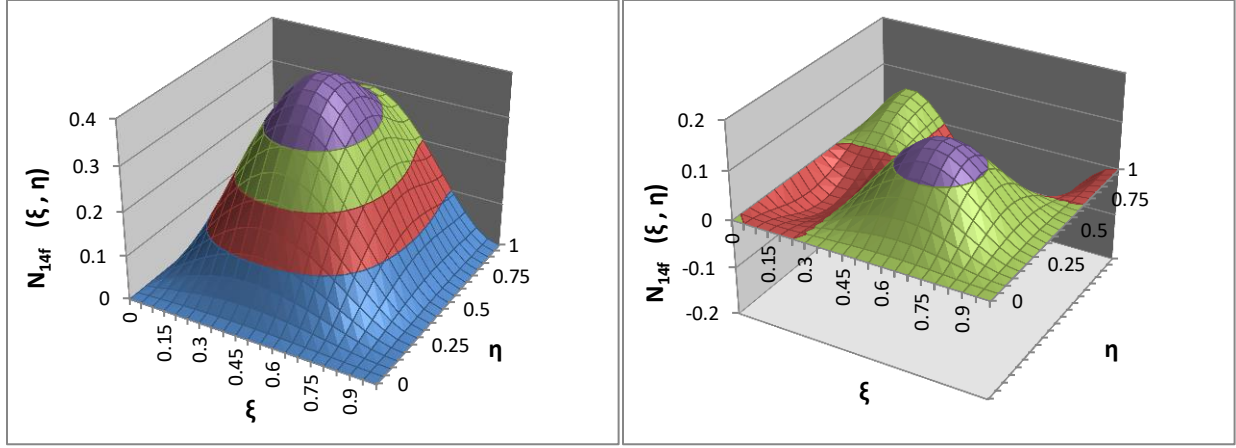


Figure 49: Shape function N_{14f} at ω_1 (left) and ω_5 (right)

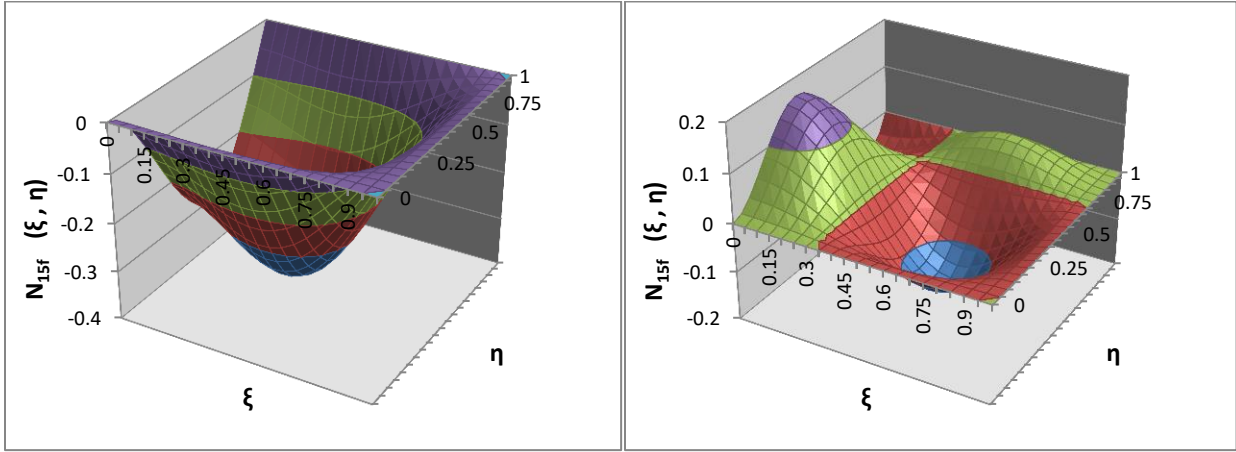


Figure 50: Shape function N_{15f} at ω_1 (left) and ω_5 (right)

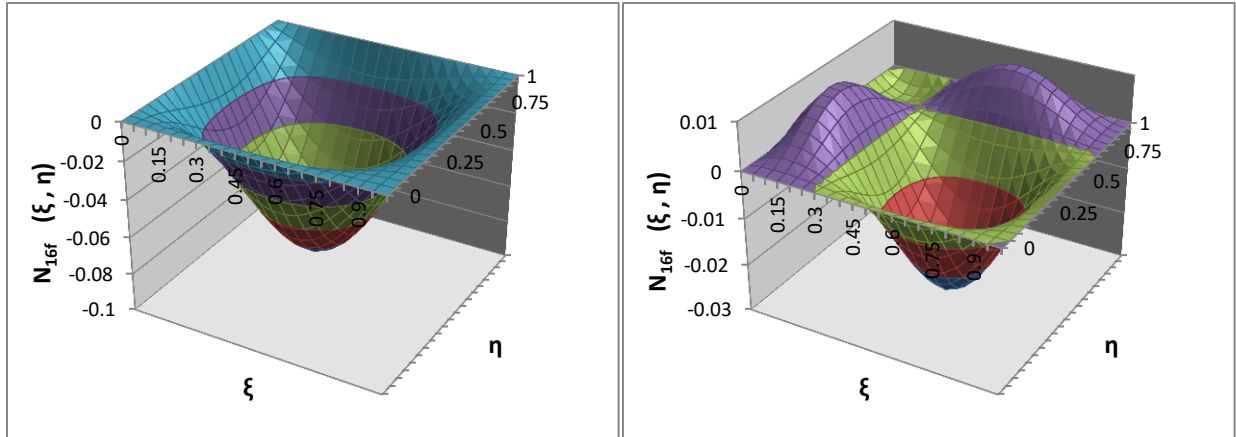


Figure 51: Shape function N_{16f} at ω_1 (left) and ω_5 (right)

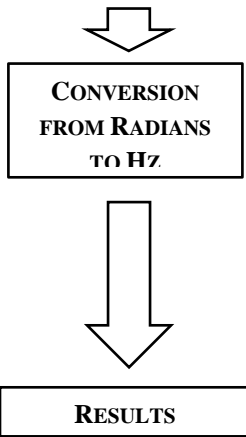
Appendix B: QDFE Solution for a Simply Supported (S-S-S-S) Plate

MATLAB[®] program flowchart and sample results for the QDFE solution

Table 32: MATLAB[®] program flow chart and sample results for S-S-S-S plate using QDFE

Flow Chart	Excerpts from MATLAB [®] Code	Equation
<div style="border: 1px solid black; padding: 5px; text-align: center; width: 100px; margin: 0 auto;">INPUT</div> <div style="text-align: center; margin: 10px 0;">↓</div>	<pre>syms ax ay bx by k e omega % Constants Rho = 7800; F = 200e9; h = 0.004; v = 0.3; D = (F*(h^3))/(12*(1-(v^2))); c1 = 0.6; c2 = 0.4; ax = abs(sqrt(-1)*(sqrt(c1*omega*sqrt(Rho*h/D)))); ay = abs(sqrt(-1)*(sqrt(c2*omega*sqrt(Rho*h/D)))); bx = (sqrt(c1*omega*sqrt(Rho*h/D))); by = (sqrt(c2*omega*sqrt(Rho*h/D))); % Element stiffness contribution from bending moment Mx x001 = zeros(16); x001 = dkt*d2k; x01 = subs (x001,k,1) - subs (x001,k,0); x1 = int (x01,e,0,1); g1 = (D*b)/(a^3); mx1 = g1*x1; x002 = zeros(16); x002 = dkt*d2e; x02 = subs (x002,k,1) - subs (x002,k,0); x2 = int (x02,e,0,1); g2 = (D*v)/(a*b); mx2 = g2*x2; mx = mx1+mx2;</pre>	Eqs. 31 to 34
<div style="border: 1px solid black; padding: 5px; text-align: center; width: 100px; margin: 0 auto;">ELEMENT STIFFNESS AND MASS MATRICES</div> <div style="text-align: center; margin: 10px 0;">↓</div>	<pre>% Global stiffness and mass matrix for single element model Kg = zeros(16); Mg = zeros(16); Kg = ke; Mg = me;</pre>	Eqs. 113 to 132
<div style="border: 1px solid black; padding: 5px; text-align: center; width: 100px; margin: 0 auto;">GLOBAL STIFFNESS AND MASS MATRICES</div> <div style="text-align: center; margin: 10px 0;">↓</div>	<pre>% Application of S-S-S-S boundary conditions for i = 1:3:10; Kg(:,i) = []; Kg(i,:) = []; Mg(:,i) = []; Mg(i,:) = []; end K = Kg; M = Mg;</pre>	Eq. 78
<div style="border: 1px solid black; padding: 5px; text-align: center; width: 100px; margin: 0 auto;">BOUNDARY CONDITIONS</div> <div style="text-align: center; margin: 10px 0;">↓</div>	<pre>% Eigenvalue problem DTM = det (K-(omega^2)*M);</pre>	N/A
<div style="border: 1px solid black; padding: 5px; text-align: center; width: 100px; margin: 0 auto;">EIGENVALUE PROBLEM</div> <div style="text-align: center; margin: 10px 0;">↓</div>	<pre>% Determination of roots EQN = @(omega) DTM; d1 = bisection(EQN,471,785); d2 = bisection(EQN,785,1099); d3 = bisection(EQN,1099,1727); d4 = bisection(EQN,1727,2042); d5 = bisection(EQN,2042,2356);</pre>	Eq. 133
<div style="border: 1px solid black; padding: 5px; text-align: center; width: 100px; margin: 0 auto;">DETERMINATION OF ROOTS</div> <div style="text-align: center; margin: 10px 0;">↓</div>		N/A

Table 32 continued: MATLAB[®] program flow chart and sample results for S-S-S-S plate using QDFE

Flow Chart	Excerpts from MATLAB [®] Code	Equation
 <p>CONVERSION FROM RADIANs TO Hz</p> <p>RESULTS</p>	<pre>% Converting from Radians to Hz freq_rad = zeros(5,1); freq_rad(1,1) = d1; freq_rad(2,1) = d2; freq_rad(3,1) = d3; freq_rad(4,1) = d4; freq_rad(5,1) = d5; freq_hz = 1/(2*pi()) * freq_rad freq = floor(freq_hz * 100)/100</pre>	N/A
	<pre>Command Window >> Thinplatevibration freq = 86.6800 166.6300 266.9900 300.1400 346.4900 fx >></pre>	N/A

MATLAB® code for a simply supported, thin, homogeneous, rectangular plate analysed using a single element QDFE model

```
% Free vibration of a thin, homogeneous, rectangular plate under S-S-S-S
boundary condition
% Toronto, Ontario, Canada, 2019
% Copyright Heenkenda Jayasinghe 2019

syms ax ay bx by ke omega

Rho = 7800;
F = 200e9;
h = 0.004;
v = 0.3;
D = (F*(h^3))/(12*(1-(v^2)));
c1 = 0.6;
c2 = 0.4;
ax = abs(sqrt(-1)*(sqrt(c1*omega*sqrt(Rho*h/D))));
ay = abs(sqrt(-1)*(sqrt(c2*omega*sqrt(Rho*h/D))));
bx = (sqrt(c1*omega*sqrt(Rho*h/D)));
by = (sqrt(c2*omega*sqrt(Rho*h/D)));

% Shape Functions: Node 1

N1 = ((cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e - 1)) - cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e - 1)))/by - (by*sinh(by)*sin(ay*(e - 1)))/ay)*(cos(ax*k) + cosh(bx*k) - cos(ax)*cosh(bx*(k - 1)) - cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k - 1)))/bx - (bx*sinh(bx)*sin(ax*(k - 1)))/ax))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N2 = -((bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k - 1))) + ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k - 1))))*(cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e - 1)) - cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e - 1)))/by - (by*sinh(by)*sin(ay*(e - 1)))/ay))/((ax*bx*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N3 = -((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1))) + ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e - 1))))*(cos(ax*k) + cosh(bx*k) - cos(ax)*cosh(bx*(k - 1)) - cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k - 1)))/bx - (bx*sinh(bx)*sin(ax*(k - 1)))/ax))/((ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N4 = ((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1))) + ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e - 1))))*(bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k - 1))) + ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k - 1)))))/((ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
```

```
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));
```

```
% Shape Functions: Node 2
```

```
N5 = ((cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e - 1)) - cosh(by)*cos(ay*(e
- 1)) + (ay*sin(ay)*sinh(by*(e - 1)))/by - (by*sinh(by)*sin(ay*(e -
1)))/ay)*(cos(ax*(k - 1)) + cosh(bx*(k - 1)) - cos(ax*k)*cosh(bx) -
cosh(bx*k)*cos(ax) - (ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));
```

```
N6 = ((bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) +
ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) - sinh(bx*k)*cos(ax)))*(cos(ay*e) +
cosh(by*e) - cos(ay)*cosh(by*(e - 1)) - cosh(by)*cos(ay*(e - 1)) +
(ay*sin(ay)*sinh(by*(e - 1)))/by - (by*sinh(by)*sin(ay*(e -
1)))/ay))/(ax*bx*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));
```

```
N7 = -((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1)))
+ ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(cos(ax*(k - 1)) + cosh(bx*(k - 1)) - cos(ax*k)*cosh(bx) -
cosh(bx*k)*cos(ax) - (ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N8 = -((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1)))
+ ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) +
ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) -
sinh(bx*k)*cos(ax)))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
% Shape Functions: Node 3
```

```
N9 = ((cos(ay*(e - 1)) + cosh(by*(e - 1)) - cos(ay*e)*cosh(by) -
cosh(by*e)*cos(ay) - (ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay)*(cos(ax*(k - 1)) + cosh(bx*(k - 1)) -
cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) - (ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));
```

```
N10 = ((bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) +
ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) - sinh(bx*k)*cos(ax)))*(cos(ay*(e -
1)) + cosh(by*(e - 1)) - cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) -
(ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```

N11 = ((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) - sinh(by*e)*cos(ay)))*(cos(ax*(k -
1)) + cosh(bx*(k - 1)) - cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) -
(ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N12 = ((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) -
sin(ax*k)*cosh(bx)) + ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) -
sinh(bx*k)*cos(ax)))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

% Shape Functions: Node 4

```

N13 = ((cos(ax*k) + cosh(bx*k) - cos(ax)*cosh(bx*(k - 1)) -
cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k - 1)))/bx -
(bx*sinh(bx)*sin(ax*(k - 1)))/ax)*(cos(ay*(e - 1)) + cosh(by*(e - 1)) -
cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) - (ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

```

```

N14 = -((bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k - 1)))
+ ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))))*(cos(ay*(e - 1)) + cosh(by*(e - 1)) - cos(ay*e)*cosh(by) -
cosh(by*e)*cos(ay) - (ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N15 = ((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) - sinh(by*e)*cos(ay)))*(cos(ax*k) +
cosh(bx*k) - cos(ax)*cosh(bx*(k - 1)) - cosh(bx)*cos(ax*(k - 1)) +
(ax*sin(ax)*sinh(bx*(k - 1)))/bx - (bx*sinh(bx)*sin(ax*(k -
1)))/ax))/(ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

```

```

N16 = -((bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k - 1)))
+ ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))))*(by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

% Shape Function Derivatives

% Node 1

```
N1k = -((cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e - 1)) -
cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e - 1)))/by -
(by*sinh(by)*sin(ay*(e - 1)))/ay)*(ax*sin(ax*k) - bx*sinh(bx*k) -
ax*cosh(bx)*sin(ax*(k - 1)) - ax*sin(ax)*cosh(bx*(k - 1)) +
bx*cos(ax)*sinh(bx*(k - 1)) + bx*sinh(bx)*cos(ax*(k -
1))))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N1kk = ((cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e - 1)) -
cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e - 1)))/by -
(by*sinh(by)*sin(ay*(e - 1)))/ay)*(bx^2*cosh(bx*k) - ax^2*cos(ax*k) +
ax^2*cosh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*cosh(bx*(k - 1)) +
ax*bx*sin(ax)*sinh(bx*(k - 1)) + ax*bx*sinh(bx)*sin(ax*(k -
1))))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N1e = -((ay*sin(ay*e) - by*sinh(by*e) - ay*cosh(by)*sin(ay*(e - 1)) -
ay*sin(ay)*cosh(by*(e - 1)) + by*cos(ay)*sinh(by*(e - 1)) +
by*sinh(by)*cos(ay*(e - 1)))*(cos(ax*k) + cosh(bx*k) - cos(ax)*cosh(bx*(k -
1)) - cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k - 1)))/bx -
(bx*sinh(bx)*sin(ax*(k - 1)))/ax))/(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N1ee = ((by^2*cosh(by*e) - ay^2*cos(ay*e) + ay^2*cosh(by)*cos(ay*(e - 1)) -
by^2*cos(ay)*cosh(by*(e - 1)) + ay*by*sin(ay)*sinh(by*(e - 1)) +
ay*by*sinh(by)*sin(ay*(e - 1)))*(cos(ax*k) + cosh(bx*k) - cos(ax)*cosh(bx*(k -
1)) - cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k - 1)))/bx -
(bx*sinh(bx)*sin(ax*(k - 1)))/ax))/(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N1ke = ((ay*sin(ay*e) - by*sinh(by*e) - ay*cosh(by)*sin(ay*(e - 1)) -
ay*sin(ay)*cosh(by*(e - 1)) + by*cos(ay)*sinh(by*(e - 1)) +
by*sinh(by)*cos(ay*(e - 1)))*(ax*sin(ax*k) - bx*sinh(bx*k) -
ax*cosh(bx)*sin(ax*(k - 1)) - ax*sin(ax)*cosh(bx*(k - 1)) +
bx*cos(ax)*sinh(bx*(k - 1)) + bx*sinh(bx)*cos(ax*(k -
1))))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N1k2e = -((ay*sin(ay*e) - by*sinh(by*e) - ay*cosh(by)*sin(ay*(e - 1)) -
ay*sin(ay)*cosh(by*(e - 1)) + by*cos(ay)*sinh(by*(e - 1)) +
by*sinh(by)*cos(ay*(e - 1)))*(bx^2*cosh(bx*k) - ax^2*cos(ax*k) +
ax^2*cosh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*cosh(bx*(k - 1)) +
ax*bx*sin(ax)*sinh(bx*(k - 1)) + ax*bx*sinh(bx)*sin(ax*(k -
1))))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N1ke2 = -((by^2*cosh(by*e) - ay^2*cos(ay*e) + ay^2*cosh(by)*cos(ay*(e - 1)) -
by^2*cos(ay)*cosh(by*(e - 1)) + ay*by*sin(ay)*sinh(by*(e - 1)) +
ay*by*sinh(by)*sin(ay*(e - 1)))*(ax*sin(ax*k) - bx*sinh(bx*k) -
```

```

ax*cosh(bx)*sin(ax*(k - 1)) - ax*sin(ax)*cosh(bx*(k - 1)) +
bx*cos(ax)*sinh(bx*(k - 1)) + bx*sinh(bx)*cos(ax*(k -
1)))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N1k2e2 = ((by^2*cosh(by*e) - ay^2*cos(ay*e) + ay^2*cosh(by)*cos(ay*(e - 1)) -
by^2*cos(ay)*cosh(by*(e - 1)) + ay*by*sin(ay)*sinh(by*(e - 1)) +
ay*by*sinh(by)*sin(ay*(e - 1)))*(bx^2*cosh(bx*k) - ax^2*cos(ax*k) +
ax^2*cosh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*cosh(bx*(k - 1)) +
ax*bx*sin(ax)*sinh(bx*(k - 1)) + ax*bx*sinh(bx)*sin(ax*(k -
1)))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N1k3 = ((cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e - 1)) -
cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e - 1)))/by -
(by*sinh(by)*sin(ay*(e - 1)))/ay)*(ax^3*sin(ax*k) + bx^3*sinh(bx*k) -
ax^3*cosh(bx)*sin(ax*(k - 1)) - bx^3*cos(ax)*sinh(bx*(k - 1)) +
ax*bx^2*sin(ax)*cosh(bx*(k - 1)) + ax^2*bx*sinh(bx)*cos(ax*(k -
1)))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N1e3 = ((ay^3*sin(ay*e) + by^3*sinh(by*e) - ay^3*cosh(by)*sin(ay*(e - 1)) -
by^3*cos(ay)*sinh(by*(e - 1)) + ay*by^2*sin(ay)*cosh(by*(e - 1)) +
ay^2*by*sinh(by)*cos(ay*(e - 1)))*(cos(ax*k) + cosh(bx*k) -
cos(ax)*cosh(bx*(k - 1)) - cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k
- 1)))/bx - (bx*sinh(bx)*sin(ax*(k - 1)))/ax)/((2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N1k4 = ((ax^4*cos(ax*k) + bx^4*cosh(bx*k) - ax^4*cosh(bx)*cos(ax*(k - 1)) -
bx^4*cos(ax)*cosh(bx*(k - 1)) + ax*bx^3*sin(ax)*sinh(bx*(k - 1)) -
ax^3*bx*sinh(bx)*sin(ax*(k - 1)))*(cos(ay*e) + cosh(by*e) -
cos(ay)*cosh(by*(e - 1)) - cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e
- 1)))/by - (by*sinh(by)*sin(ay*(e - 1)))/ay)/((2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N1e4 = ((cos(ax*k) + cosh(bx*k) - cos(ax)*cosh(bx*(k - 1)) -
cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k - 1)))/bx -
(bx*sinh(bx)*sin(ax*(k - 1)))/ax)*(ay^4*cos(ay*e) + by^4*cosh(by*e) -
ay^4*cosh(by)*cos(ay*(e - 1)) - by^4*cos(ay)*cosh(by*(e - 1)) +
ay*by^3*sin(ay)*sinh(by*(e - 1)) - ay^3*by*sinh(by)*sin(ay*(e -
1)))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

% Node 2

```

N2k = -((bx*(ax*cosh(bx)*cos(ax*(k - 1)) - ax*cos(ax*k) +
bx*sin(ax)*sinh(bx*(k - 1))) - ax*(bx*cosh(bx*k) - bx*cos(ax)*cosh(bx*(k -
1)) + ax*sinh(bx)*sin(ax*(k - 1)))*(cos(ay*e) + cosh(by*e) -
cos(ay)*cosh(by*(e - 1)) - cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e
- 1)))/by - (by*sinh(by)*sin(ay*(e - 1)))/ay)/((ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```


$$N2kk = -((bx*(ax^2*\sin(ax*k) - ax^2*\cosh(bx)*\sin(ax*(k - 1)) + bx^2*\sin(ax)*\cosh(bx*(k - 1))) - ax*(bx^2*\sinh(bx*k) + ax^2*\sinh(bx)*\cos(ax*(k - 1)) - bx^2*\cos(ax)*\sinh(bx*(k - 1))))*(\cos(ay*e) + \cosh(by*e) - \cos(ay)*\cosh(by*(e - 1)) - \cosh(by)*\cos(ay*(e - 1)) + (ay*\sin(ay)*\sinh(by*(e - 1)))/by - (by*\sinh(by)*\sin(ay*(e - 1)))/ay))/ (ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/ (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/ (ay*by) - 2));$$

$$N2e = ((bx*(\cosh(bx)*\sin(ax*(k - 1)) - \sin(ax*k) + \sin(ax)*\cosh(bx*(k - 1))) + ax*(\cos(ax)*\sinh(bx*(k - 1)) - \sinh(bx*k) + \sinh(bx)*\cos(ax*(k - 1))))*(ay*\sin(ay*e) - by*\sinh(by*e) - ay*\cosh(by)*\sin(ay*(e - 1)) - ay*\sin(ay)*\cosh(by*(e - 1)) + by*\cos(ay)*\sinh(by*(e - 1)) + by*\sinh(by)*\cos(ay*(e - 1)))/ (ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/ (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/ (ay*by) - 2));$$

$$N2ee = -((bx*(\cosh(bx)*\sin(ax*(k - 1)) - \sin(ax*k) + \sin(ax)*\cosh(bx*(k - 1))) + ax*(\cos(ax)*\sinh(bx*(k - 1)) - \sinh(bx*k) + \sinh(bx)*\cos(ax*(k - 1))))*(by^2*\cosh(by*e) - ay^2*\cos(ay*e) + ay^2*\cosh(by)*\cos(ay*(e - 1)) - by^2*\cos(ay)*\cosh(by*(e - 1)) + ay*by*\sin(ay)*\sinh(by*(e - 1)) + ay*by*\sinh(by)*\sin(ay*(e - 1)))/ (ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/ (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/ (ay*by) - 2));$$

$$N2ke = ((bx*(ax*\cosh(bx)*\cos(ax*(k - 1)) - ax*\cos(ax*k) + bx*\sin(ax)*\sinh(bx*(k - 1))) - ax*(bx*\cosh(bx*k) - bx*\cos(ax)*\cosh(bx*(k - 1)) + ax*\sinh(bx)*\sin(ax*(k - 1))))*(ay*\sin(ay*e) - by*\sinh(by*e) - ay*\cosh(by)*\sin(ay*(e - 1)) - ay*\sin(ay)*\cosh(by*(e - 1)) + by*\cos(ay)*\sinh(by*(e - 1)) + by*\sinh(by)*\cos(ay*(e - 1)))/ (ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/ (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/ (ay*by) - 2));$$

$$N2k2e = ((bx*(ax^2*\sin(ax*k) - ax^2*\cosh(bx)*\sin(ax*(k - 1)) + bx^2*\sin(ax)*\cosh(bx*(k - 1))) - ax*(bx^2*\sinh(bx*k) + ax^2*\sinh(bx)*\cos(ax*(k - 1)) - bx^2*\cos(ax)*\sinh(bx*(k - 1))))*(ay*\sin(ay*e) - by*\sinh(by*e) - ay*\cosh(by)*\sin(ay*(e - 1)) - ay*\sin(ay)*\cosh(by*(e - 1)) + by*\cos(ay)*\sinh(by*(e - 1)) + by*\sinh(by)*\cos(ay*(e - 1)))/ (ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/ (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/ (ay*by) - 2));$$

$$N2ke2 = -((bx*(ax*\cosh(bx)*\cos(ax*(k - 1)) - ax*\cos(ax*k) + bx*\sin(ax)*\sinh(bx*(k - 1))) - ax*(bx*\cosh(bx*k) - bx*\cos(ax)*\cosh(bx*(k - 1)) + ax*\sinh(bx)*\sin(ax*(k - 1))))*(by^2*\cosh(by*e) - ay^2*\cos(ay*e) + ay^2*\cosh(by)*\cos(ay*(e - 1)) - by^2*\cos(ay)*\cosh(by*(e - 1)) + ay*by*\sin(ay)*\sinh(by*(e - 1)) + ay*by*\sinh(by)*\sin(ay*(e - 1)))/ (ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/ (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/ (ay*by) - 2));$$

$$N2k2e2 = -((bx*(ax^2*\sin(ax*k) - ax^2*\cosh(bx)*\sin(ax*(k - 1)) + bx^2*\sin(ax)*\cosh(bx*(k - 1))) - ax*(bx^2*\sinh(bx*k) + ax^2*\sinh(bx)*\cos(ax*(k - 1)) - bx^2*\cos(ax)*\sinh(bx*(k - 1))))*(by^2*\cosh(by*e) - ay^2*\cos(ay*e) + ay^2*\cosh(by)*\cos(ay*(e - 1)) - by^2*\cos(ay)*\cosh(by*(e - 1)) + ay*by*\sin(ay)*\sinh(by*(e - 1)) + ay*by*\sinh(by)*\sin(ay*(e - 1)))/ (ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/ (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/ (ay*by) - 2));$$

```
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by - 2));
```

```
N2k3 = -( (bx*(ax^3*cos(ax*k) - ax^3*cosh(bx)*cos(ax*(k - 1)) +
bx^3*sin(ax)*sinh(bx*(k - 1))) + ax*(bx^3*cos(ax)*cosh(bx*(k - 1)) -
bx^3*cosh(bx*k) + ax^3*sinh(bx)*sin(ax*(k - 1))) )*(cos(ay*e) + cosh(by*e) -
cos(ay)*cosh(by*(e - 1)) - cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e
- 1)))/by - (by*sinh(by)*sin(ay*(e - 1)))/ay))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by - 2)));
```

```
N2e3 = -( (bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k -
1))) + ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))) )*(ay^3*sin(ay*e) + by^3*sinh(by*e) - ay^3*cosh(by)*sin(ay*(e - 1)) -
by^3*cos(ay)*sinh(by*(e - 1)) + ay*by^2*sin(ay)*cosh(by*(e - 1)) +
ay^2*by*sinh(by)*cos(ay*(e - 1)))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by - 2)));
```

```
N2k4 = -( (bx*(ax^4*cosh(bx)*sin(ax*(k - 1)) - ax^4*sin(ax*k) +
bx^4*sin(ax)*cosh(bx*(k - 1))) + ax*(ax^4*sinh(bx)*cos(ax*(k - 1)) -
bx^4*sinh(bx*k) + bx^4*cos(ax)*sinh(bx*(k - 1))) )*(cos(ay*e) + cosh(by*e) -
cos(ay)*cosh(by*(e - 1)) - cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e
- 1)))/by - (by*sinh(by)*sin(ay*(e - 1)))/ay))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by - 2)));
```

```
N2e4 = -( (bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k -
1))) + ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))) )*(ay^4*cos(ay*e) + by^4*cosh(by*e) - ay^4*cosh(by)*cos(ay*(e - 1)) -
by^4*cos(ay)*cosh(by*(e - 1)) + ay*by^3*sin(ay)*sinh(by*(e - 1)) -
ay^3*by*sinh(by)*sin(ay*(e - 1)))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by - 2)));
```

```
% Node 3
```

```
N3k = ((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1)))
+ ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(ax*sin(ax*k) - bx*sinh(bx*k) - ax*cosh(bx)*sin(ax*(k - 1)) -
ax*sin(ax)*cosh(bx*(k - 1)) + bx*cos(ax)*sinh(bx*(k - 1)) +
bx*sinh(bx)*cos(ax*(k - 1)))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by - 2)));
```

```
N3kk = -( (by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e -
1))) + ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))) )*(bx^2*cosh(bx*k) - ax^2*cos(ax*k) + ax^2*cosh(bx)*cos(ax*(k - 1)) -
bx^2*cos(ax)*cosh(bx*(k - 1)) + ax*bx*sin(ax)*sinh(bx*(k - 1)) +
ax*bx*sinh(bx)*sin(ax*(k - 1)))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by - 2)));
```

```
N3e = -( (by*(ay*cosh(by)*cos(ay*(e - 1)) - ay*cos(ay*e) +
by*sin(ay)*sinh(by*(e - 1))) - ay*(by*cosh(by*e) - by*cos(ay)*cosh(by*(e -
```

```

1)) + ay*sinh(by)*sin(ay*(e - 1)))*(cos(ax*k) + cosh(bx*k) -
cos(ax)*cosh(bx*(k - 1)) - cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k
- 1)))/bx - (bx*sinh(bx)*sin(ax*(k - 1)))/ax))/ (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N3ee = -((by*(ay^2*sin(ay*e) - ay^2*cosh(by)*sin(ay*(e - 1)) +
by^2*sin(ay)*cosh(by*(e - 1))) - ay*(by^2*sinh(by*e) +
ay^2*sinh(by)*cos(ay*(e - 1)) - by^2*cos(ay)*sinh(by*(e - 1))))*(cos(ax*k) +
cosh(bx*k) - cos(ax)*cosh(bx*(k - 1)) - cosh(bx)*cos(ax*(k - 1)) +
(ax*sin(ax)*sinh(bx*(k - 1)))/bx - (bx*sinh(bx)*sin(ax*(k -
1)))/ax))/ (ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

N3ke = ((by*(ay*cosh(by)*cos(ay*(e - 1)) - ay*cos(ay*e) +
by*sin(ay)*sinh(by*(e - 1))) - ay*(by*cosh(by*e) - by*cos(ay)*cosh(by*(e -
1)) + ay*sinh(by)*sin(ay*(e - 1))))*(ax*sin(ax*k) - bx*sinh(bx*k) -
ax*cosh(bx)*sin(ax*(k - 1)) - ax*sin(ax)*cosh(bx*(k - 1)) +
bx*cos(ax)*sinh(bx*(k - 1)) + bx*sinh(bx)*cos(ax*(k -
1))))/ (ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N3k2e = -((by*(ay*cosh(by)*cos(ay*(e - 1)) - ay*cos(ay*e) +
by*sin(ay)*sinh(by*(e - 1))) - ay*(by*cosh(by*e) - by*cos(ay)*cosh(by*(e -
1)) + ay*sinh(by)*sin(ay*(e - 1))))*(bx^2*cosh(bx*k) - ax^2*cos(ax*k) +
ax^2*cosh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*cosh(bx*(k - 1)) +
ax*bx*sin(ax)*sinh(bx*(k - 1)) + ax*bx*sinh(bx)*sin(ax*(k -
1))))/ (ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N3ke2 = ((by*(ay^2*sin(ay*e) - ay^2*cosh(by)*sin(ay*(e - 1)) +
by^2*sin(ay)*cosh(by*(e - 1))) - ay*(by^2*sinh(by*e) +
ay^2*sinh(by)*cos(ay*(e - 1)) - by^2*cos(ay)*sinh(by*(e - 1))))*(ax*sin(ax*k)
- bx*sinh(bx*k) - ax*cosh(bx)*sin(ax*(k - 1)) - ax*sin(ax)*cosh(bx*(k - 1)) +
bx*cos(ax)*sinh(bx*(k - 1)) + bx*sinh(bx)*cos(ax*(k -
1))))/ (ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N3k2e2 = -((by*(ay^2*sin(ay*e) - ay^2*cosh(by)*sin(ay*(e - 1)) +
by^2*sin(ay)*cosh(by*(e - 1))) - ay*(by^2*sinh(by*e) +
ay^2*sinh(by)*cos(ay*(e - 1)) - by^2*cos(ay)*sinh(by*(e -
1))))*(bx^2*cosh(bx*k) - ax^2*cos(ax*k) + ax^2*cosh(bx)*cos(ax*(k - 1)) -
bx^2*cos(ax)*cosh(bx*(k - 1)) + ax*bx*sin(ax)*sinh(bx*(k - 1)) +
ax*bx*sinh(bx)*sin(ax*(k - 1))))/ (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N3k3 = -((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e -
1))) + ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(ax^3*sin(ax*k) + bx^3*sinh(bx*k) - ax^3*cosh(bx)*sin(ax*(k - 1)) -
bx^3*cos(ax)*sinh(bx*(k - 1)) + ax*bx^2*sin(ax)*cosh(bx*(k - 1)) +
ax^2*bx*sinh(bx)*cos(ax*(k - 1))))/ (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N3e3 = -( (by*(ay^3*cos(ay*e) - ay^3*cosh(by)*cos(ay*(e - 1)) +
by^3*sin(ay)*sinh(by*(e - 1))) + ay*(by^3*cos(ay)*cosh(by*(e - 1)) -
by^3*cosh(by*e) + ay^3*sinh(by)*sin(ay*(e - 1))) )*(cos(ax*k) + cosh(bx*k) -
cos(ax)*cosh(bx*(k - 1)) - cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k
- 1)))/bx - (bx*sinh(bx)*sin(ax*(k - 1)))/ax))/ (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N3k4 = -( (by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e -
1))) + ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))) )*(ax^4*cos(ax*k) + bx^4*cosh(bx*k) - ax^4*cosh(bx)*cos(ax*(k - 1)) -
bx^4*cos(ax)*cosh(bx*(k - 1)) + ax*bx^3*sin(ax)*sinh(bx*(k - 1)) -
ax^3*bx*sinh(bx)*sin(ax*(k - 1))) )/ (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N3e4 = -( (by*(ay^4*cosh(by)*sin(ay*(e - 1)) - ay^4*sin(ay*e) +
by^4*sin(ay)*cosh(by*(e - 1))) + ay*(ay^4*sinh(by)*cos(ay*(e - 1)) -
by^4*sinh(by*e) + by^4*cos(ay)*sinh(by*(e - 1))) )*(cos(ax*k) + cosh(bx*k) -
cos(ax)*cosh(bx*(k - 1)) - cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k
- 1)))/bx - (bx*sinh(bx)*sin(ax*(k - 1)))/ax))/ (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

% Node 4

```

N4k = ((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1)))
+ ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(bx*(ax*cosh(bx)*cos(ax*(k - 1)) - ax*cos(ax*k) + bx*sin(ax)*sinh(bx*(k
- 1))) - ax*(bx*cosh(bx*k) - bx*cos(ax)*cosh(bx*(k - 1)) +
ax*sinh(bx)*sin(ax*(k - 1))))/ (ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N4kk = ((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1)))
+ ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(bx*(ax^2*sin(ax*k) - ax^2*cosh(bx)*sin(ax*(k - 1)) +
bx^2*sin(ax)*cosh(bx*(k - 1))) - ax*(bx^2*sinh(bx*k) +
ax^2*sinh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*sinh(bx*(k -
1)))))/ (ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

```

```

N4e = ((bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k - 1)))
+ ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))))*(by*(ay*cosh(by)*cos(ay*(e - 1)) - ay*cos(ay*e) + by*sin(ay)*sinh(by*(e
- 1))) - ay*(by*cosh(by*e) - by*cos(ay)*cosh(by*(e - 1)) +
ay*sinh(by)*sin(ay*(e - 1)))))/ (ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N4ee = ((bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k - 1)))
+ ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))))*(by*(ay^2*sin(ay*e) - ay^2*cosh(by)*sin(ay*(e - 1)) +

```

```

by^2*sin(ay)*cosh(by*(e - 1))) - ay*(by^2*sinh(by*e) +
ay^2*sinh(by)*cos(ay*(e - 1)) - by^2*cos(ay)*sinh(by*(e -
1)))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

N4ke = ((by*(ay*cosh(by)*cos(ay*(e - 1)) - ay*cos(ay*e) +
by*sin(ay)*sinh(by*(e - 1))) - ay*(by*cosh(by*e) - by*cos(ay)*cosh(by*(e -
1)) + ay*sinh(by)*sin(ay*(e - 1))))*(bx*(ax*cosh(bx)*cos(ax*(k - 1)) -
ax*cos(ax*k) + bx*sin(ax)*sinh(bx*(k - 1))) - ax*(bx*cosh(bx*k) -
bx*cos(ax)*cosh(bx*(k - 1)) + ax*sinh(bx)*sin(ax*(k -
1)))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

N4k2e = ((by*(ay*cosh(by)*cos(ay*(e - 1)) - ay*cos(ay*e) +
by*sin(ay)*sinh(by*(e - 1))) - ay*(by*cosh(by*e) - by*cos(ay)*cosh(by*(e -
1)) + ay*sinh(by)*sin(ay*(e - 1))))*(bx*(ax^2*sin(ax*k) -
ax^2*cosh(bx)*sin(ax*(k - 1)) + bx^2*sin(ax)*cosh(bx*(k - 1))) -
ax*(bx^2*sinh(bx*k) + ax^2*sinh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*sinh(bx*(k
- 1)))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

N4ke2 = ((bx*(ax*cosh(bx)*cos(ax*(k - 1)) - ax*cos(ax*k) +
bx*sin(ax)*sinh(bx*(k - 1))) - ax*(bx*cosh(bx*k) - bx*cos(ax)*cosh(bx*(k -
1)) + ax*sinh(bx)*sin(ax*(k - 1))))*(by*(ay^2*sin(ay*e) -
ay^2*cosh(by)*sin(ay*(e - 1)) + by^2*sin(ay)*cosh(by*(e - 1))) -
ay*(by^2*sinh(by*e) + ay^2*sinh(by)*cos(ay*(e - 1)) - by^2*cos(ay)*sinh(by*(e
- 1)))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

N4k2e2 = ((by*(ay^2*sin(ay*e) - ay^2*cosh(by)*sin(ay*(e - 1)) +
by^2*sin(ay)*cosh(by*(e - 1))) - ay*(by^2*sinh(by*e) +
ay^2*sinh(by)*cos(ay*(e - 1)) - by^2*cos(ay)*sinh(by*(e -
1))))*(bx*(ax^2*sin(ax*k) - ax^2*cosh(bx)*sin(ax*(k - 1)) +
bx^2*sin(ax)*cosh(bx*(k - 1))) - ax*(bx^2*sinh(bx*k) +
ax^2*sinh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*sinh(bx*(k -
1)))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

N4k3 = ((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1)))
+ ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(bx*(ax^3*cos(ax*k) - ax^3*cosh(bx)*cos(ax*(k - 1)) +
bx^3*sin(ax)*sinh(bx*(k - 1)) + ax*(bx^3*cos(ax)*cosh(bx*(k - 1)) -
bx^3*cosh(bx*k) + ax^3*sinh(bx)*sin(ax*(k -
1)))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

N4e3 = ((bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k - 1)))
+ ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))))*(by*(ay^3*cos(ay*e) - ay^3*cosh(by)*cos(ay*(e - 1)) +

```

```
by^3*sin(ay)*sinh(by*(e - 1))) + ay*(by^3*cos(ay)*cosh(by*(e - 1)) -
by^3*cosh(by*e) + ay^3*sinh(by)*sin(ay*(e -
1)))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2)))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2)))/(ay*by) - 2));
```

```
N4k4 = ((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1)))
+ ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(bx*(ax^4*cosh(bx)*sin(ax*(k - 1)) - ax^4*sin(ax*k) +
bx^4*sin(ax)*cosh(bx*(k - 1)) + ax*(ax^4*sinh(bx)*cos(ax*(k - 1)) -
bx^4*sinh(bx*k) + bx^4*cos(ax)*sinh(bx*(k -
1)))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2)))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2)))/(ay*by) - 2));
```

```
N4e4 = ((bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k - 1)))
+ ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))))*(by*(ay^4*cosh(by)*sin(ay*(e - 1)) - ay^4*sin(ay*e) +
by^4*sin(ay)*cosh(by*(e - 1)) + ay*(ay^4*sinh(by)*cos(ay*(e - 1)) -
by^4*sinh(by*e) + by^4*cos(ay)*sinh(by*(e -
1)))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2)))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2)))/(ay*by) - 2));
```

% Node 5

```
N5k = -((cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e - 1)) -
cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e - 1)))/by -
(by*sinh(by)*sin(ay*(e - 1)))/ay)*(ax*sin(ax*(k - 1)) - bx*sinh(bx*(k - 1)) +
ax*cosh(bx*k)*sin(ax) - ax*sin(ax*k)*cosh(bx) - bx*cos(ax*k)*sinh(bx) +
bx*sinh(bx*k)*cos(ax)))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2)))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2)))/(ay*by) - 2));
```

```
N5kk = -((cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e - 1)) -
cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e - 1)))/by -
(by*sinh(by)*sin(ay*(e - 1)))/ay)*(ax^2*cos(ax*(k - 1)) - bx^2*cosh(bx*(k -
1)) - ax^2*cos(ax*k)*cosh(bx) + bx^2*cosh(bx*k)*cos(ax) +
ax*bx*sin(ax*k)*sinh(bx) + ax*bx*sinh(bx*k)*sin(ax)))/((2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)))/(ay*by) - 2));
```

```
N5e = -((ay*sin(ay*e) - by*sinh(by*e) - ay*cosh(by)*sin(ay*(e - 1)) -
ay*sin(ay)*cosh(by*(e - 1)) + by*cos(ay)*sinh(by*(e - 1)) +
by*sinh(by)*cos(ay*(e - 1)))*(cos(ax*(k - 1)) + cosh(bx*(k - 1)) -
cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) - (ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2)))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2)))/(ay*by) - 2));
```

```
N5ee = ((by^2*cosh(by*e) - ay^2*cos(ay*e) + ay^2*cosh(by)*cos(ay*(e - 1)) -
by^2*cos(ay)*cosh(by*(e - 1)) + ay*by*sin(ay)*sinh(by*(e - 1)) +
ay*by*sinh(by)*sin(ay*(e - 1)))*(cos(ax*(k - 1)) + cosh(bx*(k - 1)) -
cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) - (ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
```

$$\frac{bx^2}{ax \cdot bx - 2} \cdot (2 \cos(ay) \cosh(by) + (\sin(ay) \sinh(by) \cdot (ay^2 - by^2)) / (ay \cdot by - 2));$$

$$\begin{aligned} N5ke = & ((ay \cdot \sin(ay \cdot e) - by \cdot \sinh(by \cdot e) - ay \cdot \cosh(by) \cdot \sin(ay \cdot (e - 1)) - \\ & ay \cdot \sin(ay) \cdot \cosh(by \cdot (e - 1)) + by \cdot \cos(ay) \cdot \sinh(by \cdot (e - 1)) + \\ & by \cdot \sinh(by) \cdot \cos(ay \cdot (e - 1))) \cdot (ax \cdot \sin(ax \cdot (k - 1)) - bx \cdot \sinh(bx \cdot (k - 1)) + \\ & ax \cdot \cosh(bx \cdot k) \cdot \sin(ax) - ax \cdot \sin(ax \cdot k) \cdot \cosh(bx) - bx \cdot \cos(ax \cdot k) \cdot \sinh(bx) + \\ & bx \cdot \sinh(bx \cdot k) \cdot \cos(ax)) / ((2 \cdot \cos(ax) \cdot \cosh(bx) + (\sin(ax) \cdot \sinh(bx) \cdot (ax^2 - \\ & bx^2)) / (ax \cdot bx - 2) \cdot (2 \cdot \cos(ay) \cdot \cosh(by) + (\sin(ay) \cdot \sinh(by) \cdot (ay^2 - \\ & by^2)) / (ay \cdot by - 2)); \end{aligned}$$

$$\begin{aligned} N5k2e = & ((ay \cdot \sin(ay \cdot e) - by \cdot \sinh(by \cdot e) - ay \cdot \cosh(by) \cdot \sin(ay \cdot (e - 1)) - \\ & ay \cdot \sin(ay) \cdot \cosh(by \cdot (e - 1)) + by \cdot \cos(ay) \cdot \sinh(by \cdot (e - 1)) + \\ & by \cdot \sinh(by) \cdot \cos(ay \cdot (e - 1))) \cdot (ax^2 \cdot \cos(ax \cdot (k - 1)) - bx^2 \cdot \cosh(bx \cdot (k - 1)) - \\ & ax^2 \cdot \cos(ax \cdot k) \cdot \cosh(bx) + bx^2 \cdot \cosh(bx \cdot k) \cdot \cos(ax) + ax \cdot bx \cdot \sin(ax \cdot k) \cdot \sinh(bx) \\ & + ax \cdot bx \cdot \sinh(bx \cdot k) \cdot \sin(ax)) / ((2 \cdot \cos(ax) \cdot \cosh(bx) + (\sin(ax) \cdot \sinh(bx) \cdot (ax^2 - \\ & bx^2)) / (ax \cdot bx - 2) \cdot (2 \cdot \cos(ay) \cdot \cosh(by) + (\sin(ay) \cdot \sinh(by) \cdot (ay^2 - \\ & by^2)) / (ay \cdot by - 2)); \end{aligned}$$

$$\begin{aligned} N5ke2 = & -((by^2 \cdot \cosh(by \cdot e) - ay^2 \cdot \cos(ay \cdot e) + ay^2 \cdot \cosh(by) \cdot \cos(ay \cdot (e - 1)) - \\ & by^2 \cdot \cos(ay) \cdot \cosh(by \cdot (e - 1)) + ay \cdot by \cdot \sin(ay) \cdot \sinh(by \cdot (e - 1)) + \\ & ay \cdot by \cdot \sinh(by) \cdot \sin(ay \cdot (e - 1))) \cdot (ax \cdot \sin(ax \cdot (k - 1)) - bx \cdot \sinh(bx \cdot (k - 1)) + \\ & ax \cdot \cosh(bx \cdot k) \cdot \sin(ax) - ax \cdot \sin(ax \cdot k) \cdot \cosh(bx) - bx \cdot \cos(ax \cdot k) \cdot \sinh(bx) + \\ & bx \cdot \sinh(bx \cdot k) \cdot \cos(ax)) / ((2 \cdot \cos(ax) \cdot \cosh(bx) + (\sin(ax) \cdot \sinh(bx) \cdot (ax^2 - \\ & bx^2)) / (ax \cdot bx - 2) \cdot (2 \cdot \cos(ay) \cdot \cosh(by) + (\sin(ay) \cdot \sinh(by) \cdot (ay^2 - \\ & by^2)) / (ay \cdot by - 2)); \end{aligned}$$

$$\begin{aligned} N5k2e2 = & -((by^2 \cdot \cosh(by \cdot e) - ay^2 \cdot \cos(ay \cdot e) + ay^2 \cdot \cosh(by) \cdot \cos(ay \cdot (e - 1)) \\ & - by^2 \cdot \cos(ay) \cdot \cosh(by \cdot (e - 1)) + ay \cdot by \cdot \sin(ay) \cdot \sinh(by \cdot (e - 1)) + \\ & ay \cdot by \cdot \sinh(by) \cdot \sin(ay \cdot (e - 1))) \cdot (ax^2 \cdot \cos(ax \cdot (k - 1)) - bx^2 \cdot \cosh(bx \cdot (k - 1)) \\ & - ax^2 \cdot \cos(ax \cdot k) \cdot \cosh(bx) + bx^2 \cdot \cosh(bx \cdot k) \cdot \cos(ax) + \\ & ax \cdot bx \cdot \sin(ax \cdot k) \cdot \sinh(bx) + ax \cdot bx \cdot \sinh(bx \cdot k) \cdot \sin(ax)) / ((2 \cdot \cos(ax) \cdot \cosh(bx) + \\ & (\sin(ax) \cdot \sinh(bx) \cdot (ax^2 - bx^2)) / (ax \cdot bx - 2) \cdot (2 \cdot \cos(ay) \cdot \cosh(by) + \\ & (\sin(ay) \cdot \sinh(by) \cdot (ay^2 - by^2)) / (ay \cdot by - 2)); \end{aligned}$$

$$\begin{aligned} N5k3 = & -((\cos(ay \cdot e) + \cosh(by \cdot e) - \cos(ay) \cdot \cosh(by \cdot (e - 1)) - \\ & \cosh(by) \cdot \cos(ay \cdot (e - 1)) + (ay \cdot \sin(ay) \cdot \sinh(by \cdot (e - 1))) / by - \\ & (by \cdot \sinh(by) \cdot \sin(ay \cdot (e - 1))) / ay) \cdot (ax^3 \cdot \sin(ax \cdot k) \cdot \cosh(bx) - bx^3 \cdot \sinh(bx \cdot (k \\ & - 1)) - ax^3 \cdot \sin(ax \cdot (k - 1)) + bx^3 \cdot \sinh(bx \cdot k) \cdot \cos(ax) + \\ & ax \cdot bx^2 \cdot \cosh(bx \cdot k) \cdot \sin(ax) + \\ & ax^2 \cdot bx \cdot \cos(ax \cdot k) \cdot \sinh(bx)) / ((2 \cdot \cos(ax) \cdot \cosh(bx) + (\sin(ax) \cdot \sinh(bx) \cdot (ax^2 - \\ & bx^2)) / (ax \cdot bx - 2) \cdot (2 \cdot \cos(ay) \cdot \cosh(by) + (\sin(ay) \cdot \sinh(by) \cdot (ay^2 - \\ & by^2)) / (ay \cdot by - 2)); \end{aligned}$$

$$\begin{aligned} N5e3 = & ((ay^3 \cdot \sin(ay \cdot e) + by^3 \cdot \sinh(by \cdot e) - ay^3 \cdot \cosh(by) \cdot \sin(ay \cdot (e - 1)) - \\ & by^3 \cdot \cos(ay) \cdot \sinh(by \cdot (e - 1)) + ay \cdot by^2 \cdot \sin(ay) \cdot \cosh(by \cdot (e - 1)) + \\ & ay^2 \cdot by \cdot \sinh(by) \cdot \cos(ay \cdot (e - 1))) \cdot (\cos(ax \cdot (k - 1)) + \cosh(bx \cdot (k - 1)) - \\ & \cos(ax \cdot k) \cdot \cosh(bx) - \cosh(bx \cdot k) \cdot \cos(ax) - (ax \cdot \sinh(bx \cdot k) \cdot \sin(ax)) / bx + \\ & (bx \cdot \sin(ax \cdot k) \cdot \sinh(bx)) / ax) / ((2 \cdot \cos(ax) \cdot \cosh(bx) + (\sin(ax) \cdot \sinh(bx) \cdot (ax^2 - \\ & bx^2)) / (ax \cdot bx - 2) \cdot (2 \cdot \cos(ay) \cdot \cosh(by) + (\sin(ay) \cdot \sinh(by) \cdot (ay^2 - \\ & by^2)) / (ay \cdot by - 2)); \end{aligned}$$

$$\begin{aligned} N5k4 = & ((\cos(ay \cdot e) + \cosh(by \cdot e) - \cos(ay) \cdot \cosh(by \cdot (e - 1)) - \\ & \cosh(by) \cdot \cos(ay \cdot (e - 1)) + (ay \cdot \sin(ay) \cdot \sinh(by \cdot (e - 1))) / by - \\ & (by \cdot \sinh(by) \cdot \sin(ay \cdot (e - 1))) / ay) \cdot (ax^4 \cdot \cos(ax \cdot (k - 1)) + bx^4 \cdot \cosh(bx \cdot (k - \end{aligned}$$

```

1)) - ax^4*cos(ax*k)*cosh(bx) - bx^4*cosh(bx*k)*cos(ax) -
ax*bx^3*sinh(bx*k)*sin(ax) +
ax^3*bx*sin(ax*k)*sinh(bx))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

```

```

N5e4 = ((cos(ax*(k - 1)) + cosh(bx*(k - 1)) - cos(ax*k)*cosh(bx) -
cosh(bx*k)*cos(ax) - (ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax)*(ay^4*cos(ay*e) + by^4*cosh(by*e) -
ay^4*cosh(by)*cos(ay*(e - 1)) - by^4*cos(ay)*cosh(by*(e - 1)) +
ay*by^3*sin(ay)*sinh(by*(e - 1)) - ay^3*by*sinh(by)*sin(ay*(e -
1)))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

% Node 6

```

N6k = ((bx*(ax*cos(ax*(k - 1)) - ax*cos(ax*k)*cosh(bx) +
bx*sinh(bx*k)*sin(ax)) - ax*(bx*cosh(bx*k)*cos(ax) - bx*cosh(bx*(k - 1)) +
ax*sin(ax*k)*sinh(bx)))*(cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e - 1)) -
cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e - 1)))/by -
(by*sinh(by)*sin(ay*(e - 1)))/ay))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N6kk = ((bx*(ax^2*sin(ax*k)*cosh(bx) - ax^2*sin(ax*(k - 1)) +
bx^2*cosh(bx*k)*sin(ax)) - ax*(ax^2*cos(ax*k)*sinh(bx) - bx^2*sinh(bx*(k -
1)) + bx^2*sinh(bx*k)*cos(ax)))*(cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e -
1)) - cosh(by)*cos(ay*(e - 1)) + (ay*sin(ay)*sinh(by*(e - 1)))/by -
(by*sinh(by)*sin(ay*(e - 1)))/ay))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N6e = -((bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) +
ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) -
sinh(bx*k)*cos(ax)))*(ay*sin(ay*e) - by*sinh(by*e) - ay*cosh(by)*sin(ay*(e -
1)) - ay*sin(ay)*cosh(by*(e - 1)) + by*cos(ay)*sinh(by*(e - 1)) +
by*sinh(by)*cos(ay*(e - 1)))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N6ee = ((bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) +
ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) -
sinh(bx*k)*cos(ax)))*(by^2*cosh(by*e) - ay^2*cos(ay*e) +
ay^2*cosh(by)*cos(ay*(e - 1)) - by^2*cos(ay)*cosh(by*(e - 1)) +
ay*by*sin(ay)*sinh(by*(e - 1)) + ay*by*sinh(by)*sin(ay*(e -
1)))/(ax*bx*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N6ke = -((bx*(ax*cos(ax*(k - 1)) - ax*cos(ax*k)*cosh(bx) +
bx*sinh(bx*k)*sin(ax)) - ax*(bx*cosh(bx*k)*cos(ax) - bx*cosh(bx*(k - 1)) +
ax*sin(ax*k)*sinh(bx)))*(ay*sin(ay*e) - by*sinh(by*e) - ay*cosh(by)*sin(ay*(e -
1)) - ay*sin(ay)*cosh(by*(e - 1)) + by*cos(ay)*sinh(by*(e - 1)) +
by*sinh(by)*cos(ay*(e - 1)))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```


$$N6k2e = -((bx*(ax^2*\sin(ax*k)*\cosh(bx) - ax^2*\sin(ax*(k-1)) + bx^2*\cosh(bx*k)*\sin(ax)) - ax*(ax^2*\cos(ax*k)*\sinh(bx) - bx^2*\sinh(bx*(k-1)) + bx^2*\sinh(bx*k)*\cos(ax)))*(ay*\sin(ay*e) - by*\sinh(by*e) - ay*\cosh(by)*\sin(ay*(e-1)) - ay*\sin(ay)*\cosh(by*(e-1)) + by*\cos(ay)*\sinh(by*(e-1)) + by*\sinh(by)*\cos(ay*(e-1))))/(ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2));$$

$$N6ke2 = ((bx*(ax*\cos(ax*(k-1)) - ax*\cos(ax*k)*\cosh(bx) + bx*\sinh(bx*k)*\sin(ax)) - ax*(bx*\cosh(bx*k)*\cos(ax) - bx*\cosh(bx*(k-1)) + ax*\sin(ax*k)*\sinh(bx)))*(by^2*\cosh(by*e) - ay^2*\cos(ay*e) + ay^2*\cosh(by)*\cos(ay*(e-1)) - by^2*\cos(ay)*\cosh(by*(e-1)) + ay*by*\sin(ay)*\sinh(by*(e-1)) + ay*by*\sinh(by)*\sin(ay*(e-1))))/(ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2));$$

$$N6k2e2 = ((bx*(ax^2*\sin(ax*k)*\cosh(bx) - ax^2*\sin(ax*(k-1)) + bx^2*\cosh(bx*k)*\sin(ax)) - ax*(ax^2*\cos(ax*k)*\sinh(bx) - bx^2*\sinh(bx*(k-1)) + bx^2*\sinh(bx*k)*\cos(ax)))*(by^2*\cosh(by*e) - ay^2*\cos(ay*e) + ay^2*\cosh(by)*\cos(ay*(e-1)) - by^2*\cos(ay)*\cosh(by*(e-1)) + ay*by*\sin(ay)*\sinh(by*(e-1)) + ay*by*\sinh(by)*\sin(ay*(e-1))))/(ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2));$$

$$N6k3 = ((bx*(ax^3*\cos(ax*k)*\cosh(bx) - ax^3*\cos(ax*(k-1)) + bx^3*\sinh(bx*k)*\sin(ax)) + ax*(bx^3*\cosh(bx*(k-1)) - bx^3*\cosh(bx*k)*\cos(ax) + ax^3*\sin(ax*k)*\sinh(bx)))*(cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e-1)) - cosh(by)*cos(ay*(e-1)) + (ay*\sin(ay)*\sinh(by*(e-1)))/by - (by*\sinh(by)*\sin(ay*(e-1)))/ay))/(ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2));$$

$$N6e3 = ((bx*(\sin(ax*(k-1)) + \cosh(bx*k)*\sin(ax) - \sin(ax*k)*\cosh(bx)) + ax*(\sinh(bx*(k-1)) + \cos(ax*k)*\sinh(bx) - \sinh(bx*k)*\cos(ax)))*(ay^3*\sin(ay*e) + by^3*\sinh(by*e) - ay^3*\cosh(by)*\sin(ay*(e-1)) - by^3*\cos(ay)*\sinh(by*(e-1)) + ay*by^2*\sin(ay)*\cosh(by*(e-1)) + ay^2*by*\sinh(by)*\cos(ay*(e-1))))/(ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2));$$

$$N6k4 = ((bx*(ax^4*\sin(ax*(k-1)) - ax^4*\sin(ax*k)*\cosh(bx) + bx^4*\cosh(bx*k)*\sin(ax)) + ax*(bx^4*\sinh(bx*(k-1)) + ax^4*\cos(ax*k)*\sinh(bx) - bx^4*\sinh(bx*k)*\cos(ax)))*(cos(ay*e) + cosh(by*e) - cos(ay)*cosh(by*(e-1)) - cosh(by)*cos(ay*(e-1)) + (ay*\sin(ay)*\sinh(by*(e-1)))/by - (by*\sinh(by)*\sin(ay*(e-1)))/ay))/(ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2));$$

$$N6e4 = ((bx*(\sin(ax*(k-1)) + \cosh(bx*k)*\sin(ax) - \sin(ax*k)*\cosh(bx)) + ax*(\sinh(bx*(k-1)) + \cos(ax*k)*\sinh(bx) - \sinh(bx*k)*\cos(ax)))*(ay^4*\cos(ay*e) + by^4*\cosh(by*e) - ay^4*\cosh(by)*\cos(ay*(e-1)) - by^4*\cos(ay)*\cosh(by*(e-1)) + ay*by^3*\sin(ay)*\sinh(by*(e-1)) - ay^3*by*\sinh(by)*\sin(ay*(e-1))))/(ax*bx*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2));$$

% Node 7

```
N7k = ((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1)))
+ ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(ax*sin(ax*(k - 1)) - bx*sinh(bx*(k - 1)) + ax*cosh(bx*k)*sin(ax) -
ax*sin(ax*k)*cosh(bx) - bx*cos(ax*k)*sinh(bx) +
bx*sinh(bx*k)*cos(ax))/ (ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2
- bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));
```

```
N7kk = ((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1)))
+ ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(ax^2*cos(ax*(k - 1)) - bx^2*cosh(bx*(k - 1)) - ax^2*cos(ax*k)*cosh(bx)
+ bx^2*cosh(bx*k)*cos(ax) + ax*bx*sin(ax*k)*sinh(bx) +
ax*bx*sinh(bx*k)*sin(ax))/ (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N7e = -((by*(ay*cosh(by)*cos(ay*(e - 1)) - ay*cos(ay*e) +
by*sin(ay)*sinh(by*(e - 1))) - ay*(by*cosh(by*e) - by*cos(ay)*cosh(by*(e -
1)) + ay*sinh(by)*sin(ay*(e - 1))))*(cos(ax*(k - 1)) + cosh(bx*(k - 1)) -
cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) - (ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/ (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N7ee = -((by*(ay^2*sin(ay*e) - ay^2*cosh(by)*sin(ay*(e - 1)) +
by^2*sin(ay)*cosh(by*(e - 1))) - ay*(by^2*sinh(by*e) +
ay^2*sinh(by)*cos(ay*(e - 1)) - by^2*cos(ay)*sinh(by*(e - 1))))*(cos(ax*(k -
1)) + cosh(bx*(k - 1)) - cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) -
(ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/ (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N7ke = ((by*(ay*cosh(by)*cos(ay*(e - 1)) - ay*cos(ay*e) +
by*sin(ay)*sinh(by*(e - 1))) - ay*(by*cosh(by*e) - by*cos(ay)*cosh(by*(e -
1)) + ay*sinh(by)*sin(ay*(e - 1))))*(ax*sin(ax*(k - 1)) - bx*sinh(bx*(k - 1))
+ ax*cosh(bx*k)*sin(ax) - ax*sin(ax*k)*cosh(bx) - bx*cos(ax*k)*sinh(bx) +
bx*sinh(bx*k)*cos(ax))/ (ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2
- bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));
```

```
N7k2e = ((by*(ay*cosh(by)*cos(ay*(e - 1)) - ay*cos(ay*e) +
by*sin(ay)*sinh(by*(e - 1))) - ay*(by*cosh(by*e) - by*cos(ay)*cosh(by*(e -
1)) + ay*sinh(by)*sin(ay*(e - 1))))*(ax^2*cos(ax*(k - 1)) - bx^2*cosh(bx*(k -
1)) - ax^2*cos(ax*k)*cosh(bx) + bx^2*cosh(bx*k)*cos(ax) +
ax*bx*sin(ax*k)*sinh(bx) +
ax*bx*sinh(bx*k)*sin(ax))/ (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N7ke2 = ((by*(ay^2*sin(ay*e) - ay^2*cosh(by)*sin(ay*(e - 1)) +
by^2*sin(ay)*cosh(by*(e - 1))) - ay*(by^2*sinh(by*e) +
ay^2*sinh(by)*cos(ay*(e - 1)) - by^2*cos(ay)*sinh(by*(e - 1))))*(ax*sin(ax*(k
```

```
- 1)) - bx*sinh(bx*(k - 1)) + ax*cosh(bx*k)*sin(ax) - ax*sin(ax*k)*cosh(bx) -
bx*cos(ax*k)*sinh(bx) + bx*sinh(bx*k)*cos(ax)) / (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));
```

```
N7k2e2 = ((by*(ay^2*sin(ay*e) - ay^2*cosh(by)*sin(ay*(e - 1)) +
by^2*sin(ay)*cosh(by*(e - 1))) - ay*(by^2*sinh(by*e) +
ay^2*sinh(by)*cos(ay*(e - 1)) - by^2*cos(ay)*sinh(by*(e -
1))))*(ax^2*cos(ax*(k - 1)) - bx^2*cosh(bx*(k - 1)) - ax^2*cos(ax*k)*cosh(bx)
+ bx^2*cosh(bx*k)*cos(ax) + ax*bx*sin(ax*k)*sinh(bx) +
ax*bx*sinh(bx*k)*sin(ax)) / (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));
```

```
N7k3 = ((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1)))
+ ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(ax^3*sin(ax*k)*cosh(bx) - bx^3*sinh(bx*(k - 1)) - ax^3*sin(ax*(k - 1))
+ bx^3*sinh(bx*k)*cos(ax) + ax*bx^2*cosh(bx*k)*sin(ax) +
ax^2*bx*cos(ax*k)*sinh(bx)) / (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));
```

```
N7e3 = -((by*(ay^3*cos(ay*e) - ay^3*cosh(by)*cos(ay*(e - 1)) +
by^3*sin(ay)*sinh(by*(e - 1))) + ay*(by^3*cos(ay)*cosh(by*(e - 1)) -
by^3*cosh(by*e) + ay^3*sinh(by)*sin(ay*(e - 1))))*(cos(ax*(k - 1)) +
cosh(bx*(k - 1)) - cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) -
(ax*sinh(bx*k)*sin(ax)) / bx +
(bx*sin(ax*k)*sinh(bx)) / ax) / (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));
```

```
N7k4 = -((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e -
1))) + ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(ax^4*cos(ax*(k - 1)) + bx^4*cosh(bx*(k - 1)) - ax^4*cos(ax*k)*cosh(bx)
- bx^4*cosh(bx*k)*cos(ax) - ax*bx^3*sinh(bx*k)*sin(ax) +
ax^3*bx*sin(ax*k)*sinh(bx)) / (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));
```

```
N7e4 = -((by*(ay^4*cosh(by)*sin(ay*(e - 1)) - ay^4*sin(ay*e) +
by^4*sin(ay)*cosh(by*(e - 1))) + ay*(ay^4*sinh(by)*cos(ay*(e - 1)) -
by^4*sinh(by*e) + by^4*cos(ay)*sinh(by*(e - 1))))*(cos(ax*(k - 1)) +
cosh(bx*(k - 1)) - cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) -
(ax*sinh(bx*k)*sin(ax)) / bx +
(bx*sin(ax*k)*sinh(bx)) / ax) / (ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));
```

% Node 8

```
N8k = -((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e - 1)))
+ ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(bx*(ax*cos(ax*(k - 1)) - ax*cos(ax*k)*cosh(bx) +
bx*sinh(bx*k)*sin(ax)) - ax*(bx*cosh(bx*k)*cos(ax) - bx*cosh(bx*(k - 1)) +
ax*sin(ax*k)*sinh(bx))) / (ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
```

$$(\sin(ax) \sinh(bx) (ax^2 - bx^2)) / (ax \cdot bx - 2) * (2 \cos(ay) \cosh(by) + (\sin(ay) \sinh(by) (ay^2 - by^2)) / (ay \cdot by - 2));$$

$$\begin{aligned} N8kk = & -((by * (\cosh(by) * \sin(ay * (e - 1)) - \sin(ay * e) + \sin(ay) * \cosh(by * (e - 1))) + ay * (\cos(ay) * \sinh(by * (e - 1)) - \sinh(by * e) + \sinh(by) * \cos(ay * (e - 1)))) * (bx * (ax^2 * \sin(ax * k) * \cosh(bx) - ax^2 * \sin(ax * (k - 1)) + bx^2 * \cosh(bx * k) * \sin(ax)) - ax * (ax^2 * \cos(ax * k) * \sinh(bx) - bx^2 * \sinh(bx * (k - 1)) + bx^2 * \sinh(bx * k) * \cos(ax))) / (ax * ay * bx * by * (2 * \cos(ax) * \cosh(bx) + (\sin(ax) * \sinh(bx) (ax^2 - bx^2)) / (ax \cdot bx - 2) * (2 * \cos(ay) * \cosh(by) + (\sin(ay) * \sinh(by) (ay^2 - by^2)) / (ay \cdot by - 2)); \end{aligned}$$

$$\begin{aligned} N8e = & -((by * (ay * \cosh(by) * \cos(ay * (e - 1)) - ay * \cos(ay * e) + by * \sin(ay) * \sinh(by * (e - 1))) - ay * (by * \cosh(by * e) - by * \cos(ay) * \cosh(by * (e - 1)) + ay * \sinh(by) * \sin(ay * (e - 1)))) * (bx * (\sin(ax * (k - 1)) + \cosh(bx * k) * \sin(ax) - \sin(ax * k) * \cosh(bx)) + ax * (\sinh(bx * (k - 1)) + \cos(ax * k) * \sinh(bx) - \sinh(bx * k) * \cos(ax))) / (ax * ay * bx * by * (2 * \cos(ax) * \cosh(bx) + (\sin(ax) * \sinh(bx) (ax^2 - bx^2)) / (ax \cdot bx - 2) * (2 * \cos(ay) * \cosh(by) + (\sin(ay) * \sinh(by) (ay^2 - by^2)) / (ay \cdot by - 2)); \end{aligned}$$

$$\begin{aligned} N8ee = & -((by * (ay^2 * \sin(ay * e) - ay^2 * \cosh(by) * \sin(ay * (e - 1)) + by^2 * \sin(ay) * \cosh(by * (e - 1))) - ay * (by^2 * \sinh(by * e) + ay^2 * \sinh(by) * \cos(ay * (e - 1)) - by^2 * \cos(ay) * \sinh(by * (e - 1)))) * (bx * (\sin(ax * (k - 1)) + \cosh(bx * k) * \sin(ax) - \sin(ax * k) * \cosh(bx)) + ax * (\sinh(bx * (k - 1)) + \cos(ax * k) * \sinh(bx) - \sinh(bx * k) * \cos(ax))) / (ax * ay * bx * by * (2 * \cos(ax) * \cosh(bx) + (\sin(ax) * \sinh(bx) (ax^2 - bx^2)) / (ax \cdot bx - 2) * (2 * \cos(ay) * \cosh(by) + (\sin(ay) * \sinh(by) (ay^2 - by^2)) / (ay \cdot by - 2)); \end{aligned}$$

$$\begin{aligned} N8ke = & -((by * (ay * \cosh(by) * \cos(ay * (e - 1)) - ay * \cos(ay * e) + by * \sin(ay) * \sinh(by * (e - 1))) - ay * (by * \cosh(by * e) - by * \cos(ay) * \cosh(by * (e - 1)) + ay * \sinh(by) * \sin(ay * (e - 1)))) * (bx * (ax * \cos(ax * (k - 1)) - ax * \cos(ax * k) * \cosh(bx) + bx * \sinh(bx * k) * \sin(ax)) - ax * (bx * \cosh(bx * k) * \cos(ax) - bx * \cosh(bx * (k - 1)) + ax * \sin(ax * k) * \sinh(bx))) / (ax * ay * bx * by * (2 * \cos(ax) * \cosh(bx) + (\sin(ax) * \sinh(bx) (ax^2 - bx^2)) / (ax \cdot bx - 2) * (2 * \cos(ay) * \cosh(by) + (\sin(ay) * \sinh(by) (ay^2 - by^2)) / (ay \cdot by - 2)); \end{aligned}$$

$$\begin{aligned} N8k2e = & -((by * (ay * \cosh(by) * \cos(ay * (e - 1)) - ay * \cos(ay * e) + by * \sin(ay) * \sinh(by * (e - 1))) - ay * (by * \cosh(by * e) - by * \cos(ay) * \cosh(by * (e - 1)) + ay * \sinh(by) * \sin(ay * (e - 1)))) * (bx * (ax^2 * \sin(ax * k) * \cosh(bx) - ax^2 * \sin(ax * (k - 1)) + bx^2 * \cosh(bx * k) * \sin(ax)) - ax * (ax^2 * \cos(ax * k) * \sinh(bx) - bx^2 * \sinh(bx * (k - 1)) + bx^2 * \sinh(bx * k) * \cos(ax))) / (ax * ay * bx * by * (2 * \cos(ax) * \cosh(bx) + (\sin(ax) * \sinh(bx) (ax^2 - bx^2)) / (ax \cdot bx - 2) * (2 * \cos(ay) * \cosh(by) + (\sin(ay) * \sinh(by) (ay^2 - by^2)) / (ay \cdot by - 2)); \end{aligned}$$

$$\begin{aligned} N8ke2 = & -((by * (ay^2 * \sin(ay * e) - ay^2 * \cosh(by) * \sin(ay * (e - 1)) + by^2 * \sin(ay) * \cosh(by * (e - 1))) - ay * (by^2 * \sinh(by * e) + ay^2 * \sinh(by) * \cos(ay * (e - 1)) - by^2 * \cos(ay) * \sinh(by * (e - 1)))) * (bx * (ax * \cos(ax * (k - 1)) - ax * \cos(ax * k) * \cosh(bx) + bx * \sinh(bx * k) * \sin(ax)) - ax * (bx * \cosh(bx * k) * \cos(ax) - bx * \cosh(bx * (k - 1)) + ax * \sin(ax * k) * \sinh(bx))) / (ax * ay * bx * by * (2 * \cos(ax) * \cosh(bx) + (\sin(ax) * \sinh(bx) (ax^2 - bx^2)) / (ax \cdot bx - 2) * (2 * \cos(ay) * \cosh(by) + (\sin(ay) * \sinh(by) (ay^2 - by^2)) / (ay \cdot by - 2)); \end{aligned}$$

```

N8k2e2 = -((by*(ay^2*sin(ay*e) - ay^2*cosh(by)*sin(ay*(e - 1)) +
by^2*sin(ay)*cosh(by*(e - 1))) - ay*(by^2*sinh(by*e) +
ay^2*sinh(by)*cos(ay*(e - 1)) - by^2*cos(ay)*sinh(by*(e -
1))))*(bx*(ax^2*sin(ax*k)*cosh(bx) - ax^2*sin(ax*(k - 1)) +
bx^2*cosh(bx*k)*sin(ax)) - ax*(ax^2*cos(ax*k)*sinh(bx) - bx^2*sinh(bx*(k -
1)) + bx^2*sinh(bx*k)*cos(ax))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N8k3 = -((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e -
1))) + ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(bx*(ax^3*cos(ax*k)*cosh(bx) - ax^3*cos(ax*(k - 1)) +
bx^3*sinh(bx*k)*sin(ax)) + ax*(bx^3*cosh(bx*(k - 1)) -
bx^3*cosh(bx*k)*cos(ax) +
ax^3*sin(ax*k)*sinh(bx))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N8e3 = -((by*(ay^3*cos(ay*e) - ay^3*cosh(by)*cos(ay*(e - 1)) +
by^3*sin(ay)*sinh(by*(e - 1))) + ay*(by^3*cos(ay)*cosh(by*(e - 1)) -
by^3*cosh(by*e) + ay^3*sinh(by)*sin(ay*(e - 1))))*(bx*(sin(ax*(k - 1)) +
cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) + ax*(sinh(bx*(k - 1)) +
cos(ax*k)*sinh(bx) - sinh(bx*k)*cos(ax))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N8k4 = -((by*(cosh(by)*sin(ay*(e - 1)) - sin(ay*e) + sin(ay)*cosh(by*(e -
1))) + ay*(cos(ay)*sinh(by*(e - 1)) - sinh(by*e) + sinh(by)*cos(ay*(e -
1))))*(bx*(ax^4*sin(ax*(k - 1)) - ax^4*sin(ax*k)*cosh(bx) +
bx^4*cosh(bx*k)*sin(ax)) + ax*(bx^4*sinh(bx*(k - 1)) +
ax^4*cos(ax*k)*sinh(bx) -
bx^4*sinh(bx*k)*cos(ax))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N8e4 = -((by*(ay^4*cosh(by)*sin(ay*(e - 1)) - ay^4*sin(ay*e) +
by^4*sin(ay)*cosh(by*(e - 1))) + ay*(ay^4*sinh(by)*cos(ay*(e - 1)) -
by^4*sinh(by*e) + by^4*cos(ay)*sinh(by*(e - 1))))*(bx*(sin(ax*(k - 1)) +
cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) + ax*(sinh(bx*(k - 1)) +
cos(ax*k)*sinh(bx) - sinh(bx*k)*cos(ax))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

% Node 9

```

N9k = -((cos(ay*(e - 1)) + cosh(by*(e - 1)) - cos(ay*e)*cosh(by) -
cosh(by*e)*cos(ay) - (ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay)*(ax*sin(ax*(k - 1)) - bx*sinh(bx*(k - 1)) +
ax*cosh(bx*k)*sin(ax) - ax*sin(ax*k)*cosh(bx) - bx*cos(ax*k)*sinh(bx) +
bx*sinh(bx*k)*cos(ax)))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

```

```

N9kk = -((cos(ay*(e - 1)) + cosh(by*(e - 1)) - cos(ay*e)*cosh(by) -
cosh(by*e)*cos(ay) - (ay*sinh(by*e)*sin(ay))/by +

```



```

N9k3 = -((cos(ay*(e - 1)) + cosh(by*(e - 1)) - cos(ay*e)*cosh(by) -
cosh(by*e)*cos(ay) - (ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay)*(ax^3*sin(ax*k)*cosh(bx) - bx^3*sinh(bx*(k - 1))
- ax^3*sin(ax*(k - 1)) + bx^3*sinh(bx*k)*cos(ax) + ax*bx^2*cosh(bx*k)*sin(ax)
+ ax^2*bx*cos(ax*k)*sinh(bx)))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2
- bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

```

```

N9e3 = -((ay^3*sin(ay*e)*cosh(by) - by^3*sinh(by*(e - 1)) - ay^3*sin(ay*(e -
1)) + by^3*sinh(by*e)*cos(ay) + ay*by^2*cosh(by*e)*sin(ay) +
ay^2*by*cos(ay*e)*sinh(by))*(cos(ax*(k - 1)) + cosh(bx*(k - 1)) -
cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) - (ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

```

```

N9k4 = ((ax^4*cos(ax*(k - 1)) + bx^4*cosh(bx*(k - 1)) -
ax^4*cos(ax*k)*cosh(bx) - bx^4*cosh(bx*k)*cos(ax) -
ax*bx^3*sinh(bx*k)*sin(ax) + ax^3*bx*sin(ax*k)*sinh(bx))*(cos(ay*(e - 1)) +
cosh(by*(e - 1)) - cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) -
(ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

```

```

N9e4 = ((ay^4*cos(ay*(e - 1)) + by^4*cosh(by*(e - 1)) -
ay^4*cos(ay*e)*cosh(by) - by^4*cosh(by*e)*cos(ay) -
ay*by^3*sinh(by*e)*sin(ay) + ay^3*by*sin(ay*e)*sinh(by))*(cos(ax*(k - 1)) +
cosh(bx*(k - 1)) - cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) -
(ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

```

% Node 10

```

N10k = ((bx*(ax*cos(ax*(k - 1)) - ax*cos(ax*k)*cosh(bx) +
bx*sinh(bx*k)*sin(ax)) - ax*(bx*cosh(bx*k)*cos(ax) - bx*cosh(bx*(k - 1)) +
ax*sin(ax*k)*sinh(bx)))*(cos(ay*(e - 1)) + cosh(by*(e - 1)) -
cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) - (ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N10kk = ((bx*(ax^2*sin(ax*k)*cosh(bx) - ax^2*sin(ax*(k - 1)) +
bx^2*cosh(bx*k)*sin(ax)) - ax*(ax^2*cos(ax*k)*sinh(bx) - bx^2*sinh(bx*(k -
1)) + bx^2*sinh(bx*k)*cos(ax)))*(cos(ay*(e - 1)) + cosh(by*(e - 1)) -
cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) - (ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N10e = -((bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) +
ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) -
sinh(bx*k)*cos(ax)))*(ay*sin(ay*(e - 1)) - by*sinh(by*(e - 1)) +

```

$$\frac{ay \cosh(by^*e) \sin(ay) - ay \sin(ay^*e) \cosh(by) - by \cos(ay^*e) \sinh(by) + by \sinh(by^*e) \cos(ay)}{(ax^*bx^*(2^*\cos(ax)^*\cosh(bx) + (\sin(ax)^*\sinh(bx)^*(ax^2 - bx^2)))/(ax^*bx) - 2)^*(2^*\cos(ay)^*\cosh(by) + (\sin(ay)^*\sinh(by)^*(ay^2 - by^2)))/(ay^*by) - 2)};$$

$$N10ee = -((bx^*(\sin(ax^*(k-1)) + \cosh(bx^*k)^*\sin(ax) - \sin(ax^*k)^*\cosh(bx)) + ax^*(\sinh(bx^*(k-1)) + \cos(ax^*k)^*\sinh(bx) - \sinh(bx^*k)^*\cos(ax)))*(ay^2*\cos(ay^*(e-1)) - by^2*\cosh(by^*(e-1)) - ay^2*\cos(ay^*e)^*\cosh(by) + by^2*\cosh(by^*e)^*\cos(ay) + ay^*by^*\sin(ay^*e)^*\sinh(by) + ay^*by^*\sinh(by^*e)^*\sin(ay)))/(ax^*bx^*(2^*\cos(ax)^*\cosh(bx) + (\sin(ax)^*\sinh(bx)^*(ax^2 - bx^2)))/(ax^*bx) - 2)^*(2^*\cos(ay)^*\cosh(by) + (\sin(ay)^*\sinh(by)^*(ay^2 - by^2)))/(ay^*by) - 2);$$

$$N10ke = -((bx^*(ax^*\cos(ax^*(k-1)) - ax^*\cos(ax^*k)^*\cosh(bx) + bx^*\sinh(bx^*k)^*\sin(ax)) - ax^*(bx^*\cosh(bx^*k)^*\cos(ax) - bx^*\cosh(bx^*(k-1)) + ax^*\sin(ax^*k)^*\sinh(bx)))*(ay^*\sin(ay^*(e-1)) - by^*\sinh(by^*(e-1)) + ay^*\cosh(by^*e)^*\sin(ay) - ay^*\sin(ay^*e)^*\cosh(by) - by^*\cos(ay^*e)^*\sinh(by) + by^*\sinh(by^*e)^*\cos(ay)))/(ax^*bx^*(2^*\cos(ax)^*\cosh(bx) + (\sin(ax)^*\sinh(bx)^*(ax^2 - bx^2)))/(ax^*bx) - 2)^*(2^*\cos(ay)^*\cosh(by) + (\sin(ay)^*\sinh(by)^*(ay^2 - by^2)))/(ay^*by) - 2);$$

$$N10k2e = -((bx^*(ax^2*\sin(ax^*k)^*\cosh(bx) - ax^2*\sin(ax^*(k-1)) + bx^2*\cosh(bx^*k)^*\sin(ax)) - ax^*(ax^2*\cos(ax^*k)^*\sinh(bx) - bx^2*\sinh(bx^*(k-1)) + bx^2*\sinh(bx^*k)^*\cos(ax)))*(ay^*\sin(ay^*(e-1)) - by^*\sinh(by^*(e-1)) + ay^*\cosh(by^*e)^*\sin(ay) - ay^*\sin(ay^*e)^*\cosh(by) - by^*\cos(ay^*e)^*\sinh(by) + by^*\sinh(by^*e)^*\cos(ay)))/(ax^*bx^*(2^*\cos(ax)^*\cosh(bx) + (\sin(ax)^*\sinh(bx)^*(ax^2 - bx^2)))/(ax^*bx) - 2)^*(2^*\cos(ay)^*\cosh(by) + (\sin(ay)^*\sinh(by)^*(ay^2 - by^2)))/(ay^*by) - 2);$$

$$N10ke2 = -((bx^*(ax^*\cos(ax^*(k-1)) - ax^*\cos(ax^*k)^*\cosh(bx) + bx^*\sinh(bx^*k)^*\sin(ax)) - ax^*(bx^*\cosh(bx^*k)^*\cos(ax) - bx^*\cosh(bx^*(k-1)) + ax^*\sin(ax^*k)^*\sinh(bx)))*(ay^2*\cos(ay^*(e-1)) - by^2*\cosh(by^*(e-1)) - ay^2*\cos(ay^*e)^*\cosh(by) + by^2*\cosh(by^*e)^*\cos(ay) + ay^*by^*\sin(ay^*e)^*\sinh(by) + ay^*by^*\sinh(by^*e)^*\sin(ay)))/(ax^*bx^*(2^*\cos(ax)^*\cosh(bx) + (\sin(ax)^*\sinh(bx)^*(ax^2 - bx^2)))/(ax^*bx) - 2)^*(2^*\cos(ay)^*\cosh(by) + (\sin(ay)^*\sinh(by)^*(ay^2 - by^2)))/(ay^*by) - 2);$$

$$N10k2e2 = -((bx^*(ax^2*\sin(ax^*k)^*\cosh(bx) - ax^2*\sin(ax^*(k-1)) + bx^2*\cosh(bx^*k)^*\sin(ax)) - ax^*(ax^2*\cos(ax^*k)^*\sinh(bx) - bx^2*\sinh(bx^*(k-1)) + bx^2*\sinh(bx^*k)^*\cos(ax)))*(ay^2*\cos(ay^*(e-1)) - by^2*\cosh(by^*(e-1)) - ay^2*\cos(ay^*e)^*\cosh(by) + by^2*\cosh(by^*e)^*\cos(ay) + ay^*by^*\sin(ay^*e)^*\sinh(by) + ay^*by^*\sinh(by^*e)^*\sin(ay)))/(ax^*bx^*(2^*\cos(ax)^*\cosh(bx) + (\sin(ax)^*\sinh(bx)^*(ax^2 - bx^2)))/(ax^*bx) - 2)^*(2^*\cos(ay)^*\cosh(by) + (\sin(ay)^*\sinh(by)^*(ay^2 - by^2)))/(ay^*by) - 2);$$

$$N10k3 = ((bx^*(ax^3*\cos(ax^*k)^*\cosh(bx) - ax^3*\cos(ax^*(k-1)) + bx^3*\sinh(bx^*k)^*\sin(ax)) + ax^*(bx^3*\cosh(bx^*(k-1)) - bx^3*\cosh(bx^*k)^*\cos(ax) + ax^3*\sin(ax^*k)^*\sinh(bx)))*(cos(ay^*(e-1)) + cosh(by^*(e-1)) - cos(ay^*e)^*\cosh(by) - cosh(by^*e)^*\cos(ay) - (ay^*\sinh(by^*e)^*\sin(ay))/by + (by^*\sin(ay^*e)^*\sinh(by))/ay))/(ax^*bx^*(2^*\cos(ax)^*\cosh(bx) + (\sin(ax)^*\sinh(bx)^*(ax^2 - bx^2)))/(ax^*bx) - 2)^*(2^*\cos(ay)^*\cosh(by) + (\sin(ay)^*\sinh(by)^*(ay^2 - by^2)))/(ay^*by) - 2);$$


```

N10e3 = -((bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) +
ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) -
sinh(bx*k)*cos(ax)))*(ay^3*sin(ay*e)*cosh(by) - by^3*sinh(by*(e - 1)) -
ay^3*sin(ay*(e - 1)) + by^3*sinh(by*e)*cos(ay) + ay*by^2*cosh(by*e)*sin(ay) +
ay^2*by*cos(ay*e)*sinh(by)))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N10k4 = ((bx*(ax^4*sin(ax*(k - 1)) - ax^4*sin(ax*k)*cosh(bx) +
bx^4*cosh(bx*k)*sin(ax)) + ax*(bx^4*sinh(bx*(k - 1)) +
ax^4*cos(ax*k)*sinh(bx) - bx^4*sinh(bx*k)*cos(ax)))*(cos(ay*(e - 1)) +
cosh(by*(e - 1)) - cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) -
(ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N10e4 = ((bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) +
ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) -
sinh(bx*k)*cos(ax)))*(ay^4*cos(ay*(e - 1)) + by^4*cosh(by*(e - 1)) -
ay^4*cos(ay*e)*cosh(by) - by^4*cosh(by*e)*cos(ay) -
ay*by^3*sinh(by*e)*sin(ay) +
ay^3*by*sin(ay*e)*sinh(by)))/(ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

% Node 11

```

N11k = -((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(ax*sin(ax*(k - 1)) - bx*sinh(bx*(k - 1)) +
ax*cosh(bx*k)*sin(ax) - ax*sin(ax*k)*cosh(bx) - bx*cos(ax*k)*sinh(bx) +
bx*sinh(bx*k)*cos(ax)))/(ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

```

```

N11kk = -((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(ax^2*cos(ax*(k - 1)) - bx^2*cosh(bx*(k - 1)) -
ax^2*cos(ax*k)*cosh(bx) + bx^2*cosh(bx*k)*cos(ax) + ax*bx*sin(ax*k)*sinh(bx)
+ ax*bx*sinh(bx*k)*sin(ax)))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N11e = ((by*(ay*cos(ay*(e - 1)) - ay*cos(ay*e)*cosh(by) +
by*sinh(by*e)*sin(ay)) - ay*(by*cosh(by*e)*cos(ay) - by*cosh(by*(e - 1)) +
ay*sin(ay*e)*sinh(by)))*(cos(ax*(k - 1)) + cosh(bx*(k - 1)) -
cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) - (ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N11ee = ((by*(ay^2*sin(ay*e)*cosh(by) - ay^2*sin(ay*(e - 1)) +
by^2*cosh(by*e)*sin(ay)) - ay*(ay^2*cos(ay*e)*sinh(by) - by^2*sinh(by*(e -
1)) + by^2*sinh(by*e)*cos(ay)))*(cos(ax*(k - 1)) + cosh(bx*(k - 1)) -

```

```

cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) - (ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax)/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N11ke = -((by*(ay*cos(ay*(e - 1)) - ay*cos(ay*e)*cosh(by) +
by*sinh(by*e)*sin(ay)) - ay*(by*cosh(by*e)*cos(ay) - by*cosh(by*(e - 1)) +
ay*sin(ay*e)*sinh(by)))*(ax*sin(ax*(k - 1)) - bx*sinh(bx*(k - 1)) +
ax*cosh(bx*k)*sin(ax) - ax*sin(ax*k)*cosh(bx) - bx*cos(ax*k)*sinh(bx) +
bx*sinh(bx*k)*cos(ax))/(ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2
- bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

N11k2e = -((by*(ay*cos(ay*(e - 1)) - ay*cos(ay*e)*cosh(by) +
by*sinh(by*e)*sin(ay)) - ay*(by*cosh(by*e)*cos(ay) - by*cosh(by*(e - 1)) +
ay*sin(ay*e)*sinh(by)))*(ax^2*cos(ax*(k - 1)) - bx^2*cosh(bx*(k - 1)) -
ax^2*cos(ax*k)*cosh(bx) + bx^2*cosh(bx*k)*cos(ax) + ax*bx*sin(ax*k)*sinh(bx)
+ ax*bx*sinh(bx*k)*sin(ax))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N11ke2 = -((by*(ay^2*sin(ay*e)*cosh(by) - ay^2*sin(ay*(e - 1)) +
by^2*cosh(by*e)*sin(ay)) - ay*(ay^2*cos(ay*e)*sinh(by) - by^2*sinh(by*(e -
1)) + by^2*sinh(by*e)*cos(ay)))*(ax*sin(ax*(k - 1)) - bx*sinh(bx*(k - 1)) +
ax*cosh(bx*k)*sin(ax) - ax*sin(ax*k)*cosh(bx) - bx*cos(ax*k)*sinh(bx) +
bx*sinh(bx*k)*cos(ax))/(ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2
- bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

N11k2e2 = -((by*(ay^2*sin(ay*e)*cosh(by) - ay^2*sin(ay*(e - 1)) +
by^2*cosh(by*e)*sin(ay)) - ay*(ay^2*cos(ay*e)*sinh(by) - by^2*sinh(by*(e -
1)) + by^2*sinh(by*e)*cos(ay)))*(ax^2*cos(ax*(k - 1)) - bx^2*cosh(bx*(k - 1))
- ax^2*cos(ax*k)*cosh(bx) + bx^2*cosh(bx*k)*cos(ax) +
ax*bx*sin(ax*k)*sinh(bx) +
ax*bx*sinh(bx*k)*sin(ax))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N11k3 = -((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(ax^3*sin(ax*k)*cosh(bx) - bx^3*sinh(bx*(k - 1)) -
ax^3*sin(ax*(k - 1)) + bx^3*sinh(bx*k)*cos(ax) + ax*bx^2*cosh(bx*k)*sin(ax) +
ax^2*bx*cos(ax*k)*sinh(bx))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N11e3 = ((by*(ay^3*cos(ay*e)*cosh(by) - ay^3*cos(ay*(e - 1)) +
by^3*sinh(by*e)*sin(ay)) + ay*(by^3*cosh(by*(e - 1)) -
by^3*cosh(by*e)*cos(ay) + ay^3*sin(ay*e)*sinh(by)))*(cos(ax*(k - 1)) +
cosh(bx*(k - 1)) - cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) -
(ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax)/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N11k4 = ((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(ax^4*cos(ax*(k - 1)) + bx^4*cosh(bx*(k - 1)) -
ax^4*cos(ax*k)*cosh(bx) - bx^4*cosh(bx*k)*cos(ax) -
ax*bx^3*sinh(bx*k)*sin(ax) +
ax^3*bx*sin(ax*k)*sinh(bx)))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N11e4 = ((by*(ay^4*sin(ay*(e - 1)) - ay^4*sin(ay*e)*cosh(by) +
by^4*cosh(by*e)*sin(ay)) + ay*(by^4*sinh(by*(e - 1)) +
ay^4*cos(ay*e)*sinh(by) - by^4*sinh(by*e)*cos(ay)))*(cos(ax*(k - 1)) +
cosh(bx*(k - 1)) - cos(ax*k)*cosh(bx) - cosh(bx*k)*cos(ax) -
(ax*sinh(bx*k)*sin(ax))/bx +
(bx*sin(ax*k)*sinh(bx))/ax))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

% Node 12

```

N12k = ((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(bx*(ax*cos(ax*(k - 1)) - ax*cos(ax*k)*cosh(bx) +
bx*sinh(bx*k)*sin(ax)) - ax*(bx*cosh(bx*k)*cos(ax) - bx*cosh(bx*(k - 1)) +
ax*sin(ax*k)*sinh(bx)))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N12kk = ((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(bx*(ax^2*sin(ax*k)*cosh(bx) - ax^2*sin(ax*(k - 1)) +
bx^2*cosh(bx*k)*sin(ax)) - ax*(ax^2*cos(ax*k)*sinh(bx) - bx^2*sinh(bx*(k -
1)) + bx^2*sinh(bx*k)*cos(ax)))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N12e = ((by*(ay*cos(ay*(e - 1)) - ay*cos(ay*e)*cosh(by) +
by*sinh(by*e)*sin(ay)) - ay*(by*cosh(by*e)*cos(ay) - by*cosh(by*(e - 1)) +
ay*sin(ay*e)*sinh(by)))*(bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) -
sin(ax*k)*cosh(bx)) + ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) -
sinh(bx*k)*cos(ax)))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N12ee = ((bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) +
ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) -
sinh(bx*k)*cos(ax)))*(by*(ay^2*sin(ay*e)*cosh(by) - ay^2*sin(ay*(e - 1)) +
by^2*cosh(by*e)*sin(ay)) - ay*(ay^2*cos(ay*e)*sinh(by) - by^2*sinh(by*(e -
1)) + by^2*sinh(by*e)*cos(ay)))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

```

```

N12ke = ((by*(ay*cos(ay*(e - 1)) - ay*cos(ay*e)*cosh(by) +
by*sinh(by*e)*sin(ay)) - ay*(by*cosh(by*e)*cos(ay) - by*cosh(by*(e - 1)) +
ay*sin(ay*e)*sinh(by)))*(bx*(ax*cos(ax*(k - 1)) - ax*cos(ax*k)*cosh(bx) +

```

```

bx*sinh(bx*k)*sin(ax)) - ax*(bx*cosh(bx*k)*cos(ax) - bx*cosh(bx*(k - 1)) +
ax*sin(ax*k)*sinh(bx))) / (ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));

N12k2e = ((by*(ay*cos(ay*(e - 1)) - ay*cos(ay*e)*cosh(by) +
by*sinh(by*e)*sin(ay)) - ay*(by*cosh(by*e)*cos(ay) - by*cosh(by*(e - 1)) +
ay*sin(ay*e)*sinh(by))) * (bx*(ax^2*sin(ax*k)*cosh(bx) - ax^2*sin(ax*(k - 1)) +
bx^2*cosh(bx*k)*sin(ax)) - ax*(ax^2*cos(ax*k)*sinh(bx) - bx^2*sinh(bx*(k -
1)) + bx^2*sinh(bx*k)*cos(ax))) / (ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));

N12ke2 = ((bx*(ax*cos(ax*(k - 1)) - ax*cos(ax*k)*cosh(bx) +
bx*sinh(bx*k)*sin(ax)) - ax*(bx*cosh(bx*k)*cos(ax) - bx*cosh(bx*(k - 1)) +
ax*sin(ax*k)*sinh(bx))) * (by*(ay^2*sin(ay*e)*cosh(by) - ay^2*sin(ay*(e - 1)) +
by^2*cosh(by*e)*sin(ay)) - ay*(ay^2*cos(ay*e)*sinh(by) - by^2*sinh(by*(e -
1)) + by^2*sinh(by*e)*cos(ay))) / (ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));

N12k2e2 = ((by*(ay^2*sin(ay*e)*cosh(by) - ay^2*sin(ay*(e - 1)) +
by^2*cosh(by*e)*sin(ay)) - ay*(ay^2*cos(ay*e)*sinh(by) - by^2*sinh(by*(e -
1)) + by^2*sinh(by*e)*cos(ay))) * (bx*(ax^2*sin(ax*k)*cosh(bx) - ax^2*sin(ax*(k -
1)) + bx^2*cosh(bx*k)*sin(ax)) - ax*(ax^2*cos(ax*k)*sinh(bx) -
bx^2*sinh(bx*(k - 1)) +
bx^2*sinh(bx*k)*cos(ax))) / (ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));

N12k3 = ((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay))) * (bx*(ax^3*cos(ax*k)*cosh(bx) - ax^3*cos(ax*(k - 1)) +
bx^3*sinh(bx*k)*sin(ax)) + ax*(bx^3*cosh(bx*(k - 1)) -
bx^3*cosh(bx*k)*cos(ax) +
ax^3*sin(ax*k)*sinh(bx))) / (ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));

N12e3 = ((bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) +
ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) -
sinh(bx*k)*cos(ax))) * (by*(ay^3*cos(ay*e)*cosh(by) - ay^3*cos(ay*(e - 1)) +
by^3*sinh(by*e)*sin(ay)) + ay*(by^3*cosh(by*(e - 1)) -
by^3*cosh(by*e)*cos(ay) +
ay^3*sin(ay*e)*sinh(by))) / (ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2)) / (ay*by) - 2));

N12k4 = ((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay))) * (bx*(ax^4*sin(ax*(k - 1)) - ax^4*sin(ax*k)*cosh(bx) +
bx^4*cosh(bx*k)*sin(ax)) + ax*(bx^4*sinh(bx*(k - 1)) +
ax^4*cos(ax*k)*sinh(bx) -
bx^4*sinh(bx*k)*cos(ax))) / (ax*ay*bx*by*(2*cos(ax)*cosh(bx) +

```

```
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by - 2));
```

```
N12e4 = ((bx*(sin(ax*(k - 1)) + cosh(bx*k)*sin(ax) - sin(ax*k)*cosh(bx)) +
ax*(sinh(bx*(k - 1)) + cos(ax*k)*sinh(bx) -
sinh(bx*k)*cos(ax)))*(by*(ay^4*sin(ay*(e - 1)) - ay^4*sin(ay*e)*cosh(by) +
by^4*cosh(by*e)*sin(ay)) + ay*(by^4*sinh(by*(e - 1)) +
ay^4*cos(ay*e)*sinh(by) -
by^4*sinh(by*e)*cos(ay)))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by - 2)));
```

% Node 13

```
N13k = -((ax*sin(ax*k) - bx*sinh(bx*k) - ax*cosh(bx)*sin(ax*(k - 1)) -
ax*sin(ax)*cosh(bx*(k - 1)) + bx*cos(ax)*sinh(bx*(k - 1)) +
bx*sinh(bx)*cos(ax*(k - 1)))*(cos(ay*(e - 1)) + cosh(by*(e - 1)) -
cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) - (ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by - 2)));
```

```
N13kk = ((bx^2*cosh(bx*k) - ax^2*cos(ax*k) + ax^2*cosh(bx)*cos(ax*(k - 1)) -
bx^2*cos(ax)*cosh(bx*(k - 1)) + ax*bx*sin(ax)*sinh(bx*(k - 1)) +
ax*bx*sinh(bx)*sin(ax*(k - 1)))*(cos(ay*(e - 1)) + cosh(by*(e - 1)) -
cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) - (ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by - 2)));
```

```
N13e = -((cos(ax*k) + cosh(bx*k) - cos(ax)*cosh(bx*(k - 1)) -
cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k - 1)))/bx -
(bx*sinh(bx)*sin(ax*(k - 1)))/ax)*(ay*sin(ay*(e - 1)) - by*sinh(by*(e - 1)) +
ay*cosh(by*e)*sin(ay) - ay*sin(ay*e)*cosh(by) - by*cos(ay*e)*sinh(by) +
by*sinh(by*e)*cos(ay))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by - 2)));
```

```
N13ee = -((cos(ax*k) + cosh(bx*k) - cos(ax)*cosh(bx*(k - 1)) -
cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k - 1)))/bx -
(bx*sinh(bx)*sin(ax*(k - 1)))/ax)*(ay^2*cos(ay*(e - 1)) - by^2*cosh(by*(e -
1)) - ay^2*cos(ay*e)*cosh(by) + by^2*cosh(by*e)*cos(ay) +
ay*by*sin(ay*e)*sinh(by) + ay*by*sinh(by*e)*sin(ay))/(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by - 2)));
```

```
N13ke = ((ax*sin(ax*k) - bx*sinh(bx*k) - ax*cosh(bx)*sin(ax*(k - 1)) -
ax*sin(ax)*cosh(bx*(k - 1)) + bx*cos(ax)*sinh(bx*(k - 1)) +
bx*sinh(bx)*cos(ax*(k - 1)))*(ay*sin(ay*(e - 1)) - by*sinh(by*(e - 1)) +
ay*cosh(by*e)*sin(ay) - ay*sin(ay*e)*cosh(by) - by*cos(ay*e)*sinh(by) +
by*sinh(by*e)*cos(ay))/(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by - 2)));
```

$$\begin{aligned} N13k2e = & -((ay*\sin(ay*(e-1)) - by*\sinh(by*(e-1)) + ay*\cosh(by*e)*\sin(ay) \\ & - ay*\sin(ay*e)*\cosh(by) - by*\cos(ay*e)*\sinh(by) + \\ & by*\sinh(by*e)*\cos(ay))*(bx^2*\cosh(bx*k) - ax^2*\cos(ax*k) + \\ & ax^2*\cosh(bx)*\cos(ax*(k-1)) - bx^2*\cos(ax)*\cosh(bx*(k-1)) + \\ & ax*bx*\sin(ax)*\sinh(bx*(k-1)) + ax*bx*\sinh(bx)*\sin(ax*(k-1))) / ((2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - \\ & 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N13ke2 = & ((ax*\sin(ax*k) - bx*\sinh(bx*k) - ax*\cosh(bx)*\sin(ax*(k-1)) - \\ & ax*\sin(ax)*\cosh(bx*(k-1)) + bx*\cos(ax)*\sinh(bx*(k-1)) + \\ & bx*\sinh(bx)*\cos(ax*(k-1)))*(ay^2*\cos(ay*(e-1)) - by^2*\cosh(by*(e-1)) - \\ & ay^2*\cos(ay*e)*\cosh(by) + by^2*\cosh(by*e)*\cos(ay) + ay*by*\sin(ay*e)*\sinh(by) \\ & + ay*by*\sinh(by*e)*\sin(ay)) / ((2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N13k2e2 = & -((bx^2*\cosh(bx*k) - ax^2*\cos(ax*k) + ax^2*\cosh(bx)*\cos(ax*(k-1)) - \\ & bx^2*\cos(ax)*\cosh(bx*(k-1)) + ax*bx*\sin(ax)*\sinh(bx*(k-1)) + \\ & ax*bx*\sinh(bx)*\sin(ax*(k-1)))*(ay^2*\cos(ay*(e-1)) - by^2*\cosh(by*(e-1)) - \\ & ay^2*\cos(ay*e)*\cosh(by) + by^2*\cosh(by*e)*\cos(ay) + \\ & ay*by*\sin(ay*e)*\sinh(by) + ay*by*\sinh(by*e)*\sin(ay)) / ((2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N13k3 = & ((ax^3*\sin(ax*k) + bx^3*\sinh(bx*k) - ax^3*\cosh(bx)*\sin(ax*(k-1)) - \\ & bx^3*\cos(ax)*\sinh(bx*(k-1)) + ax*bx^2*\sin(ax)*\cosh(bx*(k-1)) + \\ & ax^2*bx*\sinh(bx)*\cos(ax*(k-1)))*(cos(ay*(e-1)) + cosh(by*(e-1)) - \\ & cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) - (ay*\sinh(by*e)*\sin(ay))/by + \\ & (by*\sin(ay*e)*\sinh(by))/ay) / ((2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N13e3 = & -((cos(ax*k) + cosh(bx*k) - cos(ax)*cosh(bx*(k-1)) - \\ & cosh(bx)*cos(ax*(k-1)) + (ax*\sin(ax)*\sinh(bx*(k-1)))/bx - \\ & (bx*\sinh(bx)*\sin(ax*(k-1)))/ax)*(ay^3*\sin(ay*e)*cosh(by) - by^3*\sinh(by*(e-1)) - \\ & ay^3*\sin(ay*(e-1)) + by^3*\sinh(by*e)*cos(ay) + \\ & ay*by^2*\cosh(by*e)*\sin(ay) + \\ & ay^2*by*\cos(ay*e)*\sinh(by)) / ((2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N13k4 = & ((ax^4*\cos(ax*k) + bx^4*\cosh(bx*k) - ax^4*\cosh(bx)*\cos(ax*(k-1)) - \\ & bx^4*\cos(ax)*\cosh(bx*(k-1)) + ax*bx^3*\sin(ax)*\sinh(bx*(k-1)) - \\ & ax^3*bx*\sinh(bx)*\sin(ax*(k-1)))*(cos(ay*(e-1)) + cosh(by*(e-1)) - \\ & cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) - (ay*\sinh(by*e)*\sin(ay))/by + \\ & (by*\sin(ay*e)*\sinh(by))/ay) / ((2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N13e4 = & ((ay^4*\cos(ay*(e-1)) + by^4*\cosh(by*(e-1)) - \\ & ay^4*\cos(ay*e)*\cosh(by) - by^4*\cosh(by*e)*\cos(ay) - \\ & ay*by^3*\sinh(by*e)*\sin(ay) + ay^3*by*\sin(ay*e)*\sinh(by))*(cos(ax*k) + \\ & cosh(bx*k) - cos(ax)*cosh(bx*(k-1)) - cosh(bx)*cos(ax*(k-1)) + \\ & (ax*\sin(ax)*\sinh(bx*(k-1)))/bx - (bx*\sinh(bx)*\sin(ax*(k-1)))/ax); \end{aligned}$$

```
1))))/ax))/((2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
% Node 14
```

```
N14k = -(bx*(ax*cosh(bx)*cos(ax*(k - 1)) - ax*cos(ax*k) +
bx*sin(ax)*sinh(bx*(k - 1))) - ax*(bx*cosh(bx*k) - bx*cos(ax)*cosh(bx*(k -
1)) + ax*sinh(bx)*sin(ax*(k - 1))))*(cos(ay*(e - 1)) + cosh(by*(e - 1)) -
cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) - (ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay))/ (ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N14kk = -(bx*(ax^2*sin(ax*k) - ax^2*cosh(bx)*sin(ax*(k - 1)) +
bx^2*sin(ax)*cosh(bx*(k - 1))) - ax*(bx^2*sinh(bx*k) +
ax^2*sinh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*sinh(bx*(k - 1))))*(cos(ay*(e -
1)) + cosh(by*(e - 1)) - cos(ay*e)*cosh(by) - cosh(by*e)*cos(ay) -
(ay*sinh(by*e)*sin(ay))/by +
(by*sin(ay*e)*sinh(by))/ay))/ (ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N14e = (bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k - 1)))
+ ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))))*(ay*sin(ay*(e - 1)) - by*sinh(by*(e - 1)) + ay*cosh(by*e)*sin(ay) -
ay*sin(ay*e)*cosh(by) - by*cos(ay*e)*sinh(by) +
by*sinh(by*e)*cos(ay))/ (ax*bx*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2
- bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));
```

```
N14ee = (bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k -
1))) + ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))))*(ay^2*cos(ay*(e - 1)) - by^2*cosh(by*(e - 1)) - ay^2*cos(ay*e)*cosh(by)
+ by^2*cosh(by*e)*cos(ay) + ay*by*sin(ay*e)*sinh(by) +
ay*by*sinh(by*e)*sin(ay))/ (ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N14ke = (bx*(ax*cosh(bx)*cos(ax*(k - 1)) - ax*cos(ax*k) +
bx*sin(ax)*sinh(bx*(k - 1))) - ax*(bx*cosh(bx*k) - bx*cos(ax)*cosh(bx*(k -
1)) + ax*sinh(bx)*sin(ax*(k - 1))))*(ay*sin(ay*(e - 1)) - by*sinh(by*(e - 1))
+ ay*cosh(by*e)*sin(ay) - ay*sin(ay*e)*cosh(by) - by*cos(ay*e)*sinh(by) +
by*sinh(by*e)*cos(ay))/ (ax*bx*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2
- bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));
```

```
N14k2e = (bx*(ax^2*sin(ax*k) - ax^2*cosh(bx)*sin(ax*(k - 1)) +
bx^2*sin(ax)*cosh(bx*(k - 1))) - ax*(bx^2*sinh(bx*k) +
ax^2*sinh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*sinh(bx*(k - 1))))*(ay*sin(ay*(e
- 1)) - by*sinh(by*(e - 1)) + ay*cosh(by*e)*sin(ay) - ay*sin(ay*e)*cosh(by) -
by*cos(ay*e)*sinh(by) + by*sinh(by*e)*cos(ay))/ (ax*bx*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

$$\begin{aligned} N14ke2 = & ((bx*(ax*\cosh(bx)*\cos(ax*(k-1)) - ax*\cos(ax*k) + \\ & bx*\sin(ax)*\sinh(bx*(k-1))) - ax*(bx*\cosh(bx*k) - bx*\cos(ax)*\cosh(bx*(k-1)) + \\ & ax*\sinh(bx)*\sin(ax*(k-1))))*(ay^2*\cos(ay*(e-1)) - by^2*\cosh(by*(e-1)) - \\ & ay^2*\cos(ay*e)*\cosh(by) + by^2*\cosh(by*e)*\cos(ay) + \\ & ay*by*\sin(ay*e)*\sinh(by) + \\ & ay*by*\sinh(by*e)*\sin(ay)) / (ax*bx*(2*\cos(ax)*\cosh(bx) + \\ & (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + \\ & (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N14k2e2 = & ((bx*(ax^2*\sin(ax*k) - ax^2*\cosh(bx)*\sin(ax*(k-1)) + \\ & bx^2*\sin(ax)*\cosh(bx*(k-1))) - ax*(bx^2*\sinh(bx*k) + \\ & ax^2*\sinh(bx)*\cos(ax*(k-1)) - bx^2*\cos(ax)*\sinh(bx*(k-1))))*(ay^2*\cos(ay*(e-1)) - \\ & by^2*\cosh(by*(e-1)) - ay^2*\cos(ay*e)*\cosh(by) + by^2*\cosh(by*e)*\cos(ay) + \\ & ay*by*\sin(ay*e)*\sinh(by) + \\ & ay*by*\sinh(by*e)*\sin(ay)) / (ax*bx*(2*\cos(ax)*\cosh(bx) + \\ & (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + \\ & (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N14k3 = & -((bx*(ax^3*\cos(ax*k) - ax^3*\cosh(bx)*\cos(ax*(k-1)) + \\ & bx^3*\sin(ax)*\sinh(bx*(k-1))) + ax*(bx^3*\cos(ax)*\cosh(bx*(k-1)) - \\ & bx^3*\cosh(bx*k) + ax^3*\sinh(bx)*\sin(ax*(k-1))))*(\cos(ay*(e-1)) + \\ & \cosh(by*(e-1)) - \cos(ay*e)*\cosh(by) - \cosh(by*e)*\cos(ay) - \\ & (ay*\sinh(by*e)*\sin(ay))/by + \\ & (by*\sin(ay*e)*\sinh(by))/ay) / (ax*bx*(2*\cos(ax)*\cosh(bx) + \\ & (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + \\ & (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N14e3 = & ((bx*(\cosh(bx)*\sin(ax*(k-1)) - \sin(ax*k) + \sin(ax)*\cosh(bx*(k-1))) + \\ & ax*(\cos(ax)*\sinh(bx*(k-1)) - \sinh(bx*k) + \sinh(bx)*\cos(ax*(k-1))))*(ay^3*\sin(ay*e)*\cosh(by) - \\ & by^3*\sinh(by*(e-1)) - ay^3*\sin(ay*(e-1)) + by^3*\sinh(by*e)*\cos(ay) + \\ & ay*by^2*\cosh(by*e)*\sin(ay) + ay^2*by*\cos(ay*e)*\sinh(by)) / (ax*bx*(2*\cos(ax)*\cosh(bx) + \\ & (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + \\ & (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N14k4 = & -((bx*(ax^4*\cosh(bx)*\sin(ax*(k-1)) - ax^4*\sin(ax*k) + \\ & bx^4*\sin(ax)*\cosh(bx*(k-1))) + ax*(ax^4*\sinh(bx)*\cos(ax*(k-1)) - \\ & bx^4*\sinh(bx*k) + bx^4*\cos(ax)*\sinh(bx*(k-1))))*(\cos(ay*(e-1)) + \\ & \cosh(by*(e-1)) - \cos(ay*e)*\cosh(by) - \cosh(by*e)*\cos(ay) - \\ & (ay*\sinh(by*e)*\sin(ay))/by + \\ & (by*\sin(ay*e)*\sinh(by))/ay) / (ax*bx*(2*\cos(ax)*\cosh(bx) + \\ & (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + \\ & (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N14e4 = & -((bx*(\cosh(bx)*\sin(ax*(k-1)) - \sin(ax*k) + \sin(ax)*\cosh(bx*(k-1))) + \\ & ax*(\cos(ax)*\sinh(bx*(k-1)) - \sinh(bx*k) + \sinh(bx)*\cos(ax*(k-1))))*(ay^4*\cos(ay*(e-1)) + \\ & by^4*\cosh(by*(e-1)) - ay^4*\cos(ay*e)*\cosh(by) - by^4*\cosh(by*e)*\cos(ay) - \\ & ay*by^3*\sinh(by*e)*\sin(ay) + ay^3*by*\sin(ay*e)*\sinh(by)) / (ax*bx*(2*\cos(ax)*\cosh(bx) + \\ & (\sin(ax)*\sinh(bx)*(ax^2 - bx^2)) / (ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + \\ & (\sin(ay)*\sinh(by)*(ay^2 - by^2)) / (ay*by) - 2)); \end{aligned}$$

% Node 15

```

N15k = -((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(ax*sin(ax*k) - bx*sinh(bx*k) - ax*cosh(bx)*sin(ax*(k -
1)) - ax*sin(ax)*cosh(bx*(k - 1)) + bx*cos(ax)*sinh(bx*(k - 1)) +
bx*sinh(bx)*cos(ax*(k - 1))))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N15kk = ((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(bx^2*cosh(bx*k) - ax^2*cos(ax*k) +
ax^2*cosh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*cosh(bx*(k - 1)) +
ax*bx*sin(ax)*sinh(bx*(k - 1)) + ax*bx*sinh(bx)*sin(ax*(k -
1))))/(ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N15e = ((by*(ay*cos(ay*(e - 1)) - ay*cos(ay*e)*cosh(by) +
by*sinh(by*e)*sin(ay)) - ay*(by*cosh(by*e)*cos(ay) - by*cosh(by*(e - 1)) +
ay*sin(ay*e)*sinh(by)))*(cos(ax*k) + cosh(bx*k) - cos(ax)*cosh(bx*(k - 1)) -
cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k - 1)))/bx -
(bx*sinh(bx)*sin(ax*(k - 1)))/ax))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N15ee = ((by*(ay^2*sin(ay*e)*cosh(by) - ay^2*sin(ay*(e - 1)) +
by^2*cosh(by*e)*sin(ay)) - ay*(ay^2*cos(ay*e)*sinh(by) - by^2*sinh(by*(e -
1)) + by^2*sinh(by*e)*cos(ay)))*(cos(ax*k) + cosh(bx*k) - cos(ax)*cosh(bx*(k
- 1)) - cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k - 1)))/bx -
(bx*sinh(bx)*sin(ax*(k - 1)))/ax))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N15ke = -((by*(ay*cos(ay*(e - 1)) - ay*cos(ay*e)*cosh(by) +
by*sinh(by*e)*sin(ay)) - ay*(by*cosh(by*e)*cos(ay) - by*cosh(by*(e - 1)) +
ay*sin(ay*e)*sinh(by)))*(ax*sin(ax*k) - bx*sinh(bx*k) - ax*cosh(bx)*sin(ax*(k
- 1)) - ax*sin(ax)*cosh(bx*(k - 1)) + bx*cos(ax)*sinh(bx*(k - 1)) +
bx*sinh(bx)*cos(ax*(k - 1))))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N15k2e = ((by*(ay*cos(ay*(e - 1)) - ay*cos(ay*e)*cosh(by) +
by*sinh(by*e)*sin(ay)) - ay*(by*cosh(by*e)*cos(ay) - by*cosh(by*(e - 1)) +
ay*sin(ay*e)*sinh(by)))*(bx^2*cosh(bx*k) - ax^2*cos(ax*k) +
ax^2*cosh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*cosh(bx*(k - 1)) +
ax*bx*sin(ax)*sinh(bx*(k - 1)) + ax*bx*sinh(bx)*sin(ax*(k -
1))))/(ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N15ke2 = -((by*(ay^2*sin(ay*e)*cosh(by) - ay^2*sin(ay*(e - 1)) +
by^2*cosh(by*e)*sin(ay)) - ay*(ay^2*cos(ay*e)*sinh(by) - by^2*sinh(by*(e -
1)) + by^2*sinh(by*e)*cos(ay)))*(ax*sin(ax*k) - bx*sinh(bx*k) -
ax*cosh(bx)*sin(ax*(k - 1)) - ax*sin(ax)*cosh(bx*(k - 1)) +
bx*cos(ax)*sinh(bx*(k - 1)) + bx*sinh(bx)*cos(ax*(k -

```

```
1))))/(ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N15k2e2 = ((by*(ay^2*sin(ay*e)*cosh(by) - ay^2*sin(ay*(e - 1)) +
by^2*cosh(by*e)*sin(ay)) - ay*(ay^2*cos(ay*e)*sinh(by) - by^2*sinh(by*(e -
1)) + by^2*sinh(by*e)*cos(ay)))*(bx^2*cosh(bx*k) - ax^2*cos(ax*k) +
ax^2*cosh(bx)*cos(ax*(k - 1)) - bx^2*cos(ax)*cosh(bx*(k - 1)) +
ax*bx*sin(ax)*sinh(bx*(k - 1)) + ax*bx*sinh(bx)*sin(ax*(k -
1))))/(ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N15k3 = ((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(ax^3*sin(ax*k) + bx^3*sinh(bx*k) -
ax^3*cosh(bx)*sin(ax*(k - 1)) - bx^3*cos(ax)*sinh(bx*(k - 1)) +
ax*bx^2*sin(ax)*cosh(bx*(k - 1)) + ax^2*bx*sinh(bx)*cos(ax*(k -
1))))/(ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N15e3 = ((by*(ay^3*cos(ay*e)*cosh(by) - ay^3*cos(ay*(e - 1)) +
by^3*sinh(by*e)*sin(ay)) + ay*(by^3*cosh(by*(e - 1)) -
by^3*cosh(by*e)*cos(ay) + ay^3*sin(ay*e)*sinh(by)))*(cos(ax*k) + cosh(bx*k) -
cos(ax)*cosh(bx*(k - 1)) - cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k
- 1)))/bx - (bx*sinh(bx)*sin(ax*(k - 1)))/ax))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N15k4 = ((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(ax^4*cos(ax*k) + bx^4*cosh(bx*k) -
ax^4*cosh(bx)*cos(ax*(k - 1)) - bx^4*cos(ax)*cosh(bx*(k - 1)) +
ax*bx^3*sin(ax)*sinh(bx*(k - 1)) - ax^3*bx*sinh(bx)*sin(ax*(k -
1))))/(ay*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) -
2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N15e4 = ((by*(ay^4*sin(ay*(e - 1)) - ay^4*sin(ay*e)*cosh(by) +
by^4*cosh(by*e)*sin(ay)) + ay*(by^4*sinh(by*(e - 1)) +
ay^4*cos(ay*e)*sinh(by) - by^4*sinh(by*e)*cos(ay)))*(cos(ax*k) + cosh(bx*k) -
cos(ax)*cosh(bx*(k - 1)) - cosh(bx)*cos(ax*(k - 1)) + (ax*sin(ax)*sinh(bx*(k
- 1)))/bx - (bx*sinh(bx)*sin(ax*(k - 1)))/ax))/(ay*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

% Node 16

```
N16k = -((bx*(ax*cosh(bx)*cos(ax*(k - 1)) - ax*cos(ax*k) +
bx*sin(ax)*sinh(bx*(k - 1))) - ax*(bx*cosh(bx*k) - bx*cos(ax)*cosh(bx*(k -
1)) + ax*sinh(bx)*sin(ax*(k - 1))))*(by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay)
- sin(ay*e)*cosh(by)) + ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));
```

```
N16kk = -((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
```

$$\begin{aligned} & \sinh(by*e)*\cos(ay)))*(bx*(ax^2*\sin(ax*k) - ax^2*\cosh(bx)*\sin(ax*(k-1)) + \\ & bx^2*\sin(ax)*\cosh(bx*(k-1))) - ax*(bx^2*\sinh(bx*k) + \\ & ax^2*\sinh(bx)*\cos(ax*(k-1)) - bx^2*\cos(ax)*\sinh(bx*(k-1))))/(ax*ay*bx*by*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - \\ & bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - \\ & by^2))/(ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N16e = & -((bx*(\cosh(bx)*\sin(ax*(k-1)) - \sin(ax*k) + \sin(ax)*\cosh(bx*(k-1))) + ax*(\cos(ax)*\sinh(bx*(k-1)) - \sinh(bx*k) + \sinh(bx)*\cos(ax*(k-1))))* \\ & (by*(ay*\cos(ay*(e-1)) - ay*\cos(ay*e)*\cosh(by) + by*\sinh(by*e)*\sin(ay)) - ay*(by*\cosh(by*e)*\cos(ay) - by*\cosh(by*(e-1)) + \\ & ay*\sin(ay*e)*\sinh(by)))/(ax*ay*bx*by*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - \\ & bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N16ee = & -((bx*(\cosh(bx)*\sin(ax*(k-1)) - \sin(ax*k) + \sin(ax)*\cosh(bx*(k-1))) + ax*(\cos(ax)*\sinh(bx*(k-1)) - \sinh(bx*k) + \sinh(bx)*\cos(ax*(k-1))))* \\ & (by*(ay^2*\sin(ay*e)*\cosh(by) - ay^2*\sin(ay*(e-1)) + by^2*\cosh(by*e)*\sin(ay)) - ay*(ay^2*\cos(ay*e)*\sinh(by) - by^2*\sinh(by*(e-1)) + \\ & by^2*\sinh(by*e)*\cos(ay)))/(ax*ay*bx*by*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - \\ & bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N16ke = & -((bx*(ax*\cosh(bx)*\cos(ax*(k-1)) - ax*\cos(ax*k) + bx*\sin(ax)*\sinh(bx*(k-1))) - ax*(bx*\cosh(bx*k) - bx*\cos(ax)*\cosh(bx*(k-1)) + \\ & ax*\sinh(bx)*\sin(ax*(k-1))))*(by*(ay*\cos(ay*(e-1)) - ay*\cos(ay*e)*\cosh(by) + by*\sinh(by*e)*\sin(ay)) - ay*(by*\cosh(by*e)*\cos(ay) - by*\cosh(by*(e-1)) + \\ & ay*\sin(ay*e)*\sinh(by)))/(ax*ay*bx*by*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - \\ & bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N16k2e = & -((by*(ay*\cos(ay*(e-1)) - ay*\cos(ay*e)*\cosh(by) + by*\sinh(by*e)*\sin(ay)) - ay*(by*\cosh(by*e)*\cos(ay) - by*\cosh(by*(e-1)) + \\ & ay*\sin(ay*e)*\sinh(by)))*(bx*(ax^2*\sin(ax*k) - ax^2*\cosh(bx)*\sin(ax*(k-1)) + bx^2*\sin(ax)*\cosh(bx*(k-1))) - ax*(bx^2*\sinh(bx*k) + \\ & ax^2*\sinh(bx)*\cos(ax*(k-1)) - bx^2*\cos(ax)*\sinh(bx*(k-1))))/(ax*ay*bx*by*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - \\ & bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N16ke2 = & -((bx*(ax*\cosh(bx)*\cos(ax*(k-1)) - ax*\cos(ax*k) + bx*\sin(ax)*\sinh(bx*(k-1))) - ax*(bx*\cosh(bx*k) - bx*\cos(ax)*\cosh(bx*(k-1)) + \\ & ax*\sinh(bx)*\sin(ax*(k-1))))*(by*(ay^2*\sin(ay*e)*\cosh(by) - ay^2*\sin(ay*(e-1)) + by^2*\cosh(by*e)*\sin(ay)) - ay*(ay^2*\cos(ay*e)*\sinh(by) - by^2*\sinh(by*(e-1)) + \\ & by^2*\sinh(by*e)*\cos(ay)))/(ax*ay*bx*by*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - \\ & bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2)); \end{aligned}$$

$$\begin{aligned} N16k2e2 = & -((bx*(ax^2*\sin(ax*k) - ax^2*\cosh(bx)*\sin(ax*(k-1)) + bx^2*\sin(ax)*\cosh(bx*(k-1))) - ax*(bx^2*\sinh(bx*k) + \\ & ax^2*\sinh(bx)*\cos(ax*(k-1)) - bx^2*\cos(ax)*\sinh(bx*(k-1))))*(by*(ay^2*\sin(ay*e)*\cosh(by) - ay^2*\sin(ay*(e-1)) + \\ & by^2*\cosh(by*e)*\sin(ay)) - ay*(ay^2*\cos(ay*e)*\sinh(by) - by^2*\sinh(by*(e-1)) + by^2*\sinh(by*e)*\cos(ay)))/(ax*ay*bx*by*(2*\cos(ax)*\cosh(bx) + (\sin(ax)*\sinh(bx)*(ax^2 - \\ & bx^2))/(ax*bx) - 2)*(2*\cos(ay)*\cosh(by) + (\sin(ay)*\sinh(by)*(ay^2 - by^2))/(ay*by) - 2)); \end{aligned}$$

```

1)) + by^2*sinh(by*e)*cos(ay)))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N16k3 = -((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(bx*(ax^3*cos(ax*k) - ax^3*cosh(bx)*cos(ax*(k - 1)) +
bx^3*sin(ax)*sinh(bx*(k - 1))) + ax*(bx^3*cos(ax)*cosh(bx*(k - 1)) -
bx^3*cosh(bx*k) + ax^3*sinh(bx)*sin(ax*(k -
1)))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

N16e3 = -((bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k -
1))) + ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))))*(by*(ay^3*cos(ay*e)*cosh(by) - ay^3*cos(ay*(e - 1)) +
by^3*sinh(by*e)*sin(ay)) + ay*(by^3*cosh(by*(e - 1)) -
by^3*cosh(by*e)*cos(ay) +
ay^3*sin(ay*e)*sinh(by))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

N16k4 = -((by*(sin(ay*(e - 1)) + cosh(by*e)*sin(ay) - sin(ay*e)*cosh(by)) +
ay*(sinh(by*(e - 1)) + cos(ay*e)*sinh(by) -
sinh(by*e)*cos(ay)))*(bx*(ax^4*cosh(bx)*sin(ax*(k - 1)) - ax^4*sin(ax*k) +
bx^4*sin(ax)*cosh(bx*(k - 1))) + ax*(ax^4*sinh(bx)*cos(ax*(k - 1)) -
bx^4*sinh(bx*k) + bx^4*cos(ax)*sinh(bx*(k -
1)))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) + (sin(ax)*sinh(bx)*(ax^2 -
bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) + (sin(ay)*sinh(by)*(ay^2 -
by^2))/(ay*by) - 2));

N16e4 = -((bx*(cosh(bx)*sin(ax*(k - 1)) - sin(ax*k) + sin(ax)*cosh(bx*(k -
1))) + ax*(cos(ax)*sinh(bx*(k - 1)) - sinh(bx*k) + sinh(bx)*cos(ax*(k -
1))))*(by*(ay^4*sin(ay*(e - 1)) - ay^4*sin(ay*e)*cosh(by) +
by^4*cosh(by*e)*sin(ay)) + ay*(by^4*sinh(by*(e - 1)) +
ay^4*cos(ay*e)*sinh(by) -
by^4*sinh(by*e)*cos(ay))))/(ax*ay*bx*by*(2*cos(ax)*cosh(bx) +
(sin(ax)*sinh(bx)*(ax^2 - bx^2))/(ax*bx) - 2)*(2*cos(ay)*cosh(by) +
(sin(ay)*sinh(by)*(ay^2 - by^2))/(ay*by) - 2));

% Shape function and derivative vectors

N = [N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N12 N13 N14 N15 N16];
Nt = N';
dk = [N1k N2k N3k N4k N5k N6k N7k N8k N9k N10k N11k N12k N13k N14k N15k
N16k];
dkt = dk';
de = [N1e N2e N3e N4e N5e N6e N7e N8e N9e N10e N11e N12e N13e N14e N15e
N16e];
det = de';
d2k = [N1kk N2kk N3kk N4kk N5kk N6kk N7kk N8kk N9kk N10kk N11kk N12kk N13kk
N14kk N15kk N16kk];
d2kt = d2k';
d2e = [N1ee N2ee N3ee N4ee N5ee N6ee N7ee N8ee N9ee N10ee N11ee N12ee N13ee
N14ee N15ee N16ee];

```

```

d2et = d2e';
dke = [N1ke N2ke N3ke N4ke N5ke N6ke N7ke N8ke N9ke N10ke N11ke N12ke N13ke
N14ke N15ke N16ke];
dket = dke';
dk2e = [N1k2e N2k2e N3k2e N4k2e N5k2e N6k2e N7k2e N8k2e N9k2e N10k2e N11k2e
N12k2e N13k2e N14k2e N15k2e N16k2e];
dk2et = dk2e';
dke2 = [N1ke2 N2ke2 N3ke2 N4ke2 N5ke2 N6ke2 N7ke2 N8ke2 N9ke2 N10ke2 N11ke2
N12ke2 N13ke2 N14ke2 N15ke2 N16ke2];
dke2t = dke2';
dk3 = [N1k3 N2k3 N3k3 N4k3 N5k3 N6k3 N7k3 N8k3 N9k3 N10k3 N11k3 N12k3 N13k3
N14k3 N15k3 N16k3];
dk3t = dk3';
de3 = [N1e3 N2e3 N3e3 N4e3 N5e3 N6e3 N7e3 N8e3 N9e3 N10e3 N11e3 N12e3 N13e3
N14e3 N15e3 N16e3];
de3t = de3';
dk4 = [N1k4 N2k4 N3k4 N4k4 N5k4 N6k4 N7k4 N8k4 N9k4 N10k4 N11k4 N12k4 N13k4
N14k4 N15k4 N16k4];
dk4t = dk4';
de4 = [N1e4 N2e4 N3e4 N4e4 N5e4 N6e4 N7e4 N8e4 N9e4 N10e4 N11e4 N12e4 N13e4
N14e4 N15e4 N16e4];
de4t = de4';
dk2e2 = [N1k2e2 N2k2e2 N3k2e2 N4k2e2 N5k2e2 N6k2e2 N7k2e2 N8k2e2 N9k2e2
N10k2e2 N11k2e2 N12k2e2 N13k2e2 N14k2e2 N15k2e2 N16k2e2];
dk2e2t = dk2e2';

```

```

% Element stiffness contribution from bending moment Mx

```

```

x001 = zeros(16);
x001 = dkt*d2k;
x01 = subs (x001,k,1) - subs (x001,k,0);
x1 = int (x01,e,0,1);
g1 = (D*b)/(a^3);
mx1 = g1*x1;

```

```

x002 = zeros(16);
x002 = dkt*d2e;
x02 = subs (x002,k,1) - subs (x002,k,0);
x2 = int (x02,e,0,1);
g2 = (D*v)/(a*b);
mx2 = g2*x2;

```

```

mx = mx1+mx2;

```

```

% Element stiffness contribution from bending moment My

```

```

y001 = zeros(16);
y001 = det*d2k;
y01 = subs (y001,e,1) - subs (y001,e,0);
y1 = int (y01,k,0,1);
g3 = (D*v)/(a*b);
my1 = g3*y1;

```

```

y002 = zeros(16);
y002 = det*d2e;

```

```

y02 = subs (y002,e,1) - subs (y002,e,0);
y2 = int (y02,k,0,1);
g4 = (D*a)/(b^3);
my2 = g4*y2;

my = my1+my2;

% Element stiffness contribution from twisting moment Mxy

xy001 = zeros(16);
xy001 = dkt*dke;
xy01 = subs (xy001,e,1) - subs (xy001,e,0);
xy1 = int (xy01,k,0,1);
g5 = (D*(1-v))/(a*b);
mxy1 = g5*xy1;

xy002 = zeros(16);
xy002 = det*dke;
xy02 = subs (xy002,k,1) - subs (xy002,k,0);
xy2 = int (xy02,e,0,1);
g6 = (D*(1-v))/(a*b);
mxy2 = g6*xy2;

mxy = mxy1+mxy2;

% Element stiffness contribution from shear force Qx

sfx001 = zeros(16);
sfx001 = Nt*dk3;
sfx01 = subs (sfx001,k,1) - subs (sfx001,k,0);
sfx1 = int (sfx01,e,0,1);
g7 = -(D*b)/(a^3);
qx1 = g7*sfx1;

sfx002 = zeros(16);
sfx002 = Nt*dke2;
sfx02 = subs (sfx002,k,1) - subs (sfx002,k,0);
sfx2 = int (sfx02,e,0,1);
g8 = -(D)/(a*b);
qx2 = g8*sfx2;

qx = qx1+qx2;

% Element stiffness contribution from shear force Qy

sfy001 = zeros(16);
sfy001 = Nt*de3;
sfy01 = subs (sfy001,e,1) - subs (sfy001,e,0);
sfy1 = int (sfy01,k,0,1);
g9 = -(D*a)/(b^3);
qy1 = g9*sfy1;

sfy002 = zeros(16);
sfy002 = Nt*dk2e;
sfy02 = subs (sfy002,e,1) - subs (sfy002,e,0);

```

```

sfy2 = int (sfy02,k,0,1);
g10 = -(D)/(a*b);
qy2 = g10*sfy2;

qy = qy1+qy2;

% Residual element stiffness matrices

r001 = zeros(16);
r001 = Nt*dk4;
r01 = int(r001,k,0,1);
r1 = int(r01,e,0,1);
g11 = (D*b)/(a^3);
rg1 = g11*r1;

r002 = zeros(16);
r002 = Nt*dk2e2;
r02 = int(r002,k,0,1);
r2 = int(r02,e,0,1);
g12 = (2*D)/(a*b);
rg2 = g12*r2;

r003 = zeros(16);
r003 = Nt*de4;
r03 = int(r003,k,0,1);
r3 = int(r03,e,0,1);
g13 = (D*a)/(b^3);
rg3 = g13*r3;

rg = rg1+rg2+rg3;

ke = mx+my+mxy+qx+qy+rg;

% Element mass matrix mk

mk001 = zeros(16);
mk001 = Nt*N;
mk01 = int(mk001,k,0,1);
mk1 = int (mk01,e,0,1);
g14 = (a*b*Rho*h*(omg^2));
me = g14*mk1;

ke = mx+my+mxy+qx+qy+rg;

% Global stiffness and mass matrix for single element model

Kg = zeros(16);
Mg = zeros(16);
Kg = ke;
Mg = me;

% Application of S-S-S-S boundary conditions

for i = 1:3:10;

```

```

Kg(:,i) = [];
Kg(i,:) = [];
Mg(:,i) = [];
Mg(i,:) = [];
end

K = Kg;
M = Mg;

% Eigenvalue problem

DTM = det (K-(omega^2)*M);

% Determination of roots

EQN = @(omega) DTM;
d1 = bisection(EQN,471,785);
d2 = bisection(EQN,785,1099);
d3 = bisection(EQN,1099,1727);
d4 = bisection(EQN,1727,2042);
d5 = bisection(EQN,2042,2356);

% Converting from Radians to Hz

freq_rad = zeros(5,1);
freq_rad(1,1) = d1;
freq_rad(2,1) = d2;
freq_rad(3,1) = d3;
freq_rad(4,1) = d4;
freq_rad(5,1) = d5;

freq_hz = 1/(2*pi())* freq_rad;

freq = floor(freq_hz * 100)/100

```

```

function p = bisection(f,a,b)

if f(a)*f(b)>0
    disp('Error! There is no solution between these two input values')
else
    p = (a + b)/2;
    err = abs(f(p));
    while err > 1e-7
        if f(a)*f(p)<0
            b = p;
        else
            a = p;
        end
        p = (a + b)/2;
        err = abs(f(p));
    end
end

```

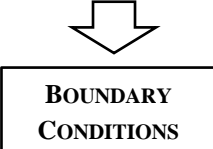
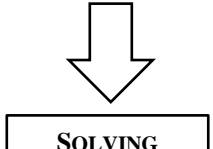
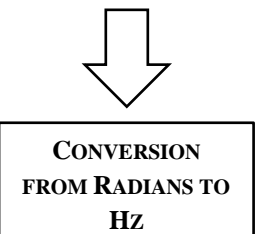
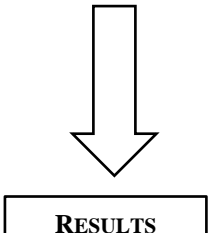

Appendix C: DCM Solution for a Simply Supported (S-S-S-S) Plate

MATLAB[®] program flowchart and sample results for the DCM solution

Table 33: MATLAB[®] program flow chart and sample results for S-S-S-S plate using DCM

Flow Chart	Excerpts from MATLAB [®] Code	Equation
<div style="border: 1px solid black; padding: 5px; text-align: center; width: 100px; margin: 0 auto;">INPUT</div> <div style="text-align: center; margin: 20px 0;">↓</div>	<pre>syms ax ay bx by x y omega % Constants Rho = 7800; F = 200e9; h = 0.004; v = 0.3; D = (F*(h^3))/(12*(1-(v^2))); c1 = 0.6; c2 = 0.4; ax = abs(sqrt(-1)*(sqrt(c1*omega*sqrt(Rho*h/D)))); ay = abs(sqrt(-1)*(sqrt(c2*omega*sqrt(Rho*h/D)))); bx = (sqrt(c1*omega*sqrt(Rho*h/D))); by = (sqrt(c2*omega*sqrt(Rho*h/D)));</pre>	Eqs. 31 to 34
<div style="border: 1px solid black; padding: 5px; text-align: center; width: 100px; margin: 0 auto;">DERIVATIVES OF SOLUTION VECTOR</div> <div style="text-align: center; margin: 20px 0;">↓</div>	<pre>% Solution vector G = zeros(1,16); G = [sin(ax*x)*sin(ay*y) sin(ax*x)*cos(ay*y) sin(ax*x)* % Derivation with respect to x, y and xy thetax = zeros(1,16); thetay = zeros(1,16); thetaxy = zeros(1,16); thetax = diff(G,x,1); thetay = diff(G,y,1); thetaxy = diff(thetax,y,1);</pre>	Eqs. 39 and 134-137
<div style="border: 1px solid black; padding: 5px; text-align: center; width: 100px; margin: 0 auto;">EVALUATING DERIVATIVES AT EACH NODE</div> <div style="text-align: center; margin: 20px 0;">↓</div>	<pre>% Evaluation at node 1 Ln11 = subs(G,x,0) L11 = subs(Ln11,y,0) Ln12 = subs(thetax,x,0) L12 = subs(Ln12,y,0) Ln13 = subs(thetay,x,0) L13 = subs(Ln13,y,0) Ln14 = subs(thetaxy,x,0) L14 = subs(Ln14,y,0)</pre>	Eq. 138
<div style="border: 1px solid black; padding: 5px; text-align: center; width: 100px; margin: 0 auto;">DCM MATRIX</div> <div style="text-align: center; margin: 20px 0;">↓</div>	<pre>% Solution matrix Tg = zeros(16); Tg = [L11 L12 L13 L14 L21 L22 L23 L24 L31 L32 L33 L34 L41 L42 L43 L44];</pre>	Eq. 139

Table 33 continued: MATLAB[®] program flow chart and sample results for S-S-S-S plate using DCM

Flow Chart	Excerpts from MATLAB [®] Code	Equation
 <p>BOUNDARY CONDITIONS</p>	<pre>% Application of S-S-S-S boundary conditions for i = 1:3:10; Tg(:,i) = []; Tg(i,:) = []; end T = Tg;</pre>	N/A
 <p>SOLVING</p>	<pre>% Solving for natural frequencies detT = det(T); EQN = @(omega) detT; d1 = bisection(EQN,471,785); d2 = bisection(EQN,785,1099); d3 = bisection(EQN,1099,1727); d4 = bisection(EQN,1727,2042); d5 = bisection(EQN,2042,2356);</pre>	Eqs. 140
 <p>CONVERSION FROM RADIANS TO HZ</p>	<pre>% Converting from Radians to Hz freq_rad = zeros(5,1); freq_rad(1,1) = d1; freq_rad(2,1) = d2; freq_rad(3,1) = d3; freq_rad(4,1) = d4; freq_rad(5,1) = d5; freq_hz = 1/(2*pi())* freq_rad freq = floor(freq_hz * 100)/100</pre>	N/A
 <p>RESULTS</p>	<pre>Command Window >> DCMthinplate freq = 86.6800 166.6300 266.9900 300.1400 346.4900 fx >></pre>	N/A

MATLAB® code for a simply supported, thin, homogeneous, rectangular plate analysed using a single element DCM model

```
% Dynamic Coefficient Matrix method for a thin, homogeneous, rectangular  
plate under S-S-S-S boundary condition  
% Toronto, Ontario, Canada, 2019  
% Copyright Heenkenda Jayasinghe 2019
```

```
syms ax ay bx by xy omega
```

```
% Constants
```

```
Rho = 7800;  
F = 200e9;  
h = 0.004;  
v = 0.3;  
D = (F*(h^3))/(12*(1-(v^2)));  
c1 = 0.6;  
c2 = 0.4;  
ax = abs(sqrt(-1)*(sqrt(c1*omega*sqrt(Rho*h/D))));  
ay = abs(sqrt(-1)*(sqrt(c2*omega*sqrt(Rho*h/D))));  
bx = (sqrt(c1*omega*sqrt(Rho*h/D)));  
by = (sqrt(c2*omega*sqrt(Rho*h/D)));
```

```
% Solution vector
```

```
G = zeros(1,16);  
G = [sin(ax*x)*sin(ay*y) sin(ax*x)*cos(ay*y) sin(ax*x)*sinh(by*y)  
sin(ax*x)*cosh(by*y) cos(ax*x)*sin(ay*y) cos(ax*x)*cos(ay*y)  
cos(ax*x)*sinh(by*y) cos(ax*x)*cosh(by*y) sinh(bx*x)*sin(ay*y)  
sinh(bx*x)*cos(ay*y) sinh(bx*x)*sinh(by*y) sinh(bx*x)*cosh(by*y)  
cosh(bx*x)*sin(ay*y) cosh(bx*x)*cos(ay*y) cosh(bx*x)*sinh(by*y)  
cosh(bx*x)*cosh(by*y)];
```

```
% Vector of unknown coefficients
```

```
E = zeros(16,1);  
E = [E11  
E12  
E13  
E14  
E21  
E22  
E23  
E24  
E31  
E32  
E33  
E34  
E41  
E42  
E43  
E44];
```

```

% Displacement amplitude

diam = zeros(16,1);
diam = [w1
        tx1
        ty1
        txy1
        w2
        tx2
        ty2
        txy2
        w3
        tx3
        ty3
        txy3
        w4
        tx4
        ty4
        txy4];

% Derivation with respect to x, y and xy

thetax = zeros(1,16);
thetay = zeros(1,16);
thetaxy = zeros(1,16);

thetax = diff(G,x,1);
thetay = diff(G,y,1);
thetaxy = diff(thetax,y,1);
% Evaluation at node 1

Ln11 = subs(G,x,0)
L11 = subs(Ln11,y,0)

Ln12 = subs(thetax,x,0)
L12 = subs(Ln12,y,0)

Ln13 = subs(thetay,x,0)
L13 = subs(Ln13,y,0)

Ln14 = subs(thetaxy,x,0)
L14 = subs(Ln14,y,0)

% Evaluation at node 2

Ln21 = subs(G,x,0.6)
L21 = subs(Ln21,y,0)

Ln22 = subs(thetax,x,0.6)
L22 = subs(Ln22,y,0)

Ln23 = subs(thetay,x,0.6)

```

```

L23 = subs(Ln23,y,0)

Ln24 = subs(thetaxy,x,0.6)
L24 = subs(Ln24,y,0)

% Evaluation at node 3

Ln31 = subs(G,x,0.6)
L31 = subs(Ln31,y,0.4)

Ln32 = subs(thetax,x,0.6)
L32 = subs(Ln32,y,0.4)

Ln33 = subs(thetay,x,0.6)
L33 = subs(Ln33,y,0.4)

Ln34 = subs(thetaxy,x,0.6)
L34 = subs(Ln34,y,0.4)

% Evaluation at node 4

Ln41 = subs(G,x,0)
L41 = subs(Ln41,y,0.4)

Ln42 = subs(thetax,x,0)
L42 = subs(Ln42,y,0.4)

Ln43 = subs(thetay,x,0)
L43 = subs(Ln43,y,0.4)
Ln44 = subs(thetaxy,x,0)
L44 = subs(Ln44,y,0.4)

% Solution matrix

Tg = zeros(16);

Tg = [L11
      L12
      L13
      L14
      L21
      L22
      L23
      L24
      L31
      L32
      L33
      L34
      L41
      L42
      L43
      L44];

```

```

% Application of S-S-S-S boundary conditions

for i = 1:3:10;

Tg(:,i) = [];
Tg(i,:) = [];

end

T = Tg;

% Solving for natural frequencies

detT = det(T);
EQN = @(omega) detT;
d1 = bisection(EQN,471,785);
d2 = bisection(EQN,785,1099);
d3 = bisection(EQN,1099,1727);
d4 = bisection(EQN,1727,2042);
d5 = bisection(EQN,2042,2356);

% Converting from Radians to Hz

freq_rad = zeros(5,1);
freq_rad(1,1) = d1;
freq_rad(2,1) = d2;
freq_rad(3,1) = d3;
freq_rad(4,1) = d4;
freq_rad(5,1) = d5;

freq_hz = 1/(2*pi())* freq_rad

freq = floor(freq_hz * 100)/100



---


function p = bisection(f,a,b)

if f(a)*f(b)>0
    disp('Error! There is no solution between these two input values')
else
    p = (a + b)/2;
    err = abs(f(p));
    while err > 1e-7
        if f(a)*f(p)<0
            b = p;
        else
            a = p;
        end
        p = (a + b)/2;
        err = abs(f(p));
    end
end
end

```

REFERENCES

1. E. F. F. Chladni. (1802). *Die Akustik*, Leipzig.
2. A. W. Leissa. (1969). NASA SP-169. *Vibration of Plates*. Washington, D.C.: Office of Technology Utilization.
3. W. Leissa. (1977). Recent research in plate vibrations: classical theory. *The Shock and Vibration Digest*. 9(10), 13–24.
4. W. Leissa. (1977). Recent research in plate vibrations: complicating effects. *The Shock and Vibration Digest*. 9(11), 21–35.
5. W. Leissa. (1981). Plate vibration research, 1976–1980: classical theory. *The Shock and Vibration Digest*. 13(9), 11–22.
6. W. Leissa. (1981). Plate vibration research, 1976–1980: complicating effects. *The Shock and Vibration Digest*. 13(10), 19–36.
7. W. Leissa. (1987). Recent research in plate vibrations, 1981–1985, Part II: complicating effects. *The Shock and Vibration Digest*. 19(3), 10–24.
8. W. Bert. (1976). Dynamics of composite and sandwich panels, Parts I and II. *The Shock and Vibration Digest*. 8(11), 15–24.
9. W. Bert. (1979). Recent research in composite and sandwich plate dynamics. *The Shock and Vibration Digest*. 11(10), 13–23.
10. W. Bert. (1982). Research on dynamics of composite and sandwich plates. *The Shock and Vibration Digest*. 14(10), 17–34.
11. W. Bert. (1985). Research on dynamic behavior of composite and sandwich plates, part IV. *The Shock and Vibration Digest*. 17(11), 3–25.
12. W. Bert. (1991). Research on dynamic behavior of composite and sandwich plates, V: part I. *The Shock and Vibration Digest*. 23(6), 3–14.
13. W. Bert. (1991). Research on dynamic behavior of composite and sandwich plates, V: part II. *The Shock and Vibration Digest*. 23(7), 9–21.
14. J. N. Reddy. (1985). A review of the literature on finite-element modelling of laminated composite plates and shells. *The Shock and Vibration Digest*. 17(4), 3–8.

15. R. K. Kapania and S. Raciti. (1989). Recent advances in analysis of laminated beams and plates, part II: vibrations and wave propagation. *American Institute of Aeronautics and Astronautics Journal*. 27, 935–946.
16. S. Timoshenko and S. Woinowsky-Krieger. (1959). *Theory of Plates and Shells*. McGraw-Hill. New York.
17. S.G. Lekhnitkii. (1968). *Anisotropic plates*. Gordon and Breach Science Publishers, New York.
18. J.R. Vinson and R.L. Sierakowski. (1986). *The Behavior of Structures Composed of Composite Materials*, Martinus Nijhoff Publishers, Boston, Massachusetts.
19. J.E. Ashton and J.M. Whitney. (1970). *Theory of Laminated Plates*, Technomic Publishing Company Inc. Stamford, Connecticut.
20. E. Reissner. (1945). "The Effect of Transverse Shear Deformation on the Bending of Elastic Plates," *Journal of Applied Mechanics*, 12, 69-77(1945).
21. R.D. Mindlin. (1951). "Influence of Rotatory Inertia and Shear on Flexural Motions of Isotropic Elastic Plates," *Journal of Applied Mechanics*, 18, 31-38.
22. H. Reismann. (1988). *Elastic Plates*. John Wiley & Sons Inc. New York.
23. G. Bergan and X. Wang. (1984). Quadrilateral plate bending elements with shear deformations. *Computers and Structures*. 19, 25–34.
24. J. M. Whitney and N. J. Pagano. (1970). *American Society of Mechanical Engineers Journal of Applied Mechanics*. 37, 1031–1036.
25. P. C. Yang, C. H. Norris and Y. Stavsky. (1966). Elastic wave propagation in heterogeneous plates. *International Journal of Solids and Structures*. 2, 665–684.
26. J.N. Reddy. (1984). A Simple Higher-Order Theory for Laminated Composite Plates. *Journal of Applied Mechanics*, 45, 745-752(1984).
27. A.V. Krishnamurthy. (1987). Flexure of Composite Plates. *Composite Structure*. 7, 161-177.
28. R. B. Nelson and D. R. Lorch. (1974). A refined theory for laminated orthotropic plates. *American Society of Mechanical Engineers Journal of Applied Mechanics*. 41, 177–183.

29. K. H. Lo, R. M. Christensen and E. M. Wu. (1977). A higher-order theory of plate deformation, part 1: homogeneous plates/part 2: laminated plates. Transactions of the American Society of Mechanical Engineers, Journal of Applied Mechanics. 44, 663–676.
30. M. Levinson. (1980). An accurate simple theory of the statics and dynamics of elastic plates. Mechanics Research Communications. 7, 343–350.
31. M. V. V. Murthy. (1981). An improved transverse shear deformation theory for laminated anisotropic plate. NASA Technical Paper 1903.
32. J. L. Doong, C. Lee and C. P. Fung. (1991). Vibration and stability of laminated plates based on a modified plate theory. Journal of Sound and Vibration. 151, 193–201.
33. J.N. Reddy and N.D. Phan. (1985). Stability and Vibration of Isotropic, Orthotropic and Laminated Plates according to a Higher Order Shear Deformation Theory. Journal of Sound and Vibration. 98, 157-170.
34. N.R. Senthilnathan, S.P. Lim, K.H. Lee and S.T. Chow. (1988). Vibration of Laminated Orthotropic Plates Using a Simplified Higher-Order Deformation Theory. Composite Structures. 10, 211-229.
35. C.W. Bert and T.L. Chen. (1978). Effect of Shear Deformation on Vibration of Antisymmetric Angle-Ply Laminated Rectangular Plates," Int. Journal of Solids and Structures. 14, 465-473.
36. I.D. Kaplevatsky and V.O. Shestopal. (1982). "Bending and Buckling of Multilayer Thin Plates. Acta Mechanica. 43, 169-176.
37. Ji-Liang Doong, Tsyr-Jang Chen and Lien-Wen Chen. (1987). Vibration and Stability of an Initially Stressed Laminated Plate Based on a Higher Order Deformation Theory. Composite Structures. 7, 285-310.
38. C.W. Bert. (1977). Optimal Design of a Composite-Material Plate to Maximize Its Fundamental Frequency. Journal of Sound and Vibration. 50, 229-237.
39. S. Adali. (1984). Design of Shear-Deformable Antisymmetric Angle-Ply Laminates to Maximize the Fundamental Frequency and Frequency Separation. Composite Structures. 2, 349-369.

40. M. Levy. (1899). Sur L'equilibre elasticque d'une Plaque Rectangulaire. C.R. Acad. Science. 129, 535-539.
41. S. Timoshenko and J.M. Gere. (1961). Theory of Elastic Stability, McGraw-Hill Publishing Company, New York.
42. P.S. Bulson. (1970). The Stability of Flat Plate, Chatto & Windus Limited. London.
43. J.N. Reddy, A.A. Khdeir and L. Librescu. (1987). Levy Type Solutions for Symmetrically Laminated Rectangular Plates Using the First Order Shear Deformation Theory. Journal of Applied Mechanics. 54, 741-742.
44. A.A. Khdeir. (1988). Free Vibration and Buckling of Symmetric Cross-Ply Laminated Plates by an Exact Method. Journal of Sound and Vibration. 126, 447-461.
45. R.F. Palardy and A.N. Palazotto. (1990). Buckling and Vibration of Composite Plates Using the Levy Method. Composite Structures. 14, 61-86.
46. A.A. Khdeir. (1988). Free Vibration of Antisymmetric Angle-Ply Laminated Plates Including Various Boundary Conditions. Journal of Sound and Vibration. 122, 377-388.
47. A.A. Khdeir. (1989). Free Vibration and Buckling of Unsymmetric Cross-Ply Laminated Plates Using a Refined Theory. Journal of Sound and Vibration. 128, 377-395.
48. J.N. Reddy and A.A. Khedir. (1989). Buckling and vibration of laminated composite plates using various plate theories. American Institute of Aeronautics and Astronautics Journal. 27, 1808–1817.
49. D.J. Gorman. (1982). Free Vibration Analysis of Rectangular Plates. Elsevier/North-Holland, New York.
50. D.J. Gorman. (1983). A Highly Accurate Analytical Solution for Free Vibration Analysis of Simply Supported Right Triangular Plates. Journal of Sound and Vibration. 89, 107-118.
51. H.T. Saliba. (1986). Free Vibration Analysis of Non-Rectangular Quadrilateral Plates. Ph.D. Thesis, University of Ottawa.
52. H.T. Saliba. (1988). Transverse Free Vibration of Fully Clamped Symmetrical Trapezoidal Plates. Journal of Sound and Vibration. 126, 237-247.
53. D.J. Gorman. (1990). Accurate Free Vibration Analysis of Clamped Orthotropic Plates by the Method of Superposition. Journal of Sound and Vibration. 140(3), 391-411.

54. S.D. Yu and W.L. Cleghorn. (1991). Study of Free Vibration of Orthotropic Rectangular Plates. Proceedings of the 19th CANCAM Conference, Winnipeg, Manitoba. 1, 318-319.
55. S.D. Yu and W.L. Cleghorn. (1993). Generic Free Vibration of Orthotropic Rectangular Plates with Combinations of Clamped and Simply Supported Edges. Journal of Sound and Vibration. 163 (3), 439-450.
56. S.D. Yu and W.L. Cleghorn. (1991). Free Vibration Analysis of Clamped Orthotropic Rectangular Plates Subjected to Constant In-Plane Loads. Presented at 4SME Pressure Vessels and Piping Conference, San Diego, California. Published in Piping Components Analysis: Piping and Structural Dynamics, PVP-Vol. 218, 113-119.
57. S.D. Yu and W.L. Cleghorn. (1992). Generic Buckling Analysis of Orthotropic Plates with Clamped and Simply Supported Edges. Proceedings of the Second International Congress on Recent Developments in Air-& Structure- Bourne Sound and Vibration. Auburn University, Auburn, AL, 729-736, 1992.
58. W.L. Cleghorn and S.D. Yu. (1992). Analysis of Buckling of Rectangular Plates Using the Method of Superposition. Transactions of GSME. 16(2), 185-199.
59. W.L. Cleghorn and S.D. Yu. (1991). Effect of Shear Deformation on Free Vibration of Rectangular Plates. Presented at ASME Biennial Conference on Mechanical Vibration & Noise, Miami, Florida. Published in Machinery Dynamics and Element Vibrations, Del-Vol.36. 185-191.
60. S.D. Yu and W.L. Cleghorn. (1992). Accurate Free Vibration Analysis of Clamped Mindlin Plates Using the Method of Superposition. Proceedings of the 11th Symposium on EAM. 226-230.
61. W.L. Cleghorn and S.D. Yu. (1992). Accurate Analysis of Free Vibration of Rectangular Mindlin Plates with Clamped and Simply Supported Edges. Presented at HSTAM, National Technical University of Athens, Greece.
62. S.D. Yu. W.L. Cleghorn, and R.G. Fenton. (1993). Buckling of Clamped Rectangular Mindlin Plates. Proceedings of CANCAM. 465-466.
63. S.D. Yu and W.L. Cleghorn. (1993). Accurate Free Vibration Analysis of Rectangular Mindlin Plates Using the Method of Superposition. Transactions of CSME. 17(2), 243-255.

64. S.D. Yu, W.L. Cleghorn, R.G. Fenton. (1994). Accurate Analysis of Free Vibration and Buckling of Clamped Symmetric Cross-Ply Laminates. *AIAA Journal*. 32(11), 2300-2308.
65. Y. Mochida and S. Ilanko. (2008). Bounded natural frequencies of completely free rectangular plates. *Journal of Sound and Vibration*. 311, pp. 1–8.
66. S. Ilanko. (2006). On the bounds of Gorman's Superposition method of free vibration analysis. *Journal of Sound Vibration*. 294, pp. 418–420.
67. M. Boscolo and J. R. Banerjee, 'Dynamic stiffness formulation for plates using first order shear deformation theory', presented at the 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Orlando, Florida, 12 - 15 April 2010.
68. M. Boscolo and J. R. Banerjee. (2011). Dynamic stiffness method for exact inplane free vibration analysis of plates and plate assemblies. *Journal of Sound and Vibration*. 330 (12), 2928-2936.
69. M. Boscolo and J. R. Banerjee. (2012). Dynamic stiffness formulation for composite Mindlin plates for exact modal analysis of structures. Part I: Theory. *Computers & Structures*. 96-97, 61-73.
70. M. Boscolo and J. R. Banerjee. (2012). Dynamic stiffness formulation for composite Mindlin plates for exact modal analysis of structures. Part II: Results and applications. *Computers & Structures*. 96-97, 74-83.
71. F.A. Fazzolari, J. R. Banerjee and M. Boscolo. (2013). Buckling of composite plate assemblies using higher order shear deformation theory—An exact method of solution. *Thin-Walled Structures*. 71, 18-34.
72. F.A. Fazzolari, M. Boscolo and J. R. Banerjee. (2013). An exact dynamic stiffness element using a higher order shear deformation theory for free vibration analysis of composite plate assemblies. *Composite Structures*. 96, 262-278.
73. M. Boscolo and J.R. Banerjee. (2014). Layer-wise dynamic stiffness solution for free vibration analysis of laminated composite plate. *Journal of Sound and Vibration*. 333(16), 200-227.

74. A. Pagani et al. (2014). Free vibration analysis of composite plates by higher-order 1D dynamic stiffness elements and experiments. *Composite Structures*. 118, 654-663.
75. J.R. Banerjee et al. (2015). Dynamic stiffness matrix of a rectangular plate for the general case. *Journal of Sound and Vibration*. 342, 177-199.
76. X. Liu and J. R. Banerjee. (2015). An exact spectral-dynamic stiffness method for free flexural vibration analysis of orthotropic composite plate assemblies – Part I: Theory. *Composite Structures*. 132, 1274-1287.
77. X. Liu and J. R. Banerjee. (2015). An exact spectral-dynamic stiffness method for free flexural vibration analysis of orthotropic composite plate assemblies – Part II: Applications. *Composite Structures*. 132, 1288-1302.
78. X. Liu and J. R. Banerjee. (2016). Free vibration analysis for plates with arbitrary boundary conditions using a novel spectral-dynamic stiffness method. *Computers & Structures*. 164, 108-126.
79. E. J. Ruggiero. (2005). Modelling and control of SPIDER satellite components. PhD thesis. Virginia Polytechnic Institute and State University (USA)
80. P. Cawley and R. D. Adams. (1978). The predicted and experimental natural modes of free-free cfrp plates. *Journal of Composite Materials*. 12, 336– 347.
81. J. N. Reddy. (1979). Free vibration of antisymmetric, angle-ply laminated plates including transverse shear deformation by the finite element method. *Journal of Sound and Vibration*. 66, 565–576.
82. W. Bert, J. N. Reddy, W. C. Chao and V. S. Reddy. (1981). Vibration of thick rectangular plates of bimodulus composite materials. *Transactions of the American Society of Mechanical Engineers, Journal of Applied Mechanics*. 48, 371–376.
83. J.N. Reddy. (1982). Large amplitude flexural vibration of layered composite plates with cut-outs. *Journal of Sound and Vibration*. 83, 1–10.
84. J.N. Reddy and T. Kuppusamy. (1984). Natural vibration of laminated anisotropic plates. *Journal of Sound and Vibration*. 94, 63–69.
85. Y.V.K.S. Rao and G. Singh. (1986). Vibration of corner supported thick composite plates. *Journal of Sound and Vibration*. 111, 510–514.

86. C. N. Chang and F. K. Chiang. (1988). Vibration analysis of a thick plate with an interior cut-out by a finite element method. *Journal of Sound and Vibration*. 125, 477–486.
87. L. C. Shiau and J. T. Chang. (1991). Transverse shear effect on vibration of laminated plate using higher-order plate element. *Computers and Structures*. 39, 735–740.
88. M. Ganapathi, T. K. Vardan and B. S. Sarma. (1991). Nonlinear flexural vibrations of laminated orthotropic plates. *Computers and Structures*. 39, 685–688.
89. Y. K. Cheung and W. L. Kwok. (1975). Dynamic analysis of circular and sector thick, layered plates. *Journal of Sound and Vibration*. 42, 147–158.
90. L. W. Chen and C. C. Chen. (1989). Asymmetric vibration and dynamic stability of bi-modulus thick annular plates. *Computers and Structures*. 31, 1013–1022.
91. J. Vivoli. (1972). Vibrations de plaques et potentiels de couches. *Acustica*. 26, 305-314.
92. J. Vivoli and P. Filipi. (1974). Eigenfrequencies of thin plates and layer potentials. *The Journal of Acoustical Society of America*. 55, 562.
93. Y. Niwa et al. (1982). Determination of eigenvalues by boundary element methods. *Developments in boundary element methods*, Vol. 2. Applied Science Publishers, Chap 7.
94. G. I. K. Wong and J. R. Hutchinson. (1981). An improved boundary element method for plate vibrations. *Proceedings of the 3rd International Seminar on Boundary Element Method*. Irvine, California. Springer – Verlag.
95. G. Bazine. (1980). A mixed boundary integral – finite element approach to plate vibration problems. *Mechanics Research Communications*. 7, 141-150.
96. D. Nardini and C. A. Brebbia. (1982). A new approach to free vibration analysis using boundary elements. *Proceedings of the 4th International Conference on BEM*. Southampton, England. Springer – Verlag, Berlin. pp. 313-326.
97. A. Kanarachos and Ch. Provatidis. (1987). Performance of mass matrices for the BEM dynamic analysis of wave propagation problems. *Computational Methods of Applied Mechanical Engineering* 63, 155–165.
98. S. Ahmad and P.K. Banerjee. (1986). Free vibration analysis using BEM particular integrals. *Journal of Engineering Mechanical ASCE* 112, 682–695.

99. T.W. Davies and F.A.Moslehy. (1994). Modal analysis of plates using the dual reciprocity boundary element method. *Engineering of Analysis Boundary Elements*. 14, 357–362.
- 100.Y. K. Cheung and S. Chakrabarti. (1972). Free vibration of thick, layered rectangular plates by a finite layer method. *Journal of Sound and Vibration*. 21, 277–284.
- 101.W. L. Kwok and Y. K. Cheung. (1974). Analyses of circular and annular laminated thick plates. *Proceedings of the International Conference on Finite Element Methods in Engineering*, University of New South Wales, Sydney.
- 102.J. Dawe. (1978). Finite strip models for vibration of Mindlin plates. *Journal of Sound and Vibration*. 59, 441–452.
- 103.L. Roufaeil and D. J. Dawe. (1980). Vibration analyses of rectangular plates by the finite strip method. *Computers and Structures*. 12, 833–842.
- 104.M. S. Cheung and M. Y. T. Chan. (1981). Static and dynamic analysis of thin and thick sectorial plates by the finite strip method. *Computers and Structures*. 14, 79–88.
- 105.T. Mizusawa. (1991). Vibration of thick annular sector plates using semi-numerical methods. *Journal of Sound and Vibration*. 150, 245–259.
- 106.T. Mizusawa. (1993). Vibration of rectangular Mindlin plates with tapered thickness by the spline strip method. *Computers and Structures*. 46, 451–463.
- 107.S. M. Hashemi. (1998). Free-vibrational analysis of rotating beam-like structures: A dynamic finite element approach. PhD thesis. Laval University (Canada).
- 108.S.M. Hashemi and J.M. Richard. (2000). A DFE method for free vibrations of bending – torsion coupled beams. *Aerospace Science and Technology*. 4, 41 – 55.
- 109.S.M. Hashemi and A. Roach. (2010). A Dynamic Finite Element for the Free Vibration Analysis of Extension-Torsion Coupled Composite Beams. *Mathematics in Engineering, Science and Aerospace (MESA), The Transdisciplinary International Journal*. 1(3), 221 – 239.
- 110.S.M. Hashemi and E.J. Adique. (2010). A Quasi–Exact Dynamic Finite Element for Free Vibration Analysis of Sandwich Beams. *Applied Composite Materials*. 17, 259 – 269.

- 111.S. Borneman and S.M. Hashemi. (2004). Dynamic Finite Element (DFE) Formulation for the Free Vibration Analysis of Laminated Composite Wing– Beams. *Journal of Sound and Vibration*.
- 112.S.M. Hashemi and D. Pereira. (2007). A Meshless Dynamic Finite Element for Beam Vibrations including Rotary Inertia. *Proceedings of the 7th International Conference on Computational Structures Technology*. Stirlingshire, UK: Civil – Comp Press. Paper 127.
- 113.S.M. Hashemi and M.J. Richard. (2000). Free vibrational analysis of axially loaded bending–torsion coupled beams: a dynamic finite element. *Computers and Structures*.77, 711 – 24.
- 114.H. Jayasinghe, T. Kashani, and S.M. Hashemi. Coupled Flexural–Torsional Free Vibration Analysis of Pre-Loaded Beams- A Dynamic Finite Element Method. *Proceedings of the 3rd International Conference of Acoustics and Vibration, ISAV 2013, Tehran, Iran*.
- 115.T. Kashani, H. Jayasinghe, S.M. Hashemi. Stability Analysis of Beams Subjected to Axial Load and End Moment- A Dynamic Finite Element. *Proceedings of the 3rd International Conference of Acoustics and Vibration, ISAV 2013, Tehran, Iran*.
- 116.K. F. Graff. (1991). *Wave Motion in Elastic Solids*. Dover Publications Inc. New York.
- 117.K. H. Huebner et al. (2001). *The Finite Element Method for Engineers*. John Wiley & Sons Inc. New York.
- 118.A. W. Leissa. (1973). The Free Vibration of Rectangular Plates. *Journal of Sound and Vibration*. 31(3), 257-293.
- 119.E. Ventsel and T. Krauthammer. (2001). *Thin Plates and Shells: Theory, Analysis and Applications*, Marcel Dekker Inc., New York, NY.
- 120.Irvine, T., Thin Plate Vibration – Rectangular Elements. Retrieved from www.vibrationdata.com. Last Accessed 20th October 2018.