Isogeometric Analysis vs. NURBS- Mapped Finite Elements: A Study on Selected Examples

Bianca E. Pirani[1], Nathan Shauer[1]

[1]*Dept. of Structures, School of Civil Engineering, Architecture and Urban Planning, University of Campinas (UNICAMP)*
224 Saturnino de Brito Street, Campinas, SP, 13083-889, Brazil
b258868@dac.unicamp.br, shauer@unicamp.br

Abstract. This work will focus on the application of Isogeometric Analysis (IGA), a numerical method that integrates finite element analysis (FEA) with CAD geometric representations. It will be demonstrated how NURBS basis functions are used to describe the geometry and simultaneously construct the approximation space for the solution. The methodology will be illustrated through a numerical example, where the advantages, challenges, and results obtained with this approach will be discussed. As a counterpoint, a finite element method (FEM) approach will be explored in future work, in which the geometric mapping is performed with NURBS, but the approximation space uses traditional hierarchical shape functions. The effectiveness of the IGA will be evaluated based on the analyses of representative numerical examples.

Keywords: Isogeometric Analysis, IGA, NURBS, Finite Element Analysis, FEA.

1 Introduction

1.1 Context

Computer-Aided Design CAD has its origins in the work of engineers like Pierre Bézier and Paul de Casteljau in the French automotive industry. This field evolved to represent complex, free-form curves and surfaces with high precision, eventually standardizing technologies like Non-Uniform Rational B-Splines (NURBS).

In parallel, Finite Element Analysis (FEA) became the standard for numerical simulation. Its methodology represents geometries through meshes of simple elements and uses piecewise polynomial functions to approximate the physical behavior.

The crucial issue is that these two fields, despite dealing with the same engineering objects, grew up independently and adopted entirely different mathematical languages for their geometric descriptions. This divergence between the exact spline-based geometries of CAD and the approximate polynomial-based meshes of FEA created a fundamental challenge in the modern design-to-analysis workflow. The idea for Isogeometric Analysis emerged from the need to bridge this historical gap and create a single, unified geometric representation for both design and analysis.

In the book, "Isogeometric Analysis: Toward Integration of CAD and FEA", Hughes, Cottrell, and Bazilevs [1] detail the depth of this challenge. They point out that the process of converting CAD geometry into an analysis mesh has become a significant hindrance in the overall workflow, estimating that for complex engineering designs, the task consumes over 80% of the total analysis time. Beyond the time consumed, the authors emphasize that the resulting finite element mesh is fundamentally an approximation of the original CAD geometry. This geometric inaccuracy, in turn, can introduce significant errors in critical simulations. Shell buckling analyses, for example, are very sensitive to geometric imperfections, as are boundary layer phenomena in aerodynamics and the precise representation of sliding contact between bodies.

1.2 Isogeometric Analysis

Isogeometric Analysis (IGA) was introduced as a new computational methodology. The fundamental goal of IGA is to unify the worlds of Computer-Aided Design (CAD) and Finite Element Analysis (FEA) by eliminating the need for the traditional meshing process.

The core principle of IGA is the use of a single, common basis for both geometric representation and numerical analysis. It employs the same sophisticated functions used in CAD systems—typically Non-Uniform Rational B-Splines (NURBS)—to precisely describe the geometry and, simultaneously, to construct the approximation space for the solution fields (such as displacement, temperature, or pressure).

This approach builds upon the classical isoparametric concept from FEA, but reverses its logic. In traditional FEA, a basis chosen for analysis (e.g., Lagrange polynomials) is used to approximate the geometry. In IGA, the basis that exactly represents the geometry is adopted for the analysis. By creating this direct link, IGA allows analysis to be performed directly on the original CAD geometry, preserving its exactness through all stages of simulation and refinement [4].

1.3 Objective

The main objective of this work is to show, through a practical application, the Isogeometric Analysis (IGA) methodology, where a NURBS basis is utilized for both the exact representation of the geometry and the construction of the solution's approximation space. The analysis will focus on the implementation process, documenting the practical advantages observed, the technical challenges faced, and the efficiency of the method. To contextualize the results, the study will also explore a comparison with a finite element approach that, while utilizing the geometric mapping via NURBS, employs traditional shape functions.

2 Theoretical Fundamentals

2.1 Traditional Finite Element Method Shape Functions

The traditional Finite Element Method (FEM) discretizes a domain into a mesh and uses piecewise polynomial shape functions, such as Lagrange polynomials, to approximate the solution. These functions are interpolatory at the nodes, with the basis function $L_j(x)$ associated with node j defined by the formula, (for k+1 nodes):

$$L_j(x) = \prod_{i=0, i\neq j}^{k} \frac{x - x_i}{x_j - x_i}$$

The resulting basis is C^0-continuous between elements. This guarantees a continuous solution but results in discontinuous derivatives. This can limit accuracy in problems where quantities like stress or flux are important.

2.2 B-splines

B-splines, the foundational building blocks of NURBS, are highly flexible piecewise polynomial functions. Their definition begins with a knot vector, a sequence of coordinates in a parameter space that dictates element boundaries and continuity. From this vector, the basis functions are generated recursively via the Cox-de Boor formula [5]. The process starts with piecewise constant functions (polynomial order p = 0) and for higher orders ($p \geq 1$) is defined as:

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

Unlike traditional finite element shape functions, B-splines offer significant advantages, most notably a high degree of controllable smoothness ($C^{\,p-m}$ continuity) across element boundaries, where p is the order of the polynomial and m is the multiplicity of the knot. Geometries are then constructed as a linear combination of these basis functions and a set of control points, that provides an intuitive way to manipulate the shape of the final smooth curve or surface, as illustrated in the figures below.

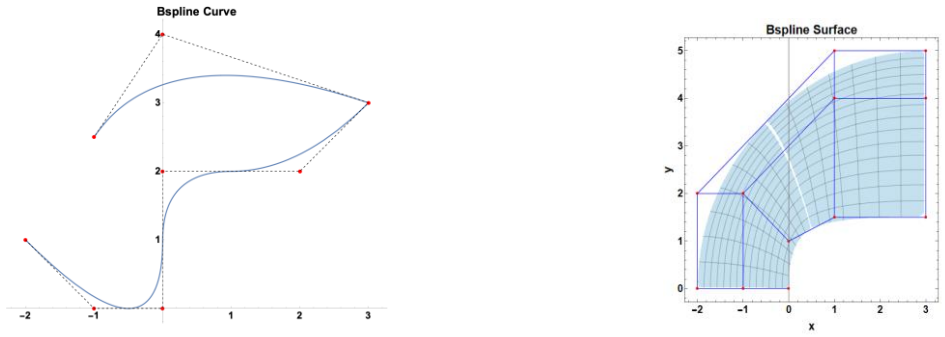


Figure 1. Illustration of a B-spline curve (left) defined by its control points (red points) and a B-spline surface (right) constructed from a mesh of control points.

The approximation quality of the B-spline basis is illustrated in Figure 2, which visually compares how solutions from traditional bilinear FEM and quadratic B-splines converge toward the exact solution as the number of basis functions is increased.

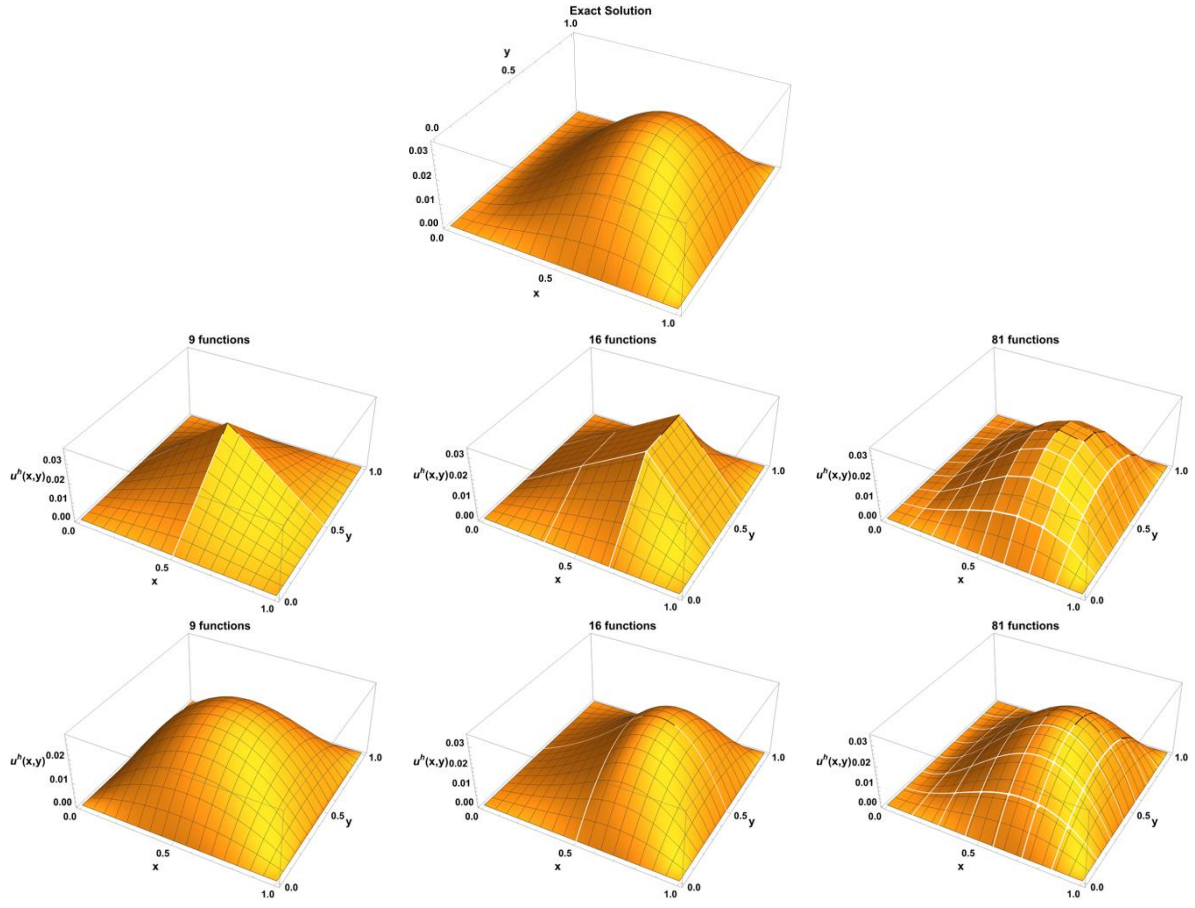


Figure 2. The first row displays the exact solution. The second and third rows show the approximate solutions using traditional FEM bilinear shape functions and quadratic B-spline basis functions as the approximation space, respectively. Each column corresponds to a discretization with 9, 16, and 81 basis functions.

2.3 NURBS

To exactly represent common engineering shapes, such as circles and spheres, which B-splines cannot do, Non-Uniform Rational B-splines (NURBS) are used. They are defined as:

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi)w_i}{W(\xi)} = \frac{N_{i,p}(\xi)w_i}{\sum_{j=1}^{n} N_{j,p}(\xi)w_j}$$

where i and j are the indices for the basis functions and their corresponding weights (w), and n is the total number of control points [5].

This structure, which adds a weight for each control point, provides a powerful and unified method for representing complex free-form shapes and exact analytical shapes. B-splines are simply a special case of NURBS, in which all weights are equal. The figures 3 and 4 illustrate the use of this basis to describe circular geometries, showing examples of NURBS curves and surfaces and the smooth, locally-supported basis functions used to construct them.

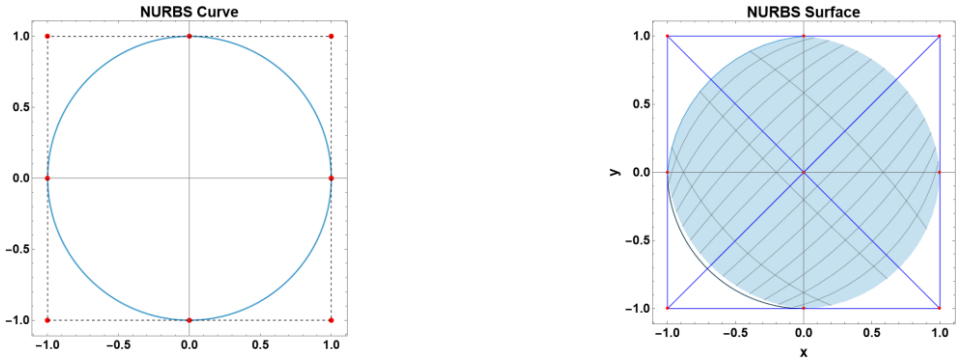


Figure 3. Illustration of a NURBS curve (left) defined by its control points (red points) and a B-spline surface (right) constructed from a mesh of control points.

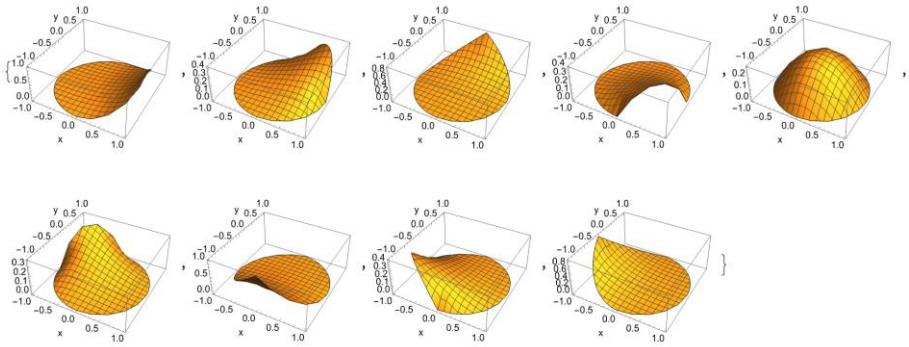


Figure 4. Illustration of the local support of each NURBS basis function.

3 Methodology and Case Study

3.1 Problem Definition

The governing problem is the Poisson equation in a domain $\Omega \subset \mathbb{R}^2$, given by: $-\Delta u = f$. For the verification of the method, the manufactured solution approach was employed, where an exact solution is defined a priori:

$$u_{\text{exact}}(x, y) = x(1-x)y(1-y)\sin(x)$$

The source term f is subsequently calculated as $f = -\Delta u_{\text{exact}}$.

The computational domain (Ω), shown in Figure 5, is a 90° circular annulus sector, whose curved geometry can be exactly represented by a single NURBS patch with quadratic (p = 2) basis functions in the circumferential direction and linear (q = 1) in the radial direction.

Dirichlet boundary conditions were imposed on the entire domain boundary using the penalty method. A high penalty parameter, $\beta = 10^8$, was used to enforce the satisfaction of the boundary conditions.

3.2 Integration Strategy

The assembly of the load vector and especially the stiffness matrix in IGA, which requires the numerical integration of products of NURBS basis functions (and their derivatives), can be a computationally costly and time-consuming process. To ensure efficiency, Gaussian Quadrature is adopted.

Due to the piecewise nature of NURBS functions, whose continuity can be reduced at the knots, the integral over the parametric domain is decomposed into a sum of integrals over the knot spans. A quadrature rule is then applied separately to each interval delimited by a pair of distinct and consecutive knots. In this study, a 3-point rule was used in each direction per interval.

3.3 Convergence Analysis

To evaluate the convergence of the numerical solution, a strategy analogous to h-refinement in traditional FEM is implemented, which consists of inserting new knots into the parametric space to increase the density of the approximation mesh. A central aspect of IGA is that the domain geometry remains identical and exact throughout the refinement process.

The effectiveness of the method was quantified by calculating the error in the L_2 norm, defined as:

$$E = \left|\left|u_{exact} - u_{aprox}\right|\right|_{L_2} = \sqrt{\int_{\Omega} \left(u_{exact} - u_{aprox}\right)^2 d\Omega}$$

To determine the expected convergence rate for this specific analysis, we apply the theory relating the polynomial order to the error.

The simulation employs a 2D tensor product of B-spline basis functions with different polynomial orders in each parametric direction: $p_\xi = 2$ and $p_\eta = 1$. The global convergence is governed by the minimum order, $p_{min} = \min(p_\xi, p_\eta) = 1$.

Based on the theoretical rate of $k_h = p_{min} + 1$, the convergence with respect to element size (h) is expected to be 2. Next, this rate is converted to the expected rate with respect to the total number of equations (N), denoted by k_{eq}. This is done by dividing k_h by the problem's dimension, which in the studied example is 2. Therefore:

$$k_{eq} = \frac{k_h}{2} = 1$$

This corresponds to the slope of -1 on the log-log convergence plot.

The convergence behavior is analyzed by plotting the error E as a function of the total number of basis functions for each mesh refinement level in Figure 6.

4 Results and Discussion

The results of the study are presented below. Figure 5 illustrates the qualitative convergence, where the approximate solutions for successively h-refined meshes (with 6, 12, and 462 basis functions) visually approach the exact solution. The quantitative convergence is detailed in Figure 6, which shows the decay of the error in the L^2 norm as a function of the number of basis functions on a log-log scale. The descending linear trend confirms the algebraic convergence of the method. The values on the curve represent the convergence rates calculated between the refinement points. The method is converging to the expected rate of 1 from above for this problem, with rates as high as 1.5 for the first few meshes, until it reaches the asymptotic region where the expected rate of 1 is obtained as seen in Figure 6.

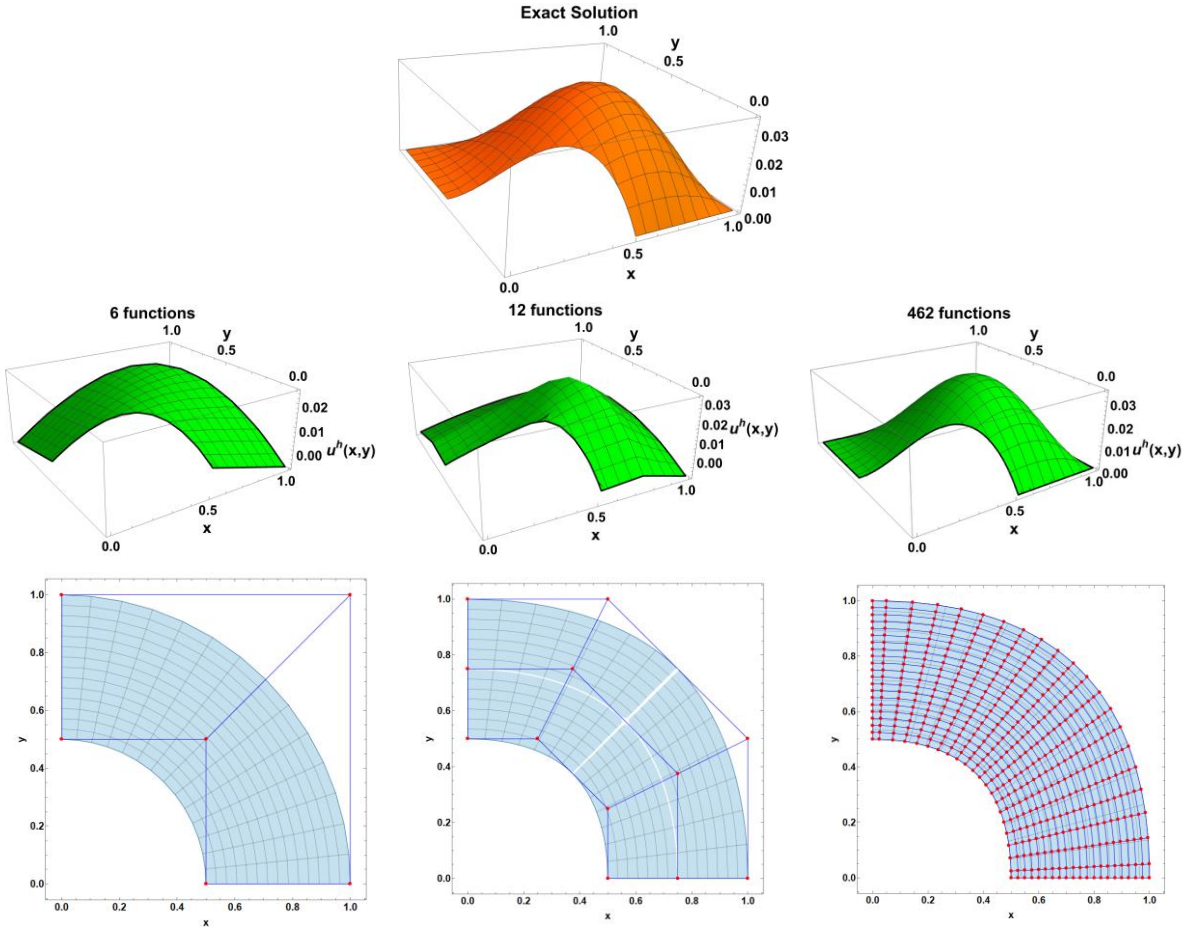


Figure 5. Comparison between the exact solution (first row) and the IGA approximate solutions (second row) for 6, 12, and 462 basis functions. The third row displays the corresponding control point mesh for each refinement level.

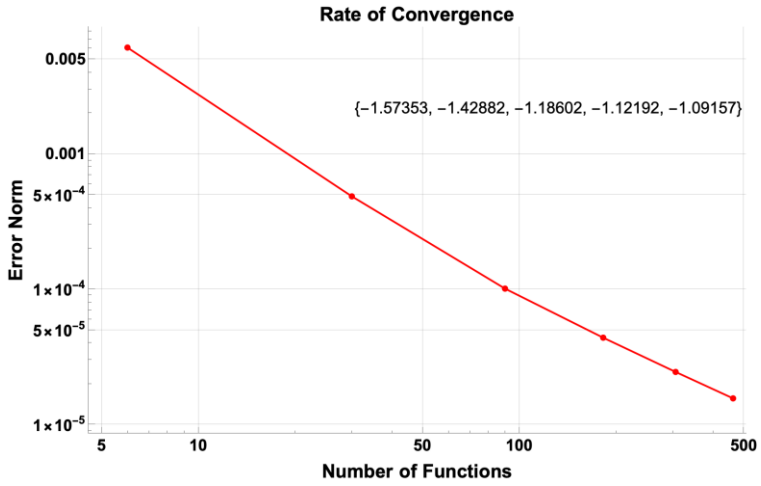


Figure 6. The convergence rate of IGA at each discretization.

5 Conclusions

In this work, Isogeometric Analysis was applied to solve the Poisson equation on a curved domain, using a manufactured solution for error verification. The convergence of the method was analyzed through uniform h-

refinement, and the results demonstrated the precision and robustness of the formulation, achieving the expected convergence rates. Future work will include the analysis of additional examples and the comparison of IGA's performance with alternative methods.

One of the main highlights of the study was the ability of the NURBS basis functions to perfectly represent the presented geometry. This characteristic eliminates the geometric discretization error, a fundamental advantage of IGA.

Acknowledgements. This work was supported by the National Council for Scientific and Technological Development (CNPq). The author, Bianca, would like to thank Prof. Nathan Shauer and Prof. Philippe R. B. Devloo for their invaluable support and guidance. Thanks are also extended to the members of the Computational Mechanics Laboratory at Unicamp for their helpful contributions and the collaborative environment.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. Isogeometric Analysis: Toward Integration of CAD and FEA. John Wiley & Sons, 2009.
[2] L. Carlon, "Isogeometric Analysis: Knot Insertion for NURBS". The Bug-Free Blog, 2021.
[3] V. P. Nguyen, C. Anitescu, S. P. A. Bordas, and T. Rabczuk, "Isogeometric analysis: An overview and computer implementation aspects". Archives of Computational Methods in Engineering, vol. 22, pp. 81–116, 2015.
[4] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," Computer Methods in Applied Mechanics and Engineering, vol. 194, pp. 4135–4195, 2005.
[5] T. N. Nguyen, "Isogeometric finite element analysis based on bezier extraction of NURBS and T-splines," Ph.D. dissertation, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2011.