

STATIC STRUCTURAL AND MODAL ANALYSIS USING
ISOGEOMETRIC ANALYSIS

SANGAMESH GONDEGAON*, HARI K. VORUGANTI

*Department of Mechanical Engineering, National Institute of Technology,
Warangal, India-506004*

[Received 28 September 2016. Accepted 12 December 2016]

ABSTRACT: Isogeometric Analysis (IGA) is a new analysis method for unification of Computer Aided Design (CAD) and Computer Aided Engineering (CAE). With the use of NURBS basis functions for both modelling and analysis, the bottleneck of meshing is avoided and a seamless integration is achieved. The CAD and computational geometry concepts in IGA are new to the analysis community. Though, there is a steady growth of literature, details of calculations, explanations and examples are not reported. The content of the paper is complimentary to the existing literature and addresses the gaps. It includes summary of the literature, overview of the methodology, step-by-step calculations and Matlab codes for example problems in static structural and modal analysis in 1-D and 2-D. At appropriate places, comparison with the Finite Element Analysis (FEM) is also included, so that those familiar with FEM can appreciate IGA better.

KEY WORDS: Isogeometric analysis, B-spline, finite element analysis, computer aided design, Matlab.

1. INTRODUCTION

The finite element method (FEM) is a numerical method for finding an approximate solution for partial differential equations. In FEM, the geometry of the domain is divided into a set of elements (mesh). But it is difficult to divide a complicated geometry into primitive elements (Ex: triangular and quadrilateral element). Also, meshing is not an exact representation of the geometry. It is an approximate representation which causes inaccurate solution. Also, mesh generation for complex shapes is time consuming and it is observed that 80% of the overall analysis time is spent on meshing. Hence, meshing is the biggest bottleneck in the use of finite element method [1].

To overcome the above issue, a new concept of Isogeometric analysis (IGA) was introduced by Hughes et al. [1]. Isogeometric analysis is the bridge between the Computer Aided Design (CAD) and the Finite Element Method (FEM). The idea of

*Corresponding author e-mail: gsangu_sjce@yahoo.co.in

IGA is to use B-spline basis functions for representing both geometry and field variable. It completely eliminates the concept of meshing, as the geometry is defined as parametric representation with B-spline basis functions. This helps in exact modelling of geometric entities, which results in better solution and also reduces a lot of computation time. B-splines are the most widespread technology in CAD programs and thus, they are used as basis functions for isogeometric analysis. Second attractive feature of B-spline is that, each p^{th} order function has $p-1$ derivatives at element boundary. In addition to this, NURBS basis function can be refined and their order can be easily elevated without affecting the geometry or its parameterization. In contrast to finite element method, which has two methods of refinement, isogeometric methodology has three methods of refinement: knot insertion, degree elevation and k-refinements. K-refinement increases the inter-element continuity as compared to standard C^0 continuity of conventional finite element and therefore, it has superior accuracy and efficiency as compared to the standard p-refinement.

Over the recent years, a lot of work is carried out in the field of isogeometric analysis. The first paper on IGA is published by Hughes *et al.* [1], in which authors proposed isogeometric methodology. Authors also mentioned different refinement techniques and their effect on IGA results. Different structural mechanics problems are solved for the purpose of validation and convergence study.

A vast variety of literature related to defining the geometry with the help of B-spline basis function can be found [2, 3]. A study is being conducted in relation to the smoothness, continuity and refinement [4]. Efforts are also being made into correcting certain deficiencies of B-spline by using T-splines to create a single patch watertight geometry, which can be locally refined. In addition to T-spline, certain alternatives have also been found such as PHT-splines and LR-splines [5]. PHT-splines have been used to solve the problems in elasticity for continua and thin structures. It is easier to do adaptive refinement using PHT-splines. Though, T-splines are favourable for local adaptive refinement but the procedure for knot insertion is a bit complex, this issue is addressed in [6] by using hierarchical T-spline refinement algorithms. This method generates a seamless CAD-FEA integration for very complex geometries.

Since tight coupling of FEA and CAD model in IGA, there is a lot of scope in the field of structural optimization and some distinctive works are presented in [7, 8]. A strategy for shape optimization, using isogeometric analysis is proposed by Wall *et al.* [7]. They studied analytical sensitivities of B-spline discretization on IGA result, using gradient-based optimization algorithms. In Hassani *et al.* [8], shape optimization for both 2-D and 3-D cases using isogeometric analysis is mentioned. Isogeometric methodology has shown its prominence in different fields of applications. The smoothness of NURBS basis functions has made it possible for rotation free formulation of plate or shell elements [9]. The smoothness of higher order NURBS

basis function has also made it significant for the analysis of fluid structure interaction problems [10]. Isogeometric methodology is also proved to be fruitful in solving PDEs that involves fourth order derivatives of the field variable, such as Hill-Cahnard equation [11]. In addition to this, higher order NURBS basis functions are significant in modelling the electronic structure of semiconducting materials [12]. NURBS has also shown significant improvement in the analysis of structural vibration problems in terms of robustness and accuracy, by using k-refinement as compared to higher order p-refinement [13]. Similar results are evident in the analysis of structural vibration of thin plates [14].

It is evident, that isogeometric analysis is spreading in various fields, but IGA method involves the concepts of CAD and Computational Geometry. Hence, it is difficult for the analysis community (familiar with FEM) to easily understand and to appreciate. Though several research articles have appeared, it is difficult for researchers to understand the methodology completely from the published works. In order to further enhance the concept of isogeometric analysis, a detailed methodology with corresponding MATLAB code is presented for different types of static structural and vibration problems. The flow of the paper is as follows: second section includes overview of B-spline; in third section, introduction to isogeometric formulation is mentioned; detailed step-wise application of IGA for solving static bar and plate problems with corresponding MATLAB code is mentioned in section four. Similarly, in section 5, this methodology is applied for dynamic analysis of beam and plate along with MATLAB code. In the final section conclusions are drawn.

2. B-SPLINE

The objective of this section is to give an overview of B-splines. B-Splines are piece-wise polynomial, formed by linear combination of basis functions and control points. B-spline is divided into pieces at distinct points called knots.

2.1. KNOT VECTOR

A knot vector is a set of non-decreasing knots, which break a B-spline into sub-domains. These sub-domains are called knot-spans, which are similar to elements as in FEM. Knot vector is of the form $T = \{\xi_1 \dots \xi_{n+p+1}\}$, where p is the degree of the B-spline, n is number of control points and ξ_i is a knot value. If the spacing between any two consecutive knots is equal, then it is called a uniform knot vector and otherwise, it is called a non-uniform knot vector. A knot vector is said to be open, if first and last knots are repeated p times. In IGA, geometry is modelled using B-spline with open knot vector, because basis functions, which are formed with the help of open knot vector are interpolatory at the ends. This property is helpful for applying boundary conditions in IGA.

2.2. BASIS FUNCTIONS

Based on the degree of the B-spline basis function and knot vector, B-Spline basis functions are calculated by Cox-de-Boor recursive formula [15].

For $p = 1$

$$(1) \quad N_{i,1} = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{Otherwise} \end{cases}$$

For $p > 1$

$$(2) \quad N_{i,p} = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1} + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}.$$

B-Spline basis functions are C^{p-1} continuous, if internal knots are not repeated. If knot has a multiplicity of k , then they are C^{p-k} continuous at that knot. In a similar fashion if a particular knot has multiplicity of p , basis function are C^0 continuous or in other words, they are interpolatory at that knot. Basis functions are all positive for the whole domain, as it is evident in Fig. 1. Another property of basis functions is partition of unity, i.e. the summation of basis functions at a parameter value will be always one.

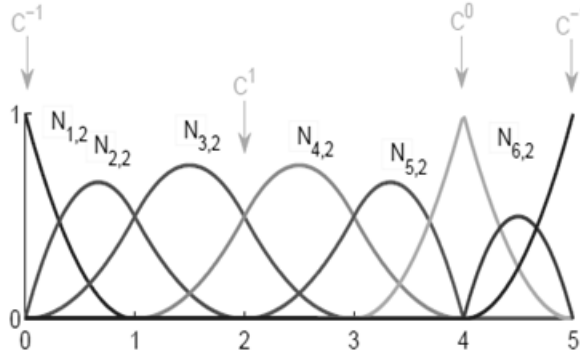


Fig. 1. B-Spline basis functions.

2.3. B-SPLINE CURVE

B-Spline curve, $C(\xi)$ of order p is constructed by the linear combination of basis functions and control points. The curve is given by

$$(3) \quad C(\xi) = \sum_{i=1}^n N_{i,p}(\xi) B_i,$$

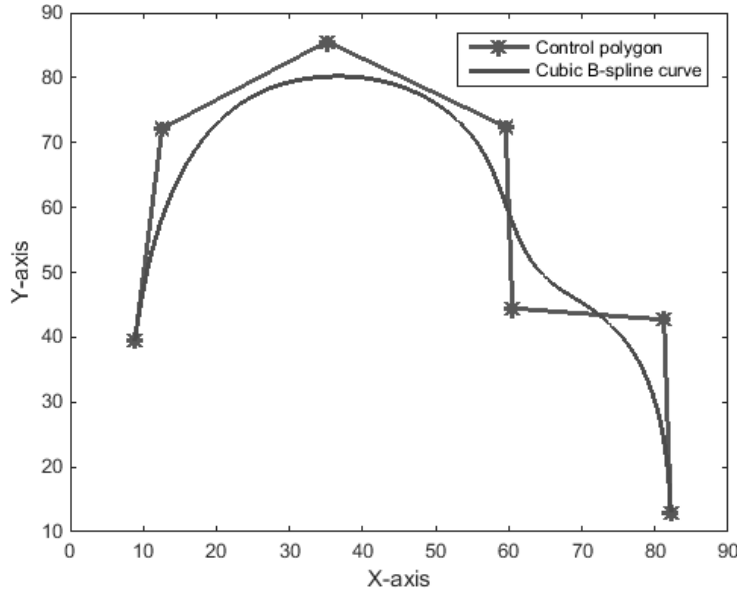


Fig. 2. Cubic B-Spline curve with 7 control points.

where B_i is the i^{th} control point and N is B-spline basis function. The resulting B-spline curve does not need to interpolate the control points. However, one can force it to do so by using open knot vector with sufficient multiplicity to ensure that the basis functions, and hence the B-spline curve, will be $C^{p-m} = C^0$ -continuous. A cubic open B-Spline curve is shown in Fig. 2.

2.4. B-SPLINE SURFACE

B-Spline surface $S(\xi, \eta)$ is constructed in the form of tensor product with knot vector in two directions, $T = \{\xi_1, \dots, \xi_{n+p+1}\}$, $H = \{\eta_1, \dots, \eta_{m+q+1}\}$ and $n \times m$ set of control points $B_{i,j}$.

$$(4) \quad S(\xi, \eta) = \sum_{i=1}^n N_{i,p}(\xi) \sum_{j=1}^m M_{j,q}(\eta) B_{i,j}.$$

Here p and q are the degrees of basis functions for $N_{i,p}$ and $M_{j,q}$, respectively.

2.5. NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Non-Uniform Rational B-splines (NURBS) uses non-uniform knot vector for calculating B-spline basis function. NURBS is the projection of B-spline in R^p with

homogeneous control points onto R^{p-1} . NURBS basis functions are defined as follows:

$$(5) \quad R_i^p = \frac{N_{i,p} w_i}{\sum_{i=1}^n N_{i,p} w_i},$$

where w is the corresponding weight and NURBS curve is defined as

$$(6) \quad C = \sum_{i=1}^n R_i^p B_i.$$

where B_i is the control point for B-Spline curve. In a similar way, NURBS surface and solids are obtained. NURBS inherits all the properties of B-Spline. If all the weights are equal to one, NURBS becomes B-Splines. The continuity and supports of NURBS basis functions are same as that of B-Splines.

3. ISOGEOMETRIC ANALYSIS FORMULATION

In this section, brief introduction to isogeometric analysis formulation is given. As mentioned earlier, the idea of IGA is to use B-spline basis function for representing both geometry and field variable. This is opposite of iso-parametric formulation of standard finite element method procedure, where the basis functions derived for the field variable are used for approximating the geometry as well.

3.1. RELEVANT SPACES INVOLVED IN IGA

In classical finite element analysis, different domains involved are physical mesh, physical elements and parent domain. The physical mesh is where the geometry is represented with the help of nodes and elements. The physical mesh is divided into non-overlapping physical elements. The parent element is where integration is performed by utilizing Gaussian quadrature rule. All physical elements are mapped to the same parent element. The physical elements are defined by the nodal coordinates, and the degrees of freedom are the values of the basis function at the nodes. Due to compact support, the local basis functions only have support on neighbouring elements. The basis functions are interpolating the nodes and are often called shape functions. In isogeometric analysis, different working domains are physical mesh, control mesh, parameter space and parent element, all shown in Fig. 3.

The physical space is where the actual geometry is represented by a linear combination of the basis functions and the control points. The basis functions are usually not interpolating the control points. The physical mesh is a decomposition of the geometry and can be divided into elements in two different ways; either divided it by

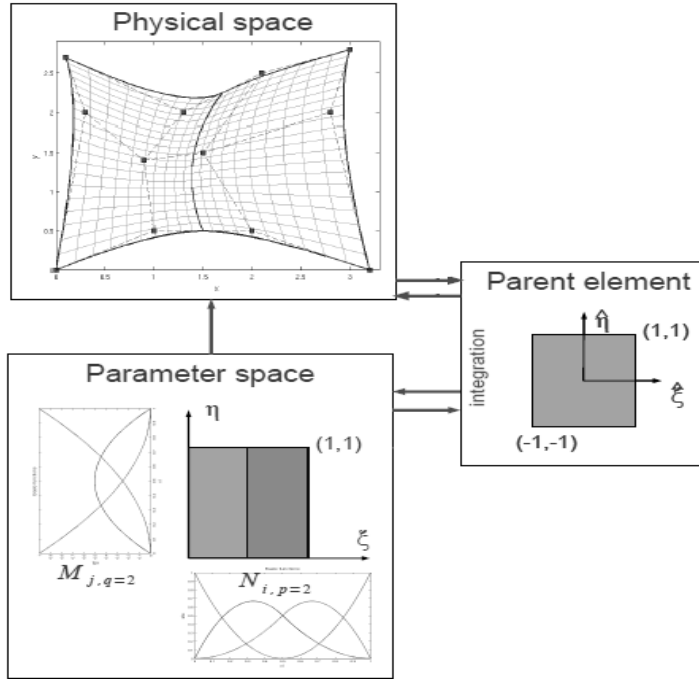


Fig. 3. Different spaces in IGA.

patches or by knot spans. A patch can be thought of as a sub-domain. Patches are curves in 1D, surfaces in 2D and volumes in 3D. Each patch can again be divided into knot spans.

The control mesh is defined by the control points. The control mesh interpolates all control points. It controls the geometry, but generally does not coincide with the physical mesh. In 1D, the control elements are straight lines between two control points. In 2D, the mesh consists of bi-linear quadrilaterals, defined by four control points. In 3D, the control elements are tri-linear hexahedral, defined by eight control points. The control variables are located at the control points and are the degrees of freedom. Control meshes may be severely distorted, while the physical geometry still remains well defined.

The parameter space is where the NURBS basis functions are defined and where the local elements are given by the knots. The parameter space is local to patches. Parameter space usually ranges from 0 to 1. The relation for the mapping between physical mesh and parameter space is given in Eq. (4).

The parent element is the constant area $[-1,1] \times [-1,1]$ and where numerical integration is performed. We map ξ and η in the parameter space to $\hat{\xi}$ and $\hat{\eta}$ in the

parent element to make it easier to exploit Gaussian quadrature. The mapping from the parent element to the parameter space is given by

$$(7) \quad \begin{aligned} \xi &= \frac{1}{2}[(\xi_{i+1} - \xi_i)\hat{\xi} + (\xi_{i+1} + \xi_i)], \\ \eta &= \frac{1}{2}[(\eta_{j+1} - \eta_j)\hat{\eta} + (\eta_{j+1} + \eta_j)]. \end{aligned}$$

3.2. REFINEMENT TECHNIQUES

The B-Spline basis functions can be enriched by three different types of refinement techniques. These are knot insertion, degree elevation and k-refinement. The first two techniques are equivalent to h- and p-refinement of FEM, respectively; the last one has no equivalent in standard FEM. It is to be noted, that after the refinement, the curve or surface remains same geometrically and parametrically.

KNOT INSERTION

This is similar to h-refinement. In this technique, additional knot is inserted in knot vector, which results in an extra knot span. Knot values, which are present in the knot vector, can also be repeated in this way, thereby increasing its multiplicity, but results in reduced continuity of the basis functions. Since knot insertion splits existing elements into new ones, it is similar to h-refinement. However, it differs in the number of new basis functions, that are created and also inter-element continuity.

DEGREE ELEVATION

This is equivalent to p-refinement. In this method, the degree of the basis function, used for representing geometry is increased. Since the B-spline basis has C^{p-m} continuity between two elements, the multiplicity of the knots should also be increased to preserve continuity in B-spline. So, in order elevation, the knot multiplicity value is increased by one, but no new knot value is added. As with knot insertion, neither the parameterization nor the geometry is changed.

K-REFINEMENT

A potentially more powerful type of refinement technique, which is unique to the B-spline basis function is k-refinement. Basically, k-refinement is a degree elevation strategy, which takes advantage of the fact that degree elevation and knot insertion do not commute. In k-refinement, a unique knot value is added between two distinct knot values in a B-spline curve of order p , and afterwards degree of B-spline is elevated to q . The reason for degree elevation from p to q is that the multiplicity of every knot value is increased, so that discontinuity in the p^{th} derivative of the basis

functions is preserved. That is, the basis function still has $p - 1$ continuous derivatives, even though the polynomial order is now q . The number of elements formed in k-refinement is much less as compared to the p-refinement. Also, additional smoothness in k-refinement is helpful in problems like free vibration of structures and bifurcation buckling of thin beams, plates and shells.

3.3. BOUNDARY CONDITIONS

As in finite element analysis, there are two types of boundary conditions: Dirichlet boundary condition and Neumann boundary conditions. Boundary conditions imposed on primary unknown variable (Ex: deformation, temperature, etc.) are known as Dirichlet boundary conditions. Neumann boundary conditions are imposed on the derivative of primary variable (Ex: slope, heat flux, etc.). Boundary conditions of the form $u = 0$ are called homogeneous Dirichlet boundary conditions, where u can be any primary variable. These types of conditions are enforced by assigning the corresponding control variables as zeros. Boundary conditions of the form $u = u_1$ are called non-homogeneous Dirichlet boundary conditions. These conditions can be also imposed by setting the corresponding control variables as u_1 . Assuming open knot vectors, both types of Dirichlet boundary conditions can be satisfied if control variables are at free end or at corner points due to the Kronecker delta property.

If the Dirichlet boundary conditions are to be imposed at any other point (other than end points/ curves) of the domain, special techniques are used namely penalty method, Lagrange multiplier method and least squares minimization. Alternative way is to use h-refinement over the boundary of the domain over which Dirichlet boundary condition is to be imposed. This method is simple to implement, but sometime results in minor error because boundary conditions are satisfied partially. Imposition of Neumann boundary condition in IGA is the same as in FEA, these conditions are naturally satisfied in the weak form.

3.4. FLOWCHART

The flowchart of the conventional finite element code is shown in Fig. 4. An existing finite element code can be converted to single-patch isogeometric analysis code by slight modifications in the boxes shown in grey colour. Since, geometry is defined with the use of NURBS, control points will be the input in place of nodes. The connectivity information will change as NURBS basis functions differs from Lagrange's basis functions and very few basis functions are active in a given knot span. The derivative of the NURBS basis functions depends on the Cox-de-Boor recursive formula. Also, NURBS basis function are used for post-processing of the results.

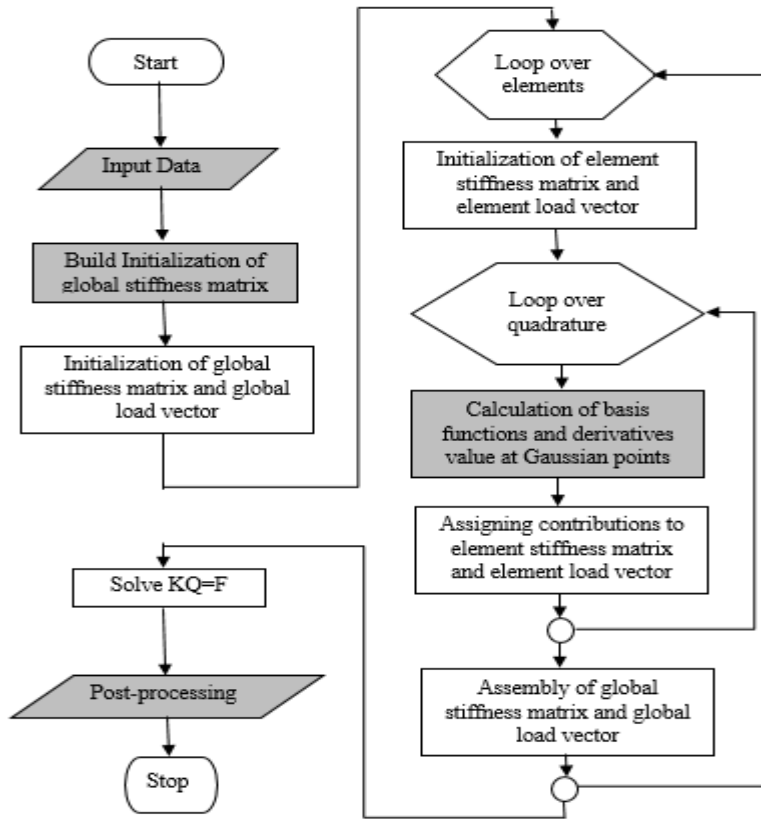


Fig. 4. Flowchart of FEM code and its modification to suite IGA.

4. STATIC STRUCTURAL ANALYSIS

In this section, static structural analysis of bar with self-weight and 2-D plate under plane stress condition is solved. The basic procedure is explained step wise using corresponding MATLAB implementation.

4.1. BAR WITH SELF-WEIGHT

The problem of vertically hanging bar with self-weight is considered to explain Isogeometric analysis methodology for solving one dimensional problems. The governing differential equation for the above problem is given by

$$\frac{d}{dx} \left(E(x)A(x) \frac{du}{dx} \right) = \rho g,$$

where E is Young's modulus, A is the area of the cross section, ρ is the density of the bar, g is the acceleration, due to gravity and u is the field variable. Young's modulus, cross sectional area and product of density and acceleration, due to gravity is taken as unity. Since, the cross sectional area and Young's modulus is constant (not varying along the length), it can be taken out of the differential term.

In case of one dimensional problem the basis functions for finite element analysis and Isogeometric analysis for linear elements are the same. To provide more clarity on IGA methodology, the above problem is modelled using quadratic B-spline with two knot spans. The B-spline data, used for solving the problem is given in Table 1.

Table 1. B-spline data for Bar problem

Degree of Basis function	2
Control Points (x_i, y_i)	[0,0], [0.5,0], [1.5,0], [2,0]
Knot vector	[0 0 0 0.5 1 1 1]
No. of knot spans	2

4.1.1. BASIS FUNCTION

The bar is modelled using B-spline curve of second degree with four control points is given in Table 1, which results in two knot spans. B-spline basis functions are calculated, using the recursive relations given in Eqs. (1) and (2). Matlab function basisfun to find the basis functions of B-spline is presented in Appendix A. The input for the function is order of the B-spline (p) and number of control points (n). The output will be $p + 1$ number of basis functions for each knot span. For the above case,

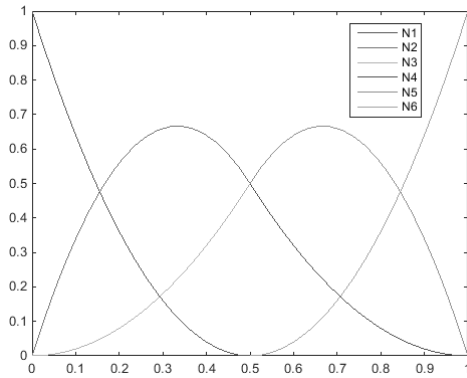


Fig. 5. IGA Basis functions

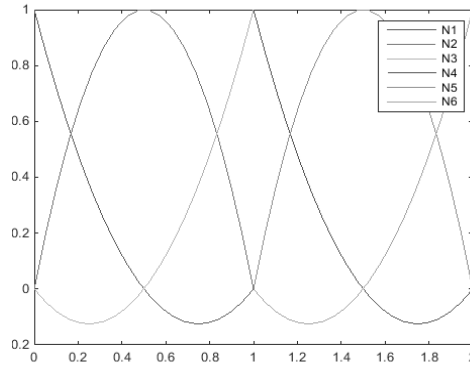


Fig. 6. FEA Basis functions

total of six quadratic basis functions are generated — three basis functions for each of two knot spans, as given in Table 2. B-spline basis functions, calculated for two knot spans are plotted in Fig. 5. To compare with the basis functions used in FEA, three noded quadratic basis functions for two elements are shown in Fig. 6.

Table 2. Basis functions for Bar problem

Knot span	Basis functions		
[0 0.5]	$N_1 = (2\xi - 1)^2,$	$N_2 = -2\xi (3\xi - 2),$	$N_3 = 2\xi^2$
[0.5 1]	$N_4 = (2\xi - 2)(\xi - 1),$	$N_5 = -6\xi^2 + 8\xi - 2,$	$N_6 = (2\xi - 1)^2$

4.1.2. ELEMENT STIFFNESS MATRIX

The element stiffness matrix is calculated by solving the weak form of the governing differential equation and considering the bilinear and linear form separately. The weak form of the above mention problem is given by

$$(8) \quad AE \int_{x_a}^{x_b} \left(\frac{dw}{dx} \frac{du}{dx} \right) dx - AE w \frac{du}{dx} \Big|_{x_a}^{x_b} = \int_{x_a}^{x_b} (\rho g w) dx,$$

where w is the weighing function. Now, considering the first term for finding the element stiffness matrix

$$(9) \quad k_e = AE \int_{x_a}^{x_b} \left(\frac{dw}{dx} \frac{du}{dx} \right) dx.$$

The element stiffness matrix for the first knot span, i.e. [0 0.5] is calculated by using first three B-spline basis functions (N_1, N_2 & N_3). Let first basis function N_1 be weighing function, then Eq. (9) becomes

$$(10) \quad k_{11} = AE \int_0^{0.5} \left(\frac{d(2\xi - 1)^2}{dx} \left(\frac{d(2\xi - 1)^2}{dx} u_1 + \frac{(-2\xi(3\xi - 2))}{dx} u_2 + \frac{2\xi^2}{dx} u_3 \right) \right) dx.$$

Since, B-Spline Basis functions are defined in parametric domain, Jacobian is to be calculated to transform it from physical domain to parametric domain. A point on the B-spline curve is represented by linear combination of basis functions and control points

$$x = N_i x_i + N_{i+1} x_{i+1} + \dots + N_{p+1} x_{p+1}.$$

The first knot span consists of first three basis functions (N_1, N_2, N_3) with the corresponding three control points (x_1, x_2, x_3), so point any point on first knot span is given by

$$(11) \quad x = N_1x_1 + N_2x_2 + N_3x_3.$$

Now, by differentiation

$$(12) \quad \frac{dx}{d\xi} = \frac{dN_1}{d\xi}x_1 + \frac{dN_2}{d\xi}x_2 + \frac{dN_3}{d\xi}x_3,$$

upon substituting values of x_1, x_2 and x_3 , Eq. (12) becomes

$$(13) \quad J_1 = \frac{dx}{d\xi} = 2.$$

Also, by using partial derivative, it can be written as

$$(14) \quad \frac{dN}{dx} = \frac{\partial N}{\partial \xi} \frac{\partial \xi}{\partial x} = \frac{\partial N}{\partial \xi} J_1^{-1}.$$

Substituting these values in Eq. (10) and by integrating between limits $[0, 0.5]$, it gives

$$(15) \quad k_{11} = 1.333u_1 - u_2 - 0.333u_3.$$

Similarly, by taking N_2 and N_3 as weight functions

$$(16) \quad k_{12} = -u_1 + u_2,$$

$$(17) \quad k_{13} = -0.333u_1 + 0.333u_3.$$

By combining Eq. (15), Eq. (16) and Eq. (17), the element stiffness matrix for the first knot span is given by

$$(18) \quad k_1 = \begin{bmatrix} 1.333 & -1 & -0.333 \\ -1 & 1 & 0 \\ -0.333 & 0 & 0.333 \end{bmatrix}.$$

Similarly, element stiffness matrix for knot span $[0.5 \ 1]$ is given by

$$(19) \quad k_2 = \begin{bmatrix} 0.333 & 0 & -0.333 \\ 0 & 1 & -1 \\ -0.333 & -1 & 1.333 \end{bmatrix}.$$

Matlab code *element_stiffness_ID* is presented in Appendix B to find the element stiffness matrix. The output of the code is a cell of size (cp, I) with each cell element contains element stiffness matrix for a given knot span.

4.1.3. ASSEMBLY OF GLOBAL STIFFNESS MATRIX

In finite element method, the continuity between the two elements is always C^0 , i.e. point continuity between the elements. But in isogeometric analysis, C^{p-m} continuity is maintained, where p is the degree of the B-spline and m is the multiplicity of the knot. For example, let us consider a case of quadratic bar element. In FEA, elements are C^0 continuous, so two neighbouring elements share only one node as shown in Fig. 7. But, in the case of IGA, elements are C^1 continuous, thus two neighbouring elements share two control points of each knot span, as shown in Fig. 8.

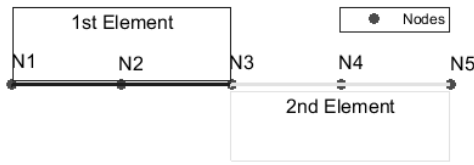


Fig. 7. FEA assembly.

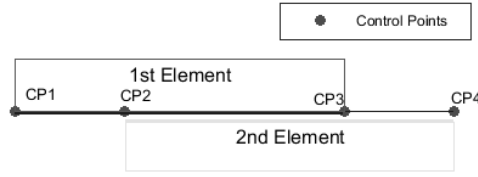


Fig. 8. IGA assembly.

Now for the case of bar, second and third control points are active in both first and second element. The resultant global stiffness matrix for the bar is given by

$$(20) \quad K = \begin{bmatrix} 1.333 & -1 & -0.333 & 0 \\ -1 & 1.333 & 0 & -0.333 \\ -0.333 & 0 & 1.333 & -1 \\ 0 & -0.333 & -1 & 1.333 \end{bmatrix}.$$

Matlab code *global_stiffness* to calculate the global stiffness matrix from the element stiffness matrix is mentioned in Appendix C. The inputs for the code are element stiffness matrices in cell format, obtained from the previous code (*element_stiffness_ID*). The output will be the global stiffness matrix of size (n, n) .

Similarly, the global load vector from the element force vectors is calculated. The global load vector for this case is given by

$$(21) \quad F = [0.333 \quad 0.667 \quad 0.667 \quad 0.333]^T.$$

4.1.4. BOUNDARY CONDITION AND SOLUTION

Elimination method used in IGA for imposing the boundary conditions is the same as in FEA. The columns and rows representing dofs of both global stiffness matrix and global load vector are eliminated. The required field variable, i. e. displacement for the bar is calculated by solving reduced global stiffness matrix and reduced global

load vector. The system of linear equations is solved by using Matlab in-built command *mldivide* (\backslash). The displacement vector, which consists of displacement values at knots for the above mentioned bar problem is shown below:

$$(22) \quad u = [0 \quad 1.0 \quad 2.0 \quad 2.0]'.$$

It is to be noted, that the solution vector U is not equal to nodal solution of FEM. The way, control points define the curve along with basis function, though they themselves are not on the curve. The displacements values at knots define the curve (solution). Hence, these may be interpreted as control points of the displacement curve.

For validation of IGA methodology, IGA results are compared with the exact solution for displacement and stress, which are given by Eq. (23) and (24), respectively

$$(23) \quad u(x) = \frac{\rho g}{E} \left(Lx - \frac{x^2}{2} \right),$$

$$(24) \quad \sigma(x) = \rho g(L - x).$$

The same problem is solved using finite element method for comparison. The bar is modelled using two quadratic elements and the nodal solution is given by

$$U = [0 \quad 0.875 \quad 1.5 \quad 1.875 \quad 2]'$$

Displacement results, obtained from IGA and FEM with analytical solution are plotted in Figs. 9 and 10, respectively. The stress developed in the bar for both cases are shown in Figs. 11 and 12. It can be observed from the figures that the results obtained from IGA for bar with self-weight is in a good agreement with that one of

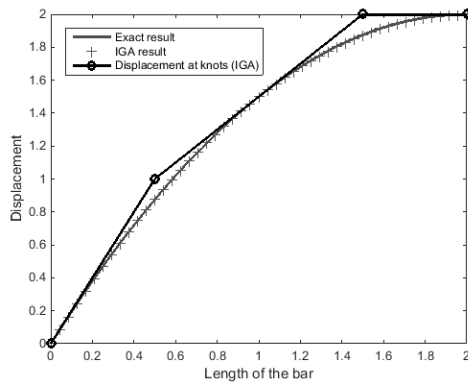


Fig. 9. IGA displacement result.

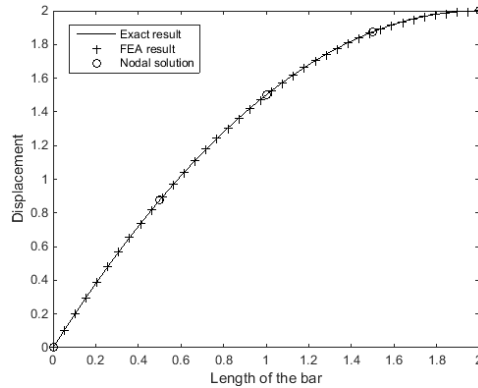


Fig. 10. FEA displacement result.

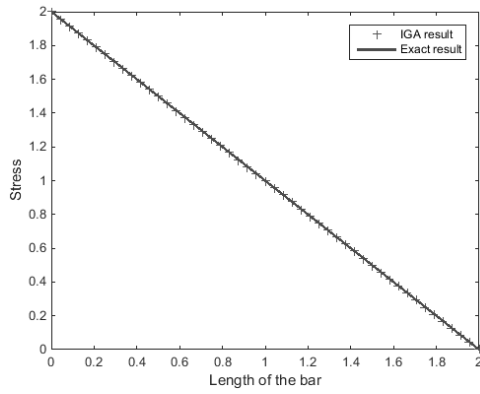


Fig. 11. IGA stress result.

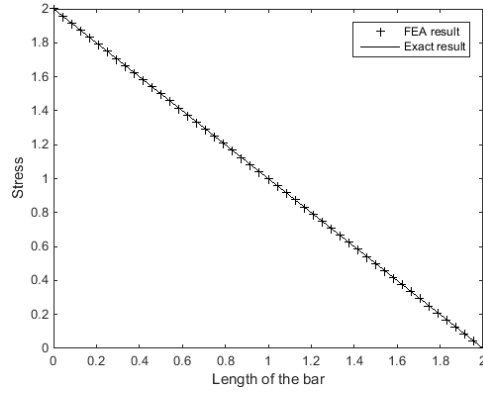


Fig. 12. FEA stress result.

the exact solution. Also, it can be observed that two quadratic elements are required in order to capture the results accurately using FEM. Two quadratic FEM elements results in five nodes, which is one greater than that of IGA control points.

4.2. PLATE PROBLEM

To explain the case of two dimension problems in IGA, a square plate made up of steel material (Young's modulus of 210×10^9 N/m² and Poisson's ratio is 0.33) with unit length and breadth is taken under consideration. The traction force of 10 kN/m is applied at both ends in x -direction. The plate is modelled by taking two linear elements in both u and v directions. Since two linear elements are considered, three control points are required along each parametric direction, which results in a total nine control points. The B-spline surface data is mentioned in Table 3. The plate

Table 3. B-spline surface data

Degree of basis function in u direction	1
Degree of basis function in v direction	1
Control Points (x_i, y_i)	[0,0],[0.5,0], [1,0], [0,0.5], [0.5,0.5], [1,0.5], [0,1], [0.5,1], [1,1]
Knot vector in u direction	[0 0 0.5 1 1]
Knot vector in v direction	[0 0 0.5 1 1]
No. of elements in u direction	2
No. of elements in v direction	2
Total no. of elements	4

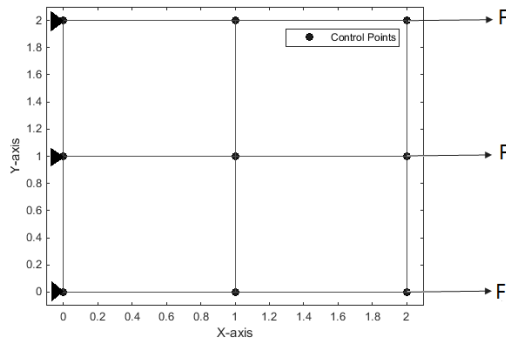


Fig. 13. Square plate problem.

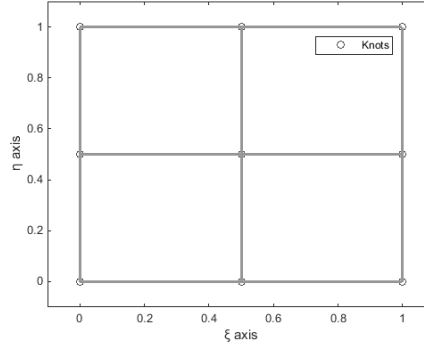


Fig. 14. Parameter space.

problem is shown in Fig. 13, corresponding parameter space of the plate with knots is shown in Fig. 14.

4.2.1. BASIS FUNCTIONS

The plate is modelled as bilinear B-spline surface. There will be two set of basis functions, one in ξ (N_i) and another in η (M_i) direction. The method to find the basis functions for B-spline surface is similar to that of B-spline curve. But in 2-D case, basis functions are calculated in two directions. Basis functions are evaluated with the use of same Matlab function basisfun, as used in previous case. Basis functions calculated for the plate problem are given in Table 4.

Table 4. Basis functions for B-spline surface

Knot span	Basis functions			
[0 0.5]	$N_1 = 1 - 2\xi$,	$N_2 = 2\xi$,	$M_1 = 1 - 2\eta$,	$M_2 = 2\eta$
[0.5 1]	$N_3 = 2 - 2\xi$,	$N_4 = 2\xi - 1$,	$M_3 = 2 - 2\eta$,	$M_4 = 2\eta - 1$

4.2.2. ELEMENT STIFFNESS MATRIX

The element stiffness matrix for two dimensional problems using theory of minimum potential energy is calculated by

$$(25) \quad k_e = \int_{\Omega} B^T D B d\Omega,$$

where D is the elasticity matrix, which depends on the plane stress or plane strain conditions and B is the strain-displacement matrix, which is given by

$$(26) \quad B = \begin{bmatrix} R_x & 0 \\ 0 & R_y \\ R_y & R_x \end{bmatrix},$$

where R_x and R_y are the derivatives of basis function with respect to x and y , respectively. For example, R vector for the first element $[(0 \leq \xi < 0.5) \& (0 \leq \eta < 0.5)]$ is given by below equation:

$$(27) \quad R = [N_1 M_1 \quad N_2 M_1 \quad N_1 M_2 \quad N_2 M_2].$$

The domain of Gaussian points for the numerical integration is $[-1, 1]$. But the parametric domain range is $[0, 1]$. So, the Gaussian points are mapped from parent domain to the parametric domain for numerical integration. The relation for the mapping is given by Eqs. (5) and (6).

If two point Gaussian numerical integration is considered, than the Gaussian points are $[-0.577 \ 0.577]$ and weights are $[1 \ 1]$. For first Gaussian point of first element $[(0 \leq \xi < 0.5) \& (0 \leq \eta < 0.5)]$, $\bar{\xi} = -0.577$ and $\bar{\eta} = -0.577$, the corresponding point in parameter domain from Eqs. (5) and (6) is given by $\xi = 0.3943$ and $\eta = 0.3943$.

The Jacobian for the mapping of the derivatives from parent domain to the parametric domain is given by

$$(28) \quad |J_{\bar{\xi}, \bar{\eta}}| = \frac{1}{4}(\xi_{i+1} - \xi_i)(\eta_{j+1} - \eta_j).$$

Using the above relation, Jacobian for the first element is

$$(29) \quad |J_{\bar{\xi}, \bar{\eta}}|_1 = \frac{1}{4} (0.5 - 0) (0.5 - 0) = 0.0625.$$

Next step is to find the derivative of basis functions with respect to two parameters (ξ, η) at the Gaussian points. The derivative of basis function for first element with respect to ξ and η is given by

$$\begin{aligned} [R_\xi]_1 &= [4\eta - 2 \quad 2 - 4\eta \quad -4\eta \quad 4\eta], \\ [R_\eta]_1 &= [4\xi - 2 \quad -4\xi \quad 2 - 4\xi \quad 4\xi]. \end{aligned}$$

The value of derivative at first Gaussian point $[\xi = 0.3943 \ \eta = 0.3943]$ for numerical integration is given by

$$\begin{aligned} [R_\xi]_1 &= [-0.4228 \quad 0.4228 \quad -1.5772 \quad 1.5772], \\ [R_\eta]_1 &= [-0.4228 \quad 0.4228 \quad -1.5772 \quad 1.5772]. \end{aligned}$$

The derivatives calculated above are in parameter domain, but from Eq. (25) it can be seen that the derivative of basis functions should be in physical domain. The Jacobian for the mapping of the derivatives from parametric domain to the physical domain is given by

$$(30) \quad x = N_1 M_1 x_1 + N_2 M_1 x_2 + N_2 M_2 x_3 + N_1 M_2 x_4,$$

$$(31) \quad \frac{\partial x}{\partial \xi} = \frac{\partial}{\partial \xi} [N_1 M_1 \quad N_2 M_1 \quad N_2 M_2 \quad N_1 M_2] [x_1 \quad x_2 \quad x_3 \quad x_4]',$$

$$(32) \quad \frac{dx}{d\xi} = 1, \quad \frac{dx}{d\eta} = 0.$$

Similarly, $\frac{dy}{d\xi} = 0, \frac{dy}{d\eta} = 1$. Jacobian matrix for first knot span is given by

$$(33) \quad [J_{\xi,\eta}]_1 = \begin{bmatrix} \frac{dx}{d\xi} & \frac{dx}{d\eta} \\ \frac{dy}{d\xi} & \frac{dy}{d\eta} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Now the derivatives of shape function, with respect to physical co-ordinates is calculated by multiplying the inverse Jacobian to derivative of shape function with respect to parametric domain.

$$(34) \quad R_x = [R_\xi \quad R_\eta] J_{\xi,\eta}^{-1},$$

$$[R_x]_1 = [-0.4228 \quad 0.4228 \quad -1.5772 \quad 1.5772],$$

$$[R_y]_1 = [-0.4228 \quad 0.4228 \quad -1.5772 \quad 1.5772].$$

So, the strain displacement matrix for the first element of the plate is given by

$$(35) \quad B_e = \begin{bmatrix} [R_1]_x & [R_2]_x & [R_3]_x & [R_4]_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & [R_1]_y & [R_2]_y & [R_3]_y & [R_4]_y \\ [R_1]_y & [R_2]_y & [R_3]_y & [R_4]_y & [R_1]_x & [R_2]_x & [R_3]_x & [R_4]_x \end{bmatrix},$$

$$B_1 = \begin{bmatrix} -0.4228 & 0.4228 & -1.5772 & 1.5772 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.4228 & -1.5772 & 0.4228 & 1.5772 \\ -0.4228 & -0.4228 & -1.5772 & 0.4228 & 0.4228 & -1.5772 & 1.5772 & 1.5772 \end{bmatrix}.$$

The elasticity matrix for a plate under plane stress is given by

$$(36) \quad C = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix},$$

where ν is the Poisson's ratio, which is taken as 0.33. The element stiffness matrix is calculated by substituting the value of elasticity matrix and strain-displacement matrix in Eq. (25), which is given by

$$(37) \quad k_e = B'CB |J_{\xi,\eta}| |J_{\bar{\xi},\bar{\eta}}| wt,$$

where wt is the weight of the Gaussian point, which is considered one for all points. Matlab code *element_stiffness_2D* to find the element stiffness matrix for two dimensional using IGA is mentioned in Appendix D. The output of the code for the above mentioned plate problem is the cell of size (4, 1), each cell element contains element stiffness matrix for the corresponding element.

To compute the load vector, a boundary mesh is defined over which a traction is applied. Boundary mesh for this case consists of two linear elements. Element load vector under traction is given by relation in

$$(38) \quad f_e = \int_{\Gamma} N' t d\Gamma,$$

where t is the traction force, which is taken as 10 kN/m applied over the right edge. Numerical integration over the line integral is calculated by using three Gaussian points. The values of Gaussian points and their associate weights are

$$[\text{Points} \quad \text{Weights}] = \begin{bmatrix} 0.7746 & 0.5556 \\ -0.7746 & 0.5556 \\ 0 & 0.5556 \end{bmatrix}.$$

4.2.3. ASSEMBLY OF GLOBAL STIFFNESS MATRIX

The procedure used for stepped bar problem can be extended to two dimensional case. The control mesh with corresponding element number of the square plate is

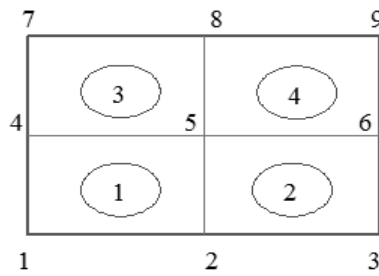


Fig. 15. Control mesh of square plate.

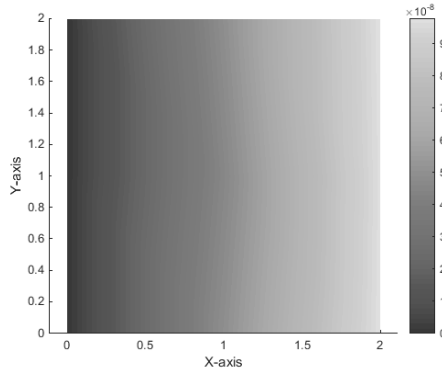
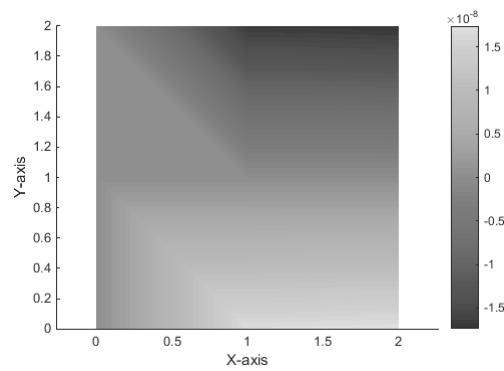
shown in Fig. 15. Each linear element matrix will have active set of four control points and each control point has two degrees of freedom (translation in x and y directions) as mentioned in Table 5. The first four degrees of freedom are along x direction and remaining four along y direction. The global stiffness matrix will be of the size $(2n, 2n)$. Same Matlab code `global_stiffness` with integration of element connectivity table can be used for the assembly process.

Table 5. Connectivity table

Element No.	Connectivity	Dofs
1	[1 2 4 5]	[1 2 4 5 10 11 13 14]
2	[2 3 5 6]	[2 3 5 6 11 12 14 15]
3	[4 5 7 8]	[4 5 7 8 13 14 16 17]
4	[5 6 8 9]	[5 6 8 9 14 15 17 18]

4.2.4. SOLUTION

The unknown variables to be calculated in this problem are displacement in x and y directions. Since left edge of the square plate is fixed, homogenous Dirichlet boundary condition is imposed. As mentioned earlier in section 3, imposition of homogenous Dirichlet boundary condition is similar to FEA and easy to impose. In the present case, rows and columns of global stiffness matrix representing degree of freedoms of left edge of the plate are eliminated. Similarly, the rows of fixed degree of freedom of the global load vector are eliminated. The required solution is obtained by solving a set of linear equations obtained from reduced global stiffness matrix and reduced global load vector.

Fig. 16. Displacement in x direction.Fig. 17. Displacement in y direction.

The displacement in the plate along x and y direction are shown in Fig. 16 and Fig. 17, respectively. The maximum displacement of the plate in x direction is 9.2×10^{-8} m and in y direction it is -1.5×10^{-8} m.

5. STRUCTURAL VIBRATION ANALYSIS

In this section, structural vibration problem of bar and beam are solved. The basic procedure is explained in steps using corresponding Matlab implementation.

5.1. NATURAL VIBRATION OF BAR

A bar of unit length which is fixed at both ends is considered to explain isogeometric analysis methodology. Material properties of the bar are assumed to be unity. First analytical solution is provided for the defined case. Results obtained from analytical solution and IGA solution are compared.

5.1.1. ANALYTICAL SOLUTION

The natural frequencies and modes are governed by

$$(39) \quad M\ddot{x} + Kx = 0,$$

where M and K are mass and stiffness matrices, respectively, x is the displacement vector and \ddot{x} is the acceleration vector. The general solution for the defined problem is given by

$$(40) \quad x = \phi_n e^{i\omega_n t},$$

where ω_n is the n^{th} natural frequency. Substituting the value of x from Eq. (39) into Eq. (40), we will get

$$(41) \quad (K - \omega_n^2 M)\phi_n = 0.$$

The bar is fixed at both the ends, i.e. $x(0) = x(1) = 0$. The exact solution for these conditions in terms of natural frequencies is

$$(42) \quad \omega_n = n\pi, \quad \text{where } n = 1, 2, 3, \dots$$

5.1.2. BASIS FUNCTION

The bar is modelled with the use of quadratic B-Spline basis functions with equi-spaced control points. B-Spline basis functions are calculated by using recursive functions given in Eqs. (1) and (2). Matlab function `basisfun` is used to find B-Spline basis functions. In order to understand the problem more precisely, a bar with three knot spans is taken. Related data is presented in Table 6.

Basis functions for each knot span are presented in Table 7.

Table 6. B-Spline Data for Bar problem

Degree of B-spline	3
Number of elements	3
Knot vector	[0 0 0 0.333 0.667 1 1 1]
Control Points (x_i, y_i)	[0, 0], [0.25, 0], [0.5, 0], [0.75, 0], [1, 0]

Table 7. B-Spline basis function for Bar problem

Knot span	Basis function
[0 0.333]	$N_1 = (3\xi - 1)^2$, $N_2 = \frac{-(3\xi(9\xi - 4))}{2}$, $N_3 = \frac{9\xi^2}{2}$
[0.333 0.667]	$N_4 = \frac{(3\xi - 2)^2}{2}$, $N_5 = \frac{-(3(6\xi^2 - 6\xi + 1))}{2}$, $N_6 = \frac{(3\xi - 1)^2}{2}$
[0.667 1]	$N_7 = \frac{(3\xi - 3)^2}{2}$, $N_8 = \frac{-(3(9\xi - 5)(\xi - 1))}{2}$, $N_9 = (3\xi - 2)^2$

5.1.3. ELEMENT STIFFNESS MATRIX

Same procedure to find element stiffness matrix is used as mentioned in bar with self-weight problem. The Jacobian for mapping the derivatives from parametric domain to physical domain is given by

$$(43) \quad J_1 = \frac{\partial x}{\partial \xi} = \frac{3}{2} - \frac{9\xi}{4}.$$

Now, by using partial derivative it can be written as

$$(44) \quad \frac{\partial N}{\partial x} = \frac{\partial N}{\partial \xi} \frac{\partial \xi}{\partial x} = \frac{\partial N}{\partial \xi} \left(\frac{3}{2} - \frac{9\xi}{4} \right)^{-1}.$$

On substituting the Jacobian and derivative value in Eq. (9), element stiffness matrix for first knot span is calculated (Appendix B)

$$(45) \quad K_1 = EA \begin{bmatrix} 3.0904 & -2.1807 & -0.9096 \\ -2.1807 & 2.3614 & -0.1807 \\ -0.9096 & -0.1807 & 1.0904 \end{bmatrix}.$$

Similarly, stiffness matrix for second and third element are

$$K_2 = EA \begin{bmatrix} 1.3333 & -0.667 & -0.667 \\ -0.667 & 1.3333 & -0.667 \\ -0.667 & -0.667 & 1.3333 \end{bmatrix},$$

$$K_3 = EA \begin{bmatrix} 1.0904 & -0.1807 & -0.9096 \\ -0.1807 & 2.3614 & -2.1807 \\ -0.9096 & -2.1807 & 3.0904 \end{bmatrix}.$$

5.1.4. GLOBAL STIFFNESS MATRIX

Assembly of global stiffness matrix is similar as that of static analysis. The global stiffness is calculated using Matlab code `global_stiffness` which is given in Appendix C,

$$K = EA \begin{bmatrix} 3.0904 & -2.1807 & -0.9096 & 0 & 0 \\ -2.1807 & 3.6948 & -0.8474 & -0.667 & 0 \\ -0.9096 & -0.8474 & 3.514 & -0.8474 & -0.9096 \\ 0 & -0.667 & -0.8474 & 3.6948 & -2.1807 \\ 0 & 0 & -0.9096 & -2.1807 & 3.0904 \end{bmatrix}.$$

5.1.5. ELEMENT MASS MATRIX

The relation of element mass matrix for an element is given by

$$(46) \quad M = \int_0^1 [N]^T [N] \rho A dx.$$

On substituting values of basis functions in Eq. (46), we get the mass terms for various elements as follows:

$$M_1 = \rho A \begin{bmatrix} 11/120 & 23/480 & 1/160 \\ 23/480 & 47/480 & 1/32 \\ 1/160 & 1/32 & 7/480 \end{bmatrix}, \quad M_2 = \rho A \begin{bmatrix} 1/80 & 13/480 & 1/480 \\ 13/480 & 9/80 & 13/480 \\ 1/480 & 13/480 & 1/80 \end{bmatrix},$$

$$M_3 = \rho A \begin{bmatrix} 7/480 & 1/32 & 1/160 \\ 1/32 & 47/480 & 23/480 \\ 1/160 & 23/480 & 11/120 \end{bmatrix}.$$

5.1.6. GLOBAL MASS MATRIX

Assembly of global mass matrix is obtained in the same manner as that of global stiffness matrix (appendix C) which is given by

$$M = \rho A \begin{bmatrix} 0.0917 & 0.0479 & 0.0063 & 0 & 0 \\ 0.0479 & 0.1104 & 0.0583 & 0.0021 & 0 \\ 0.0063 & 0.0583 & 0.1417 & 0.0583 & 0.0063 \\ 0 & 0.0021 & 0.0583 & 0.1104 & 0.0479 \\ 0 & 0 & 0.0063 & 0.0479 & 0.0917 \end{bmatrix}.$$

5.1.7. SOLUTION

The required solution is obtained by substituting the global element stiffness matrix and mass matrix in Eq. (41). Elimination method is used to impose boundary conditions. Application of boundary conditions results in an Eigen value problem. The solution of Eigen value problem is $\omega_1 = 3.143$ Hz, $\omega_2 = 6.345$ Hz, $\omega_3 = 10.1015$ Hz, which matches with analytical results ($\omega_1 = \pi$, $\omega_2 = 2\pi$, $\omega_3 = 3\pi$). In order to further refine the solution, number of elements is increased to 100. Ratio of the natural frequencies from analytical and IGA solution are plotted against the number of element (appendix E). It is evident from Fig. 18 that more than half of the modal frequencies are equal to those obtained analytically. Moreover, the rest of the frequencies' error is found to be less than 7%.

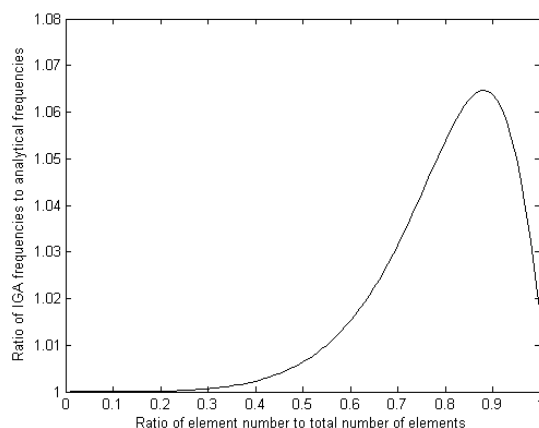


Fig. 18. Natural frequencies ratio against element number.

5.2. NATURAL VIBRATION OF BEAM

A beam fixed at both ends of unit length is considered to find natural modes of vibration, using isogeometric analysis. All material properties are considered to be unity. In finite element method, each node has two degrees of freedom i. e. slope and deflection but in isogeometric method each control point has only one degree of freedom. There is no slope in isogeometric methodology. That is the reason for beam to be called rotation free beam.

5.2.1. ANALYTICAL SOLUTION

The governing equation for finding the natural modes of vibration is as follows:

$$(47) \quad x - \omega^2 x = 0.$$

In terms of stiffness and mass, the equation will be

$$(48) \quad K - \omega^2 M = 0.$$

Boundary conditions are $x(0) = x(1) = 0$. Upon imposition of boundary conditions, exact solution in terms of frequencies will be as follows:

$$(49) \quad \omega_n = (n\pi)^2, \quad \text{where } n = 1, 2, 3, \dots$$

5.2.2. BASIS FUNCTION

The beam is modelled with a cubic B-Spline basis function having equi-spaced control points. B-Spline basis functions are calculated by using recursive functions, given in Eqs. (1) and (2). Basis functions are calculated with the use of Matlab function (appendix 7.1). In order to provide more insight on the topic, a beam of three elements is considered and modal frequencies are calculated with the help of isogeometric methodology. Related information is provided in Table 8.

Table 8. B-Spline data for Beam problem

Degree of B-spline	4
Number of knot spans	3
Knot vector	[0 0 0 0 0.333 0.667 1 1 1 1]
Control Points (x_i, y_i)	[0, 0], [0.2, 0], [0.4, 0], [0.6, 0], [0.8, 0], [1, 0]

Basis functions are generated using Matlab function *basisfun* (appendix 7.1) from above mentioned B-spline data and are given in Table 9.

Table 9. B-Spline basis function for Beam problem

Knot span	Basis function
[0 0.333]	$N_1 = (1 - 3\xi)^3,$ $N_2 = [(9\xi(21\xi^2 - 18\xi + 4))]/4,$ $N_3 = [-(9\xi^2(11\xi - 6))]/4,$ $N_4 = 9\xi^3/2$
[0.333 0.667]	$N_5 = [-(3\xi - 2)^3]/4,$ $N_6 = [3(21\xi^3 - 36\xi^2 + 18\xi - 2)]/4,$ $N_7 = [-3(21\xi^3 - 27\xi^2 + 9\xi - 1)]/4,$ $N_8 = [(3\xi - 1)^3]/4$
[0.667 1]	$N_9 = [-(3\xi - 3)^2(\xi - 1)]/2,$ $N_{10} = [9(11\xi - 5)(\xi - 1)^2]/4,$ $N_{11} = [-9(\xi - 1)(21\xi^2 - 24\xi + 7)]/4,$ $N_{12} = (3\xi - 2)^3$

5.2.3. ELEMENT STIFFNESS MATRIX

The stiffness matrix is calculated with the use of minimum potential energy method.

In case of beam, strain is given by

$$(50) \quad \epsilon = -y \frac{\partial^2 V}{\partial x^2},$$

where V is the vertical deflection, which is given by

$$(51) \quad V = N_1 v_1 + N_2 v_2 + N_3 v_3 + N_4 v_4 = [N] \{v_e\},$$

where $\{v_e\}$ is deflection vector and strain energy is given by

$$(52) \quad SE = \frac{1}{2} \{v_e\} \int \left(-y \frac{\partial^2 N}{\partial x^2} \right)^T E \left(-y \frac{\partial^2 N}{\partial x^2} \right) A dx \{v_e\}.$$

Since, $y^2 A$ is moment of inertia, i.e. I , Eq. (52) can be re-written as

$$(53) \quad SE = \frac{1}{2} \{v_e\} \int \left(\frac{\partial^2 N}{\partial x^2} \right)^T \left(\frac{\partial^2 N}{\partial x^2} \right) EI dx \{v_e\},$$

and stiffness matrix is given by

$$(54) \quad K = EI \int_0^1 \left(\frac{\partial^2 N}{\partial x^2} \right)^T \left(\frac{\partial^2 N}{\partial x^2} \right) dx.$$

Since, B-Spline Basis functions are defined in parametric domain, Jacobian is to be calculated to transform it from physical domain to parametric domain. For that, displacement vector is given as

$$(55) \quad x = N_1 x_1 + N_2 x_2 + N_3 x_3 + N_4 x_4,$$

$$(56) \quad \frac{\partial x}{\partial \xi} = \frac{9(15\xi^2 - 12\xi + 4)}{20}.$$

Now by using partial derivation

$$(57) \quad \frac{\partial N}{\partial x} = \frac{\partial N}{\partial \xi} = \frac{\partial N}{\partial \xi} \left(\frac{9(15\xi^2 - 12\xi + 4)}{20} \right)^{-1},$$

by substituting above value in Eq. (54), stiffness matrix is given by

$$K_1 = EI \begin{bmatrix} 70.93 & -116.62 & 20.45 & 25.24 \\ -116.62 & 205.03 & -60.20 & -28.20 \\ 20.45 & -60.20 & 59.05 & -19.30 \\ 25.24 & -28.21 & -19.30 & 22.27 \end{bmatrix}.$$

Similarly, stiffness matrices for remaining knot spans are given by

$$K_2 = EI \begin{bmatrix} 48.71 & -67.76 & -10.60 & 29.65 \\ -67.76 & 124.93 & -46.56 & -10.60 \\ -10.60 & -46.56 & 124.93 & -67.76 \\ 29.65 & -10.60 & -67.76 & 48.71 \end{bmatrix},$$

$$K_3 = EI \begin{bmatrix} 22.27 & -19.30 & -28.21 & 25.24 \\ -19.30 & 59.05 & -60.20 & 20.45 \\ -28.21 & -60.20 & 205.02 & -116.62 \\ 25.24 & 20.45 & -116.62 & 70.93 \end{bmatrix}.$$

5.2.4. ASSEMBLY OF GLOBAL STIFFNESS MATRIX

Same procedure is followed for assembly of stiffness matrix for beam as that of bar problem. Matlab function `global_stiffness` is used for the assembly (appendix 7.3)

$$K = EI \begin{bmatrix} 70.93 & -116.62 & 20.45 & 25.24 & 0 & 0 \\ -116.62 & 253.74 & -127.96 & -38.81 & 29.65 & 0 \\ 20.45 & -127.96 & 206.25 & -85.16 & -38.81 & 25.23 \\ 25.24 & -38.81 & -85.16 & 206.25 & -127.96 & 20.45 \\ 0 & 29.65 & -38.81 & -127.96 & 253.74 & -116.62 \\ 0 & 0 & 25.24 & 20.45 & -116.62 & 70.93 \end{bmatrix}.$$

5.2.5. ELEMENT MASS MATRIX

The mass matrix for beam knot span will be the same as that of bar. Hence, in order to obtain mass matrix the following equation is used

$$(58) \quad M = \rho A \int_0^1 [N]^T [N] dx.$$

By substituting the value of basis functions in Eq. (58), mass matrices for three knot spans are given by

$$M_1 = \rho A \begin{bmatrix} 0.07 & 0.04 & 0.007 & 0.0004 \\ 0.04 & 0.07 & 0.038 & 0.005 \\ 0.007 & 0.038 & 0.032 & 0.0055 \\ 0.0004 & 0.005 & 0.0055 & 0.001 \end{bmatrix},$$

$$M_2 = \rho A \begin{bmatrix} 0.0021 & 0.0084 & 0.0041 & 0.0001 \\ 0.0084 & 0.049 & 0.04 & 0.004 \\ 0.0041 & 0.04 & 0.049 & 0.0084 \\ 0.0001 & 0.004 & 0.0084 & 0.0021 \end{bmatrix},$$

$$M_3 = \rho A \begin{bmatrix} 0.001 & 0.0055 & 0.0049 & 0.0004 \\ 0.0055 & 0.032 & 0.038 & 0.0075 \\ 0.0049 & 0.038 & 0.0789 & 0.04 \\ 0.0004 & 0.0075 & 0.04 & 0.076 \end{bmatrix}.$$

5.2.6. ASSEMBLY OF GLOBAL MASS MATRIX

The procedure of assembly of global mass matrix is the same as that of global stiffness matrix (Appendix 7.3). Global mass matrix obtained is as follows:

$$M = \rho A \begin{bmatrix} 0.076 & 0.04 & 0.0075 & 0.0004 & 0 & 0 \\ 0.04 & 0.0811 & 0.0472 & 0.0091 & 0.0001 & 0 \\ 0.0075 & 0.0472 & 0.0827 & 0.0511 & 0.0091 & 0.0004 \\ 0.0004 & 0.0091 & 0.0511 & 0.0827 & 0.0472 & 0.0075 \\ 0 & 0.0001 & 0.0091 & 0.0472 & 0.0811 & 0.04 \\ 0 & 0 & 0.0004 & 0.0075 & 0.04 & 0.076 \end{bmatrix}.$$

5.2.7. SOLUTION

Required solution is obtained by substituting stiffness and mass matrix in Eq. (48). Elimination approach is used to impose boundary conditions. The natural frequencies obtained by the isogeometric methodology are $w_1 = 9.94$ Hz, $w_2 = 40.079$ Hz, $w_3 = 92.466$ and $w_4 = 180.042$, which satisfy the analytical solution. In order to refine the solution, number of element is increased to 99. Ratio of the natural frequencies from analytical and IGA solution are plotted against the number of element

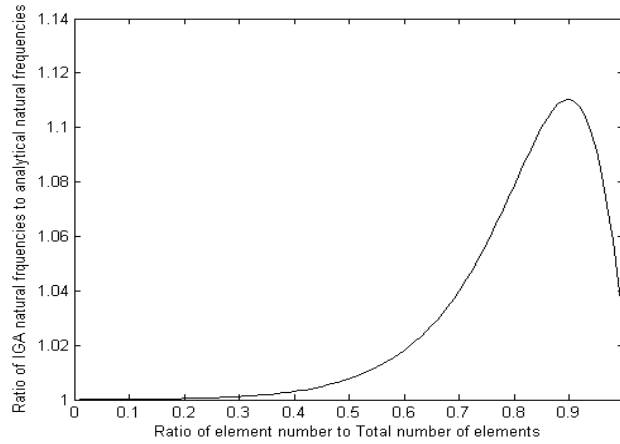


Fig. 19. Natural frequencies ratio against element number.

is plotted against the number of element (appendix 7.6). It can be seen in Fig. 19, that more than half of the modal frequencies are equal to those obtained with the help of analytical solution and even for the rest of the frequencies' error is less than 12%.

6. CONCLUDING REMARKS

NURBS is the predominant modelling technique used in most of the CAD packages. B-spline basis function has unique properties like local support, exact geometry representation, refinement techniques, etc. these are the reasons of its widespread use. Isogeometric analysis is the new analysis framework, which makes use of these NURBS properties. The idea is to use a single description of computational domain for both geometry and field variable representation to facilitate integration of CAD and CAA industries. From the past one decade, IGA has shown a lot of potential to become future alternative to FEM. In this paper, IGA methodology for the static structural and vibration cases is presented. Detailed calculation of each step with corresponding MATLAB implementation is given. It is observed (Ex: bar with self-weight problem) that the displacement solution obtained from IGA is equivalent to control points of B-spline displacement curve. Also, IGA gives better solution per degree of freedom than FEM. Differences of IGA with conventional FEM formulation are also presented in this paper.

Apart from IGA being superior to FEM in many aspects, it has some disadvantages of its own. Since, isogeometric analysis is a NURBS based method, analysts must require a lot of CAD knowledge. Multi-patch B-spline modelling of intricate shapes is a very difficult task. In IGA, exact Dirichlet boundary condition cannot be imposed on interior of the domain. Irrespective of the degree of the basis function and number of elements, the FEM solution at nodes always gives exact solution. But, that's not the case in IGA, because of non-interpolation of B-spline control points. Isogeometric analysis is computationally expensive as compared to FEM for simple one dimensional case.

Isogeometric analysis, being a new analysis technique with lot of advantages, has immense scope in the future. From literature it can be observed that, IGA is applied to relatively simple geometries, it can be applied on complex multi-patch geometries for engineering analysis. Available analysis software use different information of geometry for design and analysis. Further research can be pursued is of integrating isogeometric analysis with CAD systems and hence, avoiding meshing problems related to FEM. Applying isogeometric analysis to contact problems as these problems require smooth representation of surfaces. Also, finding an optimum set of control points for fitting a B-spline model, which gives better IGA result, is still an open problem.

APPENDIX A. **basisfun**

```

function N = basisfun(k,cp)
% Input
% P=input('Order of the B-spline')
% n=input('No of control points')
%Output
% b= Symbolic basis function in u.
% each knot span contains P+1 no of basis functions in cell format.
t=knotseq(cp,k-1);
[q,p]=size (t);
r=k;
v=(t(1,r)+t(1,r+1))/2;
c=cell(n+1,k);e=cell(n+1,1);
syms u;
a=1;b=1;
while r<p-(k-1)
for j=1:k
for i=1:(n+1)
if j==1
if v>=t(1,i)&& v<t(1,i+1)
c{i,j}=1;
else
c{i,j}=0;
end
else
y=0;
dn1=u-t(1,i);
dd1=t(1,i+j-1)-t(1,i);
if dd1 ~= 0
y=c{i,j-1}.*(dn1/dd1);
end
dn2=t(1,i+j)-u;
dd2=t(1,i+j)-t(1,i+1);
if dd2 ~= 0
y=y+c{i+1,j-1}.*(dn2/dd2);
end
c{i,j}=y;
end
end

```

```

end
end
for ad=0:k-1
    e{a+ad,1}=c{b+ad,k};
end
    a=a+k;
    b=b+1;
    r=r+1;
    v=(t(1,r)+t(1,r+1))/2;
end
    e=reshape(e,k,k)';
end

```

APPENDIX B. **element_stiffness_1D**

```

n=input('no of control points');
k=input('order of the B-spline');
cpx=input('control points in x-direction');
E=input('Youngs Modulus vector');
A=input('cross sectional area vector');
N=basisfun_garvit(k,n);
P=k-1;
uniquekntspan=unique(knotseq(n,P));
noel=length(uniquekntspan)-1;
ke=cell(noel,1);
for i=1:noel
    cpx1=cpx(i:i+P);
    x=0;
    for j=1:k
        dummy=N{i,j}*cpx1(j);
        x=dummy+x;
    end
    jacob=diff(x);
    dummy1=zeros(k);
    for j=1:k
        for j1=1:k
            dummy1(j,j1)= E(i)*A(i)*int(diff(N{i,j})*diff(N{i,j1}))/jacob,
            uniquekntspan(i),uniquekntspan(i+1));
        end
    end
end

```



```

end
    ke{i}=double(dummy1);
end

```

APPENDIX C. **global_stiffness**

```

K=zeros(n);
dummy=zeros(n);
for i=1:noel
    dummy([i:i+k-1],[i:i+k-1])=double(ke{i});
    K=dummy+K;
    dummy=zeros(n);
end

```

APPENDIX D. **element_stiffness_2D**

```

clear all
clc
disp('2D problem')
nxx=input('cord of control points in x direction: ');
nx=input('no of control points in x direction: ');
nyy=input('cord of control points in y direction: ');
ny=input('no of control points in y direction: ');
conx=input('Connectivity table');
p=input('degree of basis function in u direction: ');
q=input('degree of basis function in v direction: ');
E=input('youngs modulus: ');
nu=input('value of poisoons ratio: ');
stresstate=input('1 for planestress,2 for plainstrain');
if stresstate==1
    C=(E/(1-nu^2))*[1 nu 0;nu 1 0; 0 0 (1-nu)/2];
elseif stresstate==2
    C=(E/((1+nu)*(1-2*nu)))*[1-nu nu 0;nu 1-nu 0;0 0 1/(2-nu)];
end
% finding knot vector
[U,nonrep1]=knotseq(nx,p);
[V,nonrep2]=knotseq(ny,q);
nel=(length(nonrep1)-1)*(length(nonrep2)-1);
noctl=nx*ny;
nodof=2*noctl;

```

```

[g,wt] = gaussp(r);
jac=zeros(2);ke=zeros(8);
k=cell(nel,1);
kg=zeros(nx*ny*2);
dummy=zeros(nx*ny*2);
for i=1:nel
    for a=1:length(nonrep1)-1
        for b=1:length(nonrep2)-1
            for j=1:2
                for jjj=1:2
                    gpu=((nonrep1(a+1)+nonrep1(a))/2)+(((nonrep1(a+1)-
nonrep1(a))/2)*g(jj));
                    gpv=((nonrep2(b+1)+nonrep2(b))/2)+(((nonrep2(b+1)-
nonrep2(b))/2)*g(jjj));
                    J1=0.25*(nonrep1(a+1)-nonrep1(a))*(nonrep2(b+1)-
nonrep2(b));
                    dersu = Der1BasisFun(findspan(nx,p,gpu,U) ,gpu,p,U);
                    dersv = Der1BasisFun(findspan(ny,q,gpv,V) ,gpv,q,V);
                    qq=1;
                    for bb=1:length(dersv(1,:))
                        for aa=1:length(dersu(1,:))
                            dRdu(1,qq)=dersu(2,aa)*dersv(1,bb);
                            dRdv(1,qq)=dersv(2,bb)*dersu(1,aa);
                            qq=qq+1;
                        end
                    end
                    end
                    cpx=nxx(conx(i,:));
                    cpy=nyy(conx(i,:));
                    jac(1)=dRdu*cpx';
                    jac(4)=dRdv*cpy';
                    J2=det(jac);
                    nn=length(dersu(1,:))*length(dersv(1,:));
                    B=zeros(3,2*nn);
                    dRdx=[dRdu;dRdv]*inv(jac);
                    B(1,1:nn)=dRdx(:,1)';
                    B(2,nn+1:2*nn)=dRdx(:,2)';
                    B(3,1:nn)=dRdx(:,2)';
                    B(3,nn+1:2*nn)=dRdx(:,1)';
                    ke=ke+ J2*J1*B'*C*B;

```

```

        end
    end
end
end
k{i}=ke;
end

```

APPENDIX E. **Natural Vibration of bar**

```

clear
clc
syms u
nel=input('Enter the number of elements')
p=1;
k=3;
s=1/(nel+k-2);%difference between two control points
nv=1/(2*nel);
% finding control points
for i=0:s:1
    f(p,1)=i;
    p=p+1;
end
h=nel-1+2*k;
z=h-k;
% finding the knot vector
for i=1:h
    if i<=k
        t(1,i)=0;
    elseif i>z
        t(1,i)=1;
    else
        t(1,i)=(i-k)*2*nv;
    end
end
%finding basis functions
g=cell(k*nel,1);
g=basisfun(p,k);
% Calculating global stiffness and global mass matrix
dummy1=zeros(nel+k-1);

```

```

dummy2=zeros(nel+k-1);
KT=zeros(nel+k-1);
MT=zeros(nel+k-1);
j=1;
o=1;
q=0;
v=2*nv;
for i=1:nel
    N1(u)=g{o,1};
    N2(u)=g{o+1,1};
    N3(u)=g{o+2,1};
    du1=diff(N1)*f(j,1)+diff(N2)*f(j+1,1)+diff(N3)*f(j+2,1);
    n1=[N1 N2 N3];
    B1=diff(n1)/du1;
    k1=B1'*B1;
    k1=k1*du1;
    Ke1=int(k1,q,q+v);
    Ke1=double(Ke1(u));
    m1=n1'*n1;
    m1=m1*du1;
    Me1=int(m1,q,q+v);
    j=j+1;
    o=o+3;
    q=q+v;
    dummy1([i:i+2],[i:i+2])=Ke1;
    KT=KT+dummy1;
    dummy1=zeros(nel+k-1);
    dummy2([i:i+2],[i:i+2])=Me1(u);
    MT=MT+dummy2;
    dummy2=zeros(nel+k-1);
end
% applying boundary conditions and finding natural frequencies
w1=eig(KT(2:end-1,2:end-1),MT(2:end-1,2:end-1));
nw1=sqrt(w1);
% for plotting the ratio frequencies to element number
for i=1:nel
    na1(i,1)=nw1(i,1)/(i*pi);
    el(i,1)=i/(nel);
end
plot(el,na1,'k')

```

APPENDIX F. **Natural vibration of beam**

```

%give frequencies of fixed beam
clear
clc
syms u
nel=input('Enter the number of elements')
p=1;
k=4;
s=1/(nel+k-2);%difference between two control points
nv=1/(2*nel);
% finding control points
for i=0:s:1
    f(p,1)=i;
    p=p+1;
end
h=nel-1+2*k;
z=h-k;
% finding the knot vector
for i=1:h
    if i<=k
        t(1,i)=0;
    elseif i>z
        t(1,i)=1;
    else
        t(1,i)=(i-k)*2*nv;
    end
end
%finding basis functions
g=cell(k*nel,1);
g=basisfun(p,k);
% Calculating global stiffness and global mass matrix
dummy1=zeros(nel+k-1);
dummy2=zeros(nel+k-1);
KT=zeros(nel+k-1);
MT=zeros(nel+k-1);
j=1;o=1;q=0;v=2*nv;
for i=1:nel
    N1(u)=g{o,1};

```

```

N2(u)=g{o+1,1};
N3(u)=g{o+2,1};
N4(u)=g{o+3,1};
du1=diff(N1)*f(j,1)+diff(N2)*f(j+1,1)+diff(N3)*f(j+2,1)+diff(N4)*f(j+3,1);
n1=[N1 N2 N3 N4];
Bh1=diff(n1)/du1;
B1=diff(Bh1)/du1;
k1=B1'*B1;
k1=k1*du1;
Ke1=int(k1,q,q+v);
Ke1=double(Ke1(u));
m1=n1'*n1;
m1=m1*du1;
Me1=int(m1,q,q+v);
Me1=double(Me1(u));
j=j+1;
o=o+4;
q=q+v;
dummy1([i:i+3],[i:i+3])=Ke1;
KT=KT+dummy1;
dummy1=zeros(nel+k-1);
dummy2([i:i+3],[i:i+3])=Me1;
MT=MT+dummy2;
dummy2=zeros(nel+k-1);
end
% applying boundary conditions and finding natural frequencies
w1=eig(KT(2:end-1,2:end-1),MT(2:end-1,2:end-1));
nw1=sqrt(w1);
% for plotting the ratio frequencies to element number
for i=1:nel+1
    na1(i,1)=nw1(i,1)/(i*pi)^2;
    el(i,1)=i/(nel+1);
end
plot(el,na1,'k')

```

REFERENCES

- [1] HUGHES, T. J., J. A. COTTRELL, Y. BAZILEVS. Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement. *Computer Methods in Applied Mechanics and Engineering*, **194** (2005), No. **39**, 4135-4195.
- [2] KAGAN, P., A. FISCHER. Integrated Mechanically Based CAE System using B-Spline Finite Elements. *Computer-Aided Design*, **32** (2000), No. **8**, 539-552.
- [3] KAGAN, P., A. FISCHER, P. Z. BAR-YOSEPH. New B-Spline Finite Element Approach for Geometrical Design and Mechanical Analysis. *International Journal for Numerical Methods in Engineering*, **41** (1998), No. **3**, 435-458.
- [4] COTTRELL, J. A., T. J. R. HUGHES, A. REALI. Studies of Refinement and Continuity in Isogeometric Structural Analysis. *Computer Methods in Applied Mechanics and Engineering*, **196** (2007), No. **41**, 4160-4183.
- [5] DENG, J., F. CHEN, X. LI, C. HU, W. TONG, Z. YANG, Y. FENG. Polynomial Splines over Hierarchical T-meshes. *Graphical Models*, **70** (2008), No. **4**, 76-86.
- [6] SCHILLINGER, D., L. DEDE, M. A. SCOTT, J. A. EVANS, M. J. BORDEN, E. RANK, T. J. HUGHES. An Isogeometric Design-through-analysis Methodology based on Adaptive Hierarchical Refinement of NURBS, immersed Boundary Methods, and T-spline CAD Surfaces. *Computer Methods in Applied Mechanics and Engineering*, **249** (2012), 116-150.
- [7] WALL, W. A., M. A. FRENZEL, C. CYRON. Isogeometric Structural Shape Optimization. *Computer Methods in Applied Mechanics and Engineering*, **197** (2008), No. **33**, 2976-2988.
- [8] HASSANI, BEHROOZ, S. M. TAVAKKOLI, N. Z. MOGHADAM. Application of Isogeometric Analysis in Structural Shape Optimization. *Scientia Iranica*, **18** (2011), No. **4**, 846-852. SIMPSON, R. N., S. P. BORDAS, J. TREVELYAN, T. RABCUK, (2012).
- [9] KIENDL, J., K. U. BLETZINGER, J. LINHARD, R. WÜCHNER. Isogeometric Shell Analysis with Kirchhoff–Love Elements. *Computer Methods in Applied Mechanics and Engineering*, **198** (2009), No. **49**, 3902-3914.
- [10] NIELSEN, P. N., A. R. GERSBORG, J. GRAVESEN, N. L. PEDERSEN. Discretizations in Isogeometric Analysis of Navier–Stokes Flow. *Computer Methods in Applied Mechanics and Engineering*, **200** (2011), No. **45**, 3242-3253.
- [11] GÓMEZ, H., V. M. CALO, Y. BAZILEVS, T. J. HUGHES. Isogeometric Analysis of the Cahn–Hilliard Phase-field Model. *Computer Methods in Applied Mechanics and Engineering*, **197** (2008), No. **49**, 4333-4352.
- [12] MASUD, A., R. KANNAN. B-splines and NURBS based Finite Element Methods for Kohn–Sham Equations. *Computer Methods in Applied Mechanics and Engineering*, **241** (2012), 112-127.
- [13] COTTRELL, J. A., A. REALI, Y. BAZILEVS, T. J. HUGHES. Isogeometric Analysis of Structural Vibrations. *Computer Methods in Applied Mechanics and Engineering*, **195** (2006), No. **41**, 5257-5296.

- [14] SHOJAEE, S., E. IZADPANAH, N. VALIZADEH, J. KIENDL. Free Vibration Analysis of Thin Plates by using a NURBS-based Isogeometric Approach. *Finite Elements in Analysis and Design*, **61** (2012), 23-34.
- [15] PIEGL, L., W. TILLER. The NURBS Book, Springer Science & Business Media, 2012.