JOHANNES KEPLER
UNIVERSITÄT LINZ | JKU

TNF

Technisch-Naturwissenschaftliche
Fakultät

Isogeometric Analysis: Condition Number Estimates and Fast Solvers

DISSERTATION

zur Erlangung des akademischen Grades

Doktor

im Doktoratsstudium der

Technischen Wissenschaften

Eingereicht von:
M.Sc., M.Tech. Krishan Pratap Singh Gahalaut

Angefertigt am:
Johann Radon Institute for Computational and Applied Mathematics,
Österreichische Akademie der Wissenschaften (ÖAW)

Beurteilung:
Dr. Satyendra Kumar Tomar (Betreuung)
Prof. Lourenço Beirão da Veiga

Linz, May, 2013

Abstract

Isogeometric methods were introduced by Hughes et al. in 2005. Since its introduction these methods have been used in many practical problems in science and engineering. However, so far, only few papers appeared in the field of iterative solvers for these methods. This thesis contributes to the development of fast iterative solvers for the matrices arising in isogeometric discretization of elliptic partial differential equations.

Since the performance of iterative solvers depends on the properties of the coefficient matrix, first we focus on the study of condition number estimates for the isogeometric matrices. The bounds for the extremal eigenvalues and the spectral condition number of matrices arising in isogeometric discretizations of elliptic partial differential equations in $\Omega \subset \mathbb{R}^d, d = 2, 3,$ are given in this thesis. Using the results from Bazilevs et al. on approximation properties, stability analysis and error estimates for isogeometric discretizations, and existing finite element theory, we obtain the condition number estimates for h-refined meshes. For the h-refinement, the condition number of the stiffness matrix is bounded above and below by a constant times h^{-2}, and the condition number of the mass matrix is uniformly bounded. For the p-refinement, it is proved that the condition number is bounded above by $p^{2d}4^{pd}$ and $p^{2(d-1)}4^{pd}$ for the stiffness matrix and the mass matrix, respectively.

For large problem size, the high condition number of coefficient matrix necessitates the development of fast and robust iterative solvers. In this thesis, we present three classes of iterative solvers. At first, the multigrid methods for isogeometric discretization are presented. The smoothing property of the relaxation method, and the approximation property of the intergrid transfer operators are analyzed for two-grid and multi-grid cycles. It is shown that the convergence of the multigrid solver is independent of the discretization parameter h, and that the overall solver is of optimal complexity. Secondly, we present another class of iterative solvers called algebraic multilevel iteration (AMLI) methods for isogeometric discretizations. The formation of coarse space and complement hierarchical space is discussed for B-splines and NURBS. The numerical study of Cauchy-Bunyakowski-Schwarz constant γ, measuring the quality of splitting between coarse space and its hierarchical complement, is also presented. It is found that the convergence of the AMLI solver is independent of the discretization parameter h, and for C^{p-1} continuous basis functions the convergence rates are independent of polynomial degree p. For C^0 continuous basis functions, numerical results indicates almost p-independent convergence rates. Thirdly, some numerical results on graph theory based preconditioners, namely, Vaidya's preconditioners (maximum weight spanning tree) and Gremban and Miller's preconditioners (support tree) are also presented. However, these preconditioners do not yield h-independent results for the condition number of preconditioned system.

Zusammenfassung

Isogeometrische Verfahren wurden 2005 von Hughes et al. eingeführt. Seitdem wurden diese Methoden in vielen praktischen Problemen in Wissenschaft und Ingenieurswesen verwendet. Jedoch gibt es zu diesem Zeitpunkt nur wenige Publikationen, die sich mit iterativen Lösern für diese Methoden beschäftigen. Die vorliegende Dissertation leistet einen Beitrag zur Entwicklung schneller iterativer Löser für die Matrizen, die in isogeometrischen Diskretisierungen elliptischer partieller Differentialgleichungen auftreten.

Da das Verhalten iterativer Löser von den Eigenschaften der Koeffizientenmatrix abhängt, konzentrieren wir uns zunächst auf das Studium von Eigenwertsabschätzungen für die isogeometrischen Matrizen. Schranken für die größten und kleinsten Eigenwerte sowie für die Konditionszahl der Matrizen, die in isogeometrischen Diskretisierungen elliptischer partieller Differentialgleichungen in $\Omega \subset \mathbb{R}^d$ auftreten, werden angegeben. Mittels der Resultate von Bazilevs et al. zu Approximationseigenschaften, Stabilitätsanalyse und Fehlerabschätzungen für isogeometrische Diskretisierungen, sowie existierender Finite Elemente-Theorie, erhalten wir Konditionszahlabschätzungen für h-verfeinerte Gitter. Für h-Verfeinerung ist die Konditionszahl der Steifigkeitsmatrix von oben und unten beschränkt durch eine Konstante mal h^{-2}, und die Konditionszahl der Massenmatrix ist gleichförmig beschränkt. Für p-Verfeinerung zeigen wir, dass die Konditionszahl von oben durch $p^{2d}4^{pd}$ und die der Massenmatrix durch $p^{2(d-1)}4^{pd}$ beschränkt ist.

Für große Probleme erfordert die hohe Konditionszahl der Koeffizientenmatrix die Entwicklung schneller und robuster iterativer Löser. In dieser Dissertation stellen wir drei Klassen iterativer Löser vor. Zunächst gehen wir auf Mehrgitterverfahren für isogeometrische Diskretisierungen ein. Wir analysieren die Glättungseigenschaft des Relaxationsverfahrens und die Approximationseigenschaft des Gittertransferoperators für Zwei- und Mehrgitterzyklen. Wir zeigen, dass die Konvergenz von Mehrgitterlösern unabhängig von h ist und dass der Löser insgesamt optimale Komplexität aufweist. Zweitens stellen wir eine andere Klasse von iterativen Lösern, die algebraische Multilevel-Iteration (AMLI) für isogeometrische Diskretisierungen, vor. Wir diskutieren die Bildung grober Räume und des komplementären hierarchischen Raumes für B-Splines und NURBS. Weiters präsentieren wir eine numerische Studie der Cauchy-Bunyakowski-Schwarz-Konstante γ, welche die Qualität der Aufspaltung zwischen dem groben Raum und dessen hierarchischen Komplement misst. Es stellt sich heraus, dass die Konvergenz des AMLI-Lösers unabhängig von h ist, und dass für C^{p-1}-stetige Basisfunktionen die Konvergenzrate unabhängig vom Polynomgrad p ist. Für C^0-stetige Basisfunktionen weisen die numerischen Resultate fast p-unabhängige Konvergenzraten auf. Drittens präsentieren wir numerische Resultate für graphentheoretische Vorkonditionierer, insbesondere die Vorkonditionierer von Vaidya (*maximum weight spanning tree*) sowie Gremban und Miller (*support tree*). Jedoch liefern diese Vorkonditionierer keine h-unabhängigen Resultate für die Konditionszahl des vorkonditionierten System.

Acknowledgments

It would not have been possible to write this thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

Above all, I would like to thank my parents, my brothers and sisters. They have given me their unequivocal support throughout, as always, for which my mere expression of thanks likewise does not suffice.

This thesis would not have been possible without the help, support and patience of my adviser, Satyendra Tomar. I would like to express my thanks to him, giving me the opportunity to write this thesis, for various discussions and for organizing financial support.

The good advice, support and friendly-nature of my group member, Johannes Kraus, has been invaluable on both an academic and a personal level, for which I am extremely grateful.

I would like to thank Ulrich Langer who, as leader of the group I am employed in, made constructive suggestions on many details of this thesis. I also want to thank Walter Zulehner, Ludmil Zikatanov for helpful discussions.

I am grateful to my colleagues Manas Kar, Durga Challa, Nagaiah Chamakuri, Huidong Yang, Monika Wolfmayr, Ivan Georgiev, Ervin Karer, Martin Purrucker, Jörg Willems, Nadir Bayramov, Stefan Kleiss, Stephen Moore, Angelos Mantzaflaris and other members of my group for the good and stimulating working atmosphere. I would also like to thank Clemens Hofreither for translating the abstract of this thesis in German.

I would like to acknowledge the academic and technical support of RICAM office staff, specially, Florian Tischler, Wolfgang Forsthuber, Annette Weihs and Doris Neundinger.

I am also grateful to S Sundar and Martin Kiehl.

It would be unfair if I don't mention the following names, I deeply acknowledge the support of my friends Sahab, Pushpendra, Shikha, Kuldeep, Shipra, Gaurav, Varun S, Rakesh, Rajneesh, Ajay S, Pankaj K, Vipul, Sateesh, Somvir, Mohar, Upendra, Dheerendra S, Dheerendra O, Dharmendra, Amit, Manoj S, Ajay K, Shakti, Ajeet, Santosh, Rangnath, Hema, Manoj P, Divakar, Sandeep, Ravi, Ramakrishna, Raghavendra, Harish V, Rajkumar, Rajani, Pradeep, Varun G, IshtDev, Dhaval, Kaushik, Om, Rashmi, Aparna, Anurag, Brajbhushan, Pankaj S, and Darshana.

I express my thanks to Pappan, Harish N, Kousik, Gururaja, Mahendra, Guhan, Sridhar, Anirban, Mohar, Rajdeep, Naresh Chadha, Amal Das and my other Indian friends in Linz.

Special thanks goes to my colleague and friend Ankik Giri for his kind support academically and personally.

I would also like to thank Mini Giri and Shashi Tomar for their kind support during my stay in Linz.

At last, I don't have words to express my thanks to Sivananthan Sampath without whom this thesis could not have been completed. He has been an inspirational and motivational person during my PhD work. At times when I was needed the most, he was the one who truly supported me. Thanks a ton my dear friend Siva.

KPS Gahalaut
Linz, May 2013

Contents

Chapter 1

Introduction

Solving the large system of linear equations is a central part of many scientific calculations. Whatever physical phenomenon might be modeled, at the end of the discretization process usually a linear system must be solved for unknowns that represent the physical quantities of the underlying problem. We are particularly interested in the solution of linear system of equations which arise from the discretization of scalar elliptic partial differential equations. Finite difference methods, finite element methods, and finite volume methods are the most promising discretization schemes which have been used or are being used by engineers and scientists. Recently, in the middle of the last decade, a new discretization technique, namely, "isogeometric analysis", has been developed.

Isogeometric analysis, a hidden technique for decades in between Finite Element Analysis (FEA) and Computer Aided Designs (CAD), was introduced by Hughes et al. in 2005. The communities of FEA and CAD were established in two different domains far from each other. Isogeometric analysis fills this gap between these two technologies. It may be argued that there is nothing new in these two techniques, however their fusion resulting in isogeometric analysis has offered a great potential.

FEA was first developed in 1943 by R. Courant, who utilized the Ritz method of numerical analysis and minimization of variational calculus to obtain approximate solutions to vibration systems. Shortly thereafter, a series of papers published in 1956 by M. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp established a broader definition of numerical analysis. The finite element method was originally developed to study the stresses in complex aircraft structures. Then, it was applied to other fields of continuum mechanics, such as heat transfer, fluid mechanics, acoustics, electromagnetics, geomechanics, biomechanics, etc. FEA is used in various industries, such as aerospace, automotive, biomedical, bridges and buildings, electronics and appliances, heavy equipment and machinery, micro electromechanical systems, and sporting goods, etc.

In FEA the solution of differential equations is saught over simple regions, e.g., triangles, quadrilaterals, tetrahedrons etc., and then assembled together so that continuity is satisfied at the interconnecting nodal points of the domains. If the method is convergent, it is natural to assume that if the size of the approximating domains becomes infinitely small, the solution so obtained tending to this limit by successive mesh refinement converges towards the

analytic solution. There are two main features of FEA: approximation of solution and approximation of geometry. Typically, the approximation of geometry is done by interpolatory polynomials. The classical families of polynomials, especially the Lagrange and Hermite polynomials, are widely used in FEA. These may be considered the historical antecedents of finite elements. The integrated Legendre functions have also been used to approximate the geometry. If we use the same functions to approximate the solution of the underlying problem, the approach is called isoparametric method. Through the use of triangles, quadrilaterals, tetrahedron, etc., it became a simple matter to generate C^0-continuous finite elements using classical polynomials, and in result to get the C^0-continuous solution of the problem. A limitation of the isoparametric concept was that while it worked for C^0-continuous interpolation, it did not for C^1 or higher. There was a strong interest in the development of C^1-continuous interpolation schemes. Many researchers sought solutions to this problem and they have got noteworthy successes. However, these methods were complicated to use and expensive, and interest moved to different variational formulations to circumvent the need for C^1-continuous basis functions. In the next paragraph we discuss about the origin of the basis functions from which one obtains high continuous interpolations.

The Bézier curves were discovered simultaneously by Paul de Casteljau at Citroen and Pierre E. Bézier at Renault around late 1950s and early 1960s. The generalization of Bézier functions are Basis splines, or B-splines for short. B-splines were known and studied by N. Lobachevsky whose major contribution to mathematics is perhaps the so-called hyperbolic (non-Euclidean) geometry in late eighteenth century. However, we shall adopt a modern version developed by C. de Boor, M. Cox and L. Mansfield in late 1970s. Note that Bézier curves are special cases of B-splines. B-spline methods for curves and surfaces were first proposed in the 1940s but were seriously developed only in the 1970s. They have been studied extensively and considerably extended since the 1970s, and a lot is currently known about them. However, B-splines can only represent what polynomial parametric forms can. By introducing homogeneous coordinates making them rational, B-splines are generalized to Non-Uniform Rational B-splines (NURBS). NURBS curves are more powerful than B-spline curves since they can exactly represent circles and ellipses. NURBS are commonly used in computer-aided design (CAD), manufacturing, engineering and are part of numerous industry world-wide. NURBS tools are also found in various 3D modeling and animation software packages. CAD implementations have evolved dramatically since the introduction of NURBS. Initially, with 3D in the 1970s, it was typically limited to producing drawings similar to hand-drafted drawings. Advances in programming and computer hardware, notably solid modeling in the 1980s, have allowed more versatile applications of computers in design activities.

Computer aided designs, specially B-splines, have also been used as a basis for solving variational problems (see, e.g., Schultz, 1973; Prenter 1975; Sabin, 1997, Hoellig 2003; Kwok et al., 2001), but these efforts have been dwarfed by activity in finite element analysis. It is interesting to note that the isoparametric elements developed in the 1960s are still the most widely utilized elements in commercial FEA codes, and even in research activities in FEA. Therefore before 2005, it was still a challenge to break down the barriers between engineering design and analysis, but at the same time maintain compatibility with existing

practices. A fundamental step is to focus on one, and only one, geometric model, which can be utilized directly as an analysis model, or from which geometrically precise analysis models can be automatically built. This requires a change from classical finite element analysis to an analysis procedure based on CAD representations. This concept is referred to as Isogeometric Analysis due to the isoparametric approach is invoked from geometry to solution.

Isogeometric analysis, since its introduction, has received great attention in the computational mechanics community. The concept has the capability of leading to large steps forward in computational efficiency since effectively, the process of re-meshing is either eliminated or greatly suppressed. The geometry description of the underlying domain is adopted from a Computer Aided Design (CAD) parametrization which is usually based on Non-Uniform Rational B-splines (NURBS), and the same basis functions are employed to approximate the physical solution. Isogeometric analysis techniques have been studied and applied in diverse fields, see e.g., [2, 3, 14–24, 34, 35, 41, 57, 67, 68]. Moreover, some theoretical aspects such as approximation properties, condition number estimates have also been studied, see [14, 20, 56]. The isogeometric methods, depending on various choices of basis functions, have shown several advantages over standard Finite Element Methods (FEM). For example, some common geometries arising in engineering and applied sciences, such as circles or ellipses, are exactly represented, and complicated geometries are represented more accurately than traditional polynomial based approaches. When we compare NURBS based isogeometric analysis with standard Lagrange polynomials based finite element analysis, it leads to qualitatively more accurate results [39]. Unlike finite element analysis, where C^1 or higher order interpolation finite elements are complicated and expensive to construct, isogeometric analysis offers C^{p-k}-continuous interpolation for p-degree basis functions with knot multiplicity k. Moreover, the ease in building spaces with high inter-element regularity allows for rather small problem sizes (in terms of degrees of freedom) with respect to standard finite element methods with the same approximation properties. This implies that, in general, for same approximation properties isogeometric stiffness and mass matrices are smaller than the corresponding finite element ones. Nevertheless, isogeometric analysis discrete problems may still be very large in realistic problems of interest, which result in a large linear system of equations.

This thesis deals with the solution of linear system of equations arising from the isogeometric discretization of scalar second order elliptic problems. For simplicity, we restrict ourselves to a model Poisson problem with Dirichlet boundary conditions in an open, bounded and connected Lipschitz domain in two- and three-dimensions. The isogeometric discretization technique begins by defining a weak or variational formulation of the problem and then approximating the solution in a finite dimensional space. The finite dimensional nature of the function spaces used in discretization process leads to a coupled system of linear algebraic equations. Such systems can can be of very large size, and must be solved efficiently. Computational algorithms for finding the solution of large linear system of equations are an important part of numerical linear algebra, and play a prominent role in engineering and science.

Ancient Indian texts "Sulabh Sutra" (meaning Easy Solution Procedures) describe the method in terms of solving systems of two linear equations in two variables, see [75]. The texts have been dated from around 800 BCE to 200 CE. The recorded analysis of simultaneous equations is also found in the ancient Chinese book "Chiu-chang suan-shu" (meaning mathematical treatise in nine sections), estimated to have been written sometime around 2000 years ago. This effort culminated around 825 CE in the writing of two books, that attracted international attention, by the Arabic mathematician Muhammad ibn-Musa Al-khawarizmi. The first was "Al-Maqala fi Hisab al-jabr w'almuqabilah" (meaning An essay on Algebra and equations). The second book "Kitab al-Jam'a wal-Tafreeq bil Hisab al-Hindi" appeared in a Latin translation under the title "Algoritmi de Numero Indorum" (meaning Al-Khawarizmi Concerning the Hindu Art of Reckoning). It was based on earlier Indian and Arabic treatises. However, it was not until near the end of the 17th Century that the ideas reappeared and development really got underway. Leibnitz, one of the two founders of calculus, used determinants in 1693, and later in the 18th century Lagrange further developed the idea of determinants through his Lagrangian multipliers. Cramer presented his determinant-based formula for solving systems of linear equations (today known as Cramer's Rule) in 1750. The rule appears in his "Introduction to the Analysis of Algebraic Curves", and deals with the problem of finding the equation of an algebraic plane curve passing through some fixed points. Gauss further developed the idea of using matrices to solve systems of linear equations through the use of Gaussian elimination. Later, Wilhelm Jordan introduced the technique of Gauss-Jordan elimination. In 1848 in England, J.J. Sylvester first introduced the term "matrix", which was the Latin word for womb, as a name for an array of numbers. Matrix algebra was nurtured by the work of Arthur Cayley in 1855. Cayley studied compositions of linear transformations and was led to define matrix multiplication so that the matrix of coefficients for the composite transformation ST is the product of the matrix for S times the matrix for T. He went on to study the algebra of these compositions including matrix inverses. The famous Cayley-Hamilton theorem, which asserts that a square matrix is a root of its characteristic polynomial, was given by Cayley in his 1858 Memoir on the Theory of Matrices. After World War II, there was renewed interest in matrices, particularly on the numerical analysis of matrices with the development of modern digital computers. John von Neumann and Herman Goldstine introduced condition numbers in analyzing round-off errors in 1947. Alan Turing and von Neumann were the 20th century giants in the development of stored-program computers. Turing introduced the LU decomposition of a matrix in 1948, where L is a lower triangular matrix with 1's on the diagonal and the U is an upper triangular matrix. It is common to use LU decompositions in the solution of a sequence of systems of linear equations, each having the same coefficient matrix. The QR factorization is used in computer algorithms for various computations, such as solving equations and find eigenvalues, where Q is a matrix whose columns are orthonormal vectors and R is a square upper triangular invertible matrix with positive entries on its diagonal. These are easy and intuitive algorithms however, because of their high computational complexity these are not suitable for large practical computations.

Direct methods for solving linear systems theoretically give the exact solution in a finite number of steps but are not appropriate for solving large number of equations in a system

where a high computational cost is involved. Another disadvantage of direct methods is rounding errors: an error made in one step spreads further in all following steps. There is another class of methods to solve linear system of equations called iterative methods. These methods generate a sequence of approximate solutions and essentially involve the coefficient matrix only in the context of matrix-vector multiplication. The evaluation of an iterative method invariably focuses on how quickly the iterates converge. The study of round off errors is in general not very well developed. A reason for this is that the iterates are only approximations of the exact solution, so round off errors in general only influence the speed of convergence but not the quality of the final approximation.

The earliest reference to an iterative approach to solving $Ax = b$ appears to be contained in a letter by Gauss to his student Gerling dated 26 December 1823. It was in the context of solving least squares problems via the normal equations. In 1845 Jacobi introduced his own iterative method, again for solving normal equations for least squares problems arising in astronomical calculations. Again in the context of least squares, in 1874 Seidel published his own iterative method. The paper contains what we now call the Gauss-Seidel method, which he describes as an improvement over Jacobi's method. In the early 20th century we note the important contributions, Richardson (1910) and Liebmann (1918). These papers marked the first use of iterative methods in the solution of finite difference approximations to elliptic PDEs. Richardson's method is still well known today, and can be regarded as an acceleration of Jacobi's method by means of over- or under-relaxation factors. Liebmann's method is identical with Seidel's method.

In the early 1950s a number of new methods appeared that dramatically changed the landscape of iterative methods. In separate contributions Lanczos [76] and Hestenes and Stiefel [62] proposed different versions of what is now known as the conjugate gradient method. The method proposed by Lanczos is, for symmetric positive-definite matrices, mathematically equivalent to the conjugate gradient method, but it was described for the general case of nonsymmetric matrices. The theory of stationary iterative methods was solidly established with the work of Young starting in the 1950s. Only in the 1970s was it realized that conjugacy based methods work very well for partial differential equations, especially the elliptic type. These methods are also used in practice upto a great extant, but none of these methods gives optimal order (linear in time) method.

In 1961, a new class of methods, namely multigrid methods, were discovered. During the last five decades, these methods have been established as a powerful and efficient tool for solving linear system of equations arising in a variety of problems [30, 60, 97]. The key idea of multigrid goes back to R.P. Fedorenko in the early 1960s [52, 53], who developed the first multigrid method for solving the Poisson equation on a unit square. The first rigorous convergence proof was provided by Bakhwalov [13]. In early 1970s, the multigrid idea was generalized to variational finite difference equations and general finite element equations by Astrachancev [1] and Korneev [71]. However, the huge potential of multigrid methods was realized due to the works of Brandt [31] and Hackbusch [59, 60]. These methods use the power of classical iterative methods on different levels of the problem. Nowadays the multigrid technique is one of the most efficient methods for solving a large class of problems

including elliptic boundary value problems for partial differential equations (PDEs) or systems of PDEs. In the early 1980s, algebraic multigrid methods were introduced by Brandt, McCormick, and Ruge [32], which rebuild the multigrid algorithm based on the information that is accessible via the system of (linear) algebraic equations only. Around the same time, AMLI methods were introduced by Axelsson and Vassilevski in a series of papers [7–10]. To achieve optimal computational complexity from AMLI methods, various stabilization techniques can be used. In the original work [7, 8], the stabilization was achieved by employing properly shifted and scaled Chebyshev polynomials. This approach requires the computation of polynomial coefficients which depends on the bounds of the eigenvalues of the preconditioned system. Alternatively, some inner iterations at coarse levels can be used to stabilize the outer iterations, which lead to parameter-free AMLI methods [9, 10, 72, 85]. These methods utilize a sequence of coarse-grid problems that are obtained from repeated application of a natural (and simple) hierarchical basis transformation, which is computationally advantageous. Moreover, the underlying technique of these methods often requires only a few minor adjustments (mainly two-level hierarchical basis transformation) even if the underlying problem changes significantly.

Since the introduction of isogeometric analysis, most of its progress has been focused on the applications and discretization properties. Nevertheless, when dealing with large problems, the cost of solving the linear system of equations arising from the isogeometric discretization becomes an important issue. Clearly, the discretization matrix A gets denser with increasing polynomial degree p. Therefore, the cost of a direct solver, particularly for large problems, becomes prohibitively expensive. The most practical way to solve them is to resort to an iterative method. Since the convergence rate of such methods is strongly affected by the condition number of the system matrix A, it is important to assess this quantity as a function of the mesh size h for the h-refinement, or as a function of the degree p for the p-refinement. Note that in the p-refinement, improved approximate solutions are sought by increasing p while the mesh of the domain, and thus the maximum quadrilateral diameter h, is held fixed, whereas in the h-refinement, improved approximations are obtained by refining the mesh, and thus reducing h, while p is held fixed. In this thesis we consider both the cases, i.e. the h-refinement and the p-refinement. Our main results provide upper and lower bounds for the condition number of the stiffness matrix and the mass matrix for the h-refinement, and upper bounds for the condition number of the stiffness matrix and the mass matrix for the p-refinement. It is well known fact that for the h-refinement, when applied to second order elliptic problems on a regular mesh, the condition number of the finite element stiffness matrix scales as h^{-2}, and the condition number of the mass matrix is bounded uniformly, independent of h, see e.g. [11, 33]. This is true for a great variety of elements and independent of the dimension of the problem domain. Our results are in agreement with the fact that for h-refinement whatever may be the underlying basis functions from the respective function space, the above bounds remain same [5]. These results are useful in theoretical analysis that relate to the h-refinement. For example, in convergence analysis of multigrid methods, these results are one of the key elements in deriving convergence factors, for finite element analysis, see e.g. [29, 60, 61, 91], and for isogeometric analysis, see [55]. It is known that the order of the approximation error of the numerical solution depends on the

choice of the finite dimensional subspace, and not on the choice of its basis. Therefore, when working with finite element method or isogeometric method for elliptic problems, one should think in terms of function spaces rather than on the choice of particular basis functions. Nevertheless, the choice of the basis functions affects the condition number of the stiffness matrix and the mass matrix, which influences the performance of iterative solvers. To the best of our knowledge, there is no general theory to characterize the extremal eigenvalues or the condition number based on a set of general polynomial basis functions, see e.g. [12, 80–82]. Unlike the h-refinement, for the p-refinement the condition number heavily depends on the choice of basis functions. For different choices of basis functions the condition number may grow algebraically or exponentially. Olsen and Douglas, Jr. [86] estimated the condition number bounds of finite element matrices for tensor product elements with two choices of basis functions. For the first choice, Lagrange elements, it is proved that the condition number grows exponentially in p, whereas for the second choice, hierarchical basis functions based on Chebychev polynomials, the condition number grows rapidly but only algebraically in p. Similar results on the condition number bounds can be found in, e.g., [51, 66, 79]. Due to the larger support of NURBS basis functions, the band of the stiffness matrix corresponding to the NURBS-based isogeometric method is less sparse than the one arising from finite element procedures. Therefore, a larger condition number is expected. Our results for the p-refinement show that in isogeometric method the condition number of system matrices grows exponentially.

Condition number estimates for the isogeometric matrices advocate fast, efficient and robust iterative solvers. In this thesis, first we focus on multigrid methods for solving the linear system of equations arising from the isogeometric discretization of scalar second order elliptic problems in a single patch. These methods are based on smoothing property of classical iterative methods and approximation property of intergrid transfer operators. We analyze two-grid methods and multigrid methods. The smoothing property and approximation properties are proved for two-grid cycle. Together, these two components establish the h-independence of the two-grid solver. For the multi-grid solver, which uses the two-grid solver recursively, we recall the h-independent convergence estimates from [60]. Elliptic problems with constant and variable coefficients are solved on a single patch in two and three dimensions. Numerical results show good convergence rates independent of the mesh parameter h. However, for discretizations based on higher degree polynomials, the convergence rate are quickly deteriorated. Therefore, we further analyze algebraic multilevel iteration methods. These methods are based upon the splitting of the fine space into coarse space and its hierarchical complement. Coarse space together with its hierarchical complement forms hierarchical space. The construction of hierarchical space is not unique and it depends on the different choices of linear combination of fine basis functions. Therefore, with different choices of hierarchical spaces we get different convergence rates. We construct two different hierarchical spaces for C^0-continuous and C^{p-1}-continuous basis functions. We could obtain for C^{p-1}-continuous basis functions p-independent convergence rates. It is still a challenge for higher p and low regularity basis functions to form the hierarchical spaces in optimal sense.

In this thesis, we also study the preconditioning of the isogeometric stiffness matrix us-

ing support graph theory. It began with the remarkable work of Pravin Vaidya in the early 1990s, in which he proposed and analyzed maximum weight spanning tree preconditioners for Laplacian matrices. Vaidya chose not to publish his work, but to produce commercial software instead. His software has had a significant impact in the structural analysis community. Fortunately, John Gilbert and Gary Miller recognized the significance of Vaidya's ideas and kept them from being lost. Miller and two of his students (Keith Gremban and Steve Guattery) formalized and greatly extended the techniques that Vaidya had used. In the process, they devised a new multi-level preconditioner and also conducted an analysis of incomplete Cholesky preconditioning. The idea, to use such graph theory based preconditioners in isogeometric analysis, stems from the fact that B-splines (NURBS) basis functions have larger support. We numerically studied Vaidya's preconditioners (maximum weight spanning tree) and Gremban and Miller's preconditioners (support tree). Since preliminary results were not promising, we did not pursue this direction further.

This thesis is structured as follows. Chapter 2 contains the preliminaries and basic concepts and introduces the notations. We briefly describe B-splines and NURBS. A brief methodology of multigrid methods and AMLI methods is also given.

In Chapter 3, we give the estimates of condition number of isogeometric matrices. The condition number estimates of B-spline basis functions are also given in this section. We discuss the condition number estimates for stiffness matrix and mass matrix as a function of mesh size h, and as a function of polynomial degree p.

Chapter 4 deals with the explicit representation of B-splines on a unit interval with uniform refinement. Moreover, the multilevel structure of B-splines and NURBS basis functions is given. The B-splines transfer operators are presented from a given fine level to the next coarse level. NURBS transfer operators are obtained from B-spline transfer operators.

In Chapter 5, we introduce the multigrid methods for isogeometric discretizations. The smoothing property of the Gauss-Seidel method and the approximation properties of intergrid transfer operators are discussed . Two-grid analysis and multigrid analysis for solving isogeometric linear system of equations are also analyzed in this section.

Chapter 6 describes our work on AMLI methods for isogeometric discretizations. This chapter focuses on construction and numerical studies. The construction of hierarchical spaces is given. The numerical study of constant γ, which measures the quality of space splitting, is also given in this section.

In Chapter 7, we discuss the graph theory based preconditioners for system of equations arising in isogeometric discretizations. The implementation of Vaidya's preconditioners and Gremban and Miller's preconditioners are given. In this section we also give local preconditioning of stiffness matrix based on M-matrix based approaches and two-color based approaches.

Finally in Chapter 8 we end with some concluding remarks and give an outlook on possible future work.

Chapter 2

Preliminaries

In this chapter we briefly describe the definitions and the basics of the study material which is useful for following this thesis. We first focus on B-splines and NURBS, and then describe our model problem. For simplicity we restrict ourselves to a model Poisson problem with Dirichlet boundary conditions. A short description to isogeometric analysis is given. Finally, we shall give a short overview on multigrid and multilevel methods to solve the resulting linear system of equations.

2.1 Spline Curves

The types of curve fall into two broad categories: interpolating curves and approximating curves. Interpolation curves fit given data points exactly, i.e, these curves will pass through the points used to describe it. Lagrange polynomials, integrated Legendre polynomials are interpolatory functions and commonly used in FEM. We shall not discuss them in details. Our main focus is on approximating curves which are commonly used in isogeometric analysis. Unlike interpolating curves, an approximating curve will not necessarily pass through the data points. The fundamental unit of approximating curves is Bézier curves. Bézier curves are defined by Bernstein polynomials. To define a Bézier curve of degree p, we need to choose $p+1$ control points in space so that they roughly indicate the shape of the desired curve. Then, if it is not up to our expectation, we can move the control points around. As one or more control points are moved, the shape of the Bézier curve changes accordingly, but the curve always lies in the convex hull defined by the control points. However, we cannot easily control the curve locally, i.e., any change to an individual control point will cause changes in the curve along its full length. In addition, we cannot create a local cusp in the curve, i.e., we cannot create a sharp corner unless we create it at the beginning or end of a curve where it joins another curve. Finally, it is not possible to keep the degree of the Bézier curve fixed while adding additional points; any additional points will automatically increase the degree of the curve. The so-called B-spline curve, which is the generalization of Bézier curves, addresses each of these problems with the Bézier curve. It provides todays' powerful and useful approach to curve design. We shall discuss them in the following section.

2.1.1 B-splines

Definition 2.1. *Let $\Xi_1 = \{\xi_i : i = 1, ..., n + p + 1\}$ be a non-decreasing sequence of real numbers called the knot vector, where ξ_i is the i^{th} knot, p is the polynomial degree, and n is the number of basis function. With a knot vector in hand, the B-spline basis functions denoted by $N_i^p(\xi)$ are (recursively) defined starting with a piecewise constant $(p = 0)$*

$$N_i^0(\xi) = \begin{cases} 1 & \text{if } \xi \in [\xi_i, \xi_{i+1}), \\ 0 & \text{otherwise,} \end{cases} \tag{2.1a}$$

$$N_i^p(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_i^{p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1}^{p-1}(\xi), \tag{2.1b}$$

where $0 \leq i \leq n, p \geq 1$ and $\frac{0}{0}$ is considered as zero.

The above is usually referred as the *Cox-de Boor recursion formula*, see e.g. [45]. For a B-spline basis function of degree p, an interior knot can be repeated at most p times, and the boundary knots can be repeated at most $p + 1$ times. A knot vector for which the two boundary knots are repeated $p + 1$ times is said to be open. In this case, the basis functions are interpolatory at the first and the last knot. If the number of knots is m, the degree of the basis functions is p, and the number of basis functions of degree p is n, then $m = n + p + 1$. B-spline basis functions have the following properties.

1. Polynomial: $N_i^p(\xi)$ is a degree p polynomial in ξ.
2. Nonnegativity: For all i, p and ξ, $N_i^p(\xi)$ is non-negative.
3. Local Support: $N_i^p(\xi)$ is a non-zero polynomial only on $[\xi_i, \xi_{i+p+1})$. On any span $[\xi_i, \xi_{i+1})$, at most $p + 1$ basis functions of degree p are non-zero, namely: $N_{i-p}^p(\xi), N_{i-p+1}^p(\xi), N_{i-p+2}^p(\xi), ..., N_i^p(\xi)$.
4. Partition of Unity: The sum of all non-zero degree p basis functions on span $[\xi_i, \xi_{i+1})$ is 1.
5. Composite curve: Basis function $N_i^p(\xi)$ is a composite curve of degree p polynomials with joining points at knots in $[\xi_i, \xi_{i+p+1})$.
6. Continuity: At a knot of multiplicity k, basis function $N_i^p(\xi)$ is C^{p-k} continuous. Therefore, increasing multiplicity decreases the level of continuity, and increasing degree increases continuity.

Definition 2.2. *A B-spline curve $C(\xi)$, is defined by*

$$C(\xi) = \sum_{i=1}^{n} P_i N_i^p(\xi) \tag{2.2}$$

where $\{P_i : i = 1, ..., n\}$ are the control points and N_i^p are B-spline basis functions defined in (2.1).

B-spline curves are generalization of Bézier curves and share many important properties with them. Moreover, B-spline curves have more useful properties than Bézier curves. The list below shows some of the most important properties of B-spline curves.

1. B-spline curve $C(\xi)$ is a piecewise curve with each component a curve of degree p. This nice property allows us to design complex shapes with lower degree polynomials.

2. Convex Hull Property: A B-spline curve is contained in the convex hull of its control net. More specifically, if ξ is in knot span $[\xi_i, \xi_{i+1})$, then $C(\xi)$ is in the convex hull of control points $P_{i-p}, P_{i-p+1}, ..., P_i$.

3. Local modification: Changing the position of control point P_i only affects the curve $C(\xi)$ on interval $[\xi_i, \xi_{i+p+1})$. This local modification property is very important to curve design.

4. $C(\xi)$ is C^{p-k} continuous at a knot of multiplicity k.

5. Variation Diminishing Property: The variation diminishing property holds for B-spline curves. If the curve is in a plane (space), this means that no straight line (resp. plane) intersects a B-spline curve more times than it intersects the curve's control polyline.

6. Affine Invariance: The affine invariance property also holds for B-spline curves. If an affine transformation is applied to a B-spline curve, the result can be constructed from the affine images of its control points. When we want to apply a geometric transformation to a B-spline curve, this property states that we can apply the transformation to control points, which is quite easy, and once the transformed control points are obtained the transformed B-spline curve is the one defined by these new points. Therefore, we do not have to transform the curve.

The previous definitions are easily generalized to the higher dimensional cases by means of tensor product. Using tensor product of one-dimensional B-spline functions, a B-spline surface $S(\xi, \eta)$ is defined as follows:

$$S(\xi, \eta) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} N_{i,j}^{p_1,p_2}(\xi, \eta) P_{i,j}, \tag{2.3}$$

where $P_{i,j}$, $i = 1, 2, \ldots, n_1$, $j = 1, 2, \ldots, n_2$, denote the control points, $N_{i,j}^{p_1,p_2}$ is the tensor product of B-spline basis functions $N_i^{p_1}$ and $N_j^{p_2}$, and $\Xi_1 = \{\xi_1, \xi_2, \ldots, \xi_{n_1+p_1+1}\}$ and $\Xi_2 = \{\eta_1, \eta_2, \ldots, \eta_{n_2+p_2+1}\}$ are the corresponding knot vectors. Similarly, B-spline solids can also be defined.

2.1.2 NURBS

It is known that polynomials can not exactly describe frequently encountered shapes in engineering, particularly the conic family, e.g. circle. While B-splines (polynomials) are

flexible and have many nice properties for curve design, they are also incapable of representing such curves exactly. Such limitations are overcome by NURBS functions which can be used to exactly represent a wide array of objects. Rational representation of conics originates from projective geometry. The "coordinates" in the additional dimension are called weights, which we shall denote by w. Furthermore, let $\{P_i^w\}$ be a set of control points for a projective B-spline curve in $\mathbb{R}^3$. For the desired NURBS curve in $\mathbb{R}^2$, the weights and the control points are derived by the relations

$$w_i = (P_i^w)_3, \qquad (P_i)_d = (P_i^w)/w_i, \quad d = 1, 2, \tag{2.4}$$

where w_i is called the i^{th} weight and $(P_i)_d$ is the d^{th}-dimension component of the vector P_i. The weight function $w(\xi)$ is defined as

$$w(\xi) = \sum_{i=1}^{n} N_i^p(\xi) w_i. \tag{2.5}$$

Then, the NURBS basis functions and curve are defined by

$$R_i^p(\xi) = \frac{N_i^p(\xi) w_i}{w(\xi)}, \qquad C(\xi) = \sum_{i=1}^{n} R_i^p(\xi) P_i. \tag{2.6}$$

The NURBS surfaces are analogously defined as follows

$$S(\xi, \eta) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} R_{i,j}^{p_1,p_2}(\xi, \eta) P_{i,j}, \tag{2.7}$$

where $R_{i,j}^{p_1,p_2}$ is the tensor product of NURBS basis functions $R_i^{p_1}$ and $R_j^{p_2}$. NURBS functions also satisfy the properties of B-spline functions. For a detailed exposition see, e.g. [88, 90, 94].

2.1.3 Knot Insertion and Degree Elevation

The meaning of knot insertion is adding a new knot into the existing knot vector without changing the shape of the curve. This new knot may be equal to an existing knot and, in this case, the multiplicity of that knot is increased by one. Because of the fundamental relation $m = n + p + 1$, after adding a new knot, the value of m is increased by one and, consequently, either the number of control points or the degree of the curve must also be increased by one. Changing the degree of the curve due to the increase of knots will change the shape of the curve globally and will not be considered. Therefore, inserting a new knot causes a new control point to be added. Knot insertion is one of the most important features for B-spline curves since many other useful techniques are based on knot insertion. For example, in isogeometric analysis h-refinement takes place by inserting the knots. The second mechanism by which one can enrich the basis is degree elevation. As its name implies, the process involves raising the polynomial degree of the basis functions

used to represent the geometry. During degree elevation, the multiplicity of each knot value is increased by one, but no new knot values are added. As with knot insertion, the geometry does not changed with degree elevation. There are well established algorithms for knot insertion and degree elevation. Therefore, we shall not discuss them in details, and refer the reader to, e.g., [88, 90, 94].

2.1.4 Derivatives of Splines

Derivatives of B-splines and NURBS, see e.g. [54,88,90,94], and their conditioning are very important for the estimation of the condition number of the stiffness matrix. The recursive definition of B-spline functions allows us to seek the relationship between the derivative of a B-spline basis function and lower degree basis function.

Definition 2.3. *The derivative of i^{th} B-spline basis function defined in (2.1), is given by*

$$\frac{d}{d\xi}N_i^p(\xi) = \frac{p}{\xi_{i+p} - \xi_i}N_i^{p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}}N_{i+1}^{p-1}(\xi). \tag{2.8}$$

By repeating differentiation of (2.8), we get the general formula for any order derivative. Since we are interested in the first derivative only, we ignore further details.

The derivatives of rational functions will clearly depend on the derivatives of their non-rational counterpart. Definition 2.3 can be generalized for NURBS as follows.

Definition 2.4. *The derivative of i^{th} NURBS basis function is given by*

$$\frac{d}{d\xi}R_i^p(\xi) = w_i\frac{w(\xi)\dfrac{d}{d\xi}N_i^p(\xi) - \dfrac{d}{d\xi}w(\xi)N_i^p(\xi)}{(w(\xi)^2)}. \tag{2.9}$$

where w_i and $w(\xi)$ are defined in (2.4) and (2.5), respectively.

2.2 Fundamentals of Isogeometric Analysis

In FEA there is one notion of a mesh and one notion of an element, but an element has two representations, one in the parent domain and one in the physical domain. Elements are usually defined by their nodal coordinates and the degrees-of-freedom are usually the values of the basis functions at the nodes. Finite element basis functions are typically interpolatory and may take on positive and negative values. Finite element basis functions are often referred to as interpolation functions.

On the other hand, isogeometric analysis utilizes NURBS basis functions and two concepts about numerical meshes can be identified: control mesh and physical mesh. Control points are defined in order to control the geometry and they do not conform to the actual geometry. The control mesh looks like a finite element mesh constructed with multilinear elements. Geometry and degrees-of-freedom are represented in terms of their respective values defined

Figure 2.1: Schematic illustration of NURBS paraphernalia for a one-patch surface model. Courtesy [39].

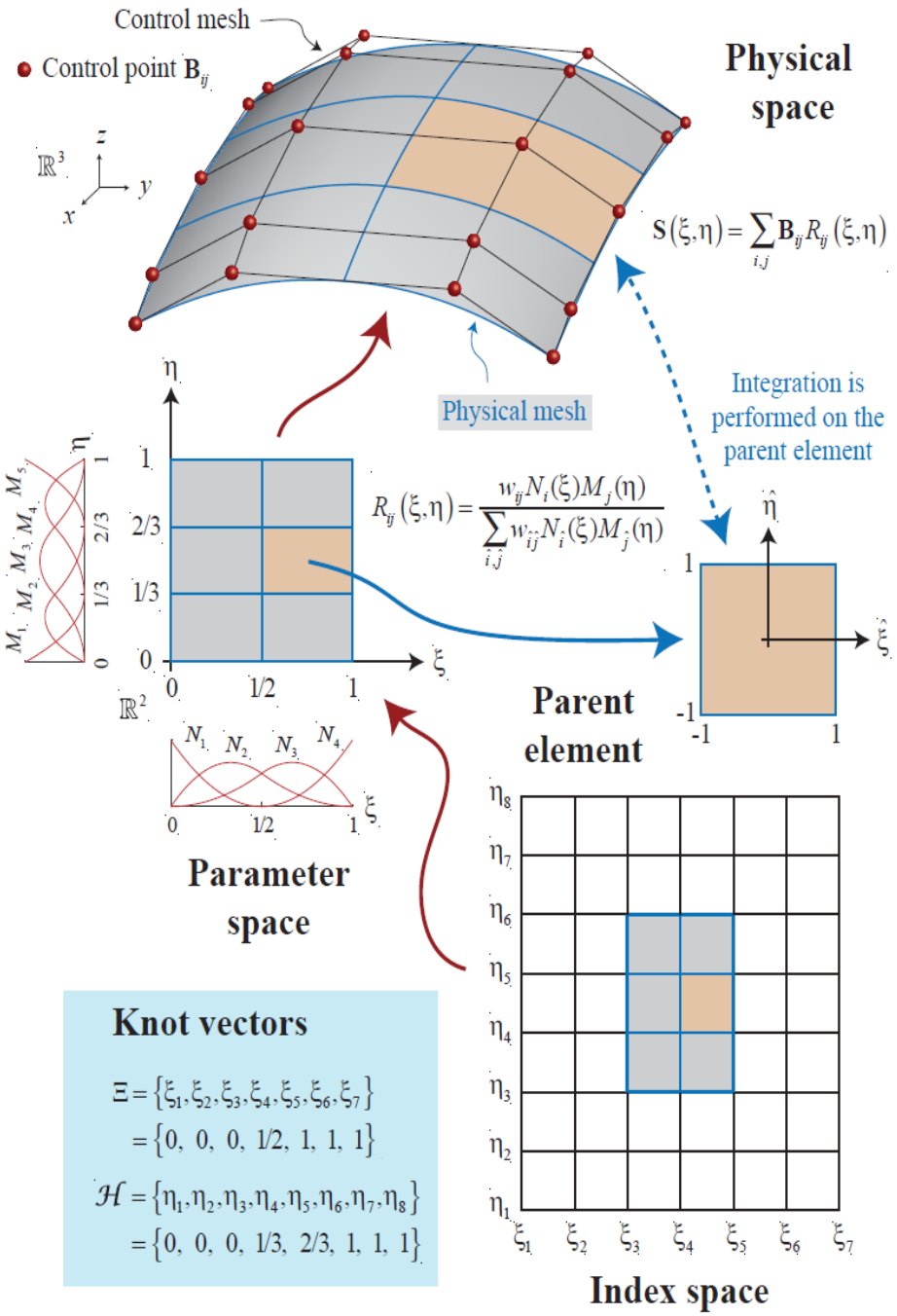


Figure 2.2: In classical finite element analysis, the parameter space is local to individual elements. Each element has its own mapping from the reference element. Courtesy [39].

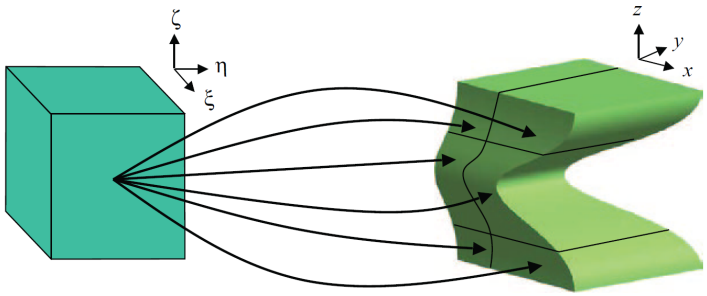
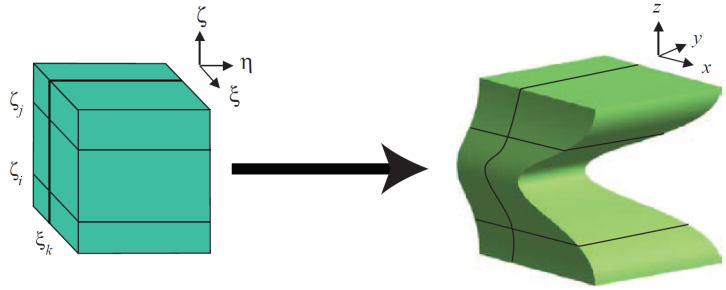


Figure 2.3: The B-spline parameter space is local to the entire patch. Internal knots partition the patch into elements. A single B-spline map takes the patch from the parameter space to the physical space. Courtesy [39].



at the control points. The physical mesh is a decomposition of the actual geometry. There are two notions of elements in the physical mesh, the patch and the knot span. The patch may be thought of as a macro-element or subdomain. Most geometries utilized for academic test cases can be modeled with a single patch. Each patch has two representations, one in a parent domain and one in physical domain. In two-dimensional topologies, a patch is a rectangle in the parent domain representation, and in three dimensions it is a cuboid.

Each patch can be decomposed into knot spans. Knot spans are bounded by knots. These define element domains where basis functions are smooth (i.e., C^{∞}). Across knots, basis functions will be C^{p-k} where p is the degree of the polynomial and k is the multiplicity of the knot in question. There is one other very important notion that is a key to understanding NURBS, the index space of a patch. It uniquely identifies each knot and discriminates among knots having multiplicity greater than one. The Figure 2.2, taken from [39], explains this mechanism.

Unlike in standard finite element analysis, the B-spline parameter space is local to patches

rather than elements. That is, the parameter space in FEA (tellingly dubbed the "reference element" or "parent element") is mapped into a single element in the physical space, and each element has its own such mapping, as in Figure 2.2. On the other hand, the B-spline mapping takes a patch of multiple elements in the parameter space into the physical space, as seen in Figure 2.3. Each element in the physical space is the image of a corresponding element in the parameter space, but the mapping itself is global to the whole patch, rather than to the elements themselves. Patches play the role of subdomains within which element types and material models are assumed to be uniform. Many simple domains can be represented by a single patch.

2.3 Model Problem and its Discretization

Let $\Omega \subset \mathbb{R}^d, d = 2, 3$, be an open, bounded and simply connected Lipschitz domain with Dirichlet boundary $\partial\Omega$. We consider the following second order scalar elliptic partial differential equation,

$$-\nabla \cdot (\mathcal{A}\nabla u) = f \quad \text{in } \Omega, \qquad u = g \quad \text{on } \partial\Omega, \tag{2.10}$$

where $f : \Omega \to \mathbb{R}$ is given. The aim is to find $u : (\Omega \cup \partial\Omega) \to \mathbb{R}$ which satisfies (2.10). There are several classes of numerical methods that lend themselves to isogeometric analysis. However, we restrict to Galerkin's formulation of the problem which is commonly used in isogeometric analysis. Isogeometric discretization has the same theoretical foundation as finite element analysis, namely the variational form of a partial differential equation. For this we define the function space, denoted by $\mathcal{S}$, as all the functions which have square integrable derivatives and also satisfy $u|_{\partial\Omega} = g$, i.e.

$$\mathcal{S} = \{u : u \in H^1(\Omega), u|_{\partial\Omega} = g\}, \tag{2.11}$$

where $H^1(\Omega) = \{u : D^\alpha u \in L^2(\Omega), |\alpha| \leq 1\}$ is the Sobolev space, $\alpha \in \mathbb{N}^d$ is a multi-index, $D^\alpha = D_1^{\alpha_1} D_2^{\alpha_2} ... D_d^{\alpha_d}$ and $D_i^j = \dfrac{\partial^j}{\partial x_i^j}$. The second collection of functions is very similar to the first one, except that we have the homogeneous counterpart of the Dirichlet boundary condition. We denote it by a set $\mathcal{V}$ defined by

$$\mathcal{V} = \{u : u \in H^1(\Omega), u|_{\partial\Omega} = 0\}, \tag{2.12}$$

We now write the variational formulation of the model problem by multiplying it by an arbitrary test function $v \in \mathcal{V}$ and integrating by parts. For a given $f \in L^2(\Omega)$: Find $u \in \mathcal{S}$ such that for all $v \in \mathcal{V}$

$$\int_\Omega \mathcal{A}\nabla u \cdot \nabla v \; d\Omega = \int_\Omega f v \; d\Omega.$$

We may rewrite above as

$$a(u, v) = L(v), \tag{2.13}$$

where

$$a(u, v) = \int_\Omega \mathcal{A}\nabla u \cdot \nabla v \; d\Omega, \quad \text{and} \quad L(v) = \int_\Omega f v \; d\Omega.$$

A few properties of $a(\cdot,\cdot)$ and $L(\cdot)$ are worth noting. The first is the symmetry of $a(\cdot,\cdot)$. It follows directly from its definition that $a(u,v) = a(v,u)$. Also, $a(\cdot,\cdot)$ is bilinear and $L(\cdot)$ is linear. That is, for all constants C_1 and C_2,

$$\begin{aligned} a(C_1u + C_2v, w) &= C_1a(u,w) + C_2a(v,w), \\ L(C_1u + C_2v) &= C_1L(u) + C_2L(v). \end{aligned} \tag{2.14}$$

Let $\mathcal{S}^h$ and $\mathcal{V}^h$ be the finite dimensional approximations of $\mathcal{S}$ and $\mathcal{V}$, respectively, i.e. $\mathcal{S}^h \subset \mathcal{S}$ and $\mathcal{V}^h \subset \mathcal{V}$. Suppose we have a given function $g^h \in \mathcal{S}^h$ such that $g^h|_{\partial\Omega} = g$, then for every $u^h \in \mathcal{S}^h$ there exists a unique $v^h \in \mathcal{V}^h$ such that

$$u^h = v^h + g^h. \tag{2.15}$$

The Galerkin form of the problem is: Find $u^h = v^h + g^h$, where $v^h \in \mathcal{V}^h$, such that for all $w^h \in \mathcal{V}^h$

$$a(u^h, w^h) = L(w^h). \tag{2.16}$$

It is well known that (2.16) is a well-posed problem and has a unique solution.

By approximating u_h and v_h using B-splines or NURBS basis functions N_i, $i = 1, 2, \ldots, n_h$, where $n_h = \mathcal{O}(h^{-2})$, the variational formulation (2.16) is transformed in to a set of linear algebraic equations

$$Au = f, \tag{2.17}$$

where A denotes the stiffness matrix obtained from the bilinear form $a(\cdot,\cdot)$, i.e.

$$A = (a_{i,j}) = (a(N_i, N_j)), \quad i,j = 1,2,3,....,n_h,$$

u denotes the vector of unknown degrees of freedom, and f denotes the right hand side (RHS) vector from the known data of the problem. It is clear that A is a real symmetric positive definite matrix.

2.4 Iterative Solvers

The system of linear equations coming from isogeometric discretization of partial differential equations is, in general, large and sparse. Direct solution methods can be impractical for such large and sparse systems because of memory and computational time requirements. Iterative methods are useful for such problems.

2.4.1 Basic Iterative Methods

The basic idea behind iterative methods for the solution of a linear system $Au = f$ is: starting from a given $u^{(k)}$, obtain a better approximation $u^{(k+1)}$ of u in a cheap way. Note that $f - Au^{(k)}$ is small if $u^{(k)}$ is close to u. This motivates the iteration process

$$u^{(k)} = u^{(k-1)} + M^{-1}(f - Au^{(k-1)}). \tag{2.18}$$

One can easily verifies that if this process converges, then the iterates $u^{(k)}$ converges to exact solution u. Rewriting of (2.18) leads to

$$Mu^{(k)} = Nu^{(k-1)} + f, \tag{2.19}$$

where the matrix N is obtained from the splitting $A = M - N$. It can easily be seen from

$$Mu = Nu + f \Longleftrightarrow Au = (M - N)u = f. \tag{2.20}$$

The classic iterative methods are based on different splittings of the coefficient matrix A.

Jacobi Method

The first classical iterative technique is called the Jacobi method, named after Carl Gustav Jacob Jacobi. This method splits A into three matrices: the diagonal D, an upper triangular U, and a lower triangular L, such that D is the same as the diagonal of A, $-U$ is the upper triangular part of A, and $-L$ is the lower triangular part of A, i.e.,

$$A = D - L - U. \tag{2.21}$$

Therefore, $Au = f$ can be rewritten

$$(D - L - U)u = f, \tag{2.22}$$

which leads to

$$Du = (L + U)u + f, \quad \Longrightarrow \quad u = D^{-1}(L + U)u + D^{-1}f. \tag{2.23}$$

This results in the iterative Jacobi technique:

$$u^{(k)} = D^{-1}(L + U)u^{(k-1)} + D^{-1}f, \quad k = 1, 2, 3, ... \tag{2.24}$$

In order to get a convergence result for the Jacobi method we need to recall the concept of diagonal dominance. We say a matrix A is strictly row diagonally dominant if

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|. \tag{2.25}$$

Theorem 2.5. *[91] If A is strictly row diagonally dominant, then the Jacobi method converges for any initial guess $u^{(0)}$.*

Gauss-Seidel Method

Gauss-Seidel method is a more efficient version of the Jacobi Method. The way of deriving the Gauss-Seidel method formally is as follows

$$(D - L)u = Uu + f, \quad \Longrightarrow \quad u = (D - L)^{-1}Uu + (D - L)^{-1}f, \tag{2.26}$$

and hence, generating the recurrence relation

$$u^{(k)} = (D - L)^{-1}Uu^{(k-1)} + (D - L)^{-1}f. \tag{2.27}$$

The convergence criteria for Gauss-Seidel iteration are a little more general than those for the Jacobi method. We have the following result.

Theorem 2.6. *[91] The Gauss-Seidel method converges for any initial guess $u^{(0)}$ if*

1. *A is strictly diagonally dominant, or*
2. *A is symmetric positive definite.*

Successive Relaxation Method

The successive relaxation method further improves the Gauss-Seidel method. The recurrence relation is given by

$$u^{(k)} = (D - \omega L)^{-1}((1 - \omega)D + \omega U)u^{(k-1)}) + (D - \omega L)^{-1}\omega f. \tag{2.28}$$

If $0 < \omega < 1$, the iterative method is known as a "successive under relaxation" and they can be used to obtain convergence when the Gauss-Seidel scheme is not convergent. For choices of $\omega > 1$ the scheme is a "successive over relaxation" and is used to accelerate the convergence of Gauss-Seidel iterations. Note that, $\omega = 1$ is simply the Gauss-Seidel iterative method. We can prove a general convergence theorem that is similar to those for the Jacobi and Gauss-Seidel methods:

Theorem 2.7. *[91] If A is symmetric positive definite then the SOR method with $0 < \omega < 2$ converges for any starting value $u^{(0)}$.*

2.4.2 Conjugate Gradient Methods

The Conjugate Gradient (CG) algorithm is one of the best known iterative techniques for solving sparse symmetric positive definite linear systems. CG is simply the method of conjugate directions where the search directions are constructed by conjugation of the residuals. The method proceeds by generating vector sequences of iterates (i.e., successive approximations to the solution), residuals corresponding to the iterates, and search directions used in updating the iterates and residuals.

The iterates $u^{(k)}$ are updated in each iteration by a multiple α_k of the search direction vector $p^{(k)}$:

$$u^{(k)} = u^{(k-1)} + \alpha_k p^{(k)}. \tag{2.29}$$

Correspondingly the residuals $r^{(k)} = f - Au^{(k)}$ are updated as

$$r^{(k)} = r^{(k-1)} - \alpha_k q^{(k)}. \tag{2.30}$$

where

$$q^{(k)} = Ap^{(k)}. \tag{2.31}$$

The choice

$$\alpha = \alpha_k = \frac{{r^{(k-1)}}^T r^{(k-1)}}{{p^{(k)}}^T A p^{(k)}} \tag{2.32}$$

minimizes ${r^{(k)}}^T A^{-1} r^{(k)}$ over all possible choices for α. The search directions are updated using the residuals

$$p^{(k)} = r^{(k)} - \beta_{k-1} p^{(k-1)}, \tag{2.33}$$

where the choice

$$\beta_k = \frac{{r^{(k)}}^T r^{(k)}}{{r^{(k-1)}}^T r^{(k-1)}} \tag{2.34}$$

ensures that $p^{(k)}$ and $Ap^{(k-1)}$, or equivalently, $r^{(k)}$ and $r^{(k-1)}$, are orthogonal.

Preconditioning is an important technique used to develop an efficient CG method solver for challenging problems in scientific computing. The idea behind preconditioning is using the CG method on an equivalent system. Thus, instead of solving $Au = f$ we solve a related problem $\tilde{A}\tilde{u} = \tilde{f}$ for which $\tilde{A}$ is chosen such that its condition number is closer to one; in other words, $\tilde{A}$ is close to the identity.

The generalized conjugate gradient (GCG) methods are iterative methods that are based on generating residuals that are formally orthogonal to each other with respect to some true or formal inner product. This includes methods that generate residuals that are minimal with respect to some norm based on an inner product. In this case the preconditioner is no longer given by a linear mapping but it can be defined by an iterative process itself.

For brevity reasons, we skip the details of CG algorithms and convergence analysis, and refer the readers to, e.g., [4, 9, 10, 73, 91].

2.5 Multigrid Methods

A multigrid method involves a hierarchy of meshes and related discretizations. The starting point of the multigrid idea is the observation that classical iteration methods (e.g. Jacobi and Gauss-Seidel schemes) have smoothing properties. Although these methods are characterized by poor global convergence rates, for errors whose length scales are comparable to the mesh size, they provide rapid damping, leaving behind smooth, longer wave-length errors. These smooth parts of the error are responsible for the poor convergence. A quantity that is smooth on a certain grid can, without any essential loss of information, also be approximated on a coarser grid. So the low-frequency error components can be effectively reduced by a coarse-grid correction procedure. Because the action of smoothing steps leaves only smooth error components, it is possible to represent them as the solution of an appropriate coarser system. Once this coarser problem is solved, its solution is interpolated back to the fine grid to correct the fine grid approximation for its low-frequency errors.

2.5.1 Two-grid Cycle

We begin with two-grid method. For smoothing on each grid one can use Gauss-Seidel. For the larger problem on the fine grid, the effect of smoothing on the low frequency (smooth) part of the solution u is small. The multigrid method transfers the current residual $r_h = f_h - Au_h$, h represents mesh size at fine level, to the coarse grid, denoted by H. We perform

Algorithm 2.1 Two-grid algorithm

1. Perform ν Gauss-Seidel steps on $A_h u_h = f_h$ at fine grid with given initial guess u_h.
2. Restrict the residual $r_h = f_h - Au_h$ to the coarse grid by $r_H = R_h^H r_h$, where R_h^H is the restriction operator.
3. Solve $A_H E_H = r_H$ exactly.
4. Interpolate E_H to get E_h by $E_h = P_H^h E_H$, where P_H^h is the prolongation operator.
5. Correct u_h as $u_h + E_h$.
6. Go to step 1.

a few smoothing step on the H grid, to approximate the coarse-grid error by E_H . Then, we interpolate E_H back to E_h on the fine grid, and make the correction to u_h , and begin again. This fine-coarse-fine loop is a two-grid V-cycle. The procedural way of a two-grid V-cycle is given in Algorithm 2.1.

2.5.2 Multigrid Cycle

For real world large scale problems, the two-grid method is impractical because the exact solution of the coarse-grid problem is required. The two-grid V-cycle extends in a natural way to more grids. It can go down to coarser grids $(2h, 4h, 8h, ...)$ and back up to $(..., 4h, 2h, h)$. For a detailed description, we introduce a sequence of grids with mesh size $h_1 > h_2 > ... > h_L > 0$, so that $h_{k-1} = 2h_k$. Here $k = 1, 2, ..., L$, is called the level number. On each level k we define the problem $A_k u_k = f_k$. Algorithm 2.2 explains the μ-cycle multigrid procedure.

We skip further details about the convergence results of multigrid theory for FEM, and refer the reader to, e.g., [4, 60, 61, 91, 97]. The convergence rates of multigrid methods in the framework of isogeometric analysis will be discussed in details in Chapter 5.

2.6 AMLI Methods

In this section we present the basic principle of AMLI methods. In what follows we will denote by $M^{(k)}$ a preconditioner for stiffness matrix $A^{(k)}$ corresponding to level k. We will also make use of the corresponding hierarchical matrix $\hat{A}^{(k)}$, which is related to $A^{(k)}$ via a two-level hierarchical basis (HB) transformation $J^{(k)}$, i.e.,

$$\hat{A}^{(k)} = J^{(k)} A^{(k)} (J^{(k)})^T. \tag{2.35}$$

The transformation matrix $J^{(k)}$ specifies the space splitting, which will be described in detail in Chapter 6. By $A_{ij}^{(k)}$ and $\hat{A}_{ij}^{(k)}$, $1 \leq i, j \leq 2$, we denote the blocks of $A^{(k)}$ and $\hat{A}^{(k)}$

Algorithm 2.2 Multigrid algorithm

1. If $k = 1$ solve $A_k u_k = f_k$ exactly.
2. Pre-smoothing steps on the fine grid: ν Gauss-Seidel steps to solve $A_k u_k = f_k$.
3. Restriction of the residual $r_k = f_k - Au_k$ to the next coarse grid by $r_{k-1} = R_k^{k-1} r_k$.
4. Set $u_{k-1} = 0$.
5. Call μ times the 'Multigrid algorithm' to solve $A_{k-1} u_{k-1} = r_{k-1}$.
6. Coarse-grid correction: $u_k = u_k + I_{k-1}^k u_{k-1}$.
7. Post-smoothing steps on the fine grid: ν Gauss-Seidel steps to solve $A_k u_k = f_k$ starting from the improved initial guess.

that correspond to the fine-coarse partitioning of degrees of freedom where the degrees of freedom associated with the coarse mesh are numbered last.

The aim is to build a multilevel preconditioner $M^{(L)}$ for the coefficient matrix $A^{(L)}$ at the level of the finest mesh that has a uniformly bounded (relative) condition number

$$\varkappa({M^{(L)}}^{-1} A^{(L)}) = \mathcal{O}(1),$$

and an optimal computational complexity, that is, linear in the number of degrees of freedom N_L at the finest mesh (grid). In order to achieve this goal hierarchical basis methods can be combined with various types of stabilization techniques.

One particular purely algebraic stabilization technique is the so-called algebraic multilevel iteration (AMLI) method, where a specially constructed matrix polynomial $p^{(k)}$ of degree ν_k can be employed at some (or all) levels k. The AMLI algorithm has been originally introduced and studied in a multiplicative form, see [7, 8].

We have the following two-level hierarchical basis representation at level k

$$\hat{A}^{(k)} = \begin{bmatrix} \hat{A}_{11}^{(k)} & \hat{A}_{12}^{(k)} \\ \hat{A}_{21}^{(k)} & \hat{A}_{22}^{(k)} \end{bmatrix} = \begin{bmatrix} A_{11}^{(k)} & \hat{A}_{12}^{(k)} \\ \hat{A}_{21}^{(k)} & A^{(k-1)} \end{bmatrix}. \tag{2.36}$$

Starting at level $l = 1$ (associated with the coarsest mesh), on which a complete LU factorization of the matrix $A^{(1)}$ is performed, we define

$$M^{(1)} := A^{(1)}. \tag{2.37}$$

Given the preconditioner $M^{(k-1)}$ at level $k-1$, the preconditioner $M^{(k)}$ at level k is then defined by

$$M^{(k)} := L^{(k)} U^{(k)}, \tag{2.38}$$

where

$$L^{(k)} := \begin{bmatrix} C_{11}^{(k)} & 0 \\ \hat{A}_{21}^{(k)} & C_{22}^{(k)} \end{bmatrix}, \quad U^{(k)} := \begin{bmatrix} I & {C_{11}^{(k)}}^{-1} \hat{A}_{12}^{(k)} \\ 0 & I \end{bmatrix}. \tag{2.39}$$

Here $C_{11}^{(k)}$ is a preconditioner for the pivot block $A_{11}^{(k)}$, and

$$C_{22}^{(k)} := A^{(k-1)} \left(I - p^{(k)}({M^{(k-1)}}^{-1} A^{(k-1)}) \right)^{-1} \tag{2.40}$$

$$0 \leq p^{(k)}(t) < 1, \quad 0 < t \leq 1, \quad p^{(k)}(0) = 1. \tag{2.41}$$

It is easily seen that (2.40) is equivalent to

$${C_{22}^{(k)}}^{-1} = {M^{(k-1)}}^{-1} q^{(k)}(A^{(k-1)} {M^{(k-1)}}^{-1}), \tag{2.42}$$

where the polynomial $q^{(k)}$ is given by

$$q^{(k)}(x) = \frac{1 - p^{(k)}(x)}{x}. \tag{2.43}$$

We note that the multilevel preconditioner defined via (2.38) is getting close to a two-level method when $q^{(k)}(x)$ closely approximates $1/x$, in which case ${C_{22}^{(k)}}^{-1} \approx {A^{(k-1)}}^{-1}$. In order to construct an efficient multilevel method, the action of ${C_{22}^{(k)}}^{-1}$ on an arbitrary vector should be much cheaper to compute (in terms of the number of arithmetic operations) than the action of ${A^{(k-1)}}^{-1}$. Optimal order solution algorithms typically require that the arithmetic work for one application of ${C_{22}^{(k)}}^{-1}$ is of the order $\mathcal{O}(N_{k-1})$ where N_{k-1} denotes the number of unknowns at level $k-1$.

It is well known from the theory introduced in [7,8] that a properly shifted and scaled Chebyshev polynomial $p^{(k)} := p_{\nu_k}$ of degree ν_k can be used to stabilize the condition number of ${M^{(k)}}^{-1} A^{(k)}$ (and thus obtain optimal order computational complexity). Other polynomials such as the best polynomial approximation of $1/x$ in uniform norm also qualify for stabilization, see, e.g., [74]. Alternatively, in the nonlinear AMLI method, see, e.g., [10], a few inner flexible conjugate gradient (FCG) type iterations (for the FCG algorithm, see also [84]) are performed in order to improve (or freeze) the residual reduction factor of the outer FCG iteration. In general, the resulting nonlinear (variable step) multilevel preconditioning method is almost equally efficient, and, because its realization does not rely on any spectral bounds, is easier to implement than the linear AMLI method (based on a stabilization polynomial). For a convergence analysis of nonlinear AMLI see, e.g., [72, 100].

Typically, the iterative solution process is of optimal order of computational complexity if the degree $\nu_k = \nu$ of the matrix polynomial (or alternatively, the number of inner iterations for nonlinear AMLI) at level k satisfies the optimality condition

$$1/\sqrt{(1-\gamma^2)} < \nu < \tau, \tag{2.44}$$

where $\tau \approx \tau_k = N_k/N_{k-1}$ denotes the reduction factor of the number of degrees of freedom, and γ denotes the constant in the strengthened Cauchy-Bunyakowski-Schwarz (CBS)

inequality. The value of τ is approximately 4 and 8 in case of two- and three-dimensional problems, respectively. For a more detailed discussion of AMLI methods, including implementation issues, see, e.g., [73, 100].

Remark 2.8. *The preconditioner defined in* (2.38) *is of multiplicative form. The introduction of AMLI methods was based on the multiplicative form, see [7–10], and is commonly used in practice. However, it is also possible to choose the preconditioner in the additive form, which is defined as follows*

$$M_A^{(k)} := \begin{bmatrix} C_{11}^{(k)} & 0 \\ 0 & C_{22}^{(k)} \end{bmatrix}. \tag{2.45}$$

In this case the optimal order of computational complexity demands that the matrix polynomial degree (or the number of inner iterations of nonlinear AMLI) satisfy the following relation

$$\sqrt{(1+\gamma)/(1-\gamma)} < \nu < \tau. \tag{2.46}$$

Chapter 3

Condition Number Estimates of Isogeometric Matrices

The condition number of matrices plays an important role in the numerical linear algebra. It measures the sensitivity of the solution of a problem to perturbations in the data. It provides an approximate upper bound on the error in a computed solution. The condition number can also be used to predict the convergence of iterative methods. The convergence of an iteration process and the existence of the solution of linear system depends on the form of the coefficient matrix. If the matrix is singular, then this system does not have a solution. On the contrary, when determinant of $A \neq 0$, it is the condition number of a matrix that decides the convergence of the approximate solution, obtained in the iteration process, to the correct solution of the equation system. The condition number is defined by the relation

$$\kappa(A) = \|A\| \cdot \|A^{-1}\| \tag{3.1}$$

The value of condition number is dependent on the choice of a matrix norm, and indirectly on the choice of a vector norm. We use the spectral norm to estimate the condition number throughout the thesis. If a condition number is very high then the matrix A is said to be ill-conditioned, otherwise it is said to be well-conditioned. A matrix with a high condition number can generate approximations with a large error.

The condition number is very crucial in the discretization techniques. It heavily depends on the choice of the basis functions. In the next section, we briefly discuss about the condition number of B-splines, which will be followed by the main results on the estimates of condition number of isogeometric matrices. We shall discuss the bounds on maximum and minimum eigenvalues of the stiffness matrix and the mass matrix. These bounds are given with respect to h-refinement and p-refinement. The numerical results will be given for h-refinement, for p-refinement, and for r-refinement.

3.1 Condition Number of B-splines

To bound the condition number of matrices resulting from isogeometric discretizations, first we need to know the bounds on B-spline basis functions in L^s-norm[1], where $s \in [1, \infty]$, which is briefly discussed in this section. We estimate the size of the coefficients of a polynomial of degree p in d dimensions when it is represented using the tensor product structure of B-spline basis functions. The condition number of a basis can be defined as follows.

Definition 3.1. *A basis $\{N_i\}$ of a normed linear space is said to be stable with respect to a vector norm if there are constants K_1 and K_2 such that for all coefficients $\{v_i\} = \{v_1, v_2, ..., v_{n_h}\}$ the following relation holds*

$$K_1^{-1}\|\{v_i\}\| \leq \Big\| \sum_i v_i N_i \Big\| \leq K_2 \|\{v_i\}\|. \tag{3.2}$$

The number $\kappa = K_1 K_2$, with K_1 and K_2 as small as possible, is called the condition number of $\{N_i\}$ with respect to the norm $\|\cdot\|$. Note that we use the symbols $\|\cdot\|$ and $\|\{\cdot\}\|$ for the norms in the vector space and the discrete vector norm, respectively.

Such condition numbers give an upper bound for magnification of error in the coefficients to the function values. Indeed, if $f = \sum_i f_i N_i \neq 0$ and $g = \sum_i g_i N_i$, then it follows immediately from (3.2) that

$$\frac{\|f - g\|}{\|f\|} \leq \kappa \frac{\|\{f_i - g_i\}\|}{\|\{f_i\}\|}.$$

More details on the approximation properties and the stability of B-splines can be found in, e.g., [63–65, 77, 78, 83, 87]. We shall use these estimates on κ to estimate the bounds on the condition number of the stiffness matrix and the mass matrix.

It is of central importance for working with B-spline basis functions that its condition number is bounded independently of the underlying knot sequence. That is, the condition number of B-splines does not depend on the multiplicity of the knots of knot vector. This fact was proved by de Boor in 1968 for the sup-norm and in 1973 for any L_s-norm, see [42–44, 47]. In [43] he gave the direct estimate that the worst condition number of a B-spline of degree p with respect to s-norm is bounded above by $p9^p$, and conjectured that the real value of κ grows like 2^p, which is seen far better than the direct estimate:

$$\kappa < p9^p \quad \text{(direct estimate)}, \tag{3.3a}$$

$$\kappa \sim 2^p \qquad \text{(conjecture)}. \tag{3.3b}$$

[1]Note that we use the subscript 's' to represent the norm in L^s-space because we reserve 'p' to represent the degree of the spline basis functions.

In [46], de Boor discussed that the exact condition number of B-spline basis may be hard to determine. Scherer and Shadrin proved that the upper bound of the condition number of a B-spline of degree p with respect to s-norm is bounded by $p^{\frac{1}{2}}4^p$, see [92], i.e.

$$\kappa < p^{\frac{1}{2}}4^p. \tag{3.4}$$

This improves de Boor's estimate $\kappa < p9^p$ and stands closer to his conjecture that $\kappa \sim 2^p$. Later, in [93], Scherer and Shadrin proved the following result.

Lemma 3.2. *For all p and all $s \in [1, \infty]$,*

$$\kappa < p2^p. \tag{3.5}$$

The above lemma confirms the de Boor's conjecture up to a polynomial factor. Further possible approaches by which the polynomial factor could be removed are also discussed in [93].

The above one-dimensional B-spline condition number can be easily generalized to d-dimensions. For the tensor product B-spline basis of degree p in d-dimensions, using (3.5) one obtains the condition number estimate as follows

$$\kappa < (p\,2^p)^d. \tag{3.6}$$

3.2 Condition Number Estimates

This section is devoted to the estimates for the condition number of the stiffness matrix and the mass matrix obtained from isogeometric discretization. We study the condition number of the stiffness matrix and the mass matrix with respect to h-refinement and p-refinement. For h-refinement, upper and lower bounds for the extremal eigenvalues and the condition number are given, whereas for p-refinement we prove upper and lower bounds for the maximum eigenvalue, lower bounds for the minimum eigenvalue, and upper bounds for the condition number. The constant C will be used often in this section for the generic constant that may take different values at different occasions, and is independent of h and p in the analysis with respect to h-refinement and p-refinement, respectively.

3.2.1 Stiffness Matrix

h-refinement

For simplicity, we begin with a two-dimensional domain. Let $\Omega := (0,1)^2$ be an open parametric domain which we will refer as a *patch*. Assume that two open knot vectors $\Xi_1 := \{0 = \xi_1, \xi_2, \xi_3, \ldots, \xi_{m_1} = 1\}$ and $\Xi_2 := \{0 = \eta_1, \eta_2, \eta_3, \ldots, \eta_{m_2} = 1\}$ are given. Associated with Ξ_1 and Ξ_2, we partition the patch Ω in to a mesh

$$\mathcal{Q}_h := \{Q = (\xi_i, \xi_{i+1}) \otimes (\eta_j, \eta_{j+1}), i = p_1{+}1, 2, \ldots, m_1{-}p_1{-}1, j = p_2{+}1, 2, \ldots, m_2{-}p_2{-}1\},$$

where Q is a two-dimensional open knot-span whose diameter is denoted by h_Q. We consider a family of quasi-uniform meshes $\{\mathcal{Q}_h\}_h$ on Ω, where $h = \max\{h_Q | Q \in \mathcal{Q}_h\}$ denotes the family index, see [14]. Furthermore, let $\mathcal{S}_h$ denote the B-spline space associated with the mesh $\mathcal{Q}_h$. Given two adjacent elements Q_1 and Q_2, by $m_{Q_1Q_2}$ we denote the number of continuous derivatives across their common face $\partial Q_1 \cap \partial Q_2$. In the analysis, we will use the following Sobolev space of order $m \in \mathbb{N}$

$$\begin{aligned}\mathcal{H}^m(\Omega) := \Big\{ & v \in L^2(\Omega) \text{ such that } v|_Q \in H^m(Q), \forall Q \in \mathcal{Q}_h, \text{ and} \\ & \nabla^i(v|_{Q_1}) = \nabla^i(v|_{Q_2}) \text{ on } \partial Q_1 \cap \partial Q_2, \\ & \forall i \in \mathbb{N} \text{ with } 0 \le i \le \min\{m_{Q_1Q_2}, m-1\}, \forall Q_1, Q_2 \text{ with } \partial Q_1 \cap \partial Q_2 \neq \emptyset \Big\},\end{aligned} \tag{3.7}$$

where ∇^i has the usual meaning of i^{th}-order partial derivative, and H^m is the usual Sobolev space of order m. The space $\mathcal{H}^m$ is equipped with the following semi-norms and norm

$$|v|^2_{\mathcal{H}^i(\Omega)} := \sum_{Q \in \mathcal{Q}_h} |v|^2_{H^i(Q)}, \quad 0 \le i \le m, \qquad \|v\|^2_{\mathcal{H}^m(\Omega)} := \sum_{i=0}^{m} |v|^2_{\mathcal{H}^i(\Omega)}.$$

On a regular mesh of size h, the condition number of the finite element equations for a second-order elliptic boundary value problem can be obtained using *inverse estimates*, see the classical texts e.g. [5, 29, 33, 37]. Therefore, similar inverse estimates are of interest in isogeometric framework using NURBS basis functions. To keep this chapter self-contained, we recall some results from [14, 94].

Theorem 3.3. *Let $\mathcal{S}_h$ be the spline space consisting of piecewise polynomials of degree p associated with uniform partitions. Then there exists a constant $C = C(shape)$, such that for all $0 \le l \le m$,*

$$\|v\|_{\mathcal{H}^m(\Omega)} \le C h^{l-m} \|v\|_{\mathcal{H}^l(\Omega)}, \quad \forall v \in \mathcal{S}_h. \tag{3.8}$$

The proof of the above theorem, for a particular case $m = 2$ and $l = 1$, is given in [14]. More general inverse inequalities can be easily derived following the same approach. By taking $m = 1$ and $l = 0$, the following can be easily derived from (3.8)

$$a(v, v) = \int_\Omega |\nabla v|^2 \le C h^{-2} \|v\|^2. \tag{3.9}$$

Under suitable conditions, the condition number related to elliptic problems in finite element analysis scales as h^{-2}, see e.g. [51,69,95]. We prove the similar result for the stiffness matrix arising in isogeometric discretization. To prove that we first shall prove the following result.

Lemma 3.4. *There exist constants C_1 and C_2 independent of h (but may depend on p), such that for all $v = \sum_{i=1}^{n_h} v_i N_i \in \mathcal{S}_h$, we have*

$$C_1 h^2 \|\{v_i\}\|^2 \le \Big\| \sum_{i=1}^{n_h} v_i N_i \Big\|^2 \le C_2 h^2 \|\{v_i\}\|^2. \tag{3.10}$$

Proof. We only consider the non-trivial case, i.e. there exists some i for which $v_i \neq 0$. For any $Q \in \mathcal{Q}_h$, there are $(p+1)^2$ basis functions with non-zero support. Let $\mathcal{I}_h^Q \equiv \{i_1^Q, i_2^Q, \ldots, i_{p+1}^Q\} \times \{j_1^Q, j_2^Q, \ldots, j_{p+1}^Q\} \subset \{1, 2, \ldots, n_h\}$ denote the index set for the basis functions which have non-zero support in Q. Also, let $\bar{v}_q = \max_{i \in \mathcal{I}_h^Q} |v_i|$ and $\bar{v} = \max_{i=1,2,\ldots,n_h} |v_i|$. Now using positivity and partition of unity properties of basis functions, the right hand side inequality can be proved as follows:

$$\begin{aligned}
\|v\|^2 = \sum_{Q \in \mathcal{Q}_h} \int_Q v^2 = \sum_{Q \in \mathcal{Q}_h} \int_Q \left(\sum_{i \in \mathcal{I}_h^Q} v_i N_i \right)^2 \leq \sum_{Q \in \mathcal{Q}_h} \int_Q \left(\bar{v}_q \sum_{i \in \mathcal{I}_h^Q} N_i \right)^2 \\
= \sum_{Q \in \mathcal{Q}_h} \int_Q \bar{v}_q^2 \leq \sum_{Q \in \mathcal{Q}_h} h_Q^2 \bar{v}_q^2 \leq \sum_{Q \in \mathcal{Q}_h} h_Q^2 \sum_{i \in \mathcal{I}_h^Q} v_i^2 \\
\leq h^2 \sum_{Q \in \mathcal{Q}_h} \sum_{i \in \mathcal{I}_h^Q} v_i^2 \leq C_2 h^2 \sum_{i=1}^{n_h} v_i^2 = C_2 h^2 \|\{v_i\}\|^2.
\end{aligned}$$

For the left hand side inequality, we have

$$\begin{aligned}
h^2 \|\{v_i\}\|^2 =& h^2 \sum_{i=1}^{n_h} v_i^2 \leq h^2 \sum_{i=1}^{n_h} \bar{v}^2 = h^2 n_h \bar{v}^2 \leq h^2 \left(\frac{C}{h} \right)^2 \bar{v}^2 = C^2 \bar{v}^2 \\
=& C^2 \|\{v_i\}\|_{L_\infty}^2 \leq C^2 K_1^2 \|v\|_{L_\infty}^2 \left(\text{using (3.2), } K_1^{-1} \|\{v_i\}\|_{L_\infty} \leq \left\| \sum v_i N_i \right\|_{L_\infty} \right) \\
\leq& C^2 K_1^2 \|v\|^2.
\end{aligned}$$

The result then follows by taking $C_1 = \left(\frac{1}{C^2 K_1^2} \right)$. □

We now turn to the problem of obtaining bounds on the extremal eigenvalues and the condition number. The main result concerning the condition number of the stiffness matrix is the following.

Theorem 3.5. *Let A be the stiffness matrix, i.e. $A = (a_{ij})$, where $a_{ij} = a(N_i, N_j) = \int_\Omega \nabla N_i \cdot \nabla N_j$, then the bounds on λ_{max} and λ_{min} are given by*

$$k_1 \leq \lambda_{max} \leq k_2, \quad \text{and} \quad k_3 h^2 \leq \lambda_{min} \leq k_4 h^2,$$

where k_1, k_2, k_3 and k_4 are constants independent of h. Furthermore, the bounds on $\kappa(A)$ are given by

$$c_1 h^{-2} \leq \kappa(A) \leq c_2 h^{-2},$$

where c_1 and c_2 are constants independent of h.

Proof. Let $v = \sum_{i=1}^{n_h} v_i N_i$. Then $a(v,v) = \{v_i\} \cdot A\{v_i\}$, where $\{v_i\} = \{v_1, v_2, ..., v_{n_h}\}$. Using inverse estimate (3.9) we get

$$\frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} = \frac{a(v,v)}{\|\{v_i\}\|^2} \leq \frac{Ch^{-2}\|v\|^2}{\|\{v_i\}\|^2} =: \overline{G}.$$

Let $\overline{G}$ be the supremum of $\frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2}$. Using (3.10) we can get the following upper and lower bounds on $\overline{G}$

$$\overline{G} = \frac{Ch^{-2}\|v\|^2}{\|\{v_i\}\|^2} \geq \frac{Ch^{-2}C_1h^2\|\{v_i\}\|^2}{\|\{v_i\}\|^2} = CC_1 = k_1$$
$$\Rightarrow \qquad \sup\sup\frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} = k_1,$$

and

$$\overline{G} = \frac{Ch^{-2}\|v\|^2}{\|\{v_i\}\|^2} \leq \frac{Ch^{-2}C_2h^2\|\{v_i\}\|^2}{\|\{v_i\}\|^2} = CC_2 = k_2$$
$$\Rightarrow \qquad \inf\sup\frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} = k_2.$$

Therefore

$$k_1 \leq \sup_{v \neq 0} \frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} \leq k_2,$$

which implies

$$k_1 \leq \lambda_{\text{max}} \leq k_2. \tag{3.11}$$

On the other hand, for the bounds on λ_{min}, by using coercivity of bilinear form $a(v,v)$ we get

$$\frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} = \frac{a(v,v)}{\|\{v_i\}\|^2} \geq \frac{\alpha\|v\|^2_{H^1}}{\|\{v_i\}\|^2} \geq \frac{\alpha\|v\|^2}{\|\{v_i\}\|^2} =: \underline{G}.$$

Assume $\underline{G}$ is the infimum of $\frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2}$. Using again (3.10), we get

$$\underline{G} = \frac{\alpha_1\|v\|^2}{\|\{v_i\}\|^2} \geq \frac{\alpha_1C_1h^2\|\{v_i\}\|^2}{\|\{v_i\}\|^2} = \alpha_1C_1h^2 = k_3h^2$$
$$\Rightarrow \qquad \sup\inf\frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} = k_3h^2,$$

and

$$\underline{G} = \frac{\alpha_1\|v\|^2}{\|\{v_i\}\|^2} \leq \frac{\alpha_1C_2h^2\|\{v_i\}\|^2}{\|\{v_i\}\|^2} = \alpha_1C_2h^2 = k_4h^2$$
$$\Rightarrow \qquad \inf\inf\frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} = k_4h^2,$$

which implies

$$k_3 h^2 \leq \inf_{v \neq 0} \frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} \leq k_4 h^2.$$

Therefore, we have

$$k_3 h^2 \leq \lambda_{\min} \leq k_4 h^2. \tag{3.12}$$

The condition number of the stiffness matrix is given by

$$\kappa(A) = \frac{\lambda_{\max}}{\lambda_{\min}}, \text{where } \lambda_{\max} = \max_{v \neq 0} \frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2}, \text{ and } \lambda_{\min} = \min_{v \neq 0} \frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2}.$$

From (3.11) and (3.12), we get

$$c_1 h^{-2} \leq \kappa(A) \leq c_2 h^{-2}, \tag{3.13}$$

which concludes the proof. □

Table 3.1: $\lambda_{\max}$ and $\lambda_{\min}$ for $\kappa(A)$

	$1/h$	2	4	8	16	32	64
$p = 2$							
C^0	$\lambda_{\max}$	2.1726	2.5607	2.6436	2.6612	2.6653	2.6663
	$\lambda_{\min}$	0.2929	0.2008	0.0726	0.0190	0.0048	0.0012
C^1	$\lambda_{\max}$	1.4222	1.4238	1.4896	1.4951	1.4991	1.4997
	$\lambda_{\min}$	0.3556	0.3556	0.2855	0.0756	0.0192	0.0048
$p = 3$							
C^0	$\lambda_{\max}$	2.1297	2.2415	2.2844	2.2961	2.2992	2.2999
	$\lambda_{\min}$	0.0284	0.0210	0.0190	0.0085	0.0021	0.0005
C^1	$\lambda_{\max}$	0.8962	1.1705	1.1910	1.2078	1.2129	1.2142
	$\lambda_{\min}$	0.0386	0.0386	0.0386	0.0191	0.0048	0.0012
C^2	$\lambda_{\max}$	1.0384	1.3698	1.5247	1.5627	1.5720	1.5743
	$\lambda_{\min}$	0.0336	0.0464	0.0522	0.0547	0.0191	0.0048
$p = 4$							
C^0	$\lambda_{\max}$	2.1002	2.1105	2.1174	2.1195	2.1200	2.1202
	$\lambda_{\min}$	0.0024	0.0019	0.0018	0.0017	0.0012	0.0003
C^1	$\lambda_{\max}$	0.8752	1.0840	1.1452	1.1606	1.1644	1.1654
	$\lambda_{\min}$	0.0030	0.0030	0.0030	0.0030	0.0021	0.0005
C^2	$\lambda_{\max}$	0.6780	0.9178	0.9847	1.0059	1.0118	1.0133
	$\lambda_{\min}$	0.0040	0.0048	0.0051	0.0052	0.0047	0.0012
C^3	$\lambda_{\max}$	0.9369	1.3334	1.7182	1.8111	1.8311	1.8357
	$\lambda_{\min}$	0.0028	0.0050	0.0072	0.0081	0.0085	0.0048

In Table 3.1 the extremal eigenvalues of the stiffness matrix, using basis functions with continuity from C^0 to C^{p-1}, are given. The extremal eigenvalues support the theoretical estimates given above, i.e. the maximum eigenvalues are independent of h, and the minimum eigenvalues asymptotically scale as h^{-2}.

p-refinement

In this section we estimate the upper bound for the condition number as a function of p. Without loss of generality, we assume a single element mesh, i.e., $Q = \Omega = (0,1)^2$. We denote $\mathcal{S}_p$ the tensor product space of spline functions of degree p. Next we provide some basic technical lemmas which will be needed later on. The following lemma is well known generalization of a theorem of Markov by Hill, Szechuan and Tamarkin, see [25, 86].

Lemma 3.6 (Schmidt's inequality)**.** *There exists a constant C (independent of p) such that for any polynomial $f(x)$ of degree p we have*

$$\int_{-1}^{1}(f'(x))^2dx \leq Cp^4\int_{-1}^{1}(f(x))^2dx. \tag{3.14}$$

No such constant C exists so that (3.14) holds for all $f(x)$ with the exponent smaller than 4.

Let $I = (-1,1)$. Then using (3.14), we have

$$\int_I \left(\frac{dN_p(\xi)}{d\xi}\right)^2 d\xi \leq Cp^4\int_I (N_p(\xi))^2 d\xi. \tag{3.15}$$

Now using (3.15), we get the following

$$\begin{aligned}\int_\Omega \nabla N_p(\xi,\eta)\cdot\nabla N_p(\xi,\eta)d\xi d\eta &= \int_I\int_I\left[\left(\frac{\partial N_p(\xi,\eta)}{\partial\xi}\right)^2+\left(\frac{\partial N_p(\xi,\eta)}{\partial\eta}\right)^2\right]d\xi d\eta \\ &\leq Cp^4\int_\Omega (N_p(\xi,\eta))^2 d\xi d\eta.\end{aligned} \tag{3.16}$$

Moreover, the following estimate directly follows from Schmidt's inequality and (3.16)

$$a(v,v) = \int_\Omega |\nabla v|^2 \leq Cp^4\|v\|^2. \tag{3.17}$$

From this we can have a similar result like in Lemma 3.4 for the p-refinement.

Lemma 3.7. *There exist constants C_1 and C_2 (independent of p), such that for all $v = \sum_{i=1}^{n_p} v_i N_i \in \mathcal{S}_p$, we have*

$$\frac{C_1}{(p^2 4^p)^2}\|\{v_i\}\|^2 \leq \Big\|\sum_{i=1}^{n_p} v_i N_i\Big\|^2 \leq C_2\|\{v_i\}\|^2, \tag{3.18}$$

Proof. From the stability of B-splines there exists a constant γ, which depends on the degree p, such that

$$\Big\|\sum_{i=1}^{n_p} v_i N_i\Big\| \leq \|\{v_i\}\| \leq \gamma\Big\|\sum_{i=1}^{n_p} v_i N_i\Big\|, \tag{3.19}$$

where $\gamma = p^2 4^p$, following (3.6). In the estimate (3.18) the right hand side inequality follows easily from nonnegativity and partition of unity properties of basis functions, and the left hand side inequality follows from (3.19). □

Now we prove the following result, analogous to Theorem 3.5, for the p-refinement of isogeometric discretization.

Theorem 3.8. *Let $\{N_i\}$ be a set of basis function of $\mathcal{S}_p$ on a unit square. Then the following upper bound on $\kappa(A)$ holds*

$$\kappa(A) \leq Cp^8 16^p.$$

Proof. We prove this theorem following the same approach as for the h-refinement estimates. Let $v = \sum_{i=1}^{n_p} v_i N_i$, where $\{v_i\} = \{v_1, v_2, ..., v_{n_p}\}$. Now using (3.17) and (3.18), we get

$$\frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} = \frac{a(v,v)}{\|\{v_i\}\|^2} \leq \frac{Cp^4\|v\|^2}{\|\{v_i\}\|^2} \leq \frac{Cp^4 C_2 \|\{v_i\}\|^2}{\|\{v_i\}\|^2} = CC_2 p^4 = Cp^4,$$

which implies that

$$\lambda_{\text{max}} = \max_{v \neq 0} \frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} \leq Cp^4. \tag{3.20}$$

To prove the lower bound for λ_{min} we use (3.18) and coercivity of bilinear form,

$$\begin{aligned}\frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} = \frac{a(v,v)}{\|\{v_i\}\|^2} &\geq \frac{\alpha\|v\|^2_{H^1}}{\|\{v_i\}\|^2} \geq \frac{\alpha\|v\|^2}{\|\{v_i\}\|^2} \\ &\geq \frac{\alpha\frac{C_1}{(p^2 4^p)^2}\|\{v_i\}\|^2}{\|\{v_i\}\|^2} = \frac{\alpha C_1}{(p^2 4^p)^2} = \frac{C}{(p^4 16^p)},\end{aligned}$$

which implies that

$$\lambda_{\text{min}} = \min_{v \neq 0} \frac{\{v_i\} \cdot A\{v_i\}}{\|\{v_i\}\|^2} \geq \frac{C}{(p^4 16^p)}. \tag{3.21}$$

From (3.20) and (3.21), we get the desired result, i.e.

$$\kappa(A) = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} \leq \frac{Cp^4}{\left(\frac{C}{(p^4 16^p)}\right)} \leq C(p^8 16^p).$$

□

Remark 3.9. *The above result can be easily generalized for higher dimensions.The bound for condition number of the stiffness matrix for d-dimensional problem is given by $(p^{4+2d}4^{pd})$.*

In the above theorem we proved upper bound on the maximum eigenvalue of the stiffness matrix using B-spline basis functions, which is independent of the choice of the basis functions (holds for all kind of basis functions irrespective to their nature). However, from numerical experiments using B-spline basis functions, we observe that λ_{max} is uniformly bounded and is independent of p. This motivates us for further investigations.

The lower bound on the minimum eigenvalues depends on the stability of the B-spline basis functions, which cannot be improved further (specially beyond the de Boor's conjecture). On the other hand the upper bound on the maximum eigenvalue directly depends on the upper bound of bilinear form $a(v,v)$. Therefore, we shall improve the bound for $a(v,v)$ given in (3.17). From the derivative of a B-spline basis function given in (2.8) we can obtain a new upper bound on the maximum eigenvalue of the stiffness matrix, which is given in the following lemma.

Lemma 3.10. *There exists a constant C independent of p, such that for all $v = \sum_{i=1}^{n_p} v_i N_i \in \mathcal{S}_p$, we have*

$$a(v,v) = \int_\Omega |\nabla v|^2 \leq Cp^2. \tag{3.22}$$

Proof. From the derivative of a B-spline basis function given in (2.8), we have

$$\frac{d}{d\xi}N_i^p(\xi) = \frac{p}{\xi_{i+p}-\xi_i}N_i^{p-1}(\xi) - \frac{p}{\xi_{i+p+1}-\xi_{i+1}}N_{i+1}^{p-1}(\xi). \tag{3.23}$$

Without loss of generality, we consider single element mesh where we have $\xi_{i+p} - \xi_i = \xi_{i+p+1} - \xi_{i+1} = 1$. Then (3.23) gives

$$\begin{aligned}
\frac{d}{d\xi}N_i^p(\xi) &= p\left(N_i^{p-1}(\xi) - N_{i+1}^{p-1}(\xi)\right) \\
\Rightarrow \left(\frac{d}{d\xi}N_i^p(\xi)\right)^2 &= p^2\left(N_i^{p-1}(\xi) - N_{i+1}^{p-1}(\xi)\right)^2 \\
&\leq p^2\left(\left(N_i^{p-1}(\xi)\right)^2 + \left(N_{i+1}^{p-1}(\xi)\right)^2\right).
\end{aligned}$$

Integrating over unit interval I, we get

$$\begin{aligned}
\int_I \left(\frac{d}{d\xi}N_i^p(\xi)\right)^2 d\xi &\leq p^2\left(\int_I \left(N_i^{p-1}(\xi)\right)^2 d\xi + \int_I \left(N_{i+1}^{p-1}(\xi)\right)^2 d\xi\right) \\
&\leq p^2\left(\|N_i^{p-1}(\xi)\|^2 + \|N_{i+1}^{p-1}(\xi)\|^2\right) \leq Cp^2.
\end{aligned} \tag{3.24}$$

Now using above we get

$$\begin{aligned}
\int_\Omega \nabla N_i^p(\xi,\eta)\cdot\nabla N_i^p(\xi,\eta)d\xi d\eta &= \int_I\int_I \left[\left(\frac{\partial N_i^p(\xi,\eta)}{\partial\xi}\right)^2 + \left(\frac{\partial N_i^p(\xi,\eta)}{\partial\eta}\right)^2\right] d\xi d\eta \\
&\leq Cp^2,
\end{aligned}$$

which concludes the proof. □

However, this bound is still not independent of p. To further improve this bound, we proceed as follows.

On a unit length interval, the B-spline basis functions of degree p in variable ξ are given by

$$N_{i,\xi}^{p} = (-1)^{i}\binom{p}{i}(\xi-1)^{p-i}\xi^{i}, \quad i = 0,1,2,...,p.$$

Similarly in variable η, we have

$$N_{j,\eta}^{p} = (-1)^{j}\binom{p}{j}(\eta-1)^{p-j}\eta^{j}, \quad j = 0,1,2,...,p.$$

B-spline basis functions in two variables on a unit square element is simply given by the tensor product:

$$N_{i,j,\xi,\eta}^{p,p} = (-1)^{i+j}\binom{p}{i}\binom{p}{j}\xi^{i}\eta^{j}(\xi-1)^{p-i}(\eta-1)^{p-j}, \quad i,j = 0,1,2,...,p.$$

We prove the following lemma for the diagonal entries of the stiffness matrix.

Lemma 3.11. *There exists a constant C independent of p, such that*

$$(A_{(i,j),(i,j)}) = a(N_{i,j,\xi,\eta}^{p,p}, N_{i,j,\xi,\eta}^{p,p}) = \int_0^1\int_0^1 \nabla N_{i,j,\xi,\eta}^{p,p}\cdot\nabla N_{i,j,\xi,\eta}^{p,p}d\xi d\eta \leq C. \tag{3.25}$$

Proof. For all $i,j = 0,1,2,...,p$ we have

$$\begin{aligned}
&a(N_{i,j,\xi,\eta}^{p,p}, N_{i,j,\xi,\eta}^{p,p}) = \int_0^1\int_0^1 \nabla N_{i,j,\xi,\eta}^{p,p}\cdot\nabla N_{i,j,\xi,\eta}^{p,p}d\xi d\eta\\
&= \int_0^1\int_0^1 \left\{\left(\frac{\partial}{\partial\xi}N_{i,j,\xi,\eta}^{p,p}\right)^2 + \left(\frac{\partial}{\partial\eta}N_{i,j,\xi,\eta}^{p,p}\right)^2\right\}d\xi d\eta\\
&= \int_0^1\int_0^1 \left[\left\{\frac{\partial}{\partial\xi}\left((-1)^{i+j}\binom{p}{i}\binom{p}{j}\xi^{i}\eta^{j}(\xi-1)^{p-i}(\eta-1)^{p-j}\right)\right\}^2 +\right.\\
&\qquad\left.\left\{\frac{\partial}{\partial\eta}\left((-1)^{i+j}\binom{p}{i}\binom{p}{j}\xi^{i}\eta^{j}(\xi-1)^{p-i}(\eta-1)^{p-j}\right)\right\}^2\right]d\xi d\eta\\
&= \int_0^1\int_0^1 \left[\left\{\binom{p}{i}\binom{p}{j}\eta^{j}(\eta-1)^{p-j}\frac{\partial}{\partial\xi}\left(\xi^{i}(\xi-1)^{p-i}\right)\right\}^2 +\right.\\
&\qquad\left.\left\{\binom{p}{i}\binom{p}{j}\xi^{i}(\xi-1)^{p-i}\frac{\partial}{\partial\eta}\left(\eta^{j}(\eta-1)^{p-j}\right)\right\}^2\right]d\xi d\eta
\end{aligned}$$

$$\begin{aligned}
&= \int_0^1\int_0^1 \left[\left\{\binom{p}{i}\binom{p}{j}\eta^{j}(\eta-1)^{p-j}\left(i\xi^{i-1}(\xi-1)^{p-i}+(p-i)\xi^{i}(\xi-1)^{p-i-1}\right)\right\}^2 +\right.\\
&\qquad\left.\left\{\binom{p}{i}\binom{p}{j}\xi^{i}(\xi-1)^{p-i}\left(j\eta^{j-1}(\eta-1)^{p-j}+(p-j)\eta^{j}(\eta-1)^{p-j-1}\right)\right\}^2\right]d\xi d\eta\\
&= I + I\!I,
\end{aligned}$$

where

$$I = \binom{p}{i}^2 \binom{p}{j}^2 \int_0^1 \int_0^1 \big\{ i\xi^{i-1}\eta^j(\xi-1)^{p-i}(\eta-1)^{p-j} + (p-i)\xi^i\eta^j(\xi-1)^{p-i-1}(\eta-1)^{p-j} \big\}^2 d\xi d\eta,$$

$$I\!I = \binom{p}{i}^2 \binom{p}{j}^2 \int_0^1 \int_0^1 \big\{ j\xi^i\eta^{j-1}(\xi-1)^{p-i}(\eta-1)^{p-j} + (p-j)\xi^i\eta^j(\xi-1)^{p-i}(\eta-1)^{p-j-1} \big\}^2 d\xi d\eta.$$

Now,

$$\begin{aligned} I &= \binom{p}{i}^2 \binom{p}{j}^2 \int_0^1 \int_0^1 \big\{ i\xi^{i-1}\eta^j(\xi-1)^{p-i}(\eta-1)^{p-j} + (p-i)\xi^i\eta^j(\xi-1)^{p-i-1}(\eta-1)^{p-j} \big\}^2 d\xi d\eta \\ &= \binom{p}{i}^2 \binom{p}{j}^2 \int_0^1 \int_0^1 \left(i^2\xi^{2(i-1)}\eta^{2j}(\xi-1)^{2(p-i)}(\eta-1)^{2(p-j)} \right) d\xi d\eta + \\ &\quad \binom{p}{i}^2 \binom{p}{j}^2 \int_0^1 \int_0^1 \left((p-i)^2\xi^{2i}\eta^{2j}(\xi-1)^{2(p-i-1)}(\eta-1)^{2(p-j)} \right) d\xi d\eta + \\ &\quad \binom{p}{i}^2 \binom{p}{j}^2 \int_0^1 \int_0^1 \left(2i(p-i)\xi^{2i-1}\eta^{2j}(\xi-1)^{2p-2i-1}(\eta-1)^{2(p-j)} \right) d\xi d\eta \\ &= I_1 + I_2 + I_3, \end{aligned}$$

where

$$\begin{aligned} I_1 &= \binom{p}{i}^2 \binom{p}{j}^2 \int_0^1 \int_0^1 \left(i^2\xi^{2(i-1)}\eta^{2j}(\xi-1)^{2(p-i)}(\eta-1)^{2(p-j)} \right) d\xi d\eta \\ &= \binom{p}{i}^2 \binom{p}{j}^2 i^2 \underbrace{\left(\int_0^1 \left(\xi^{2(i-1)}(\xi-1)^{2(p-i)} \right) d\xi \right)}_{=: I_{11}} \underbrace{\left(\int_0^1 \left(\eta^{2j}(\eta-1)^{2(p-j)} \right) d\eta \right)}_{=: I_{12}} \\ &= \binom{p}{i}^2 \binom{p}{j}^2 i^2 I_{11} I_{12}, \\ I_2 &= \binom{p}{i}^2 \binom{p}{j}^2 \int_0^1 \int_0^1 \left((p-i)^2\xi^{2i}\eta^{2j}(\xi-1)^{2(p-i-1)}(\eta-1)^{2(p-j)} \right) d\xi d\eta \\ &= \binom{p}{i}^2 \binom{p}{j}^2 (p-i)^2 \underbrace{\left(\int_0^1 \left(\xi^{2i}(\xi-1)^{2(p-i-1)} \right) d\xi \right)}_{=: I_{21}} \underbrace{\left(\int_0^1 \left(\eta^{2j}(\eta-1)^{2(p-j)} \right) d\eta \right)}_{=: I_{22}} \\ &= \binom{p}{i}^2 \binom{p}{j}^2 (p-i)^2 I_{21} I_{22}, \end{aligned}$$

and

$$\begin{aligned} I_3 &= \binom{p}{i}^2 \binom{p}{j}^2 \int_0^1 \int_0^1 \left(2i(p-i)\xi^{2i-1}\eta^{2j}(\xi-1)^{2p-2i-1}(\eta-1)^{2(p-j)}\right) d\xi d\eta \\ &= \binom{p}{i}^2 \binom{p}{j}^2 2i(p-i) \underbrace{\left(\int_0^1 \left(\xi^{2i-1}(\xi-1)^{2p-2i-1}\right) d\xi\right)}_{=: I_{31}} \underbrace{\left(\int_0^1 \left(\eta^{2j}(\eta-1)^{2(p-j)}\right) d\eta\right)}_{=: I_{32}} \\ &= \binom{p}{i}^2 \binom{p}{j}^2 2i(p-i) I_{31} I_{32}. \end{aligned}$$

We now compute the above integrals.
Case I_1: Clearly, for $i = 0, I_1 = 0$ and for $i = 1, 2, 3, ..., p$,

$$I_{11} = \int_0^1 \left(\xi^{2(i-1)}(\xi-1)^{2(p-i)}\right) d\xi = (-1)^{2(p-i)} \int_0^1 \xi^{(2i-1)-1}(1-\xi)^{(2p-2i+1)-1} d\xi.$$

Using the integration formula for beta function, i.e. $\displaystyle\int_0^1 t^{n-1}(1-t)^{m-1}dt = \frac{\Gamma(n)\Gamma(m)}{\Gamma(n+m)}$, we get

$$\begin{aligned} I_{11} &= \frac{\Gamma(2i-1)\Gamma(2p-2i+1)}{\Gamma(2p)} = \frac{(2i-2)!(2p-2i)!}{(2p-1)!} = \frac{(2i)!(2p-2i)!}{(2p-1)!(4i^2-2i)}, \\ I_{12} &= \int_0^1 \left(\eta^{2j}(\eta-1)^{2(p-j)}\right) d\eta = (-1)^{2(p-j)} \int_0^1 \eta^{(2j+1)-1}(1-\eta)^{(2p-2j+1)-1} d\eta \\ &= \frac{\Gamma(2j+1)\Gamma(2p-2j+1)}{\Gamma(2p+2)} = \frac{(2j)!(2p-2j)!}{(2p+1)!}, \end{aligned}$$

which implies

$$\begin{aligned} I_1 &= \binom{p}{i}^2 \binom{p}{j}^2 i^2 \frac{(2i)!(2p-2i)!}{(2p-1)!(4i^2-2i)} \frac{(2j)!(2p-2j)!}{(2p+1)!} \\ &\leq \frac{1}{2}\binom{p}{i}^2 \binom{p}{j}^2 \frac{(2i)!(2p-2i)!}{(2p-1)!} \frac{(2j)!(2p-2j)!}{(2p+1)!} \quad \left(\text{since } \frac{i^2}{4i^2-2i} \leq \frac{1}{2}\right). \end{aligned}$$

Case I_2: For $i = p$, we get $I_2 = 0$ and for $i = 0, 1, 2, ..., (p-1)$,

$$\begin{aligned} I_{21} &= \int_0^1 \left(\xi^{2i}(\xi-1)^{2(p-i-1)}\right) d\xi = (-1)^{2(p-i-1)} \int_0^1 \xi^{(2i+1)-1}(1-\xi)^{(2p-2i-1)-1} d\xi \\ &= \frac{\Gamma(2i+1)\Gamma(2p-2i-1)}{\Gamma(2p)} = \frac{(2i)!(2p-2i-2)!}{(2p-1)!} = \frac{(2i)!(2p-2i)!}{(2p-1)!(4(p-i)^2-2(p-i))}, \\ I_{22} &= I_{12} = \frac{(2j)!(2p-2j)!}{(2p+1)!}. \end{aligned}$$

Therefore, we get

$$I_2 = \binom{p}{i}^2 \binom{p}{j}^2 (p-i)^2 \frac{(2i)!(2p-2i)!}{(2p-1)!(4(p-i)^2-2(p-i))} \frac{(2j)!(2p-2j)!}{(2p+1)!}$$
$$\leq \frac{1}{2}\binom{p}{i}^2 \binom{p}{j}^2 \frac{(2i)!(2p-2i)!}{(2p-1)!} \frac{(2j)!(2p-2j)!}{(2p+1)!} \quad \left(\text{since } \frac{(p-i)^2}{4(p-i)^2-2(p-i)} \leq \frac{1}{2}\right).$$

Case I_3: Clearly, $I_3 = 0$ for $i = 0$ and $i = p$, and for $i = 1, 2, 3, ..., (p-1)$,

$$I_{31} = \int_0^1 \left(\xi^{2i-1}(\xi-1)^{2p-2i-1}\right) d\xi = (-1)^{2(p-i)-1} \int_0^1 \xi^{(2i)-1}(\xi-1)^{(2p-2i)-1} d\xi$$
$$= -\frac{\Gamma(2i)\Gamma(2p-2i)}{\Gamma(2p)} = -\frac{(2i-1)!(2p-2i-1)!}{(2p-1)!} = -\frac{(2i)!(2p-2i)!}{(2p-1)!(4i(p-i))},$$
$$I_{32} = I_{12} = \frac{(2j)!(2p-2j)!}{(2p+1)!},$$

which implies

$$I_3 = -\binom{p}{i}^2 \binom{p}{j}^2 2i(p-i) \frac{(2i)!(2p-2i)!}{(2p-1)!(4i(p-i))} \frac{(2j)!(2p-2j)!}{(2p+1)!}$$
$$= -\frac{1}{2}\binom{p}{i}^2 \binom{p}{j}^2 \frac{(2i)!(2p-2i)!}{(2p-1)!} \frac{(2j)!(2p-2j)!}{(2p+1)!}.$$

Now, for all $i = 0, 1, 2, ..., p$,

$$I = I_1 + I_2 + I_3$$
$$= \begin{cases} I_2, & \text{if } i = 0, \\ I_1 + I_2 + I_3, & \text{if } i = 1, 2, ..., p-1, \\ I_1, & \text{if } i = p, \end{cases}$$
$$\leq \frac{1}{2}\binom{p}{i}^2 \binom{p}{j}^2 \frac{(2i)!(2p-2i)!}{(2p-1)!} \frac{(2j)!(2p-2j)!}{(2p+1)!}$$
$$= \left\{\binom{p}{i}^2 \frac{(2i)!(2p-2i)!}{(2p-1)!}\right\} \left\{\binom{p}{j}^2 \frac{(2j)!(2p-2j)!}{(2p+1)!}\right\}$$
$$= \frac{(2p)}{(2p+1)} \left\{\binom{p}{i}^2 \frac{(2i)!(2p-2i)!}{(2p)!}\right\} \left\{\binom{p}{j}^2 \frac{(2j)!(2p-2j)!}{(2p)!}\right\}$$
$$\leq \left\{\binom{p}{i}^2 \frac{(2i)!(2p-2i)!}{(2p)!}\right\} \left\{\binom{p}{j}^2 \frac{(2j)!(2p-2j)!}{(2p)!}\right\} = I_a I_b, \text{ where}$$
$$I_a = \binom{p}{i}^2 \frac{(2i)!(2p-2i)!}{(2p)!} = \frac{p!p!}{i!i!(p-i)!(p-i)!} \frac{(2i)!(2p-2i)!}{(2p)!},$$
$$I_b = \binom{p}{j}^2 \frac{(2j)!(2p-2j)!}{(2p)!} = \frac{p!p!}{j!j!(p-j)!(p-j)!} \frac{(2j)!(2p-2j)!}{(2p)!}.$$

Now, we seek the upper bound for I_a. We prove that $I_a \leq C$ by induction on p, where C is a constant independent of p. For $p = 1$, we have $I_a = 1$ for all $i = 0, 1$, i.e. the result holds for the base case. Assume that the result holds for $p = m$ and for all $i = 0, 1, 2, ..., m$, i.e.

$$\frac{m!m!}{i!i!(m-i)!(m-i)!}\frac{(2i)!(2m-2i)!}{(2m)!} \leq C. \tag{3.26}$$

Now we shall show that the result holds for $p = m + 1$ and for all $i = 0, 1, 2, ..., m + 1$. We have

$$\frac{(m+1)!(m+1)!}{i!i!(m+1-i)!(m+1-i)!}\frac{(2i)!(2(m+1)-2i)!}{(2(m+1))!}$$

$$= \begin{cases} \dfrac{(m+1)m!(m+1)m!}{i!i!(m+1-i)(m-i)!(m+1-i)(m-i)!}\dfrac{(2i)!(2(m-i+1))(2(m-i)+1)(2m-2i)!}{(2(m+1))(2m+1)(2m)!}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } i = 0, 1, 2, ..., m, \\ \dfrac{(m+1)!(m+1)!}{(m+1)!(m+1)!(0)!(0)!}\dfrac{(2(m+1))!(0)!}{(2(m+1))!}, \text{if } i = m+1, \end{cases}$$

$$= \begin{cases} \left(\dfrac{m^2+2m+1}{4m^2+6m+2}\right)\left(\dfrac{4(m-i)^2+6(m-i)+2}{(m-i)^2+2(m-i)+1}\right)\left\{\dfrac{m!m!}{i!i!(m-i)!(m-i)!}\dfrac{(2i)!(2m-2i)!}{(2m)!}\right\}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } i = 0, 1, 2, ..., m, \\ 1, \text{if } i = m+1. \end{cases}$$

Using (3.26) and since $\left(\dfrac{m^2+2m+1}{4m^2+6m+2}\right)\left(\dfrac{4(m-i)^2+6(m-i)+2}{(m-i)^2+2(m-i)+1}\right) \leq 1$, we get for all $i = 0, 1, 2, ..., m + 1$,

$$\frac{(m+1)!(m+1)!}{i!i!(m+1-i)!(m+1-i)!}\frac{(2i)!(2(m+1)-2i)!}{(2(m+1))!} \leq C.$$

We now have $I_a \leq C$, where C is a constant independent of p. Similarly we can obtain that $I_b \leq C$. Therefore,

$$I = I_a I_b \leq C.$$

Proceeding in the same way for $I\!I$, we can prove that

$$I\!I \leq C.$$

Therefore,

$$a(N^{p,p}_{i,j,\xi,\eta}, N^{p,p}_{i,j,\xi,\eta}) = I + I\!I \leq C,$$

which concludes the proof. □

Thus, we have proved that $a(N^{p,p}_{i,j,\xi,\eta}, N^{p,p}_{i,j,\xi,\eta})$ is bounded by a constant independent of p. Since the upper bound of the diagonal entries is the upper bound of all the entries of the stiffness matrix, the maximum entry of the stiffness matrix is bounded by a constant independent of p, i.e.

$$a(N^{p,q}_{i,j,\xi,\eta}, N^{p,q}_{k,l,\xi,\eta}) \leq C. \tag{3.27}$$

Similarly, we can prove for three dimensional problem that

$$a(N^{p,q,r}_{i,j,k,\xi,\eta,\zeta}, N^{p,q,r}_{l,m,n,\xi,\eta,\zeta}) \leq C. \tag{3.28}$$

Using (3.27) and (3.28) we can give the following result.

Lemma 3.12. *The maximum eigenvalue of the stiffness matrix A can be bounded below by a constant C (independent of p), i.e.*

$$\lambda_{max}(A) \geq C.$$

Proof. We prove this by using the basics of matrix norms. The max-norm of a matrix is the element-wise norm defined by

$$\|A\|_{\max} = \max\{|a_{ij}|\}.$$

From (3.25), we have

$$\max\{|a_{ij}|\} = C,$$

where C is independent of p. By the equivalence of norms we have

$$\|A\|_2 \geq \|A\|_{\max} = C,$$

which implies

$$\lambda_{\max}(A) \geq C.$$

□

To bound $\lambda_{\max}$ from above we bound the spectral norm by the ℓ_1-norm in the following lemma.

Lemma 3.13. *For any fixed k and l such that $0 \leq k, l \leq p$ and for any $0 \leq i, j \leq p$ we have*

$$\sum_{i=0}^{p}\sum_{j=0}^{p} \left|a(N^{p,p}_{k,l,\xi,\eta}, N^{p,p}_{i,j,\xi,\eta})\right| < C,$$

where C is a constant independent of p.

Proof. We have

$$N^{p,p}_{0,0,\xi,\eta} = (1-\xi)^p(1-\eta)^p.$$

We first prove the bound for the absolute rowsum for the first row, i.e.

$$\sum_{i=0}^{p}\sum_{j=0}^{p} \left|a(N^{p,p}_{0,0,\xi,\eta}, N^{p,p}_{i,j,\xi,\eta})\right| < C,$$

where C is a constant independent of p. We have

$$\begin{aligned}
&a(N^{p,p}_{0,0,\xi,\eta}, N^{p,p}_{i,j,\xi,\eta}) \\
&= \int_0^1 \int_0^1 \nabla N^{p,p}_{0,0,\xi,\eta} \cdot \nabla N^{p,p}_{i,j,\xi,\eta} d\xi d\eta \\
&= \int_0^1 \int_0^1 \left\{ \left(\frac{\partial}{\partial \xi} N^{p,p}_{0,0,\xi,\eta} \frac{\partial}{\partial \xi} N^{p,p}_{i,j,\xi,\eta} \right) + \left(\frac{\partial}{\partial \eta} N^{p,p}_{0,0,\xi,\eta} \frac{\partial}{\partial \eta} N^{p,p}_{i,j,\xi,\eta} \right) \right\} d\xi d\eta \\
&= \int_0^1 \int_0^1 \left(\frac{\partial}{\partial \xi} N^{p,p}_{0,0,\xi,\eta} \frac{\partial}{\partial \xi} N^{p,p}_{i,j,\xi,\eta} \right) d\xi d\eta + \int_0^1 \int_0^1 \left(\frac{\partial}{\partial \eta} N^{p,p}_{0,0,\xi,\eta} \frac{\partial}{\partial \eta} N^{p,p}_{i,j,\xi,\eta} \right) d\xi d\eta \\
&= I + I\!I,
\end{aligned}$$

where

$$I = \int_0^1 \int_0^1 \left(\frac{\partial}{\partial \xi} N^{p,p}_{0,0,\xi,\eta} \frac{\partial}{\partial \xi} N^{p,p}_{i,j,\xi,\eta} \right) d\xi d\eta,$$

and

$$I\!I = \int_0^1 \int_0^1 \left(\frac{\partial}{\partial \eta} N^{p,p}_{0,0,\xi,\eta} \frac{\partial}{\partial \eta} N^{p,p}_{i,j,\xi,\eta} \right) d\xi d\eta.$$

Now

$$\begin{aligned}
I = &\int_0^1 \int_0^1 \left(-p(1-\xi)^{p-1}(1-\eta)^p\right) \\
&\left(\binom{p}{i} \binom{p}{j} \eta^j (1-\eta)^{p-j} \left(i\xi^{i-1}(1-\xi)^{p-i} - (p-i)\xi^i (1-\xi)^{p-i-1} \right) \right) d\xi d\eta \\
= &-p \binom{p}{i} \binom{p}{j} \left(\int_0^1 \eta^j (1-\eta)^{2p-j} d\eta \right) \\
&\left(\int_0^1 i\xi^{i-1}(1-\xi)^{2p-i-1} d\xi - \int_0^1 (p-i)\xi^i (1-\xi)^{2p-i-2} d\xi \right) \\
= &-p \binom{p}{i} \binom{p}{j} (I_1) (I_2 - I_3),
\end{aligned}$$

where

$$I_1 = \int_0^1 \eta^j (1-\eta)^{2p-j} d\eta, \quad I_2 = \int_0^1 i\xi^{i-1}(1-\xi)^{2p-i-1} d\xi,$$

and

$$I_3 = \int_0^1 (p-i)\xi^i (1-\xi)^{2p-i-2} d\xi.$$

For $i = 0, 1, 2, ..., p$ we have

$$\begin{aligned}
I_1 &= \int_0^1 \eta^j (1-\eta)^{2p-j} d\eta = \frac{\Gamma(j+1)\Gamma(2p-j+1)}{\Gamma(2p+2)} \\
&= \frac{(j)!(2p-j)!}{(2p+1)!} = \frac{1}{(2p+1)} \frac{(j)!(2p-j)!}{(2p)!}.
\end{aligned}$$

Clearly, $I_2 = 0$ if $i = 0$. For $i = 1, 2, ..., p$

$$\begin{aligned} I_2 &= \int_0^1 i\xi^{i-1}(1-\xi)^{2p-i-1}d\xi = i\frac{\Gamma(i)\Gamma(2p-i)}{\Gamma(2p)} \\ &= i\frac{(i-1)!(2p-i-1)!}{(2p-1)!} = \frac{2p}{(2p-i)}\frac{(i)!(2p-i)!}{(2p)!}. \end{aligned}$$

For $i = p$ we get $I_3 = 0$ and for $i = 0, 1, ..., p-1$

$$\begin{aligned} I_3 &= \int_0^1 (p-i)\xi^i(1-\xi)^{2p-i-2}d\xi \\ &= (p-i)\frac{\Gamma(i+1)\Gamma(2p-i-1)}{\Gamma(2p)} \\ &= (p-i)\frac{(i)!(2p-i-2)!}{(2p-1)!} = \frac{(p-i)(2p)}{(2p-i)(2p-i-1)}\frac{(i)!(2p-i)!}{(2p)!}. \end{aligned}$$

Therefore we have

$$I = \begin{cases} \binom{p}{j}\dfrac{p^2}{(4p^2-1)}\dfrac{(j)!(2p-j)!}{(2p)!}, & \text{if } i = 0, \\ -p\binom{p}{i}\binom{p}{j}\dfrac{2p}{(2p+1)}\dfrac{(i)!(2p-i)!}{(2p)!}\dfrac{(j)!(2p-j)!}{(2p)!}\dfrac{1}{(2p-i)}\left(1-\dfrac{(p-i)}{(2p-i-1)}\right), & \text{if } i = 1, 2, ..., p-1, \\ -\binom{p}{j}\dfrac{2p}{(2p+1)}\dfrac{(p)!(p)!}{(2p)!}\dfrac{(j)!(2p-j)!}{(2p)!}, & \text{if } i = p. \end{cases}$$

Similar expression can be easily obtained for $I\!I$. We are interested to calculate the sum

$$\sum_{i=0}^{p}\sum_{j=0}^{p}\left|a(N^{p,p}_{0,0,\xi,\eta}, N^{p,p}_{i,j,\xi,\eta})\right|.$$

For $i = 0$, we have

$$\begin{aligned} \sum_{j=0}^{p}|I| &= \sum_{j=0}^{p}\binom{p}{j}\frac{p^2}{(4p^2-1)}\frac{(j)!(2p-j)!}{(2p)!} \\ &< \frac{1}{3}\sum_{j=0}^{p}\binom{p}{j}\frac{(j)!(2p-j)!}{(2p)!}\left(\text{since } \frac{p^2}{(4p^2-1)} \le \frac{1}{3}\right) \\ &= \frac{1}{3}\sum_{j=0}^{p}\frac{(p)!(2p-j)!}{(2p)!(p-j)!} < \frac{1}{3}\left(1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+...+\frac{p!p!}{(2p)!}\right) < 1. \end{aligned}$$

For $i = 1, 2, ..., p-1$, we have

$$\begin{aligned} &\sum_{i=1}^{p-1}\sum_{j=0}^{p}|I| \\ &= \sum_{i=1}^{p-1}\sum_{j=0}^{p}\binom{p}{i}\binom{p}{j}\frac{2p^2}{(2p+1)}\frac{(i)!(2p-i)!}{(2p)!}\frac{(j)!(2p-j)!}{(2p)!}\frac{1}{(2p-i)}\left(1-\frac{(p-i)}{(2p-i-1)}\right) \end{aligned}$$

$$
\begin{aligned}
&< \sum_{i=1}^{p-1} \frac{p!(2p-i)!}{(2p)!(p-i)!} \frac{p(p-1)}{(2p-i)(2p-i-1)} \sum_{j=0}^{p} \frac{p!(2p-j)!}{(2p)!(p-j)!} \left(\text{since } \frac{2p}{(2p+1)} < 1 \right) \\
&= \sum_{i=1}^{p-1} \frac{p!(2p-i)!}{(2p)!(p-i)!} \frac{p(p-1)}{(2p-i)(2p-i-1)} \left(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + ... + \frac{p!p!}{(2p)!} \right) \\
&< \sum_{i=1}^{p-1} \frac{p!(2p-i)!}{(2p)!(p-i)!} \frac{2p(p-1)}{(2p-i)(2p-i-1)} \left(\text{since } 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + ... + \frac{p!p!}{(2p)!} < 2 \right) \\
&= \sum_{i=1}^{p-1} \frac{p-1}{2p-1} \frac{p!(2p-i-2)!}{(2p-2)!(p-i)!} < \frac{1}{2} \sum_{i=1}^{p-1} \frac{p!(2p-i-2)!}{(2p-2)!(p-i)!} \\
&< \frac{1}{2} \left(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + ... + \frac{p!(p-1)!}{(2p-2)!} \right) < 1.
\end{aligned}
$$

For $i = p$, we have

$$
\begin{aligned}
\sum_{j=0}^{p} |I| = \sum_{j=0}^{p} \binom{p}{j} \frac{2p}{(2p+1)} \frac{(p)!(p)!}{(2p)!} \frac{(j)!(2p-j)!}{(2p)!} &< \sum_{j=0}^{p} \frac{p!p!p!(2p-j)!}{(2p)!(2p)!(p-j)!} \\
&< \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + ... + \frac{(p!)^4}{((2p)!)^2} \right) < 1.
\end{aligned}
$$

Therefore, we have

$$
\sum_{i=0}^{p} \sum_{j=0}^{p} |I| < C, \tag{3.29}
$$

where C is independent of p. Similarly, we can get

$$
\sum_{i=0}^{p} \sum_{j=0}^{p} |II| < C. \tag{3.30}
$$

Therefore, from (3.29) and (3.30) we get

$$
\sum_{i=0}^{p} \sum_{j=0}^{p} \left| a(N_{0,0,\xi,\eta}^{p,p}, N_{i,j,\xi,\eta}^{p,p}) \right| < C, \tag{3.31}
$$

where C is a constant independent of p.

The above gives us the absolute row-sum for the first row of the stiffness matrix. Since on a uniform mesh the absolute rowsum for all rows of the stiffness matrix are of the same order upto a constant, we get the desired result, i.e.

$$
\sum_{i=0}^{p} \sum_{j=0}^{p} \left| a(N_{k,l,\xi,\eta}^{p,p}, N_{i,j,\xi,\eta}^{p,p}) \right| < C,
$$

for any fixed k and l such that $0 \leq k, l \leq p$ and for any $0 \leq i, j \leq p$, where C is a constant independent of p. □

Similar results can be obtained for higher dimensions. The next lemma, which gives an upper bound for the maximum eigenvalue, is a direct consequence of the above lemma.

Lemma 3.14. *The maximum eigenvalue of the stiffness matrix A can be bounded above by a constant C, independent of p, i.e.*

$$\lambda_{max}(A) \leq C.$$

Proof. We have

$$\sum_{i=0}^{p}\sum_{j=0}^{p} \left|a(N_{k,l,\xi,\eta}^{p,p}, N_{i,j,\xi,\eta}^{p,p})\right| < C,$$

where C is a constant independent of p, which implies $\|A\|_1 \leq C$. We use the following inequality between matrix norms

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}. \tag{3.32}$$

Since A is symmetric matrix, we have $\|A\|_1 = \|A\|_\infty$, which implies

$$\|A\|_2 \leq |A\|_1 \leq C.$$

Therefore,

$$\lambda_{\max}(A) \leq C.$$

□

From Lemma 3.12 and Lemma 3.14, we have the following result.

Lemma 3.15. *$\lambda_{max}(A) = C$, where C is a constant independent of p.*

Theorem 3.16. *For two dimensional problem the improved upper bound for the condition number of the stiffness matrix A is given as follows*

$$\kappa(A) \leq C(p^2 4^p)^2 = Cp^4 16^p. \tag{3.33}$$

This result can be easily generalized for higher dimensions, the bound for d-dimensional problem is given as follows

$$\kappa(A) \leq Cp^{2d} 4^{pd}. \tag{3.34}$$

Remark 3.17. *We have used the condition number of B-splines $\kappa \sim p2^p$ in reaching the above estimates. If we use the de Boor's conjecture (the condition number of B-splines $\kappa \sim 2^p$) instead, then the upper bound of the stiffness matrix can be further improved and given by*

$$\kappa(A) \leq C4^{pd}. \tag{3.35}$$

3.2.2 Mass Matrix

h-refinement

We now estimate the condition number of the mass matrix. Let M be the mass matrix, i.e. $M = (m_{ij})$, where

$$m_{ij} = (N_i, N_j) = \int_\Omega N_i N_j \qquad i, j = 1, 2, ..., n_h.$$

The following lemma gives estimates on the maximum and minimum eigenvalues of the mass matrix with respect to h.

Lemma 3.18. *For the extremal eigenvalues of the mass matrix $M = (m_{ij}) = (N_i, N_j)$, we have the following estimates*

$$C_1 h^2 \leq \lambda_{min} \leq \lambda_{max} \leq C_2 h^2,$$

where C_1, C_2 are constants independent of h. Furthermore

$$c_1 \leq \kappa(M) \leq c_2,$$

where c_1, c_2 are constants independent of h.

Proof. By recalling the result from (3.10), we can bound both the extremal eigenvalues of the mass matrix. For the minimum eigenvalue, we have

$$\frac{\{v_i\} \cdot M\{v_i\}}{\|\{v_i\}\|^2} = \frac{(v, v)}{\|\{v_i\}\|^2} \geq \frac{C_1 h^2 \|\{v\}\|^2}{\|\{v_i\}\|^2} = C_1 h^2.$$

On the other hand, for the maximum eigenvalue we have

$$\frac{\{v_i\} \cdot M\{v_i\}}{\|\{v_i\}\|^2} = \frac{(v, v)}{\|\{v_i\}\|^2} \leq \frac{C_2 h^2 \|\{v\}\|^2}{\|\{v_i\}\|^2} = C_2 h^2.$$

Therefore we have

$$C_1 h^2 \leq \lambda_{\text{min}} \leq \lambda_{\text{max}} \leq C_2 h^2,$$

which implies

$$c_1 \leq \kappa(M) \leq c_2.$$

□

p-refinement

In this section we estimate the bounds on the extremal eigenvalues and the condition number of the mass matrices for p-refinement. We first prove the following lemma

Lemma 3.19. *The mass matrix is a positive matrix, and all the entries of the mass matrix are bounded above by $\frac{C}{(2p+1)^2}$, where C is a constant independent of p.*

Proof. We have

$$
\begin{aligned}
(M_{(i,j),(k,l)}) &= (N^{p,p}_{i,j,\xi,\eta}, N^{p,p}_{k,l,\xi,\eta}) = \int_0^1 \int_0^1 N^{p,p}_{i,j,\xi,\eta} \cdot N^{p,p}_{k,l,\xi,\eta} d\xi d\eta \\
&= \int_0^1 \int_0^1 \left((-1)^{i+j} \binom{p}{i} \binom{p}{j} \xi^i \eta^j (\xi-1)^{p-i} (\eta-1)^{p-j} \right) \\
&\qquad \left((-1)^{k+l} \binom{p}{k} \binom{p}{l} \xi^k \eta^l (\xi-1)^{p-k} (\eta-1)^{p-l} \right) d\xi d\eta \\
&= \left((-1)^{i+j+k+l} \binom{p}{i} \binom{p}{j} \binom{p}{k} \binom{p}{l} \right) \\
&\qquad \int_0^1 \int_0^1 \left(\xi^{i+k} \eta^{j+l} (\xi-1)^{2p-i-k} (\eta-1)^{2p-j-l} \right) d\xi d\eta \\
&= \left((-1)^{i+j+k+l} (-1)^{2p-i-k} (-1)^{2p-j-l} \binom{p}{i} \binom{p}{j} \binom{p}{k} \binom{p}{l} \right) \\
&\qquad \left(\int_0^1 \xi^{i+k} (1-\xi)^{2p-i-k} d\xi d\eta \right) \left(\int_0^1 \eta^{j+l} (1-\eta)^{2p-j-l} d\xi d\eta \right) \\
&= \binom{p}{i} \binom{p}{j} \binom{p}{k} \binom{p}{l} \left(\int_0^1 \xi^{(i+k+1)-1} (1-\xi)^{(2p-i-k+1)-1} d\xi d\eta \right) \\
&\qquad \left(\int_0^1 \eta^{(j+l+1)-1} (1-\eta)^{(2p-j-l+1)-1} d\xi d\eta \right) \\
&= (I)\,(II)\,,
\end{aligned}
$$

where

$$
\begin{aligned}
I &= \binom{p}{i} \binom{p}{k} \left(\int_0^1 \xi^{(i+k+1)-1} (1-\xi)^{(2p-i-k+1)-1} d\xi d\eta \right) \\
&= \binom{p}{i} \binom{p}{k} \frac{\Gamma(i+k+1)\Gamma(2p-i-k+1)}{\Gamma(2p+2)} \\
&= \frac{p!p!}{i!k!(p-i)!(p-k)!} \frac{(i+k)!(2p-i-k)!}{(2p+1)!} \\
&= \frac{1}{2p+1} \underbrace{\left\{ \frac{p!p!}{i!k!(p-i)!(p-k)!} \frac{(i+k)!(2p-i-k)!}{(2p)!} \right\}}_{I_1} = \frac{1}{2p+1} I_1,
\end{aligned}
$$

and

$$
\begin{aligned}
II &= \binom{p}{j} \binom{p}{l} \left(\int_0^1 \eta^{(j+l+1)-1} (1-\eta)^{(2p-j-l+1)-1} d\xi d\eta \right) \\
&= \binom{p}{j} \binom{p}{l} \frac{\Gamma(j+l+1)\Gamma(2p-j-l+1)}{\Gamma(2p+2)} \\
&= \frac{p!p!}{j!l!(p-j)!(p-l)!} \frac{(j+l)!(2p-j-l)!}{(2p+1)!}
\end{aligned}
$$

$$= \frac{1}{2p+1} \underbrace{\left\{ \frac{p!p!}{j!l!(p-j)!(p-l)!} \frac{(j+l)!(2p-j-l)!}{(2p)!} \right\}}_{I\!I_1} = \frac{1}{2p+1} I\!I_1.$$

Now, by induction on p, we can easily obtain that (as we proved in Lemma 3.11),

$$I_1 = \left\{ \frac{p!p!}{i!k!(p-i)!(p-k)!} \frac{(i+k)!(2p-i-k)!}{(2p)!} \right\} \leq C.$$

Similarly, $I\!I_1 \leq C$. Therefore

$$(M_{(i,j),(k,l)}) \leq \frac{C}{(2p+1)^2}. \tag{3.36}$$

It is also clear that for all $p \geq 1$ and $i, k = 0, 1, 2, ..., p$, $I_1 > 0$, and $I\!I_1 > 0$. This implies that the mass matrix $(M_{(i,j),(k,l)})$ is a positive matrix. □

Lemma 3.20. *The maximum eigenvalue of the mass matrix M can be bounded below as follows*

$$\lambda_{max}(M) \geq \frac{C}{(2p+1)^2}.$$

Proof. Following the proof of Lemma 3.12 and (3.36), we get the desired result. □

To bound $\lambda_{\max}$ from above we bound the spectral norm by the ℓ_1-norm of the mass matrix. In the following lemma we first compute the ℓ_1-norm of the mass matrix.

Lemma 3.21. *For the mass matrix M on a unit square element, we have*

$$\|M\|_1 = \frac{1}{(p+1)^2}.$$

Proof. We have

$$\begin{aligned}
\|M\|_1 &= \max_{i,j} \sum_{k,l} \int_0^1 \int_0^1 N^{p,p}_{i,j,\xi,\eta} \cdot N^{p,p}_{k,l,\xi,\eta} d\xi d\eta = \max_{i,j} \sum_{k,l} (N^{p,p}_{i,j,\xi,\eta}, N^{p,p}_{k,l,\xi,\eta}) \\
&= \max_{i,j} (N^{p,p}_{i,j,\xi,\eta}, \sum_{k,l} N^{p,p}_{k,l,\xi,\eta}) = \max_{i,j} (N^{p,p}_{i,j,\xi,\eta}, 1) \qquad \Big(\text{since } \sum_{k,l} N^{p,p}_{k,l,\xi,\eta} = 1\Big) \\
&= \max_{i,j} \int_0^1 \int_0^1 N^{p,p}_{i,j,\xi,\eta} d\xi d\eta.
\end{aligned}$$

Now,

$$\begin{aligned}
&\int_0^1 \int_0^1 N^{p,p}_{i,j,\xi,\eta} d\xi d\eta = \int_0^1 \int_0^1 (-1)^{i+j} \binom{p}{i} \binom{p}{j} \xi^i \eta^j (\xi-1)^{p-i} (\eta-1)^{p-j} d\xi d\eta \\
&= \binom{p}{i} \binom{p}{j} \int_0^1 \int_0^1 (-1)^{i+j+p-i+p-j} \xi^i \eta^j (1-\xi)^{p-i} (1-\eta)^{p-j} d\xi d\eta
\end{aligned}$$

$$= \binom{p}{i}\binom{p}{j}\left(\int_0^1 \xi^{(i+1)-1}(1-\xi)^{(p-i+1)-1}d\xi\right)\left(\int_0^1 \eta^{(j+1)-1}(1-\eta)^{(p-j+1)-1}d\eta\right)$$
$$= \binom{p}{i}\binom{p}{j}\left(\frac{\Gamma(i+1)\Gamma(p-i+1)}{\Gamma(p+2)}\right)\left(\frac{\Gamma(j+1)\Gamma(p-j+1)}{\Gamma(p+2)}\right)$$
$$= \frac{p!}{i!(p-i)!}\frac{p!}{j!(p-j)!}\frac{i!(p-i)!}{(p+1)!}\frac{j!(p-j)!}{(p+1)!} = \frac{1}{(p+1)^2}.$$

The above implies

$$\max_{i,j}\int_0^1\int_0^1 N^{p,p}_{i,j,\xi,\eta}d\xi d\eta = \frac{1}{(p+1)^2},$$

which concludes the proof. □

The symmetry of M implies

$$\|M\|_\infty = \|M\|_1 = \frac{1}{(p+1)^2}. \tag{3.37}$$

Lemma 3.22. *The maximum eigenvalue of the mass matrix M can be bounded above as follows*

$$\lambda_{max}(M) \leq \frac{1}{(p+1)^2}.$$

Proof. Using the inequality (3.32) for matrix norm, Lemma 3.21 and (3.37), we get the bound on the spectral norm of M

$$\|M\|_2 \leq \frac{1}{(p+1)^2},$$

which gives the desired result. □

Remark 3.23. *In fact, we get $\lambda_{max}(M) = \frac{1}{(p+1)^2}$ by Lemma 3.19 and by [99, Lemma 2.5].*

Lemma 3.24. *There exists a constant C, independent of p, such that the minimum eigenvalue of the mass matrix M can be bounded below as follows*

$$\lambda_{min}(M) \geq \frac{C}{p^4 16^p}.$$

Proof. To bound the minimum eigenvalue from below we use the left hand side inequality of (3.18). We have

$$\frac{\{v_i\}\cdot M\{v_i\}}{\|\{v_i\}\|^2} = \frac{(v,v)}{\|\{v_i\}\|^2} \geq \frac{\frac{C}{p^4 16^p}\|\{v_i\}\|^2}{\|\{v_i\}\|^2} = \frac{C}{p^4 16^p}.$$

Therefore, $\lambda_{\min}(M) \geq \frac{C}{p^4 16^p}$, where C is a constant, independent of p. □

The following theorem gives us the upper bound for the condition number of the mass matrix.

Theorem 3.25. *The condition number of the mass matrix M is bounded above by*

$$\kappa(M) \leq Cp^2 16^p,$$

where C is a constant, independent of p.

Proof. From Lemma 3.22 and Lemma 3.24 we have

$$\frac{C}{p^4 16^p} \leq \lambda_{\text{min}} \leq \lambda_{\text{max}} \leq \frac{1}{(p+1)^2},$$

which gives the desired result. □

Remark 3.26. *The above bound can be easily generalized for d-dimensional problem, and is given as follows*

$$\kappa(M) \leq p^{2(d-1)} 4^{pd}. \tag{3.38}$$

Furthermore, following Remark 3.17, by using the de Boor's conjecture, the upper bound for the condition number of the mass matrix can be further improved and given as follows

$$\kappa(M) \leq p^{-2} 4^{pd}. \tag{3.39}$$

Remark 3.27. *We have done all the analysis on the parametric domain $(0,1)^2$. To get the results for physical domain we can define an invertible NURBS geometrical map from parametric domain to physical domain, and with suitable transformations we get the results for physical domain. For details we refer the article by Bazilevs et al. [14].*

3.3 Numerical Results

In this section we provide the numerical results. Apart from the h-refinement and the p-refinement, for which we have established theoretical results, we also provide numerical results for the r-refinement, where we have a possibility to vary the continuity of the basis functions from C^0 to C^{p-1}. As we shall see, however, the difference between the condition numbers for C^0 and C^{p-1} continuous basis functions is hardly of order p, and the results are dominated by the exponent term 4^{pd}.

The numerical discretizations are performed using the Matlab toolbox GeoPDEs [48, 49].

3.3.1 h-refinement

For the h-refinement, the condition number of the stiffness matrix is shown in Table 3.2 for C^0 and C^{p-1} continuous basis functions. Numerical results are provided from $p = 2$ to $p = 5$. The results show a different behavior than the classical finite element method for higher p. In classical finite element method, the condition number of the stiffness matrix is

Table 3.2: Condition number of the stiffness matrix A

p \ h^{-1}	2	4	8	16	32	64
C^0 inter element continuity						
2	7.41	12.75	36.40	140.00	554.99	2215.03
3	75.11	106.56	120.34	269.99	1075.42	4297.18
4	881.40	1099.74	1189.14	1214.75	1761.85	7041.29
5	11093.95	12951.15	13680.12	13886.00	13939.10	13952.48
C^{p-1} inter element continuity						
2	4.00	4.00	5.22	19.77	78.14	311.58
3	30.93	29.51	29.19	28.56	82.10	327.21
4	339.92	269.23	240.03	222.55	215.00	381.73
5	4177.20	3220.60	2148.25	1812.58	1700.63	1688.11

of order h^{-2} even for a coarse mesh-size, but in isogeometric discretizations, for higher p the condition number does not appear to be of order h^{-2} for a coarse mesh-size. This is due to the stability of B-splines. The condition number of B-splines heavily depends on polynomial degree (as explained in Sec. 3.1), and scales as $(p2^p)^d$. This factor $(p2^p)^d$ dominates the factor h^{-2} for coarse meshes. However, the numerical results support the theoretical findings asymptotically (for reasonably refined meshes) for any polynomial degree.

In Table 3.3, we present the condition number of the mass matrix for C^0 and C^{p-1} continuous basis functions. We see that the condition number is bounded uniformly by a constant independent of h, which confirms the theoretical estimates.

Table 3.3: Condition number of the mass matrix M

p \ h^{-1}	2	4	8	16	32
C^0 inter element continuity					
2	181.43	203.92	208.18	208.50	208.95
3	2295.08	2560.70	2629.67	2641.57	2641.85
4	29390.74	32123.43	32891.28	33071.24	33103.92
5	387089.92	414957.07	422941.81	424969.15	425438.49
C^{p-1} inter element continuity					
2	89.679	109.68	108.51	109.85	111.29
3	915.558	799.941	737.379	708.010	715.89
4	11773.17	6795.46	5381.96	4762.53	4750.07
5	163371.70	77448.11	42580.04	33560.40	32587.27

3.3.2 p-refinement

We perform numerical experiments to obtain the maximum and minimum eigenvalues, and the condition number of the stiffness matrix and the mass matrix. The eigenvalues and the condition number are obtained on a single unit square element.

In Table 3.4, we present the extremal eigenvalues and the condition number of the stiffness matrix for $p = 2$ to $p = 15$ (for higher p roundoff errors start contaminating the results). We observe that the maximum eigenvalue scales as a constant independent of p, and the

Table 3.4: λ_{max}, λ_{min}, and $\kappa(A)$ for $p = 2$ to $p = 15$ on one element

p	λ_{max}	λ_{min}	$\kappa(A)$	$\frac{\kappa_p(A)}{\kappa_{p-1}(A)}$
2	3.5e-01	3.5e-01	1.0e+00	–
3	4.5e-01	3.8e-02	1.1e+01	11.67
4	4.1e-01	2.9e-03	1.3e+02	11.82
5	3.5e-01	2.1e-04	1.6e+03	11.80
6	3.3e-01	1.5e-05	2.1e+04	13.31
7	3.3e-01	1.1e-06	2.9e+05	13.77
8	3.1e-01	7.8e-08	4.0e+06	13.71
9	3.0e-01	5.4e-09	5.6e+07	13.88
10	3.0e-01	3.7e-10	8.1e+08	14.33
11	3.0e-01	2.5e-11	1.1e+10	14.58
12	3.0e-01	1.7e-12	1.7e+11	14.63
13	3.0e-01	1.1e-13	2.5e+12	14.70
14	2.9e-01	8.0e-15	3.7e+13	14.77
15	2.9e-01	5.0e-16	5.5e+14	14.62

minimum eigenvalue is bounded from below by the bound given in Theorem 3.8, i.e. $\lambda_{\text{min}} \geq C/(p^4 16^p)$, and the ratio $\kappa_p(A)/\kappa_{p-1}(A)$ is bounded by 16. In Fig. 3.1, we plot these results, which confirm the behavior of condition number according to the theoretical estimates.

The extremal eigenvalues and the condition number of the mass matrix for increasing p are presented in Table 3.5. In Fig. 3.2, we plot extremal eigenvalues and the condition number of the mass matrix against polynomial degree. We perform numerical results for $p = 2$ to $p = 10$ (we stop at $p = 10$ due to roundoff errors). Numerical results confirm the theoretical estimates given in Lemma 3.22, Lemma 3.24, and Theorem 3.25.

3.3.3 r-refinement

In this section we study how the condition number behaves with increasing smoothness of basis functions, i.e., with respect to the r-refinement. We numerically compute to get the

Table 3.5: λ_{max}, λ_{min}, and $\kappa(M)$ for $p = 2$ to $p = 10$ on one element

p	λ_{max}	λ_{min}	$\kappa(M)$	$\frac{\kappa_p(M)}{\kappa_{p-1}(M)}$
2	1.1e-01	1.1e-03	1.0e+02	–
3	6.2e-02	5.1e-05	1.2e+03	12.25
4	4.0e-02	2.5e-06	1.5e+04	12.96
5	2.7e-02	1.3e-07	2.1e+05	13.44
6	2.0e-02	6.9e-09	2.9e+06	13.80
7	1.5e-02	3.7e-10	4.1e+07	14.06
8	1.2e-02	2.0e-11	5.9e+08	14.27
9	1.0e-02	1.1e-12	8.5e+09	14.44
10	8.2e-03	6.6e-14	1.2e+11	14.58

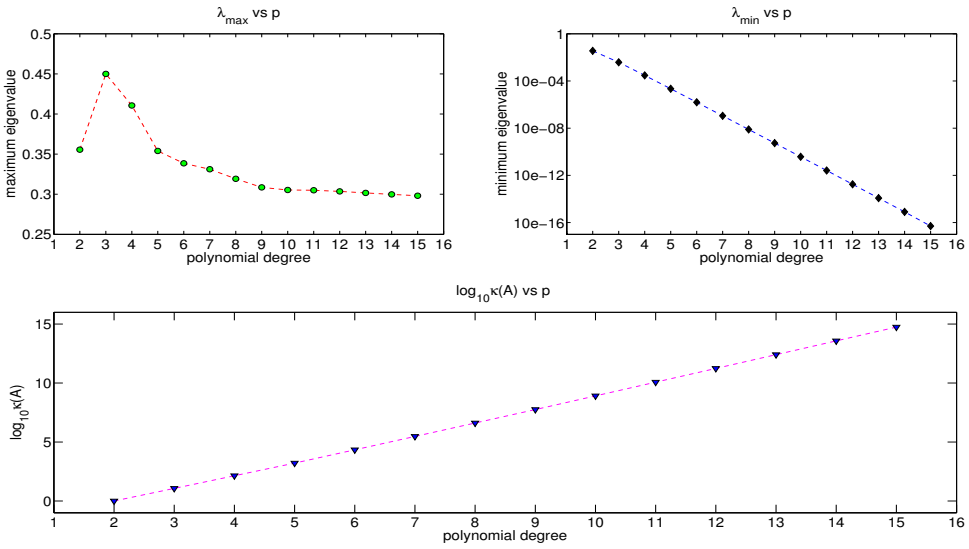


Figure 3.1: The maximum eigenvalue, the minimum eigenvalue, and the condition number of the stiffness matrix is shown on a square element for increasing p. In the top-left graph we plot the maximum eigenvalue versus polynomial degree, in the top-right the graph is given for the minimum eigenvalue versus polynomial degree, and at bottom the condition number $\kappa(A)$ is plotted against polynomial degree.

condition number of the stiffness matrix and the mass matrix for increasing smoothness for both, the h-refinement and the p-refinement. Though, one would expect the condition number to be the best and the worst in case of maximum and minimum smoothness of basis functions, respectively, however, from numerical tests we see that the best condition number is obtained for $C^{\lfloor \frac{p}{2} \rfloor}$ or $C^{\lceil \frac{p}{2} \rceil}$ continuous basis functions. Here, $C^{\lfloor \frac{p}{2} \rfloor}$ and $C^{\lceil \frac{p}{2} \rceil}$ denote the

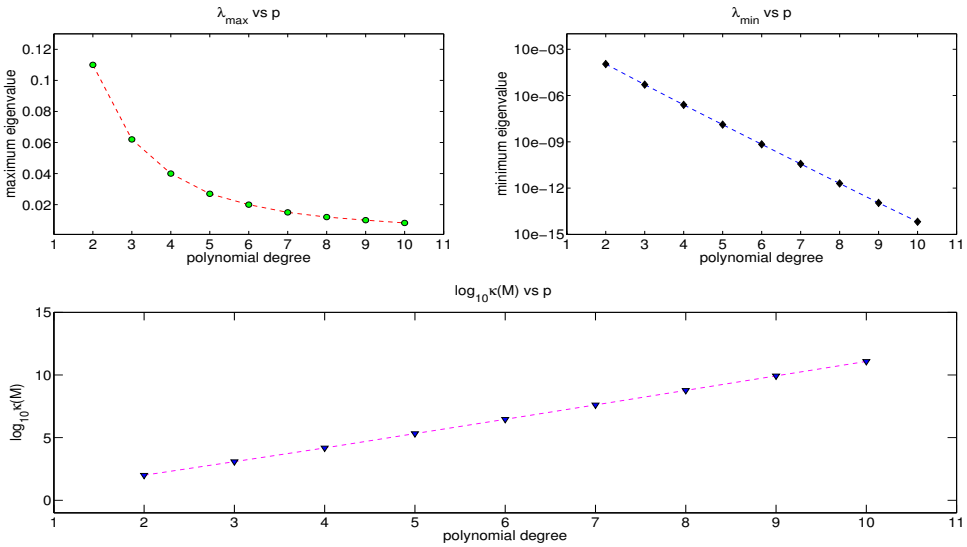


Figure 3.2: The graphs of the extremal eigenvalues of the mass matrix M against polynomial degree are given in the top left and top right. At bottom the condition number of M is plotted against polynomial degree.

floor value and ceiling value, respectively.

h-refinement

In Fig. 3.3 and Fig. 3.4, we plot the extremal eigenvalues of the stiffness matrix and the mass matrix, respectively for increasing smoothness of basis functions with respect to the h-refinement. In Table 3.6 and Table 3.7, we present the $\kappa(A)$ and $\kappa(M)$, respectively for varying smoothness of basis functions for $p = 5$. The following observation can be made from numerical results.

Table 3.6: $\kappa(A)$ for h-refinement with varying smoothness, $p = 5$

h \ r	C^0	C^1	C^2	C^3	C^4
8×8	1.4e+04	4.5e+03	1.4e+03	1.1e+03	2.1e+03
16×16	1.4e+04	4.5e+03	1.3e+03	9.8e+02	1.8e+03
32×32	1.4e+04	4.6e+03	1.3e+03	9.7e+02	1.7e+03
64×64	1.4e+04	4.6e+03	1.5e+03	9.7e+02	1.7e+03

- The maximum eigenvalue of the stiffness matrix is maximum corresponding to C^0 or C^{p-1} continuous basis function, and it is minimum for $C^{\lfloor \frac{p}{2} \rfloor}$ continuous basis func-

Table 3.7: $\kappa(M)$ for h-refinement with varying smoothness, $p = 5$

h \ r	C^0	C^1	C^2	C^3	C^4
8×8	4.2e+05	2.1e+05	7.4e+04	4.7e+04	4.3e+04
16×16	4.2e+05	2.1e+05	7.5e+04	4.5e+04	3.4e+04
32×32	4.3e+05	2.1e+05	7.5e+04	4.6e+04	3.3e+04
64×64	4.3e+05	2.1e+05	7.5e+04	4.6e+04	3.3e+04

tions. The minimum eigenvalue of the stiffness matrix keeps moving away from zero with increasing smoothness. See Fig. 3.3 and Table 3.6.

- For the mass matrix, the maximum and minimum eigenvalues uniformly keep moving away from zero with increasing smoothness of basis functions, i.e. the condition number is the best and the worst for C^{p-1} and C^0 continuous basis functions, respectively. See Fig. 3.4 and Table 3.7.

p-refinement

For p-refinement, we perform numerical computations on a 2×2 mesh (because there is no varying regularity for one element). In Table 3.8 and Table 3.9, we present the condition number of the stiffness matrix and the mass matrix, respectively for $p = 2$ to $p = 10$ with increasing continuity from C^0 to C^{p-1}. In Fig. 3.5 and Fig. 3.6, we plot the condition number of the stiffness matrix and the mass matrix, respectively with varying continuity of basis functions with all possible cases (from minimum C^0 to maximum C^{p-1}). We have the following observations from numerical results.

- The condition number of the stiffness matrix and the mass matrix is the worst for the C^0 continuous basis functions. However, for all possible cases of smoothness the condition number of the stiffness matrix and the mass matrix is not the best for maximum smoothness, i.e C^{p-1} continuous basis functions.

- In Table 3.8, for $p = 2$ to $p = 9$, the condition number of the stiffness matrix decreases from C^0 to $C^{\lfloor \frac{p}{2} \rfloor}$ continuous basis functions and then it increases up to C^{p-1}, and for $p = 10$ the condition number decreases from C^0 to $C^{\frac{p}{2}-1}$, and then increases up to C^{p-1}.

- The condition number of the mass matrix, for $p = 2$ to $p = 6$, decreases from C^0 to $C^{\lceil \frac{p}{2} \rceil}$ continuous basis functions, and then it increases up to C^{p-1}. For $p = 7$ to $p = 10$ the condition number decreases from C^0 to $C^{\lfloor \frac{p}{2} \rfloor}$, and then increases up to C^{p-1}.

- In Fig. 3.5, Fig. 3.6, Table 3.8 and Table 3.9, we see that there is a difference of a factor about p between the best and the worst condition numbers, which is negligible as the exponent term 4^{pd} is the dominating factor for higher p.

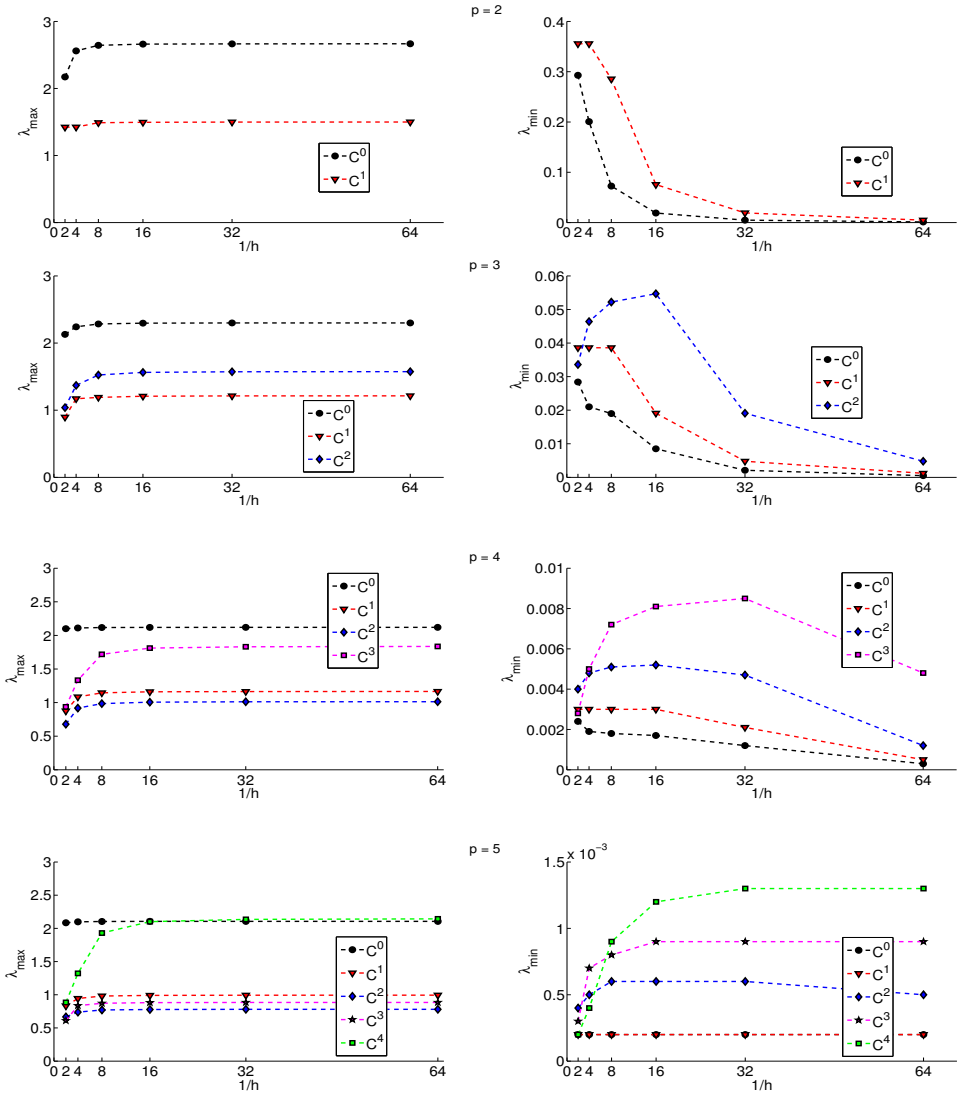


Figure 3.3: The extremal eigenvalues of the stiffness matrix A w.r.t. the h-refinement and varying smoothness.

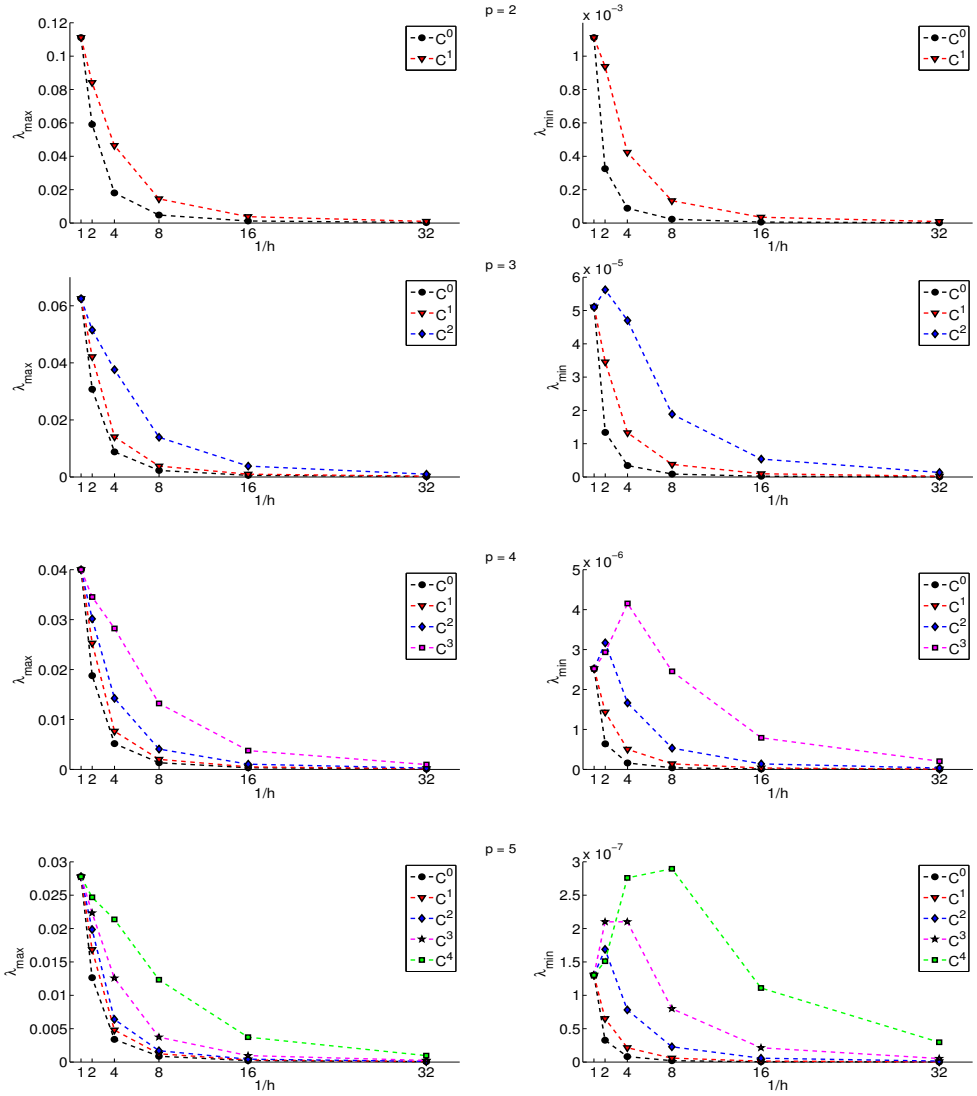


Figure 3.4: The extremal eigenvalues of the stiffness matrix M w.r.t. the h-refinement and varying smoothness.

Table 3.8: Condition number of the stiffness matrix A for $p = 2$ to $p = 10$ with continuity from C^0 to C^{p-1}, $h = 1/2$

p	C^0	C^1	C^2	C^3	C^4	C^5	C^6	C^7	C^8	C^9
2	7.4e+00	4.0e+00	–	–	–	–	–	–	–	–
3	7.5e+01	2.3e+01	3.1e+01	–	–	–	–	–	–	–
4	8.8e+02	2.9e+02	1.7e+02	3.4e+02	–	–	–	–	–	–
5	1.1e+04	3.8e+03	1.7e+03	1.9e+03	4.2e+03	–	–	–	–	–
6	1.5e+05	5.0e+04	2.0e+04	1.5e+04	2.5e+04	5.4e+04	–	–	–	–
7	2.0e+06	6.7e+05	2.6e+05	1.4e+05	1.9e+05	3.6e+05	7.5e+05	–	–	–
8	2.8e+07	9.3e+06	3.6e+06	1.6e+06	1.5e+06	2.8e+06	5.2e+06	1.0e+07	–	–
9	3.9e+08	1.3e+08	5.1e+07	2.1e+07	1.3e+07	2.1e+07	4.4e+07	7.7e+07	1.5e+08	–
10	5.6e+09	1.9e+09	7.7e+08	3.2e+08	1.4e+08	1.5e+08	3.3e+08	6.9e+08	1.1e+09	2.1e+09

Table 3.9: Condition number of the mass matrix M for $p = 2$ to $p = 10$ with continuity from C^0 to C^{p-1}, $h = 1/2$

p	C^0	C^1	C^2	C^3	C^4	C^5	C^6	C^7	C^8	C^9
2	1.8e+02	9.0e+01	–	–	–	–	–	–	–	–
3	2.3e+03	1.2e+03	9.2e+02	–	–	–	–	–	–	–
4	2.9e+04	1.8e+04	9.5e+03	1.2e+04	–	–	–	–	–	–
5	3.9e+05	2.6e+05	1.2e+05	1.1e+05	1.6e+05	–	–	–	–	–
6	5.2e+06	3.8e+06	1.7e+06	1.0e+06	1.5e+06	2.3e+06	–	–	–	–
7	7.3e+07	5.7e+07	2.6e+07	1.2e+07	1.2e+07	2.2e+07	3.4e+07	–	–	–
8	1.0e+09	8.5e+08	4.1e+08	1.6e+08	1.1e+08	1.8e+08	3.3e+08	4.9e+08	–	–
9	1.5e+10	1.3e+10	6.6e+09	2.5e+09	1.2e+09	1.4e+09	2.8e+09	5.1e+09	7.3e+09	–
10	2.1e+11	1.9e+11	1.1e+11	4.2e+10	1.6e+10	1.2e+10	2.2e+10	4.5e+10	7.8e+10	1.1e+11

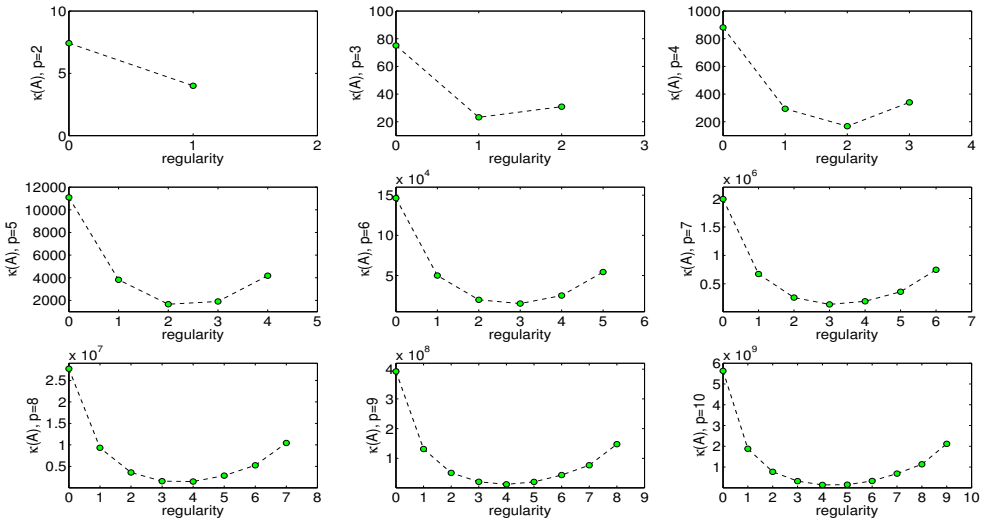


Figure 3.5: The behavior of the condition number of the stiffness matrix on a 2×2 mesh with varying continuity of basis functions on interfaces. The results are shown from $p = 2$ to $p = 10$ and with smoothness from C^0 to C^{p-1}.

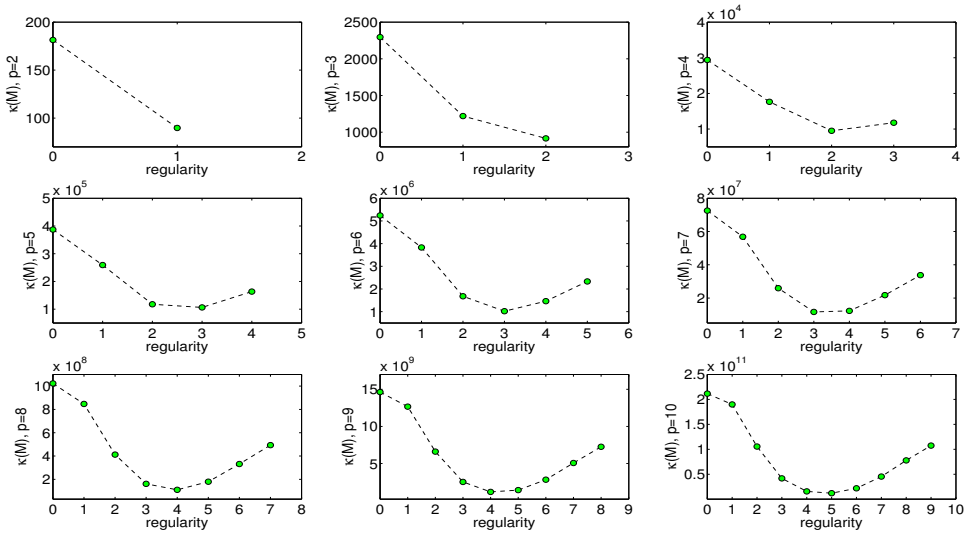


Figure 3.6: The condition number of the mass matrix is plotted on a 2×2 mesh with varying continuity of basis functions on interfaces. The results are shown from $p = 2$ to $p = 10$ and with smoothness from C^0 to C^{p-1}.

3.4 Conclusions

We have provided the bounds for the minimum eigenvalue, maximum eigenvalue and the condition numbers of the stiffness matrix and the mass matrix for the Laplace operator with the h-refinement and the p-refinement of the isogeometric discretizations based on B-spline (NURBS) basis functions. We proved that in the h-refinement, like finite element method, the condition number of the stiffness matrix scales as h^{-2}, and for the mass matrix it scales as constant independent of h. For p-refinement, we have shown that the condition number of the stiffness matrix and the mass matrix grows exponentially in p.

For p-refinement, the estimates for the maximum eigenvalues of the stiffness matrix and the mass matrix are sharp and can not be improved. However, the estimates for the minimum eigenvalues of the stiffness matrix and the mass matrix depend on the stability constant of B-splines. In reaching these estimates we have used the stability constant of B-splines as $p2^p$. Using the de Boor's conjecture (the stability constant of B-splines given by 2^p, which is the best known bound to our knowledge), these estimates can be further improved according to Remarks 3.17 and 3.26. Unfortunately, a sharp estimate for the stability constant is not known, and therefore, a sharp estimate for the minimum eigenvalue is hard to determine. Furthermore, the effect of continuity of basis functions is negligible on the condition number because the difference between the best and the worst condition number is about a factor of p, which is dwarfed by the exponent term 4^{pd}.

Chapter 4

Explicit and Multilevel Representation of B-splines and NURBS

In this chapter, we present the explicit representation of B-splines for a uniform mesh on a unit interval. The explicit representation of B-spline basis functions for a fixed mesh size h is given for $p = 2, 3, 4$ and for C^0- and C^{p-1}-continuity. Moreover, we present the multilevel structure of B-spline and NURBS. The multilevel transfer operators from fine level to coarse level are given in matrix form. The NURBS transfer operators are generated from B-splines transfer operators.

4.1 Explicit Representation for B-splines

The recursive form of B-spline basis functions, given by (2.1), is elegant and concise, and is presented in all the isogeometric related references, see e.g. [39, 67], and also in classical texts [45, 88, 90, 94]. However, this form may not be the most efficient from computational point of view, specially when dealing with large knot vectors. To the best of authors' knowledge, within the isogeometric literature there is no reference on the explicit representation of B-splines for a given mesh size h. Therefore, we present the explicit form of B-splines in terms of the mesh size (knot-span) h. Having an explicit form of basis functions is also advantageous in devising inter-grid transfer operators for multigrid and multilevel iterative solvers. For brevity reasons, we restrict ourselves to a unit interval with equal spacing. Moreover, as most of the engineering designs based on NURBS are of polynomial degree $p = 2$ and 3, we will confine ourselves up to $p = 4$ with C^0 and C^{p-1} continuous basis functions.

4.1.1 C^{p-1}-continuity

We first consider the C^{p-1} continuous case as this is the default case for knot vector with non-repeated internal knots. For B-spline functions with $p = 0$ and $p = 1$, we have the same representation as for standard piecewise constant and linear finite element functions, respectively. Quadratic B-spline basis functions, however, differ from their FEA counterparts.

They are each identical but shifted related to each other, whereas the shape of a quadratic finite element function depends on whether it corresponds to an internal node or an end node. This "homogeneous" pattern continues for the B-splines with higher-degrees.
We are interested to give an explicit representation for uniform B-spline basis functions defined on a knot vector E_k at any given level k, where $k = 1, 2, 3, ...$, with spacing $h\ (= 1/n)$, where n is the total number of knot spans. We shall use the notation $B^{p,r}_{k,i}$ for B-splines, where superscripts represent the polynomial degree and the regularity of basis functions, respectively, and the subscripts represent the level and the number of basis function, respectively. We start with level $k = 1$ with only one element. Using the definition from (2.1), at level 1, the B-spline basis functions of degree $p = 2$ on the knot vector $E_1 = \{0, 0, 0, 1, 1, 1\}$ are defined as follows

$$\begin{aligned} B^{2,p-1}_{1,1} &= (1-x)^2, \quad 0 \le x \le 1, \\ B^{2,p-1}_{1,2} &= 2x(1-x), \quad 0 \le x \le 1, \\ B^{2,p-1}_{1,3} &= x^2, \quad 0 \le x \le 1. \end{aligned} \tag{4.1}$$

The mesh refinement takes place by inserting the knots. We consider uniform refinement of E_1, i.e. inserting knots at the mid point of the knot values. At the next level $k = 2$, the basis functions on refined knot vector $E_2 = \{0, 0, 0, \frac{1}{2}, 1, 1, 1\}$ are given by

$$\begin{aligned} B^{2,p-1}_{2,1} &= \begin{cases} (1-2x)^2, & 0 \le x < \frac{1}{2}, \\ 0, & \frac{1}{2} \le x \le 1, \end{cases} \\ B^{2,p-1}_{2,2} &= \begin{cases} 2x(2-3x), & 0 \le x < \frac{1}{2}, \\ 2(1-x)^2, & \frac{1}{2} \le x \le 1, \end{cases} \\ B^{2,p-1}_{2,3} &= \begin{cases} 2x^2, & 0 \le x < \frac{1}{2}, \\ -2 + 8x - 6x^2, & \frac{1}{2} \le x \le 1, \end{cases} \\ B^{2,p-1}_{2,4} &= \begin{cases} 0, & 0 \le x < \frac{1}{2}, \\ (1-2x)^2, & \frac{1}{2} \le x \le 1. \end{cases} \end{aligned} \tag{4.2}$$

Further refinements take place in a similar way, i.e., starting with E_1, a single knot span, in the knot span E_k we will thus have 2^{k-1} knot spans. The explicit representation of B-splines at level k, where $k \ge 3$, is given by

$$B^{2,p-1}_{k,1} = \frac{1}{h^2}(h-x)^2, \quad 0 \le x < h, \quad h \le 1,$$

$$B^{2,p-1}_{k,2} = \left.\begin{cases} \dfrac{1}{2h^2}x(4h-3x), & 0 \le x < h, \\ \dfrac{1}{2h^2}(2h-x)^2, & h \le x < 2h, \end{cases}\right| \text{ for } h \le \frac{1}{2},$$

$$
\begin{aligned}
B_{k,3+i}^{2,p-1} &= \begin{cases} \dfrac{1}{2h^2}(x-ih)^2, & ih \le x < (i+1)h, \\ \dfrac{-3}{2} + \dfrac{3}{h}(x-ih) - \dfrac{1}{h^2}(x-ih)^2, & (i+1)h \le x < (i+2)h, \\ \dfrac{1}{2h^2}(3h-(x-ih))^2, & (i+2)h \le x < (i+3)h, \end{cases} \\
&\qquad \text{where} \quad i = 0, 1, 2, 3, ..., (1/h) - 3, \text{ and } h \le 1/4. \\
B_{k,n+p-1}^{2,p-1} &= \left.\begin{cases} \dfrac{1}{2h^2}(-1+2h+x)^2, & 1-2h \le x < 1-h, \\ \dfrac{-1}{2h^2}(3-4h+2(2h-3)x+3x^2), & 1-h \le x \le 1, \end{cases}\right| \text{ for } h \le \frac{1}{2}, \\
B_{k,n+p}^{2,p-1} &= \frac{1}{h^2}(1-h-x)^2, \quad 1-h \le x \le 1, \quad h \le 1.
\end{aligned}
\tag{4.3}
$$

For higher degree polynomials, we can define the explicit representation in a similar way. Again using the definition (2.1) of B-splines, for $p = 3$, at first level $k = 1$ the basis functions with C^{p-1}-continuity are given as follows

$$
\begin{aligned}
B_{1,1}^{3,p-1} &= (1-x)^3, \quad 0 \le x \le 1, \\
B_{1,2}^{3,p-1} &= 3x(1-x)^2, \quad 0 \le x \le 1, \\
B_{1,3}^{3,p-1} &= 3x^2(1-x), \quad 0 \le x \le 1, \\
B_{1,4}^{3,p-1} &= x^3, \quad 0 \le x \le 1.
\end{aligned}
\tag{4.4}
$$

At level 2, we have the following basis functions

$$
\begin{aligned}
B_{2,1}^{3,p-1} &= \begin{cases} (1-2x)^3, & 0 \le x < \frac{1}{2}, \\ 0, & \frac{1}{2} \le x \le 1, \end{cases} \\
B_{2,2}^{3,p-1} &= \begin{cases} 2x(3-9x+7x^2), & 0 \le x < \frac{1}{2}, \\ 2(1-x)^3, & \frac{1}{2} \le x \le 1, \end{cases} \\
B_{2,3}^{3,p-1} &= \begin{cases} 2x^2(3-4x), & 0 \le x < \frac{1}{2}, \\ 2(-1+x)^2(-1+4x), & \frac{1}{2} \le x \le 1, \end{cases} \\
B_{2,4}^{3,p-1} &= \begin{cases} 2x^3, & 0 \le x < \frac{1}{2}, \\ 2-12x+24x^2-14x^3, & \frac{1}{2} \le x \le 1, \end{cases} \\
B_{2,5}^{3,p-1} &= \begin{cases} 0, & 0 \le x < \frac{1}{2}, \\ (-1+2x)^3, & \frac{1}{2} \le x \le 1. \end{cases}
\end{aligned}
\tag{4.5}
$$

For all other levels k, where $k \ge 3$, the basis functions are defined below

$$
B_{k,1}^{3,p-1} = \frac{1}{h^3}(h-x)^3, \quad 0 \le x < h, \quad h \le 1,
$$

$$B_{k,2}^{3,p-1} = \left\{ \begin{array}{ll} \frac{x}{h}\left(3 - \frac{9}{2}\frac{x}{h} + \frac{7}{4}\frac{x^2}{h^2}\right), & 0 \le x < h, \\ \frac{1}{4h^3}(-2h + x)^3, & h \le x < 2h, \end{array} \right| \text{ for } h \le \frac{1}{2},$$

$$B_{k,3}^{3,p-1} = \left\{ \begin{array}{ll} \frac{1}{6}\frac{x^2}{h^2}\left(9 - \frac{11}{2}\frac{x}{h}\right), & 0 \le x < h, \\ \frac{-3}{2} + \frac{9}{2}\frac{x}{h} - 3\frac{x^2}{h^2} + \frac{7}{4}\frac{x^3}{h^3}, & h \le x < 2h, \\ \frac{1}{6h^3}(-3h + x)^3, & 2h \le x < 3h, \end{array} \right| \text{ for } h \le \frac{1}{4},$$

$$B_{k,4+i}^{3,p-1} = \begin{cases} \frac{1}{6h^3}(x - ih)^3, & ih \le x < (i+1)h, \\ \frac{2}{3} - \frac{2}{h}(x - ih) + \frac{1}{2h^2}(x - ih)^2 - \frac{1}{2h^3}(x - ih)^3, & (i+1) \le x < (i+2)h, \\ \frac{-22}{3} + \frac{10}{h}(x - ih) - \frac{4}{h^2}(x - ih)^2 + \frac{1}{2h^3}(x - ih)^3, & (i+2)h \le x < (i+3)h, \\ \frac{32}{h}\left(1 - \frac{(x - ih)}{4h}\right)^3, & (i+3)h \le x < (i+4)h, \end{cases}$$

$$\text{where} \quad i = 0, 1, 2, 3, ..., (1/h) - 4, \text{ and } h \le \frac{1}{4},$$

$$B_{k,n+p-2}^{3,p-1} = \begin{cases} \frac{1}{6h^3}(-3h + (1-x))^3, & 1 - 3h \le x < 1 - 2h, \\ \frac{-3}{2} + \frac{9}{2}\frac{(1-x)}{h} - 3\frac{(1-x)^2}{h^2} + \frac{7}{4}\frac{(1-x)^3}{h^3}, & 1 - 2h \le x < 1 - h, \\ \frac{1}{6}\frac{(1-x)^2}{h^2}\left(9 - \frac{11}{2}\frac{(1-x)}{h}\right), & 1 - h \le x \le 1, \end{cases}$$

$$\text{for } h \le \frac{1}{4},$$

$$B_{k,n+p-1}^{3,p-1} = \left\{ \begin{array}{ll} \frac{1}{4h^3}(-2h + (1-x))^3, & 1 - 2h \le x < 1 - h, \\ \frac{(1-x)}{h}\left(3 - \frac{9}{2}\frac{(1-x)}{h} + \frac{7}{4}\frac{(1-x)^2}{h^2}\right), & 1 - h \le x < 1, \end{array} \right| \text{ for } h \le \frac{1}{2},$$

$$B_{k,n+p}^{3,p-1} = \frac{1}{h^3}(h - (1-x))^3, \quad 1 - h \le x \le 1, \quad h \le 1. \tag{4.6}$$

Finally, we give the explicit representation of basis functions for $p = 4$ with C^{p-1}-continuity. At level 1, with knot E_1, the B-spline basis functions of degree $p = 4$ are given by

$$\begin{aligned} B_{1,1}^{4,p-1} &= (1-x)^4, \quad 0 \le x \le 1, \\ B_{1,2}^{4,p-1} &= 4x(1-x)^3, \quad 0 \le x \le 1, \\ B_{1,3}^{4,p-1} &= 6x^2(1-x)^2, \quad 0 \le x \le 1, \end{aligned}$$

$$\begin{aligned} B_{1,4}^{4,p-1} &= 4x^3(1-x), \quad 0 \le x \le 1, \\ B_{1,5}^{4,p-1} &= x^4, \quad 0 \le x \le 1. \end{aligned} \tag{4.7}$$

The B-splines on second level $k = 2$ with knot E_2 are defined as follows

$$\begin{aligned} B_{2,1}^{4,p-1} &= \begin{cases} (1-2x)^4, & 0 \le x < \frac{1}{2}, \\ 0, & \frac{1}{2} \le x \le 1, \end{cases} \\ B_{2,2}^{4,p-1} &= \begin{cases} 2x(4-18x+28x^2-15x^3), & 0 \le x < \frac{1}{2}, \\ 2(1-x)^4, & \frac{1}{2} \le x \le 1, \end{cases} \\ B_{2,3}^{4,p-1} &= \begin{cases} 2x^2(6-16x+11x^2), & 0 \le x < \frac{1}{2}, \\ 2(1-x)^3(-1+5x), & \frac{1}{2} \le x \le 1, \end{cases} \\ B_{2,4}^{4,p-1} &= \begin{cases} 2x^3(4-5x), & 0 \le x < \frac{1}{2}, \\ 2(1-x)^2(1-6x+11x^2), & \frac{1}{2} \le x \le 1, \end{cases} \\ B_{2,5}^{4,p-1} &= \begin{cases} 2x^4, & 0 \le x < \frac{1}{2}, \\ -2+16x-48x^2+64x^3-30x^4, & \frac{1}{2} \le x \le 1, \end{cases} \\ B_{2,6}^{4,p-1} &= \begin{cases} 0, & 0 \le x < \frac{1}{2}, \\ (1-2x)^4, & \frac{1}{2} \le x \le 1. \end{cases} \end{aligned} \tag{4.8}$$

At all other levels k, where $k \ge 3$, the basis functions of degree $p = 4$ with C^{p-1}-continuity are given by

$$B_{k,1}^{4,p-1} = \frac{1}{h^4}(h-x)^4, \quad 0 \le x < h, \quad h \le 1,$$

$$B_{k,2}^{4,p-1} = \left\{ \begin{array}{ll} \dfrac{-4x}{h}\left(-1+\dfrac{9}{4}\dfrac{x}{h}-\dfrac{7}{4}\dfrac{x^2}{h^2}+\dfrac{15}{32}\dfrac{x^3}{h^3}\right), & 0 \le x < h, \\ \dfrac{1}{8h^4}(2h-x)^4, & h \le x < 2h, \end{array} \right| \text{ for } h \le \frac{1}{2},$$

$$B_{k,3}^{4,p-1} = \left\{ \begin{array}{ll} \dfrac{1}{9}\dfrac{x^2}{h^2}\left(27-33\dfrac{x}{h}+\dfrac{85}{8}\dfrac{x^2}{h^2}\right), & 0 \le x < h, \\ \dfrac{-3}{2}+6\dfrac{x}{h}-6\dfrac{x^2}{h^2}+\dfrac{7}{3}\dfrac{x^3}{h^3}-\dfrac{23}{72}\dfrac{x^4}{h^4}, & h \le x < 2h, \\ \dfrac{1}{18h^4}(3h-x)^4, & 2h \le x < 3h, \end{array} \right| \text{ for } h \le \frac{1}{4},$$

$$B_{k,4}^{4,p-1} = \left\{ \begin{array}{ll} \dfrac{2}{3}\dfrac{x^3}{h^3}-\dfrac{25}{72}\dfrac{x^4}{h^4}, & 0 \le x < h, \\ \dfrac{2}{3}-\dfrac{8}{3}\dfrac{x}{h}+4\dfrac{x^2}{h^2}-2\dfrac{x^3}{h^3}+\dfrac{23}{72}\dfrac{x^4}{h^4}, & h \le x < 2h, \\ \dfrac{-22}{3}+\dfrac{40}{3}\dfrac{x}{h}-8\dfrac{x^2}{h^2}+2\dfrac{x^3}{h^3}-\dfrac{13}{72}\dfrac{x^4}{h^4}, & 2h \le x < 3h, \\ \dfrac{1}{24h^4}(4h-x)^4, & 3h \le x < 4h, \end{array} \right| \text{ for } h \le \frac{1}{4}.$$

$$B^{4,p-1}_{k,5+i} = \begin{cases} \frac{1}{24h^4}(x-ih)^4, & ih \le x < (i+1)h, \\ \frac{1}{24}\left(-5+\frac{20}{h}(x-ih)-\frac{30}{h^2}(x-ih)^2+\frac{20}{h^3}(x-ih)^3-\frac{4}{h^4}(x-ih)^4\right), & (i+1)h \le x < (i+2)h, \\ \frac{155}{24}-\frac{25}{2h}(x-ih)+\frac{35}{4h^2}(x-ih)^2-\frac{5}{2h^3}(x-ih)^3-\frac{1}{4h^4}(x-ih)^4, & (i+2)h \le x < (i+3)h, \\ \frac{-655}{24}+\frac{65}{2h}(x-ih)-\frac{55}{4h^2}(x-ih)^2+\frac{5}{2h^3}(x-ih)^3-\frac{1}{6h^4}(x-ih)^4, & (i+3)h \le x < (i+4)h, \\ \frac{1}{24h^4}(5h-(x-ih))^4, & (i+4)h \le x < (i+5)h, \end{cases}$$

$$\text{where} \quad i = 0, 1, 2, 3, ..., (1/h) - 5, \text{ and } h \le \frac{1}{4}.$$

$$B^{4,p-1}_{k,n+p-3} = \begin{cases} \frac{1}{24h^4}\left(4h-(1-x)\right)^4, & 1-4h \le x < 1-3h, \\ \frac{-22}{3}+\frac{40}{3}\frac{(1-x)}{h}-8\frac{(1-x)^2}{h^2}+2\frac{(1-x)^3}{h^3}-\frac{13}{72}\frac{(1-x)^4}{h^4}, & 1-3h \le x < 1-2h, \\ \frac{2}{3}-\frac{8}{3}\frac{(1-x)}{h}+4\frac{(1-x)^2}{h^2}-2\frac{(1-x)^3}{h^3}+\frac{23}{72}\frac{(1-x)^4}{h^4}, & 1-2h \le x < 1-h, \\ \frac{2}{3}\frac{(1-x)^3}{h^3}-\frac{25}{72}\frac{(1-x)^4}{h^4}, & 1-h \le x \le 1, \end{cases}$$

$$\text{for } h \le \frac{1}{4},$$

$$B^{4,p-1}_{k,n+p-2} = \begin{cases} \frac{1}{18h^4}(3h-(1-x))^4, & 1-3h \le x < 1-2h, \\ \frac{-3}{2}+6\frac{(1-x)}{h}-6\frac{(1-x)^2}{h^2}+\frac{7}{3}\frac{(1-x)^3}{h^3}-\frac{23}{72}\frac{(1-x)^4}{h^4}, & 1-2h \le x < 1-h, \\ \frac{1}{9}\frac{(1-x)^2}{h^2}\left(27-33\frac{(1-x)}{h}+\frac{85}{8}\frac{(1-x)^2}{h^2}\right), & 1-h \le x \le 1, \end{cases}$$

$$\text{for } h \le \frac{1}{4},$$

$$B^{4,p-1}_{k,n+p-1} = \begin{cases} \frac{1}{8h^4}(2h-(1-x))^4, & 1-2h \le x < 1-h, \\ \frac{-4(1-x)}{h}\left(-1+\frac{9}{4}\frac{(1-x)}{h}-\frac{7}{4}\frac{(1-x)^2}{h^2}+\frac{15}{32}\frac{(1-x)^3}{h^3}\right), & 1-h \le x \le 1, \end{cases}$$

$$\text{for } h \le \frac{1}{2},$$

$$B^{4,p-1}_{k,n+p} = \frac{1}{h^4}(h-(1-x))^4, \quad 0 \le x \le, 1 \quad h \le 1. \tag{4.9}$$

4.1.2 C^0-continuity

To reduce the continuity of the basis functions across element boundaries, the knot values are repeated upto a desired level. By repeating the internal knots k times we get the C^{p-k} continuous basis functions. In the previous section we have given the explicit representation for C^{p-1} continuity, which is the highest continuity for polynomial degree p. We now consider another extreme case, the lowest continuity, i.e. C^0 continuous basis functions. At first level $k = 1$ the C^0 continuous B-spline basis functions of degree $p = 2, 3, 4$ on a knot $E_1 = \{0, 0, 0, 1, 1, 1\}$ are same as those of C^{p-1} continuous B-spline basis functions of same degree, see (4.1), (4.4), and (4.1.1) respectively.
The explicit representation for C^0 continuous B-spline basis functions of degree $p = 2$ at level k, where $k \geq 2$ is given by

$$
\begin{aligned}
B^{2,0}_{k,1} &= \frac{1}{h^2}(h-x)^2, \quad 0 \leq x < h, \\
B^{2,0}_{k,2+2i} &= \frac{-2}{h^2}(x-ih)(h+(x-ih)), \quad (i-1)h \leq x < ih, \\
&\qquad \text{where } i = 0, 1, 2, 3, ..., (1/h) - 1, \\
B^{2,0}_{k,3+2i} &= \begin{cases} \dfrac{1}{h^2}(h+(x-ih))^2, & (i-1)h \leq x < ih, \\ \dfrac{1}{h^2}(-h+(x-ih))^2, & ih \leq x < (i+1)h, \end{cases} \\
&\qquad \text{where } i = 0, 1, 2, 3, ..., ((1/h) - 2), \\
B^{2,0}_{k,np+1} &= \frac{1}{h^2}(1-h-x)^2, \quad 1-h \leq x \leq 1.
\end{aligned}
\tag{4.10}
$$

For $p = 3$, the explicit representation for B-spline basis functions with C^0-continuity, at level k, where $k \geq 2$ is given by

$$
\begin{aligned}
B^{3,0}_{k,1} &= \frac{1}{h^3}(h-x)^3, \quad 0 \leq x < h, \\
B^{3,0}_{k,2+3i} &= \frac{3}{h}\left(-1 + \frac{1}{h}(x-ih)\right)^2, \quad ih \leq x < (i+1)h, \\
&\qquad \text{where } i = 0, 1, 2, 3, ..., (1/h) - 1, \\
B^{3,0}_{k,3+3i} &= \frac{3}{h^2}(x-ih)^2\left(1 - \frac{1}{h}(x-ih)\right), \quad ih \leq x < (i+1)h, \\
&\qquad \text{where } i = 0, 1, 2, 3, ..., (1/h) - 1, \\
B^{3,0}_{k,4+3i} &= \begin{cases} \dfrac{1}{h^3}(x-ih)^3, & ih \leq x < (i+1)h, \\ 8\left(1 - \dfrac{1}{2h}(x-ih)\right)^3, & (i+1)h \leq x < (i+2)h, \end{cases} \\
&\qquad \text{where } i = 0, 1, 2, 3, ..., ((1/h) - 2), \\
B^{3,0}_{k,np+1} &= \frac{1}{h^3}(1-h-x)^3, \quad 1-h \leq x \leq 1.
\end{aligned}
\tag{4.11}
$$

Finally, the explicit representation for C^0 continuous basis functions of degree $p = 4$ at level k, where $k \geq 2$ is given below

$$
\begin{aligned}
B^{4,0}_{k,1} &= \frac{1}{h^4}(h-x)^4, \quad 0 \leq x < h, \\
B^{4,0}_{k,2+4i} &= \frac{4}{h}(x-ih)\left(1-\frac{(x-ih)}{h}\right)^3, \quad ih \leq x < (i+1)h, \\
&\qquad \text{where } i = 0,1,2,3,...,(1/h)-1, \\
B^{4,0}_{k,3+4i} &= \frac{6}{h^2}(x-ih)^2\left(1-\frac{(x-ih)}{h}\right)^2, \quad ih \leq x < (i+1)h, \\
&\qquad \text{where } i = 0,1,2,3,...,(1/h)-1, \\
B^{4,0}_{k,4+4i} &= \frac{4}{h^3}(x-ih)^3\left(1-\frac{(x-ih)}{h}\right), \quad ih \leq x < (i+1)h, \\
&\qquad \text{where } i = 0,1,2,3,...,(1/h)-1, \\
B^{4,0}_{k,5+4i} &= \begin{cases} \dfrac{1}{h^4}(x-ih)^4, & ih \leq x < (i+1)h, \\ 16\left(1-\dfrac{1}{2h}(x-ih)\right)^4, & (i+1)h \leq x < (i+2)h, \end{cases} \\
&\qquad \text{where } i = 0,1,2,3,...,((1/h)-2), \\
B^{4,0}_{k,np+1} &= \frac{1}{h^4}(1-h-x)^4, \quad 1-h \leq x \leq 1.
\end{aligned}
\tag{4.12}
$$

4.2 Multilevel Representation of B-splines and NURBS

4.2.1 Multilevel B-splines

In this section, we present the multilevel structure of B-splines and NURBS spaces. This will be used in the construction of corresponding hierarchical spaces (i.e. splitting the fine space into coarse space and its hierarchical complement) in Section 6.1. For a two level setting, let $\mathcal{B}^{p,r}_{k-1}$ and $\mathcal{B}^{p,r}_{k}$ denote the B-spline spaces at coarse and fine level, respectively. Let $\{B^{p,r}_{k-1,i}, i = 1,2,...,n_{k-1}\}$ and $\{B^{p,r}_{k,i}, i = 1,2,...,n_k\}$ be the set of basis functions for coarse and fine space, respectively, i.e.

$$\mathcal{B}^{p,r}_{k-1} = \text{span}\{B^{p,r}_{k-1,1}, B^{p,r}_{k-1,2}, B^{p,r}_{k-1,3}, ..., B^{p,r}_{k-1,n_{k-1}}\},$$

and

$$\mathcal{B}^{p,r}_{k} = \text{span}\{B^{p,r}_{k,1}, B^{p,r}_{k,2}, B^{p,r}_{k,3}, ..., B^{p,r}_{k,n_k}\}.$$

The following result expresses coarse basis functions as the linear combination of fine basis functions.

Proposition 4.1. *Each coarse basis function $B^{p,r}_{k-1,i}, i = 1, 2, ..., n_{k-1}$, can be represented as the linear combination of the fine basis functions $\{B^{p,r}_{k,i}, i = 1, 2, ..., n_k\}$ by the following relation*

$$\mathcal{B}^{p,r}_{k-1} = G^{p,r}_k \mathcal{B}^{p,r}_k, \quad \text{i.e.,} \quad B^{p,r}_{k-1,i} = \sum_{j=1}^{n_k} g_{ij} B^{p,r}_{k,j}, \tag{4.13}$$

where $G^{p,r}_k = (g_{ij})_{n_{k-1} \times n_k}$, is called the restriction operator from a given fine level to the next coarse level for B-spline basis functions.

In the following we explain the formation of transfer operator $G^{p,r}_k$ at different levels of mesh and with increasing polynomial degree with both the extreme cases of C^{p-1} and C^0-continuity.

C^{p-1}-continuity

The B-spline basis functions $B^{2,p-1}_{1,i}, i = 1, 2, 3$, and $B^{2,p-1}_{2,i}, i = 1, 2, 3, 4$, of degree $p = 2$ on knots $E_1 = \{0, 0, 0, 1, 1, 1\}$ and $E_2 = \{0, 0, 0, \frac{1}{2}, 1, 1, 1\}$, respectively, are defined in section 4.1.1. Clearly, the total number of coarse and fine basis functions are three ($n_{k-1} = 3$) and four ($n_k = 4$), respectively. The matrix $G^{2,p-1}_2 = (g_{ij})_{3\times 4}$ is given by the following representation of coarse basis functions as the linear combination of fine basis functions.

$$\boxed{\begin{aligned} B^{2,p-1}_{1,1} &= g_{11} B^{2,p-1}_{2,1} + g_{12} B^{2,p-1}_{2,2} + g_{13} B^{2,p-1}_{2,3} + g_{14} B^{2,p-1}_{2,4}, \\ B^{2,p-1}_{1,1} &= g_{21} B^{2,p-1}_{2,1} + g_{22} B^{2,p-1}_{2,2} + g_{23} B^{2,p-1}_{2,3} + g_{24} B^{2,p-1}_{2,4}, \\ B^{2,p-1}_{1,1} &= g_{31} B^{2,p-1}_{2,1} + g_{32} B^{2,p-1}_{2,2} + g_{33} B^{2,p-1}_{2,3} + g_{34} B^{2,p-1}_{2,4}. \end{aligned}}$$

Equivalently, it can be written as

$$\mathcal{B}^{2,p-1}_1 = G^{2,p-1}_2 \mathcal{B}^{2,p-1}_2,$$

where

$$\mathcal{B}^{2,p-1}_1 = \begin{bmatrix} B^{2,p-1}_{1,1} \\ B^{2,p-1}_{1,2} \\ B^{2,p-1}_{1,3} \end{bmatrix}, G^{2,p-1}_2 = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \end{bmatrix}, \mathcal{B}^{2,p-1}_2 = \begin{bmatrix} B^{2,p-1}_{2,1} \\ B^{2,p-1}_{2,2} \\ B^{2,p-1}_{2,3} \\ B^{2,p-1}_{2,4} \end{bmatrix}.$$

For the above set of basis functions, $G^{2,p-1}_2$ is given by

$$G^{2,p-1}_2 = \frac{1}{4} \begin{bmatrix} 4 & 2 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 2 & 4 \end{bmatrix}. \tag{4.14a}$$

Similarly, the coarse basis functions for $B^{2,p-1}_{2,i}, i = 1, 2, 3, 4$, at level 2, can be obtained in terms of $B^{2,p-1}_{3,i}, i = 1, 2, ..., 6$, by the following matrix

$$G^{2,p-1}_3 = \frac{1}{4} \begin{bmatrix} 4 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 4 \end{bmatrix}. \tag{4.14b}$$

In a multilevel setting, the representation of each basis function $B_{k,i}^{2,p-1}$ at level k as the linear combination of the basis functions $B_{k+1,i}^{2,p-1}$ at level $k+1$ is given by the the following matrix $G_{k+1}^{2,p-1}$, where $3 \leq k \leq L-1$.

$$G_{k+1}^{2,p-1} = \frac{1}{4}\begin{bmatrix} 4 & 2 & & & & & & & & & & & & & \\ & 2 & 3 & 1 & & & & & & & & & & & \\ & & 1 & 3 & 3 & 1 & & & & & & & & & \\ & & & & 1 & 3 & 3 & 1 & & & & & & & \\ & & & & & & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & & & & & \\ & & & & & & & & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & & & \\ & & & & & & & & & & 1 & 3 & 3 & 1 & \\ & & & & & & & & & & & & 1 & 3 & 3 & 1 \\ & & & & & & & & & & & & & & 2 & 4 \end{bmatrix}. \tag{4.14c}$$

The size of the matrix $G_{k+1}^{2,p-1}$ is $(n_k+2) \times (n_{k+1}+2)$, where n_k and n_{k+1} are the number of total knot spans at level k and $k+1$, respectively.
For higher degree polynomials, the transfer operators can be defined in a similar way. For $p=3$, at level $k=1$ the basis functions $B_{1,i}^{3,p-1}, i=1,2,3,4$, with C^{p-1}-continuity can be represented by the following restriction operator at level $l=2$.

$$G_2^{3,p-1} = \frac{1}{2}\begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}. \tag{4.15a}$$

The transfer operator for level 3 can be written as

$$G_3^{3,p-1} = \frac{1}{16}\begin{bmatrix} 16 & 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 12 & 3 & 0 & 0 & 0 \\ 0 & 0 & 4 & 10 & 4 & 0 & 0 \\ 0 & 0 & 0 & 3 & 12 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8 & 16 \end{bmatrix}. \tag{4.15b}$$

For all levels $k+1$, where $3 \leq k \leq L-1$, we have

$$G_{k+1}^{3,p-1} = \frac{1}{16}\begin{bmatrix} 16 & 8 & & & & & & & & & & & & & \\ & 8 & 12 & 3 & & & & & & & & & & & \\ & & 4 & 11 & 8 & 2 & & & & & & & & & \\ & & & 2 & 8 & 12 & 8 & 2 & & & & & & & \\ & & & & & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & & & & & \\ & & & & & & & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & & & \\ & & & & & & & & 2 & 8 & 12 & 8 & 2 & & \\ & & & & & & & & & & 2 & 8 & 11 & 4 & \\ & & & & & & & & & & & & 3 & 12 & 8 \\ & & & & & & & & & & & & & & 8 & 16 \end{bmatrix}. \tag{4.15c}$$

The size of the matrix $G_{k+1}^{3,p-1}$ is $(n_k + 3) \times (n_{k+1} + 3)$.
Finally, we give the transfer operators for $p = 4$ with C^{p-1}-continuity. For levels 2 and 3 the transfer operators are defined as follows:

$$G_2^{4,p-1} = \frac{1}{2}\begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 \end{bmatrix}, \tag{4.16a}$$

$$G_3^{4,p-1} = \frac{1}{48}\begin{bmatrix} 48 & 24 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 24 & 36 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 30 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 30 & 12 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 36 & 24 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 24 & 48 \end{bmatrix}, \tag{4.16b}$$

respectively. For levels k, , where $3 \leq k \leq L-1$,, the transfer operator is given by the following

$$G_{k+1}^{4,p-1} = \frac{1}{48}\begin{bmatrix} 48 & 24 & & & & & & & & & & & & \\ & 24 & 36 & 9 & & & & & & & & & & \\ & & 12 & 33 & 20 & 4 & & & & & & & & \\ & & & 6 & 25 & 29 & 15 & 3 & & & & & & \\ & & & & 3 & 15 & 30 & 30 & 15 & 3 & & & & \\ & & & & & & .. & .. & .. & .. & & & & \\ & & & & & & .. & .. & .. & .. & & & & \\ & & & & & & 3 & 15 & 30 & 30 & 15 & 3 & & \\ & & & & & & & & 3 & 15 & 29 & 25 & 6 & \\ & & & & & & & & & & 4 & 20 & 33 & 12 \\ & & & & & & & & & & & & 9 & 36 \\ & & & & & & & & & & & & & 24 \end{bmatrix}, \tag{4.16c}$$

where the size of the matrix is $(n_k + 4) \times (n_{k+1} + 4)$.

Remark 4.2. *Since in the span of an internal basis function of degree p at coarse level, there are $p + 2$ full basis functions in the same span at fine level, therefore, any row of $G_{k+1}^{p,p-1}$ can have at most $p + 2$ entries.*

C^0-continuity

In section 4.1.2, we explained the explicit representation of C^0 continuous B-spline basis functions. The corresponding transfer operators are given in this section. The transfer operator $G_2^{2,0}$ for $p = 2$ with C^0-continuity at level 2 is given by

$$G_2^{2,0} = \frac{1}{4}\begin{bmatrix} 4 & 2 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 2 & 4 \end{bmatrix}. \tag{4.17a}$$

The operator $G_{k+1}^{2,0}$, where $2 \leq k \leq L-1$, is given by

$$G_{k+1}^{2,0} = \frac{1}{4}\begin{bmatrix} \boxed{\begin{matrix} 4 & 2 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 2 & 4 \end{matrix}} & & & & \\ & \boxed{\begin{matrix} 4 & 2 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 2 & 4 \end{matrix}} & & & \\ & & \ddots & \ddots & \ddots \\ & & & & \boxed{\begin{matrix} 4 & 2 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 2 & 4 \end{matrix}} \end{bmatrix}, \tag{4.17b}$$

with size $(2n_k+1) \times (2n_{k+1}+1)$. The matrix $G_{k+1}^{2,0}, k \geq 2$, has block structure with blocks $G_2^{2,0}$. The blocks are connected in such a way that if a block ends at ith row and jth column of $G_{k+1}^{2,0}$ then the next block will start at (i,j)th position of $G_{k+1}^{2,0}$ with an overlap of last entry and first entry of the corresponding blocks. Note that, the first entry and the last entry in a block are same.

The transfer operators for $p = 3$ with C^0-continuity for levels 2 and 3 are given by

$$G_2^{3,0} = \frac{1}{8}\begin{bmatrix} 8 & 4 & 2 & 1 & 0 & 0 & 0 \\ 0 & 4 & 4 & 3 & 2 & 0 & 0 \\ 0 & 0 & 2 & 3 & 4 & 4 & 0 \\ 0 & 0 & 0 & 1 & 2 & 4 & 8 \end{bmatrix}, \tag{4.18a}$$

and

$$G_3^{3,0} = \frac{1}{8}\begin{bmatrix} \boxed{\begin{matrix} 8 & 4 & 2 & 1 & 0 & 0 & 0 \\ 0 & 4 & 4 & 3 & 2 & 0 & 0 \\ 0 & 0 & 2 & 3 & 4 & 4 & 0 \\ 0 & 0 & 0 & 1 & 2 & 4 & 8 \end{matrix}} & & \\ & \boxed{\begin{matrix} 8 & 4 & 2 & 1 & 0 & 0 & 0 \\ 0 & 4 & 4 & 3 & 2 & 0 & 0 \\ 0 & 0 & 2 & 3 & 4 & 4 & 0 \\ 0 & 0 & 0 & 1 & 2 & 4 & 8 \end{matrix}} & \\ & & \ddots \end{bmatrix}, \tag{4.18b}$$

respectively. Following the same block structure as in $G_{k+1}^{2,0}$, we can generate $G_{k+1}^{3,0}$, where $3 \leq k \leq L-1$, with size $(3n_k+1) \times (3n_{k+1}+1)$. Finally for $p = 4$, we have the following transfer operators for levels 2 and 3

$$G_2^{4,0} = \frac{1}{16}\begin{bmatrix} 16 & 8 & 4 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 8 & 8 & 6 & 4 & 2 & 0 & 0 & 0 \\ 0 & 0 & 4 & 6 & 6 & 6 & 4 & 0 & 0 \\ 0 & 0 & 0 & 2 & 4 & 6 & 8 & 8 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 4 & 8 & 16 \end{bmatrix}, \tag{4.19a}$$

and

$$G_3^{4,0} = \frac{1}{16}\begin{bmatrix} \boxed{\begin{matrix} 16 & 8 & 4 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 8 & 8 & 6 & 4 & 2 & 0 & 0 & 0 \\ 0 & 0 & 4 & 6 & 6 & 6 & 4 & 0 & 0 \\ 0 & 0 & 0 & 2 & 4 & 6 & 8 & 8 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 4 & 8 & 16 \end{matrix}} & \boxed{\begin{matrix} 8 & 4 & 2 & 1 & 0 & 0 & 0 & 0 \\ 8 & 8 & 6 & 4 & 2 & 0 & 0 & 0 \\ 0 & 4 & 6 & 6 & 6 & 4 & 0 & 0 \\ 0 & 0 & 2 & 4 & 6 & 8 & 8 & 0 \\ 0 & 0 & 0 & 1 & 2 & 4 & 8 & 16 \end{matrix}} \end{bmatrix}, \tag{4.19b}$$

respectively. Similarly, repeating these blocks as in previous cases, we can generate $G_{k+1}^{4,0}$, where $3 \leq k \leq L-1$, with size $(4n_k+1) \times (4n_{k+1}+1)$.

Remark 4.3. *Note that the transfer operators are defined for one dimensional B-splines. For two- and three-dimensions, we take tensor product of these operators.*

4.2.2 Multilevel NURBS

This section presents the procedure for constructing NURBS multilevel spaces in a simplified manner. Since NURBS are generated from B-splines, its natural to construct NURBS transfer operators from B-splines transfer operators. For a two level setting, let $\mathcal{N}_{k-1}^{p,r}$ and $\mathcal{N}_k^{p,r}$ denote the NURBS spaces at coarse and fine level, respectively. Let $\{N_{k-1,i}^{p,r}, i = 1,2,...,n_{k-1}\}$ and $\{N_{k,i}^{p,r}, i = 1,2,...,n_k\}$ be the set of basis functions for coarse and fine space, respectively, i.e.

$$\mathcal{N}_{k-1}^{p,r} = \text{span}\{N_{k-1,1}^{p,r}, N_{k-1,2}^{p,r}, N_{k-1,3}^{p,r}, ..., N_{k-1,n_{k-1}}^{p,r}\},$$

and

$$\mathcal{N}_k^{p,r} = \text{span}\{N_{k,1}^{p,r}, N_{k,2}^{p,r}, N_{k,3}^{p,r}, ..., N_{k,n_k}^{p,r}\}.$$

Note that, a relation similar to Proposition 4.1 also holds for NURBS basis functions, i.e., we have

$$\mathcal{N}_{k-1}^{p,r} = R_k^{p,r}\mathcal{N}_k^{p,r}, \quad \text{i.e.,} \quad N_{k-1,i}^{p,r} = \sum_{j=1}^{n_k} r_{ij} N_{k,j}^{p,r}, \quad \forall i = 1,2,3,...,n_{k-1}, \tag{4.20}$$

where $R_k^{p,r} = (r_{ij})_{n_{k-1} \times n_k}$, is restriction operator with respect to NURBS basis functions. As NURBS are formed from B-splines and 'weights', $R_k^{p,r}$ can be obtained from $G_k^{p,r}$ and 'weights'. Using the definition of NURBS and (4.20), we have

$$\frac{w_i^{k-1} B_{k-1,i}^{p,r}}{\sum_{i'=1}^{n_{k-1}} w_{i'}^{k-1} B_{k-1,i'}^{p,r}} = \sum_{j=1}^{n_k} r_{ij} \frac{w_j^k B_{k,j}^{p,r}}{\sum_{j'=1}^{n_k} w_{j'}^k B_{k,j'}^{p,r}}, \quad \forall i = 1,2,3,...,n_{k-1}, \tag{4.21}$$

where $w_i^{k-1}, i = 1, 2, 3, ..., n_{k-1}$, and $w_j^k, j = 1, 2, 3, ..., n_k$, are the weights for coarse space and fine space, respectively. Note that the weight function $\sum_{i=1}^{n} w_i B_i$ does not change its value with respect to refinements, i.e., we have

$$\sum_{i=1}^{n_{k-1}} w_i^{k-1} B_{k-1,i}^{p,r} = \sum_{j=1}^{n_k} w_j^k B_{k,j}^{p,r}, \tag{4.22}$$

which is an important result from the refinement point of view. Now using (4.22), from (4.21) we get

$$w_i^{k-1} B_{k-1,i}^{p,r} = \sum_{j=1}^{n_k} r_{ij} w_j^k B_{k,j}^{p,r},$$

and thus

$$B_{k-1,i}^{p,r} = \sum_{j=1}^{n_k} \frac{r_{ij} w_j^k}{w_i^{k-1}} B_{k,j}^{p,r}. \tag{4.23}$$

Comparing the coefficients of $B_{k,j}^{p,r}$ in (4.13) and (4.23), we get

$$\frac{r_{ij} w_j^k}{w_i^{k-1}} = g_{ij} \Longrightarrow r_{ij} = \frac{w_i^{k-1} g_{ij}}{w_j^k}. \tag{4.24}$$

This can be equivalently written as

$$R_k^{p,r} = W_I^{k-1} G_k^{p,r} \left(W_I^k\right)^{-1}, \tag{4.25}$$

where W_I^{k-1} and W_I^k are the diagonal matrices corresponding to weights at the coarse level and fine level, respectively, and defined as follows

$$W_I^{k-1} = \begin{bmatrix} w_1^{k-1} & & & & & & \\ & w_2^{k-1} & & & & & \\ & & .. & & & & \\ & & & .. & & & \\ & & & & w_{n_{k-1}-1}^{k-1} & & \\ & & & & & w_{n_{k-1}}^{k-1} \end{bmatrix},$$

$$W_I^k = \begin{bmatrix} w_1^k & & & & & & \\ & w_2^k & & & & & \\ & & .. & & & & \\ & & & .. & & & \\ & & & & w_{n_k-1}^k & & \\ & & & & & w_{n_k}^k \end{bmatrix}.$$

The equation (4.25) gives us the NURBS operators using B-splines transfer operators and weights at coarse and fine levels. From (4.22) we can also obtain the procedure to refine the

weights as follows. We have

$$\sum_{i=1}^{n_{k-1}} w_i^{k-1} B_{k-1,i}^{p,r} = \sum_{j=1}^{n_k} w_j^k B_{k,j}^{p,r},$$

which implies

$$\sum_{i=1}^{n_{k-1}} w_i^{k-1} \sum_{j=1}^{n_k} g_{ij} B_{k,j}^{p,r} = \sum_{j=1}^{n_k} w_j^k B_{k,j}^{p,r}.$$

Comparing the coefficients of $B_{k,j}^{p,r}$ from both the sides, we get

$$w_j^k = \sum_{i=1}^{n_{k-1}} w_i^{k-1} g_{ij} \quad \text{for } j = 1, 2, ..., n_k. \tag{4.26}$$

Equivalently, this can be written in matrix form as follows

$$W^k = (G_k^{p,r})^T W^{k-1}, \tag{4.27}$$

where

$$W^k = \begin{bmatrix} w_1^k \\ w_2^k \\ \vdots \\ \vdots \\ w_{n_k-1}^k \\ w_{n_k}^k \end{bmatrix}, W^{k-1} = \begin{bmatrix} w_1^{k-1} \\ w_2^{k-1} \\ \vdots \\ \vdots \\ w_{n_{k-1}-1}^{k-1} \\ w_{n_{k-1}}^{k-1} \end{bmatrix}.$$

Using above, now we can write the NURBS operators in terms of B-spline operator and weights only at coarse level. From (4.24), we get

$$r_{ij} = \frac{g_{ij} w_i^{k-1}}{\sum_{i=1}^{n_{k-1}} w_i^{k-1} g_{ij}}. \tag{4.28}$$

In matrix form this can be written as

$$R_k^{p,r} = W_I^{k-1} G_k^{p,r} \Big(\text{diag} \left((G_k^{p,r})^T W^{k-1} \right) \Big)^{-1}. \tag{4.29}$$

Remark 4.4. *The operators $G_k^{p,r}$ and $R_k^{p,r}$ can also be used in constructing restriction operators in multigrid methods, see e.g., [55].*

Chapter 5

Multigrid Methods for Isogeometric Discretizations

Our focus in this chapter is on multigrid methods for solving the linear system of equations arising from the isogeometric discretization of scalar second order elliptic problems in a single patch. We first prove the condition number estimates of the discrete system for the h-refinement, and provide the supporting numerical results for all levels of smoothness (from C^0 to C^{p-1}). These results suggest the expected behavior from the two-(multi-)grid solver. We then prove both the components of the two-grid solver, namely the *approximation property* of the inter-grid transfer operators, and the *smoothing property* of the classical Gauss-Seidel (symmetric as well as non-symmetric) method. Together, these two components establish the h-independence of the two-grid solver. For the multi-grid solver, which uses the two-grid solver recursively, we recall the h-independent convergence estimates from [60].

Following the terminology of standard FEM, we will call the open knot-span as element wherever appropriate. Moreover, as most of the NURBS based designs in engineering use polynomial degree $p = 2$ and 3, throughout this chapter we will confine ourselves up to $p = 4$. Furthermore, throughout this chapter we use the notation $f \preceq g$ (respectively $f \succeq g$) to denote $f \leq cg$ (respectively $f \geq cg$) where the constant c is independent of the mesh parameter h and the inequality arguments, but it may depend on the polynomial degree p. For the sake of completeness of this chapter we here repeat some results from Chapter 3.

5.1 Notations

To deal with the tensor-product structure in d-dimensions, we introduce the dimension index set $\mathcal{D} := \{1, \ldots, d\}$, and the index set for knot vectors $\mathcal{K}_\alpha := \{1, 2, \ldots, k_\alpha, \alpha \in \mathcal{D}\}$. Also, let $\mathcal{N}_\alpha = \{1, 2, \ldots, n_\alpha, \alpha \in \mathcal{D}\}$ be the index set of number of basis functions in each dimension, and $p_\mathcal{D} = \{p_1, \ldots, p_d\}$ and $\mathcal{N}_\mathcal{D} = \{\otimes_{\alpha\in\mathcal{D}}\mathcal{N}_\alpha\}$ be the index set of polynomial degree and number of basis functions, respectively, in all dimensions. Now let $\tilde{\Omega} := (0, 1)^d \subset \mathbb{R}^d$ be an open parametric domain which we will refer as a *patch*. Assume that d open knot

vectors $\Xi_\alpha := \{\xi_{\alpha,\mathcal{K}_\alpha}\}, \alpha \in \mathcal{D}$, are given such that $\xi_{\alpha,1} = 0$ and $\xi_{\alpha,\kappa_\alpha} = 1$ for all $\alpha \in \mathcal{D}$. Associated with Ξ_α, $\alpha \in \mathcal{D}$, we partition the patch $\tilde{\Omega}$ in to a mesh

$$\mathcal{Q}_h := \{Q = \otimes_{\alpha\in\mathcal{D}}(\xi_{\alpha,i_\alpha}, \xi_{\alpha,i_\alpha+1}) | Q \neq \emptyset, p_\alpha + 1 \leq i_\alpha \leq k_\alpha - p_\alpha - 1\},$$

where Q is a d-dimensional open knot-span whose diameter is denoted by h_Q. We consider a family of quasi-uniform meshes $\{\mathcal{Q}_h\}_h$ on $\tilde{\Omega}$, where $h = \max\{h_Q | Q \in \mathcal{Q}_h\}$ denotes the family index, see [14].

Furthermore, let $\mathcal{B}_h$ denote the B-spline space associated with the mesh $\mathcal{Q}_h$. Since we do not consider p-refinements, we will use $\mathcal{B}_h$ to denote the mesh family $\mathcal{Q}_h$ for all polynomial degrees. The functions in $\mathcal{B}_h$ are piecewise polynomials of degree p_d in the d^{th} coordinate. Given two adjacent elements Q_1 and Q_2, by $m_{Q_1Q_2}$ we denote the number of continuous derivatives across their common $(d-1)$-dimensional face $\partial Q_1 \cap \partial Q_2$. In the analysis, we will use the following Sobolev space of order $m \in \mathbb{N}$

$$\begin{aligned}\mathcal{H}^m(\tilde{\Omega}) := \Big\{ & v \in L^2(\tilde{\Omega}) \text{ such that } v|_Q \in H^m(Q), \forall Q \in \mathcal{Q}_h, \text{ and} \\ & \nabla^i(v|_{Q_1}) = \nabla^i(v|_{Q_2}) \text{ on } \partial Q_1 \cap \partial Q_2, \\ & \forall i \in \mathbb{N} \text{ with } 0 \leq i \leq \min\{m_{Q_1Q_2}, m-1\}, \forall Q_1, Q_2 \text{ with } \partial Q_1 \cap \partial Q_2 \neq \emptyset \Big\},\end{aligned} \tag{5.1}$$

where ∇^i has the usual meaning of i^{th}-order partial derivative, and H^m is the usual Sobolev space of order m. The space $\mathcal{H}^m$ is equipped with the following semi-norms and norm

$$|v|^2_{\mathcal{H}^i(\tilde{\Omega})} := \sum_{Q\in\mathcal{Q}_h} |v|^2_{H^i(Q)}, \quad 0 \leq i \leq m, \qquad \|v\|^2_{\mathcal{H}^m(\tilde{\Omega})} := \sum_{i=0}^{m} |v|^2_{\mathcal{H}^i(\tilde{\Omega})}. \tag{5.2}$$

Clearly, for all nested meshes $\mathcal{Q}_{h_k} \subset \mathcal{Q}_{h_{k+1}}$ we have $\mathcal{B}_{h_k} \subset \mathcal{B}_{h_{k+1}}$ for all $k \geq 0$, where h_0 refers to the initial mesh. To a non-empty element $Q = \otimes_{\alpha\in\mathcal{D}}(\xi_{\alpha,i_\alpha}, \xi_{\alpha,i_\alpha+1}) \subset \tilde{\Omega}$ we associate the support extension

$$\tilde{Q} := \otimes_{\alpha\in\mathcal{D}}(\xi_{\alpha,i_\alpha-p_\alpha}, \xi_{\alpha,i_\alpha+p_\alpha+1}) \subset \tilde{\Omega}, \tag{5.3}$$

which is the union of supports of those basis functions whose support intersects Q. The restriction of $\mathcal{H}^m(\tilde{\Omega})$ to the support extension $\tilde{Q}$ is denoted by $\mathcal{H}^m(\tilde{Q})$, and is equipped with the following semi-norms and norm

$$|v|^2_{\mathcal{H}^i(\tilde{Q})} := \sum_{\substack{Q'\in\mathcal{Q}_h \\ Q'\cap\tilde{Q}\neq\emptyset}} |v|^2_{H^i(Q')}, \quad 0 \leq i \leq m, \qquad \|v\|^2_{\mathcal{H}^m(\tilde{Q})} := \sum_{i=0}^{m} |v|^2_{\mathcal{H}^i(\tilde{Q})}. \tag{5.4}$$

The NURBS space on the patch $\tilde{\Omega}$, associated with the mesh $\mathcal{Q}_h$, will be denoted by $\mathcal{R}_h$. When no ambiguity should arise, we will use the notation $\mathcal{P}_h$ to represent the polynomial space of either B-splines or NURBS.

Moreover, let the NURBS geometrical map $\mathbf{F} : \tilde{\Omega} \to \Omega$, which is a parametrization of the physical domain Ω, be given by (2.7) with suitable control points. We assume that $\mathbf{F}$ is invertible, with smooth inverse, on each element $Q \in \mathcal{Q}_h$. Therefore, each element $Q \in \mathcal{Q}_h$ is mapped into an element $K = \mathbf{F}(Q) := \{\mathbf{F}(\xi) | \xi \in Q\} \subset \Omega$, and the support extension $\tilde{Q}$ is mapped into $\tilde{K} = \mathbf{F}(\tilde{Q}) \subset \Omega$. Thereby, in the physical domain Ω we introduce the mesh $\mathcal{T}_h := \{K = \mathbf{F}(Q) | Q \in \mathcal{Q}_h\}$, where h denotes the maximum element size (hereinafter called the mesh-size) in the domain Ω. Note that the notation h is used for parametric domain as well as physical domain, however, it is a different quantity in both the contexts. Wherever needed, by h_K we will denote the element size in the physical domain. On the physical domain Ω, we denote the space of B-splines by $V_{\mathcal{B}_h}$ and the space of NURBS by $V_{\mathcal{R}_h}$, which are defined as

$$V_{\mathcal{B}_h} := \operatorname{span}\big\{\phi^{p_\mathcal{D}}_{\mathcal{N}_\mathcal{D}} = B^{p_\mathcal{D}}_{\mathcal{N}_\mathcal{D}} \circ \mathbf{F}^{-1}\big\}, \tag{5.5}$$

$$V_{\mathcal{R}_h} := \operatorname{span}\big\{\varphi^{p_\mathcal{D}}_{\mathcal{N}_\mathcal{D}} = R^{p_\mathcal{D}}_{\mathcal{N}_\mathcal{D}} \circ \mathbf{F}^{-1}\big\}. \tag{5.6}$$

When no ambiguity should arise, we will collectively denote $V_{\mathcal{B}_h}$ and $V_{\mathcal{R}_h}$ by V_h, and $\phi^{p_\mathcal{D}}_{\mathcal{N}_\mathcal{D}}$ and $\varphi^{p_\mathcal{D}}_{\mathcal{N}_\mathcal{D}}$ by $\psi^{p_\mathcal{D}}_{\mathcal{N}_\mathcal{D}}$, respectively. We will denote the number of elements (open knot-spans with non-zero measure) for a one-dimensional uniform knot vector Ξ by $n_0 \approx 1/h$. Furthermore, let n_h denote the cardinality of the space V_h. Note that for V_h with degree $p_\alpha = p$, for all $\alpha \in \mathcal{D}$, and C^{p-1} continuity, we have $n_h = (n_0 + p)^d \approx h^{-d}$.

Finally, we associate a reference support extension $\hat{Q}$ to $\tilde{Q}$ through a piecewise affine map $\mathbf{G} : \hat{Q} \to \tilde{Q}$ such that each element $Q' \in \tilde{Q}$ is the image of a unit hypercube $\mathbf{G}^{-1}(Q')$, where $\mathbf{G}^{-1}(Q') := \{\mathbf{G}^{-1}(\xi) | \xi \in Q'\}$. For brevity reasons, we omit further details (including the related spaces) related to the map $\mathbf{G}$ and refer the reader to [14].

5.2 Error Estimates

We first recall some results from [14, 94]. By l and m we shall denote integer indices with $0 \leq l \leq m \leq p + 1$.

1. **Approximation property of the spline space $\mathcal{B}_h$:** The following result is analogous to the classical result by Bramble and Hilbert.

   Lemma 5.1. *[14, Lemma 3.1] Given $Q \in \mathcal{Q}_h$, the support extension $\tilde{Q}$ as defined in* (5.3)*, and $v \in \mathcal{H}^m$, there exists an $s \in \mathcal{B}_h$ such that*

   $$|v - s|_{\mathcal{H}^l(\tilde{Q})} \preceq h_Q^{m-l} |v|_{\mathcal{H}^m(\tilde{Q})}. \tag{5.7}$$

2. **Projection operators (quasi-interpolants):** Let $\Pi_{\mathcal{B}_h} : L^2(\tilde{\Omega}) \to \mathcal{B}_h$ be a projection operator on the spline space $\mathcal{B}_h$, which is defined as follows, see [94, Chapter 12]:

   $$\Pi_{\mathcal{B}_h} v := \sum_{i_\alpha \in \mathcal{N}_\mathcal{D}} \tau_{i_\alpha} v B^{p_\mathcal{D}}_{i_\alpha}, \quad \forall v \in L^2(\tilde{\Omega}), \tag{5.8}$$

where τ_{i_α} are dual basis functionals defined as

$$\tau_{i_\alpha} B_{i'_\alpha}^{p_\mathcal{D}} = \begin{cases} 1 & \text{if } i_\alpha = i'_\alpha, \\ 0 & \text{otherwise.} \end{cases}$$

The projection operator $\Pi_{\mathcal{R}_h} : L^2(\tilde{\Omega}) \to \mathcal{R}_h$ on the NURBS space is defined as, see [14],

$$\Pi_{\mathcal{R}_h} v := \frac{\Pi_{\mathcal{B}_h}(wv)}{w}, \quad \forall v \in L^2(\tilde{\Omega}), \tag{5.9}$$

where the weight function w is defined in (2.5). Collectively, the projection operators $\Pi_{\mathcal{B}_h}$ and $\Pi_{\mathcal{R}_h}$ will be denoted by $\Pi_{\mathcal{P}_h}$. Finally, the projection operator $\Pi_{V_h} : L^2(\Omega) \to V_h$ on the physical space is defined as, see [14],

$$\Pi_{V_h} v := (\Pi_{\mathcal{P}_h}(v \circ \mathbf{F})) \circ \mathbf{F}^{-1}, \quad \forall v \in L^2(\Omega). \tag{5.10}$$

Lemma 5.2. *[94, Theorem 12.6] The projection operator $\Pi_{\mathcal{P}_h}$ has the following properties:*

$$\Pi_{\mathcal{P}_h} s = s, \qquad \forall s \in \mathcal{P}_h \qquad (\textbf{\textit{spline preserving}}), \tag{5.11a}$$

$$\|\Pi_{\mathcal{P}_h} v\|_{L^2(\tilde{\Omega})} \preceq \|v\|_{L^2(\tilde{\Omega})}, \quad \forall v \in L^2(\tilde{\Omega}) \qquad (\textbf{\textit{stability}}). \tag{5.11b}$$

3. **Interpolation error estimates:** The following lemmas concern the interpolation error estimates.

Lemma 5.3. *Let the projection operator $\Pi_{\mathcal{B}_h} : L^2(\tilde{\Omega}) \to \mathcal{B}_h$, defined by* (5.8)*, satisfy* (5.11)*. Then the following estimate holds for all $v \in \mathcal{H}^m(\tilde{\Omega})$, see [94, Theorem 12.7] and [14].*

$$|v - \Pi_{\mathcal{B}_h} v|_{\mathcal{H}^l(\tilde{\Omega})} \preceq h^{m-l} |v|_{\mathcal{H}^m(\tilde{\Omega})}. \tag{5.12}$$

For the projection operator $\Pi_{\mathcal{R}_h}$ the following result is valid for all $v \in \mathcal{H}^m(\tilde{\Omega})$, see [14]:

$$|v - \Pi_{\mathcal{R}_h} v|_{\mathcal{H}^l(\tilde{\Omega})} \preceq h^{m-l} \|v\|_{\mathcal{H}^m(\tilde{\Omega})}. \tag{5.13}$$

For the physical domain Ω we have the following result:

Lemma 5.4. *[14, Theorem 3.2] For the projection operator Π_{V_h}, the following estimate holds for all $v \in H^m(\Omega)$.*

$$\sum_{K \in \mathcal{T}_h} |v - \Pi_{V_h} v|^2_{H^l(K)} \preceq \sum_{K \in \mathcal{T}_h} h_K^{2(m-l)} \sum_{i=0}^{m} \|\nabla \mathbf{F}\|^{2(i-m)}_{L^\infty(\mathbf{F}^{-1}(K))} |v|^2_{H^i(K)}. \tag{5.14}$$

Note that the constants in (5.13) and (5.14) depend on the weight function w (and hence on the shape of the parametric domain).

Now assuming sufficient regularity (for the dual problem), a classical convergence analysis and the duality argument (*Aubin-Nitsche's trick*) easily give the following result.

Theorem 5.5. *The solution of the problem* (2.16) *satisfies the following error estimates*

$$|u - u_h|_{H^1(\Omega)} \preceq h^{m-1} \|u\|_{H^m(\Omega)}, \tag{5.15}$$

$$\|u - u_h\|_{L^2(\Omega)} \preceq h \|u - u_h\|_{H^1(\Omega)}. \tag{5.16}$$

5.3 The Discrete System

By approximating u_h and v_h using B-spline (NURBS) basis functions ψ_i, $i = 1, 2, \ldots, n_h$, where $n_h = \mathcal{O}(h^{-d})$, the weak formulation (2.16) is transformed in to a set of linear algebraic equations

$$\bar{A}_h u_h = \bar{f}_h, \tag{5.17}$$

where $\bar{A}_h$ denotes the stiffness matrix obtained from the bilinear form $a(\cdot, \cdot)$, u_h denotes the vector of unknown degrees of freedom, and $\bar{f}_h$ denotes the right hand side (RHS) vector from the known data of the problem. In the following Lemma we show the equivalence of the Euclidean norm and the maximum norm for the B-spline (NURBS) space. In this section, for ease of notations we assume uniform polynomial degree in each dimension, i.e. $p_\alpha = p$ for all $\alpha \in \mathcal{D}$, although the results are easily generalizable for non-uniform order case.

Lemma 5.6. *Let $V_h = \operatorname{span}\{\psi_i, i = 1, \ldots, n_h\}$ be the space of B-spline (NURBS) basis functions. Let $v = \sum_{i=1}^{n_h} v_i \psi_i$, where v_i are arbitrary. Then the following relation holds for all $K \in \mathcal{T}_h$*

$$\|v\|_{L^\infty(K)} \preceq \Big(\sum_{\operatorname{supp}(\{\psi_i\}) \cap K \neq \emptyset} v_i^2 \Big)^{1/2} \preceq \|v\|_{L^\infty(K)}. \tag{5.18}$$

Proof. We only consider the non-trivial case, i.e. there exists some i for which $v_i \neq 0$. For any $K \in \mathcal{T}_h$, there are at most $(p+1)^d$ basis functions with non-zero support. Let $\mathcal{I}_h^K \equiv \otimes_{\alpha \in \mathcal{D}} \{i_{\alpha,1}^K, i_{\alpha,2}^K, \ldots, i_{\alpha,p+1}^K\} \subset \{1, 2, \ldots, n_h\}$ denote the index set for the basis functions which have non-zero support in K. Also, let $\bar{v}_K = \max_{i \in \mathcal{I}_h^K} |v_i|$. Invoking the non-negativity and the partition of unity properties of basis functions, we have

$$\begin{aligned} \|v\|_{L^\infty(K)}^2 &= \big(\sup\{|v(x)| : x \in K\}\big)^2 = \Big(\sup \Big| \sum_{i \in \mathcal{I}_h^K} v_i \psi_i \Big|\Big)^2 \\ &\leq \Big(\sup \sum_{i \in \mathcal{I}_h^K} \psi_i |v_i|\Big)^2 \leq \Big(\bar{v}_K \sup \sum_{i \in \mathcal{I}_h^K} \psi_i\Big)^2 = \bar{v}_K^2 \leq \sum_{i \in \mathcal{I}_h^K} v_i^2. \end{aligned}$$

Furthermore, since $\sum_{i \in \mathcal{I}_h^K} v_i^2 \leq \sum_{i \in \mathcal{I}_h^K} \bar{v}_K^2 = (p+1)^d \bar{v}_K^2$, using the stability of B-spline basis functions [83, 93], we obtain the right hand side inequality with a constant $\gamma_1 = \mathcal{O}(p^{2d} 2^{dp})$. □

Using Sobolev inequalities, see [14], and following the standard FEM approach, see e.g. [5], we obtain the following bounds on the condition number of the matrix $\bar{A}_h$.

Lemma 5.7. *Let the basis $\{\psi_i, i = 1, \ldots, n_h\}$ satisfy* (5.18)*. Then the following relation holds*

$$h^d \preceq \lambda_{\min}(\bar{A}_h), \quad \lambda_{\max}(\bar{A}_h) \preceq h^{d-2}, \quad \kappa(\bar{A}_h) \preceq h^{-2}, \tag{5.19}$$

where $\kappa(\bar{A}_h)$ denotes the spectral condition number of $\bar{A}_h$.

From (5.19), we also note that

$$\|\bar{A}_h\| = \lambda_{\max}(\bar{A}_h) \preceq h^{d-2}, \qquad \|\bar{A}_h{}^{-1}\| = \lambda_{\max}(\bar{A}_h{}^{-1}) = 1/\lambda_{\min}(\bar{A}_h) \preceq h^{-d}, \quad (5.20)$$

where $\|\cdot\|$ denotes the spectral norm.

Table 5.1: $\lambda_{max}, \lambda_{min}$, and $\kappa(\bar{A}_h)$ for $d = 2$. Smoothness from C^0 to C^{p-1}.

	n_0	2	4	8	16	32	64
$p = 2$							
C^0	λ_{max}	2.1726	2.5607	2.6436	2.6612	2.6653	2.6663
	λ_{min}	0.2929	0.2008	0.0726	0.0190	0.0048	0.0012
	$\kappa(A)$	7.4169	12.755	36.405	140.01	555.00	2215.0
C^1	λ_{max}	1.4222	1.4238	1.4896	1.4951	1.4991	1.4997
	λ_{min}	0.3556	0.3556	0.2855	0.0756	0.0192	0.0048
	$\kappa(A)$	4.0000	4.0044	5.2173	19.768	78.142	311.58
$p = 3$							
C^0	λ_{max}	2.1297	2.2415	2.2844	2.2961	2.2992	2.2999
	λ_{min}	0.0284	0.0210	0.0190	0.0085	0.0021	0.0005
	$\kappa(A)$	75.111	106.56	120.34	269.99	1075.4	4297.2
C^1	λ_{max}	0.8962	1.1705	1.1910	1.2078	1.2129	1.2142
	λ_{min}	0.0386	0.0386	0.0386	0.0191	0.0048	0.0012
	$\kappa(A)$	23.234	30.346	30.878	63.200	252.68	1008.4
C^2	λ_{max}	1.0384	1.3698	1.5247	1.5627	1.5720	1.5743
	λ_{min}	0.0336	0.0464	0.0522	0.0547	0.0191	0.0048
	$\kappa(A)$	30.927	29.509	29.192	28.561	82.102	327.22
$p = 4$							
C^0	λ_{max}	2.1002	2.1105	2.1174	2.1195	2.1200	2.1202
	λ_{min}	0.0024	0.0019	0.0018	0.0017	0.0012	0.0003
	$\kappa(A)$	881.41	1099.7	1189.1	1214.8	1761.9	7041.3
C^1	λ_{max}	0.8752	1.0840	1.1452	1.1606	1.1644	1.1654
	λ_{min}	0.0030	0.0030	0.0030	0.0030	0.0021	0.0005
	$\kappa(A)$	293.90	364.01	384.55	389.72	545.08	2177.9
C^2	λ_{max}	0.6780	0.9178	0.9847	1.0059	1.0118	1.0133
	λ_{min}	0.0040	0.0048	0.0051	0.0052	0.0047	0.0012
	$\kappa(A)$	167.95	191.78	193.17	192.21	213.24	842.40
C^3	λ_{max}	0.9369	1.3334	1.7182	1.8111	1.8311	1.8357
	λ_{min}	0.0028	0.0050	0.0072	0.0081	0.0085	0.0048
	$\kappa(A)$	339.92	269.23	240.26	222.54	215.00	381.73

In Table 5.1, we present the extremal eigenvalues and the spectral condition number of $\bar{A}_h$ for $\Omega \subset \mathbb{R}^2$. We consider all levels of smoothness, i.e. minimum C^0 to maximum

C^{p-1} for the polynomial degrees $p = 2, 3, 4$. For h-refinement (knot insertion), we see that the extremal eigenvalues satisfy the theoretical estimates (5.19) for the discrete system of second order elliptic problems, i.e., maximum eigenvalues are constant, and the minimum eigenvalues are of $\mathcal{O}(h^2)$ asymptotically, see e.g. [33]. As mentioned earlier, for reducing the smoothness we insert multiple knots. Note that, due to a high condition number γ_1 of the B-spline basis (see proof of Lemma 5.6), for a given mesh size a higher polynomial degree adversely affects the condition number of the matrix $\bar{A}_h$, see [56].

Before proceeding further, we need to introduce some more notations which are needed for two(multi)-grid analysis. Let $k = 1, \ldots, L$, denote the level of mesh $\mathcal{T}_{h_k}$, and h_k be the associated mesh size. The discrete space of B-spline (NURBS) basis functions at level k is denoted by V_k. We assume that the meshes are nested and that $V_k \subset V_{k+1}$. The mesh-dependent inner product $(\cdot, \cdot)_k$ on V_k is defined by

$$(v, w)_k := h_k^d \sum_{i=1}^{n_k} v_i w_i, \tag{5.21}$$

where v_i and w_i denote the approximation coefficients of functions v and w, respectively, with respect to the basis of V_k. The operator $A_k : V_k \to V_k$ is defined by

$$(A_k v, w)_k = a(v, w), \quad \forall v, w \in V_k. \tag{5.22}$$

Note from (5.21)-(5.22) that $A_k = h_k^{-d} \bar{A}_h$. In terms of the operator A_k, the discrete system (5.17) can be equivalently written as

$$A_k u_k = f_k, \tag{5.23}$$

where $f_k \in V_k$ satisfies

$$(f_k, v)_k = (f, v), \quad v \in V_k. \tag{5.24}$$

Since A_k is symmetric positive definite (SPD) with respect to $(\cdot, \cdot)_k$, we define the following mesh-dependent norm

$$|\!|\!| v |\!|\!|_{s,k} := (A_k^s v, v)_k^{1/2}, \tag{5.25}$$

where A_k^s denotes the s^{th}-power of the SPD operator A_k for any $s \in \mathbb{R}$. Note that the norm $|\!|\!| \cdot |\!|\!|_{1,k}$ coincides with the energy norm $\| \cdot \|_E = \sqrt{a(\cdot, \cdot)}$. Moreover, $|\!|\!| v |\!|\!|_{2,k} = (A_k^2 v, v)_k^{1/2} = (A_k v, A_k v)_k^{1/2} = |\!|\!| A_k v |\!|\!|_{0,k}$. For the equivalence of the norm $|\!|\!| \cdot |\!|\!|_{0,k}$ with the L^2-norm we have the following result.

Lemma 5.8. *For $v \in V_k$ we have*

$$\|v\|_{L^2(\Omega)} \preceq |\!|\!| v |\!|\!|_{0,k} \preceq \|v\|_{L^2(\Omega)}. \tag{5.26}$$

Proof. Let $v = \sum_{i=1}^{n_k} v_i \psi_i$, where v_i are arbitrary. For any $K \in \mathcal{T}_h$, there are at most $(p+1)^d$ basis functions with non-zero support. Let $\bar{v}_K$ and the index set $\mathcal{I}_h^K$ be as defined in

Lemma 5.6. Also, let $\bar{v} = \max_{K\in\mathcal{T}_k} \bar{v}_K$. Using the positivity and the partition of unity properties of the basis functions, we know that $v|_K \leq \bar{v}_K$. Therefore, using $h_K \leq h_k$, we have

$$\begin{aligned}\|v\|^2_{L^2(\Omega)} &= \sum_{K\in\mathcal{T}_k} \int_K v^2 \leq \sum_{K\in\mathcal{T}_k} h_K^d \bar{v}_K^2 \leq \sum_{K\in\mathcal{T}_k} h_K^d \sum_{i\in\mathcal{I}_h^K} v_i^2 \\ &\preceq h_k^d \sum_{i=1}^{n_k} v_i^2 = |\!|\!| v |\!|\!|^2_{0,k}.\end{aligned}$$

For the right hand side inequality we have

$$|\!|\!| v |\!|\!|^2_{0,k} = h_k^d \sum_{i=1}^{n_k} v_i^2 \leq (ph_k + 1)^d \bar{v}^2 \leq (p+1)^d \bar{v}^2 \preceq \|v\|^2_{L^\infty(\Omega)},$$

where the equivalence constant, say γ_2, is $\mathcal{O}(p^{2d}2^{dp})$, see [83, 93]. The result then follows by using $\|v\|_{L^\infty(\Omega)} \leq \|v\|_{L^2(\Omega)}$. □

Note that the equivalence constant γ_2 is of the same order as γ_1 in Lemma 5.6, and can be improved up to $\mathcal{O}(p^d 2^{dp})$, see [93] for details.

To bound the spectral norm of the matrix A_k we proceed as follows. For SPD matrices we know that the eigenvalues can be estimated in terms of the Rayleigh quotients. Therefore, using the norm (5.25), the norm-equivalence relation (5.26), and the inverse inequality $\|v\|_{1,\Omega} \preceq h^{-1}\|v\|_{0,\Omega}$, we obtain

$$\|A_k\| = \lambda_{\max}(A_k) = \sup_x \frac{(x, A_k x)}{(x,x)} = \sup_{0\neq v\in V_k} \frac{|\!|\!| v |\!|\!|^2_{1,k}}{|\!|\!| v |\!|\!|^2_{0,k}} \preceq \sup_{0\neq v\in V_k} \frac{\|v\|_1^2}{\|v\|_0^2} \preceq h_k^{-2}. \tag{5.27}$$

5.4 Two-grid Analysis

In this section we present a two-grid analysis for solving the linear system (5.17). The purpose of this analysis is to show that the rate of convergence of the two-grid method for isogeometric linear system of equations is independent of the mesh-size h.

In a two-grid method, the solution of the system (5.17) is first approximated on the fine grid using a simple stationary iterative method (e.g., Jacobi or Gauss-Seidel), which is often referred to as relaxation process (or *smoother* because it smooths the error). Then, since on a coarser grid the smooth error can be well represented, and computations are cheaper, the resulting residual equation is transferred to the coarse grid and an error correction (by solving the residual equation) is computed. This error correction is then transferred back to the fine grid where it is added to the approximate solution obtained by the relaxation process. This is called the *coarse-grid correction* step. Finally, post-relaxation helps to further improve the fine-grid approximation by smoothing error components that may have been contaminated during the inter-grid transfer (from the coarse to the fine grid). The

convergence rate of any two-grid method like this depends on the efficiency of the relaxation method (smoother) and on the approximation properties of the inter-grid transfer operators, and on how well smoothing and coarse-grid correction complement each other.

For the two-grid analysis, we shall use the conventional notations h and H to denote the mesh size at the fine level and the coarse level, respectively. Together with the space of basis functions V, the SPD operator A, and the linear functional f, these notations shall be used to reflect the mesh level.

Let I_h be the identity matrix and G_h be the smoothing iteration matrix. Furthermore, let $P_h^H : V_h \to V_H$ be the orthogonal projection operator (called *restriction operator*) with respect to $a(\cdot,\cdot)$, i.e.

$$a(P_h^H v_h, w_H) = a(v_h, w_H), \quad \forall w_H \in V_H. \tag{5.28}$$

Another projection operator $P_H^h : V_H \to V_h$, called *prolongation operator*, is analogously defined. We know that the convergence of the two-grid method depends on the iteration matrix [60]

$$M = G_h^{\nu_2}(I_h - P_H^h A_H^{-1} P_h^H A_h) G_h^{\nu_1}, \tag{5.29}$$

where $A_H = P_h^H A_h P_H^h$, and ν_1 and ν_2 denote the number of *pre-* and *post-* smoothing steps, respectively. For simplicity sake (only in analysis), we take $\nu_2 = 0$. Then, for $\nu_1 = \nu$, the equation (5.29) can be written as [60]

$$M = (A_h^{-1} - P_H^h A_H^{-1} P_h^H) A_h G_h^{\nu}. \tag{5.30}$$

This break-up in to two separate parts, $A_h^{-1} - P_H^h A_H^{-1} P_h^H$ and $A_h G_h^{\nu}$, greatly helps the convergence analysis, see [60]. The factor $A_h^{-1} - P_H^h A_H^{-1} P_h^H$ is related to the *approximation property* and the factor $A_h G_h^{\nu}$ is related to the *smoothing property*.

In the following two sections we study the approximation property of the inter-grid transfer operators, and the smoothing property of the relaxation method. The h-independent convergence of the two-grid method, i.e.

$$\|M\| \leq \|A_h^{-1} - P_H^h A_H^{-1} P_h^H\| \|A_h G_h^{\nu}\| \preceq \eta(\nu), \tag{5.31}$$

where ν is defined in (5.43), is then an immediate consequence of (5.34), (5.44) and (5.47).

5.4.1 Approximation Property

To establish the approximation property we first prove the following Lemma, see e.g. [29].

Lemma 5.9. *Let $v_H := P_h^H v_h$. Then the following estimates hold for all $v_h \in V_h$.*

$$|\!|\!| v_h - v_H |\!|\!|_{0,h} \preceq h |\!|\!| v_h - v_H |\!|\!|_{1,h}, \tag{5.32a}$$

$$|\!|\!| v_h - v_H |\!|\!|_{1,h} \preceq h |\!|\!| v_h |\!|\!|_{2,h}. \tag{5.32b}$$

Proof. Using the triangle inequality, we have

$$\|\!|v_h - v_H|\!\|_{0,h} \leq \|\!|u - v_H|\!\|_{0,h} + \|\!|u - v_h|\!\|_{0,h}.$$

The inequality (5.32a) is then easily obtained by the equivalence of discrete norms and their continuous counter-parts, using (5.16), and noting that $H \leq ch$ for quasi-uniform nested meshes. For (5.32b) we proceed as follows.

$$\begin{aligned} \|\!|v_h - v_H|\!\|_{1,h}^2 = a(v_h - v_H, v_h - v_H) &= a(v_h - v_H, v_h) \\ = (v_h - v_H, A_h v_h) &\leq \|\!|v_h - v_H|\!\|_{0,h} \|\!|A_h v_h|\!\|_{0,h} \\ = \|\!|v_h - v_H|\!\|_{0,h} \|\!|v_h|\!\|_{2,h} &\preceq h \|\!|v_h - v_H|\!\|_{1,h} \|\!|v_h|\!\|_{2,h}, \end{aligned}$$

which gives the desired result. □

Combining (5.32a) and (5.32b) we get

$$\|\!|v_h - v_H|\!\|_{0,h} \preceq h^2 \|\!|v_h|\!\|_{2,h}.$$

Hence, the quality of approximation of $v_h := A_h^{-1} f_h$ by $P_H^h v_H$, where $v_H := A_H^{-1} P_h^H f_h$, can also be measured in terms of

$$\|\!|A_h^{-1} f_h - P_H^h A_H^{-1} P_h^H f_h|\!\|_{0,h} \preceq h^2 \|\!|A_h^{-1} f_h|\!\|_{2,h} = h^2 \|\!|f_h|\!\|_{0,h}. \tag{5.33}$$

Equivalently, albeit in a different terminology, see [29, 60] for details, the estimate (5.33) reads

$$\|A_h^{-1} - P_H^h A_H^{-1} P_h^H\| \preceq h^2 \preceq \|A_h\|^{-1}. \tag{5.34}$$

In Table 5.2, we present the spectral norm of $A_h^{-1} - P_H^h A_H^{-1} P_h^H$, which confirms the estimate (5.34).

Table 5.2: Illustration of the approximation property, i.e. $h^{-2}\|A_h^{-1} - P_H^h A_H^{-1} P_h^H\|$, $d = 2$.

p \ n_0	8	16	32	64
2	2.8125	2.8125	2.8125	2.8125
3	19.1435	18.2758	17.9280	17.8227
4	139.6540	122.87	117.4090	116.4410

5.4.2 Smoothing Property

In this section we recall the smoothing property of the symmetric Gauss-Seidel method. Let $A_h = D_h - L_h - U_h$ be the decomposition of the matrix A_h, where D_h denotes the diagonal matrix formed from the diagonal of A_h, and L_h and U_h denote strictly lower and strictly upper triangular matrices, respectively. From $A_h = A_h^T$ it follows that $U_h = L_h^T$.

Now consider the symmetric Gauss-Seidel iteration

$$u_h^{\nu+1} = G_h u_h^{\nu} + B_h^{-1} f_h, \qquad \forall \nu = 0, 1, \dots. \tag{5.35}$$

where the preconditioner B_h is given by

$$B_h = (D_h - L_h) D_h^{-1} (D_h - U_h) = A_h + L_h D_h^{-1} U_h, \tag{5.36}$$

and the iteration matrix G_h is given by

$$G_h = (D_h - U_h)^{-1} L_h (D_h - L_h)^{-1} U_h. \tag{5.37a}$$

It is easy to see that

$$\begin{aligned} G_h &= \left(I_h - (D_h - U_h)^{-1} A_h\right) \left(I_h - (D_h - L_h)^{-1} A_h\right) \\ &= I_h - \left((D_h - L_h) D_h^{-1} (D_h - U_h)\right)^{-1} A_h \\ &= I_h - B_h^{-1} A_h. \end{aligned} \tag{5.37b}$$

Note that if A_h is SPD (denoted by $A_h > 0$ since $(A_h x, x) > 0$ for all $x \neq 0$) then the matrix D_h and the preconditioner B_h are SPD, and we have the estimate

$$0 < A_h \leq A_h + L_h D_h^{-1} U_h = B_h. \tag{5.38}$$

Moreover,

$$(D_h)_{i,i} = h^{-d} a(\psi_i, \psi_i) \succeq h^{-d} |\psi_i|_1^2 \succeq h^{-2}, \tag{5.39}$$

because by using a Poincare type inequality on the domain $\mathrm{supp}(\psi_i)$ of characteristic size $(p+1)h \sim h$, it can be shown that $|\psi_i|_1^2 \succeq h^{d-2}$. Note that the inequality constant also depends on the stability constant and γ_1^{-2}. Therefore, using $\|D_h^{-1}\| = \max_i (D_h)_{i,i}^{-1}$ we get

$$\|D_h^{-1}\| \preceq h^2 \preceq \|A_h\|^{-1}. \tag{5.40}$$

We also note that $\|L_h\|_\infty = \max_i \sum |l_{ij}| \leq c \max_{i,j} |l_{ij}| \leq c \max_{i,j} |a_{ij}| \leq c\|A_h\|$, where l_{ij} and a_{ij} denote the entries of the matrices L_h and A_h, respectively, and c is the maximum number of non-zero entries per row (which depends on the polynomial degree p). Similarly, it can be shown that $\|L_h\|_1 \leq c\|A_h\|$. Therefore, using $\|\cdot\|^2 \leq \|\cdot\|_1 \|\cdot\|_\infty$, we get

$$\|U_h\|^2 = \|U_h^T\|^2 = \|L_h\|^2 \leq \|L_h\|_1 \|L_h\|_\infty \preceq \|A_h\|^2. \tag{5.41}$$

From (5.40) and (5.41) we get

$$\|B_h\| = \|A_h + L_h D_h^{-1} U_h\| \preceq \|A_h\|. \tag{5.42}$$

We are now in a position to prove the following lemma.

Lemma 5.10. *Let*

$$\eta(\nu) := \frac{\nu^\nu}{(\nu+1)^{(\nu+1)}}. \tag{5.43}$$

The symmetric Gauss-Seidel method (5.35) satisfies the smoothing property

$$\|A_h G_h^\nu\| \preceq \eta(\nu)\|A_h\|, \tag{5.44}$$

where the function $\eta(\nu) \to 0$ as $\nu \to \infty$.

Proof. Let $X_h := B_h^{-1/2} A_h B_h^{-1/2}$. From (5.38) it follows that $\rho(X_h) \leq 1$. Also, from [60, Lemma 6.2.1] we have $\|X_h(I_h - X_h)^\nu\| \leq \eta(\nu)$ for $0 \leq X_h = X_h^T \leq I_h$. Hence, using (5.42) we obtain

$$\begin{aligned}
\|A_h G_h^\nu\| &= \|B_h B_h^{-1} A_h G_h^\nu\| \\
&\leq \|B_h\| \|B_h^{-1} A_h (I_h - B_h^{-1} A_h)^\nu\| \\
&\preceq \|A_h\| \|B_h^{-1/2} A_h (I_h - B_h^{-1} A_h)^\nu B_h^{-1/2}\| \\
&= \|A_h\| \|X_h (I_h - X_h)^\nu\| \\
&\leq \eta(\nu)\|A_h\|,
\end{aligned} \tag{5.45}$$

which completes the proof. □

For the non-symmetric (forward) Gauss-Seidel method, with $B_h = D_h - L_h$, we proceed as follows.

Lemma 5.11. *Let $\|\cdot\|$ be a matrix norm corresponding to a vector norm. Let $G_h = I_h - B_h^{-1} A_h$ be the iteration matrix of the smoother, and $X_h = I_h - 2B_h^{-1} A_h$ be some matrix. Assume*

$$\|X_h\| \leq 1, \quad \text{and} \quad \|B_h\| \preceq \|A_h\|. \tag{5.46}$$

Then for $\nu \geq 1$ the following smoothing property holds

$$\|A_h G_h^\nu\| \preceq \sqrt{2/(\pi\nu)}\|A_h\|. \tag{5.47}$$

Proof. We have $I_h - X_h = 2B_h^{-1} A_h$, and $I_h + X_h = 2(I_h - B_h^{-1} A_h)$. Therefore,

$$(I_h - X_h)(I_h + X_h)^\nu = 2^{\nu+1} B_h^{-1} A_h (I_h - B_h^{-1} A_h)^\nu = 2^{\nu+1} B_h^{-1} A_h G_h^\nu.$$

Therefore, $A_h G_h^\nu = 2^{-(\nu+1)} B_h (I_h - X_h)(I_h + X_h)^\nu$. Now using $\|(I_h - X_h)(I_h + X_h)^\nu\| \leq 2^{\nu+1}\sqrt{2/(\pi\nu)}$ for some matrix X_h with $\|X_h\| \leq 1$ (from Reusken's Lemma [89], see also [61, Theorem 10.6.8, Lemma 10.6.9]), we get the desire result. □

In Table 5.3, we list the spectral norm of $A_h G_h^\nu$ for $\nu = 1, \ldots, 4$, symmetric Gauss-Seidel iterations, which confirms the estimate (5.45). To compare the smoothing property of symmetric Gauss-Seidel iterations with forward Gauss-Seidel iterations, since the latter is practically advantageous, in Table 5.4, we list the spectral norm of $A_h G_h^\nu$ for $\nu = 1, \ldots, 8$, forward Gauss-Seidel iterations. As one might expect, the effect of one symmetric (one forward followed by one backward) Gauss-Seidel iteration is almost the same as for two forward Gauss-Seidel iterations. In fact, we see that for higher p and smaller ν, the forward Gauss-Seidel iterations perform better than the symmetric version. Due to this reason, we will use forward Gauss-Seidel iterations in our numerical tests for multigrid convergence.

Table 5.3: Illustration of the smoothing property, i.e. $h^2\|A_h G_h^\nu\|$, for symmetric Gauss-Seidel method, $d = 2$

	$p=1$				$p=2$			
ν \ n_0	8	16	32	64	8	16	32	64
1	0.3523	0.3789	0.3889	0.3915	0.1317	0.1534	0.1597	0.1620
2	0.1312	0.1468	0.1516	0.1535	0.0462	0.0596	0.0622	0.0632
3	0.0856	0.0894	0.0929	0.0941	0.0181	0.0346	0.0376	0.0388
4	0.0563	0.0662	0.0669	0.0678	0.0071	0.0266	0.0276	0.0280
	$p=3$				$p=4$			
ν \ n_0	8	16	32	64	8	16	32	64
1	0.1948	0.1947	0.1947	0.1947	0.3775	0.3878	0.3904	0.3911
2	0.0530	0.0521	0.0520	0.0520	0.0917	0.0918	0.0918	0.0918
3	0.0253	0.0251	0.0251	0.0251	0.0364	0.0360	0.0360	0.0360
4	0.0158	0.0157	0.0163	0.0164	0.0222	0.0222	0.0222	0.0222

5.5 Multigrid Convergence

In this section we summarize some important consequences of the smoothing and approximation properties on the convergence of the classical multigrid algorithm in the setting of the isogeometric discretization. Since the proofs of the quoted convergence results can be found in [60], we confine ourselves to a short discussion without repeating any proofs.

For convenience we first consider the *symmetric case* which is the simplest to analyze. Let

$$A_k \equiv A_h = A_h^T > 0, \tag{5.48a}$$

$$P_k \equiv P_H^h = (P_h^H)^T, \tag{5.48b}$$

denote the stiffness matrix and the interpolation matrix at level k, respectively, where $1 \leq k \leq L$. Further, let the coarse grid matrix A_{k-1} satisfy the Galerkin relation

$$A_{k-1} = P_k^T A_k P_k \equiv P_h^H A_h P_H^h = A_H. \tag{5.49}$$

Moreover, assume that the preconditioner B_k is SPD, i.e.,

$$B_k \equiv B_h = B_h^T, \tag{5.50}$$

and that the smoothing iteration at level k is defined via the iteration matrix

$$G_k = I_k - B_k^{-1} A_k. \tag{5.51}$$

Then the iteration matrix of the classical multigrid algorithm with ν_1 pre- and ν_2 post-smoothing steps at level k can be recursively defined via

$$M_k(\nu_1, \nu_2) := G_k^{\nu_2} \left(I_k - P_k \left(I_{k-1} - (M_{k-1}(\nu_1, \nu_2))^\gamma \right) A_{k-1}^{-1} P_k^T A_k \right) G_k^{\nu_1}, \tag{5.52}$$

Table 5.4: Illustration of the smoothing property, i.e. $h^2\|A_h G_h^\nu\|$, for forward Gauss-Seidel method, $d = 2$

	$p=1$				$p=2$			
ν \ n_0	8	16	32	64	8	16	32	64
1	0.8917	0.9508	0.9674	0.9716	0.3508	0.3817	0.3946	0.3982
2	0.3496	0.3783	0.3888	0.3915	0.1262	0.1525	0.1595	0.1619
3	0.2007	0.2134	0.2206	0.2229	0.0738	0.0861	0.0905	0.0919
4	0.1314	0.1466	0.1516	0.1535	0.0447	0.0599	0.0622	0.0632
5	0.1065	0.1138	0.1153	0.1168	0.0260	0.0447	0.0477	0.0481
6	0.0862	0.0895	0.0930	0.0941	0.0147	0.0348	0.0377	0.0389
7	0.0697	0.0760	0.0783	0.0788	0.0082	0.0305	0.0323	0.0324
8	0.0561	0.0666	0.0671	0.0678	0.0045	0.0267	0.0277	0.0281
	$p=3$				$p=4$			
ν \ n_0	8	16	32	64	8	16	32	64
1	0.4897	0.4918	0.4945	0.4951	0.6895	0.7160	0.7218	0.7230
2	0.1758	0.1731	0.1729	0.1729	0.2766	0.2833	0.2843	0.2845
3	0.0868	0.0856	0.0854	0.0854	0.1240	0.1247	0.1257	0.1260
4	0.0510	0.0502	0.0501	0.0501	0.0743	0.0730	0.0730	0.0730
5	0.0342	0.0333	0.0332	0.0332	0.0486	0.0483	0.0483	0.0483
6	0.0249	0.0243	0.0242	0.0242	0.0349	0.0345	0.0345	0.0345
7	0.0193	0.0190	0.0190	0.0194	0.0269	0.0263	0.0263	0.0263
8	0.0159	0.0156	0.0163	0.0164	0.0214	0.0212	0.0211	0.0211

where $M_0(\nu_1, \nu_2) = 0$, cf., [60, Lemma 7.1.4]. Note that the choices $\gamma = 1$ and $\gamma = 2$ in (5.52) correspond to the classical V-cycle and W-cycle multigrid methods, respectively.

5.5.1 W-cycle Convergence

Consider the iteration matrix (5.52) of the W-cycle method, i.e., the case $\gamma = 2$, Further, for convenience, let $\nu_1 = \nu_2 = \nu/2$ on all levels k where $1 \le k \le L$. Then the following convergence result holds true, cf. [60, Theorem 7.2.3].

Theorem 5.12 (Convergence of W-cycle)**.** *Let (5.48)-(5.51) hold, and the approximation property (5.34) be satisfied on all levels $k = 1, 2, \ldots, L$ with a constant c_A, i.e.,*

$$\|A_k^{-1} - P_k A_{k-1}^{-1} P_k^T\| \le c_A \|A_k\|^{-1}. \tag{5.53}$$

If $c_A > 1$ and $\nu \le (c_A - 1)\,(1 - (1 - 1/c_A)^{2\nu})$, the contraction number of the W-cycle method ($\gamma = 2$) with $\nu/2$ pre- and $\nu/2$ post-smoothing steps can be estimated by

$$\|M_k(\nu/2, \nu/2)\| \le (1 - 1/c_A)^\nu < 1. \tag{5.54}$$

Otherwise, the smallest root $\zeta := \zeta(\nu)$ *of* $\zeta = \eta(\nu)\left(\zeta^2 + (1-\zeta^2)c_A\right)^{\nu+1}$ *satisfies*

$$\|M_k(\nu/2, \nu/2)\| \leq \zeta(\nu) \tag{5.55}$$

for all $k \geq 0$.

5.5.2 V-cycle Convergence

Next, consider the iteration matrix (5.52) of the V-cycle method, i.e., the case $\gamma = 1$. For the case of equal number of pre- and post-smoothing steps, i.e., $\nu_1 = \nu_2 = \nu/2$, we have the following convergence estimate for the V-cycle, cf. [60, Theorem 7.2.2].

Theorem 5.13 (Convergence of V-cycle)**.** *Under the assumptions of Theorem 5.12 the V-cycle method (*$\gamma = 1$*) is convergent. In the case* $\nu_1 = \nu_2 = \nu/2$ *its contraction number can be estimated by*

$$\|M_k(\nu/2, \nu/2)\| \leq \frac{c_A}{c_A + \nu} < 1. \tag{5.56}$$

For the more general case of ν_1 pre- and ν_2 post-smoothing steps, see [60, Theorem 7.2.5]. The numerical results in the next section indicate, however, that these estimates are somewhat pessimistic, and that one obtains better convergence rates in practice.

5.6 Numerical Results for Multigrid Convergence

To test the multigrid solvers' performance, we consider the following test problems, whose discretizations are performed using the Matlab toolbox GeoPDEs [48, 49].

Example 5.1. Let $\Omega = (0,1)^2$. Together with $\mathcal{A} = I$, and homogeneous Dirichlet boundary conditions, the right hand side function f is chosen such that the analytical solution of the problem is given by $u = \sin(\pi x)\sin(\pi y)$.

Example 5.2. Let $\Omega = (0,1)^2$. Together with $\mathcal{A} = x^5 \exp(10y) I$, and homogeneous Dirichlet boundary conditions, the right hand side function f is chosen such that the analytical solution of the problem is given by $u = \sin(\pi x)\sin(\pi y)$.

Example 5.3. The domain is chosen as a quarter annulus in the first Cartesian quadrant with inner radius 1 and outer radius 2, see [48]. Together with $\mathcal{A} = I$, and homogeneous Dirichlet boundary conditions, the right hand side function f is chosen such that the analytic solution is given by $u = (x^2 + y^2 - 3\sqrt{x^2+y^2} + 2)\sin(2\arctan(y/x))$.

Example 5.4. Let $\Omega = (0,1)^3$. Together with $\mathcal{A} = I$, and homogeneous Dirichlet boundary conditions, the right hand side function f is chosen such that the analytical solution of the problem is given by $u = \sin(\pi x)\sin(\pi y)\sin(\pi z)$.

Furthermore, the operator P_h^H is chosen such that the coarse basis functions are exactly represented in the space of fine basis functions. At the finest level (largest problem size), the parametric domain is divided into n_0 equal elements in each direction. The initial guess for (iteratively) solving the linear system of equations is chosen as a random vector. Let r_0 denote the initial residual vector and $r_{\rm it}$ denote the residual vector at a given multigrid iteration $n_{\rm it}$. The following stopping criteria is used

$$\frac{\|r_{\rm it}\|}{\|r_0\|} \leq 10^{-8}. \tag{5.57}$$

The average convergence factor reported in the following tables is defined as $\rho = \left(\frac{\|r_{\rm it}\|}{\|r_0\|}\right)^{1/n_{\rm it}}$.

In the tables, by M we collectively denote the multigrid method which is specified by the choice of the cycle, i.e. V- or W- or F- cycle. Moreover, p, ν, and L denote the polynomial degree, number of pre- and post- smoothing steps, and the number of mesh refinement levels, respectively. For all the test cases we take the polynomial degree $p = 2, 3, 4$.

For Example 5.1, since the geometry mapping is identity, it suffices to choose the basis functions as B-splines. To evaluate the integrals computationally, we use the Gauss-quadrature formulas with number of quadrature points $n_q = p + 1$ in each direction. This number is sufficient since the Jacobian from the mapping is constant.

We first present the ν-dependence of two-grid V-cycle method in Table 5.5. We see that for a fixed polynomial degree p and a fixed mesh size h, the number of iterations $n_{\rm it}$ of two-grid V-cycle inversely depends on the number of smoothing steps ν.
To study the effect of refinement levels on the convergence of V-cycle method, in Table 5.6, we present the average convergence factor ρ and number of iterations $n_{\rm it}$ against the number of refinement levels for a fixed $h = 1/256$, and with C^0 and C^{p-1} smoothness. As predicted by the theoretical estimates on the optimality of the V-cycle method, it is not surprising to see that ρ and $n_{\rm it}$ are practically same for all refinement levels. We do not repeat this study for W- and F- cycles, which are also of optimal order and their results for $L = 2, 4$ are presented in the next Table.
In Table 5.7, we present the average convergence factor ρ and number of iterations $n_{\rm it}$ for V-, W-, and F-cycle multigrid methods. The mesh size varies from $1/8$ to $1/64$ in each direction. We consider both the extreme cases of smoothness, namely, C^0 and C^{p-1}. As all the cycles are of optimal order, to present a comparative study of all the cases in a concise manner, we consider here only $L = 2, 4$. We make the following observations.

- For all polynomial degrees, all the approaches exhibit optimal convergence with respect to the mesh refinement, which confirm the theoretical estimates (5.31) for two-grid method, and (5.54)-(5.56) for multigrid methods.
- For a fixed mesh size, since the condition number rapidly increases with increasing polynomial degree, this affects the two-(multi-)grid convergence.
- For C^0 smoothness, for any given polynomial degree, the convergence factor is slower (and thus requires more number of iterations) as compared to the problem with C^{p-1}

Table 5.5: Poisson problem in a unit square: ν-dependence of two-grid V-cycle

ν \ n_0	8		16		32		64	
	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}
$p = 2$								
1	0.1639	11	0.1869	11	0.1819	11	0.1833	11
2	0.0286	6	0.0320	6	0.0338	6	0.0350	6
4	0.0010	3	0.0009	3	0.0010	3	0.0011	3
8	1.0e-06	2	3.0e-06	2	3.0e-06	2	3.0e-06	2
$p = 3$								
1	0.6052	37	0.5864	35	0.5987	36	0.6039	37
2	0.3659	19	0.3494	18	0.3716	19	0.3584	18
4	0.1197	9	0.1195	9	0.1385	10	0.1278	9
8	0.0212	5	0.0172	5	0.0179	5	0.0180	5
$p = 4$								
1	0.8790	143	0.8645	127	0.8586	121	0.8598	122
2	0.7763	73	0.7611	68	0.7418	62	0.7392	61
4	0.5487	31	0.5614	32	0.5611	32	0.5502	31
8	0.3293	17	0.3281	17	0.3069	16	0.3043	16

Table 5.6: Poisson problem in a unit square: V-cycle convergence, n_{it} (and ρ) versus L, $h = 1/256$; $\nu = 2$

	C^0						C^{p-1}					
	$p = 2$		$p = 3$		$p = 4$		$p = 2$		$p = 3$		$p = 4$	
L	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}
2	0.0349	6	0.4051	21	0.8143	90	0.0358	6	0.3569	18	0.7420	62
3	0.0349	6	0.4050	21	0.8144	90	0.0358	6	0.3569	18	0.7420	62
4	0.0349	6	0.4050	21	0.8144	90	0.0358	6	0.3569	18	0.7420	62
5	0.0349	6	0.4050	21	0.8144	90	0.0358	6	0.3569	18	0.7420	62
6	0.0349	6	0.4050	21	0.8144	90	0.0358	6	0.3569	18	0.7420	62
7	0.0349	6	0.4050	21	0.8144	90	0.0358	6	0.3569	18	0.7420	62
8	0.0349	6	0.4050	21	0.8144	90	0.0358	6	0.3569	18	0.7420	62

smoothness. This phenomenon, which is more prominent for higher polynomial degrees, may be attributed to an increased problem size.

- Since the V-cycle method is optimal, we see that the performance of the W-cycle for four-grids, i.e. $L = 4$, is only as good as the V-cycle method. Moreover, there is a consistent improvement of a factor about $2/3$ in the number of iterations in the F-cycle

Table 5.7: Poisson problem in a unit square: Multigrid convergence, $\nu = 2$

$M(L)$ \ n_0	8		16		32		64	
	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}
$p = 2, \quad C^0$								
$V(2)$	0.0236	5	0.0337	6	0.0340	6	0.0341	6
$V(4)$	0.0236	5	0.0338	6	0.0340	6	0.0341	6
$W(4)$	0.0236	5	0.0337	6	0.0340	6	0.0341	6
$F(4)$	0.0039	4	0.0062	4	0.0062	4	0.0063	4
$p = 2, \quad C^{p-1}$								
$V(2)$	0.0290	6	0.0351	6	0.0347	6	0.0356	6
$V(4)$	0.0290	6	0.0351	6	0.0347	6	0.0356	6
$W(4)$	0.0290	6	0.0351	6	0.0347	6	0.0356	6
$F(4)$	0.0049	4	0.0066	4	0.0065	4	0.0067	4
$p = 3, \quad C^0$								
$V(2)$	0.3762	19	0.3922	20	0.4068	21	0.4043	21
$V(4)$	0.3761	19	0.3922	20	0.4067	21	0.4043	21
$W(4)$	0.3762	19	0.3922	20	0.4068	21	0.4043	21
$F(4)$	0.2335	13	0.2506	14	0.2595	14	0.2571	14
$p = 3, \quad C^{p-1}$								
$V(2)$	0.3589	18	0.3468	18	0.3465	18	0.3546	18
$V(4)$	0.3589	18	0.3468	18	0.3465	18	0.3546	18
$W(4)$	0.3589	18	0.3468	18	0.3465	18	0.3546	18
$F(4)$	0.2150	12	0.2042	12	0.2040	12	0.2111	12
$p = 4, \quad C^0$								
$V(2)$	0.8101	88	0.8145	90	0.8122	89	0.8139	90
$V(4)$	0.8103	88	0.8147	90	0.8122	89	0.8140	90
$W(4)$	0.8103	88	0.8147	90	0.8122	89	0.8139	90
$F(4)$	0.7299	59	0.7353	60	0.7315	59	0.7343	60
$p = 4, \quad C^{p-1}$								
$V(2)$	0.7494	64	0.7679	70	0.7278	58	0.7387	61
$V(4)$	0.7493	64	0.7679	70	0.7278	58	0.7387	61
$W(4)$	0.7493	64	0.7679	70	0.7278	58	0.7387	61
$F(4)$	0.6496	43	0.6736	47	0.6220	39	0.6358	41

as compared to the number of V-cycle iterations. This compensates the additional computational cost in F-cycle to a good extent.

We now study the performance of V-cycle multigrid solver on a multi-patch geometry. This simple model case is produced by p times repetition of the knot at $h = 1/2$ (in both directions). Thereby, we get four patch fully-conforming geometry which has C^0 smoothness at $h = 1/2$ interfaces and C^{p-1} smoothness elsewhere. The coarsest mesh is fixed with $n_0 = 4$

elements in each direction and for both the refinements (2-level and 4-level). The results presented in Table 5.8 show that the convergence behavior fits nicely between the convergence behavior for global C^0 and C^{p-1} smoothness, with a bias towards C^{p-1} smoothness.

Table 5.8: Poisson problem in a unit square: V-cycle convergence on a multi-patch geometry, $\nu = 2$

	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}
$L = 2$								
p \ n_0	8		16		32		64	
2	0.0217	5	0.0284	6	0.0353	6	0.0343	6
3	0.3925	20	0.3790	19	0.3727	19	0.3651	19
4	0.8082	87	0.7756	73	0.7558	66	0.7485	64
$L = 4$								
p \ n_0	32		64		128		256	
2	0.0353	6	0.0343	6	0.0352	6	0.0357	6
3	0.3727	19	0.3651	19	0.3578	18	0.3577	18
4	0.7558	66	0.7485	64	0.7423	62	0.7448	63

We now consider Example 5.2 with variable coefficients. In Table 5.9, we present the results for V-cycle multigrid convergence for $p = 2, 3, 4$ and $L = 4$. We take the number of quadrature points $n_q = p + 2$ in both the directions so that the integrals with respect to x-variable are evaluated exactly. However, due to the exponential function, exact integration is not possible with respect to y-variable. We note that the results are qualitatively same as those with constant coefficients case (see Table 5.7).

We now consider Example 5.3 with curved boundary. The geometry for this example is represented by NURBS basis functions of degree 1 in the radial direction and of degree 2 in the angular direction, see [48]. Since the Jacobian of the geometry mapping is no more a constant, for exact integral evaluations it does not suffice to take the number of Gauss quadrature points $n_q = p + 1$ in each direction (which is clear from simple heuristic arguments). Therefore, we choose $n_q = p + 2$. From numerical experiments, it is found that this is sufficient (for up to $p = 4$) to keep the approximation error (5.34) smaller than the L^2-norm of the discretization error (which otherwise would contaminate the accuracy of two-(multi-)grid solver). Note however that this is not detrimental to the optimality of any of the methods, which can be seen from the variable coefficients case presented in Table 5.9. In Table 5.10, we present the ν-dependence of two-grid V-cycle method. In Table 5.11, we present the convergence factor and the number of iterations for V-, W-, and F- cycle multigrid methods. The mesh size again varies from $1/8$ to $1/64$ in each direction, and both the extreme cases of smoothness, namely, C^0 and C^{p-1} are considered. All the results are qualitatively similar to that of Example 5.1 with square domain.

Table 5.9: Variable coefficients elliptic problem in a unit square: V-cycle convergence, $\nu = 2$; $L = 4$

p \ n_0	8		16		32		64	
	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}
C^0								
2	0.0177	5	0.0241	5	0.0290	6	0.0322	6
3	0.3162	16	0.3872	20	0.3887	20	0.3910	20
4	0.8005	83	0.7977	82	0.8104	88	0.8121	89
C^{p-1}								
2	0.0342	6	0.0199	5	0.0306	6	0.0357	6
3	0.3067	16	0.3737	19	0.3556	18	0.3516	18
4	0.8146	90	0.7870	77	0.7257	58	0.7260	58

Table 5.10: Poisson problem in a quarter annulus: ν-dependence of two-grid V-cycle

ν \ n_0	8		16		32		64	
	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}
$p = 2$								
1	0.1926	12	0.2823	15	0.3052	16	0.3319	17
2	0.0371	6	0.0810	8	0.0931	8	0.1126	9
4	0.0014	3	0.0066	4	0.0087	4	0.0136	5
8	3.0e-06	2	4.3e-05	2	7.5e-05	2	2.3e-04	3
$p = 3$								
1	0.5858	35	0.6118	38	0.5977	36	0.6036	37
2	0.3477	18	0.3741	19	0.3575	18	0.3670	19
4	0.1196	9	0.1437	10	0.1277	9	0.1383	10
8	0.0159	5	0.0206	5	0.0181	5	0.0191	5
$p = 4$								
1	0.8703	133	0.8594	122	0.8604	123	0.8617	124
2	0.7564	66	0.7384	61	0.7408	62	0.7425	62
4	0.5767	34	0.5475	31	0.5488	31	0.5513	31
8	0.3331	17	0.3046	16	0.3054	16	0.3083	16

Finally, we consider the three-dimensional problem described in Example 5.4. The results for V-cycle multigrid method are presented in Table 5.12, which confirm the h-independence and optimality of the solver. The entries marked by † represent the cases where the results could not be obtained due to limitation on computational resources. As shown by the results of two-dimensional examples, the W- and F-cycle methods will not offer any improvement in convergence results, and are thus not repeated here.

Table 5.11: Poisson problem in a quarter annulus: Multigrid convergence, $\nu = 2$

$M(L)$ \ n_0	8		16		32		64	
	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}
$p = 2, \quad C^0$								
$V(2)$	0.0716	7	0.0977	8	0.0985	8	0.1071	9
$V(4)$	0.0716	7	0.0976	8	0.0985	8	0.1071	9
$W(4)$	0.0716	7	0.0977	8	0.0985	8	0.1071	9
$F(4)$	0.0189	5	0.0314	6	0.0325	6	0.0346	6
$p = 2, \quad C^{p-1}$								
$V(2)$	0.0371	6	0.0810	8	0.0931	8	0.1126	9
$V(4)$	0.0371	6	0.0810	8	0.0931	8	0.1126	9
$W(4)$	0.0371	6	0.0810	8	0.0931	8	0.1126	9
$F(4)$	0.0071	4	0.0225	5	0.0302	6	0.0378	6
$p = 3, \quad C^0$								
$V(2)$	0.3904	20	0.4046	21	0.3977	20	0.4046	21
$V(4)$	0.3903	20	0.4045	21	0.3975	20	0.4045	21
$W(4)$	0.3903	20	0.4046	21	0.3976	20	0.4046	21
$F(4)$	0.2415	13	0.2573	14	0.2556	14	0.2574	14
$p = 3, \quad C^{p-1}$								
$V(2)$	0.3477	18	0.3741	19	0.3575	18	0.3670	19
$V(4)$	0.3469	18	0.3741	19	0.3575	18	0.3670	19
$W(4)$	0.3472	18	0.3741	19	0.3575	18	0.3670	19
$F(4)$	0.2046	12	0.2311	13	0.2138	12	0.2246	13
$p = 4, \quad C^0$								
$V(2)$	0.8158	91	0.8184	92	0.8232	95	0.8230	95
$V(4)$	0.8145	90	0.8183	92	0.8229	95	0.8228	95
$W(4)$	0.8145	90	0.8183	92	0.8229	95	0.8228	95
$F(4)$	0.7351	60	0.7413	62	0.7461	63	0.7459	63
$p = 4, \quad C^{p-1}$								
$V(2)$	0.7564	66	0.7384	61	0.7408	62	0.7421	62
$V(4)$	0.7582	67	0.7384	61	0.7408	62	0.7422	62
$W(4)$	0.7581	67	0.7384	61	0.7408	62	0.7422	62
$F(4)$	0.6609	45	0.6355	41	0.6368	41	0.6411	42

For all the examples, we also tested the multigrid convergence for intermediate continuities C^r, i.e. $0 < r < p - 1$, and found that the results lie nicely between the results of C^0 and C^{p-1} continuities. However, they are not reported here for brevity reasons. We also remark the following on the numerical results of high polynomial degrees and where the exact solution has reduced regularity.

Remark 5.14. *It is known from finite elements literature that standard h- multigrid, which*

Table 5.12: Poisson problem in a unit cube: V-cycle multigrid convergence, $\nu = 2$

p \ n_0	8		16		32	
	ρ	n_{it}	ρ	n_{it}	ρ	n_{it}
C^0, $L = 2$						
2	0.3578	18	0.4073	21	0.4066	21
3	0.8221	147	0.8929	163	0.8947	166
4	0.9879	1514	0.9881	1540	†	†
C^{p-1}, $L = 2$						
2	0.2874	15	0.3383	17	0.3692	19
3	0.8582	121	0.8403	106	0.8431	108
4	0.9728	669	0.9751	731	0.9745	713
C^0, $L = 4$						
2	0.3685	19	0.3977	20	0.4076	21
3	0.8891	157	0.8923	162	0.8942	165
4	0.9877	1493	0.9881	1543	†	†
C^{p-1}, $L = 4$						
2	0.3339	17	0.2356	18	0.3700	19
3	0.8572	120	0.8556	110	0.8422	112
4	0.9772	797	0.9738	695	0.9740	698

is the focus of this chapter, is not suited for high polynomial degree. Most of the literature is for first and second degree polynomials only. This fact is related to the smoothing properties of the classical smoothers like Jacobi, Gauss-Seidel or Richardson methods. These methods work effectively only when the error function is oscillatory, whereas the error function gets smoother with increasing polynomial degree. For high polynomial degree, either p-multigrid should be used or different smoothers should be devised. Nevertheless, since isogeometric method in engineering applications mostly utilize second or third degree polynomials, in this study we considered polynomial degree up to $p = 4$.

Remark 5.15. *In the presence of discontinuities in the coefficients, or due to the irregular geometry (e.g. L-shaped domain), the exact solution of elliptic problems has reduced regularity and lies only in $H^{1+\epsilon}(\Omega)$, where $0 < \epsilon < 1$ depends on the strength of the singularity. Firstly, in such cases the single-patch isogeometric approach with global continuity $r > 0$ (for $p > 1$) is not so attractive. Secondly, the standard (geometric) multigrid methods are not tailored for such general problems and need special treatment. The reduced regularity negatively affects the approximation property of Lemma 5.9, and thus the overall convergence behavior of solver. Though specific problems can be treated to obtain optimal order convergence (which involves more technical results). For such problems, the multi-patch techniques, such as the tearing and interconnecting approach of Kleiss et al. [70] or BDDC approach of Beirao et al. [22], are more suitable where the multigrid solver can be used within each sub-patch.*

5.7 Conclusions

We have presented multigrid methods, with V-, W- and F- cycles, for the linear system arising from the isogeometric discretization of the scalar second order elliptic problems. For a given polynomial degree p, all multigrid cycles are of optimal complexity with respect to the mesh refinement. Despite that the condition number of the stiffness matrix grows very rapidly with the polynomial degree, these excellent results exhibit the power of multigrid methods. Nevertheless, this study can only be regarded as a first step towards utilizing the power of multigrid methods in isogeometric analysis.

Chapter 6

Algebraic Multilevel Preconditioning in Isogeometric Analysis: Construction and Numerical Studies

In this chapter, we discuss the construction of linear solvers which provide not only h-independent convergence rates but also exhibit (almost) independence on p. The presented optimal order solvers are based on algebraic multilevel iteration (AMLI) methods.
AMLI methods were introduced by Axelsson and Vassilevski in a series of papers [7–10]. The AMLI methods, which are recursive extensions of two-level multigrid methods for FEM [6], have been extensively analyzed in the context of conforming and nonconforming FEM (including discontinuous Galerkin methods). For a detailed systematic exposition of AMLI methods, see the monograph [73, 100]. To reduce the overall complexity of AMLI methods (to achieve optimal computational complexity), various stabilization techniques can be used. In the original work [7, 8], the stabilization was achieved by employing properly shifted and scaled Chebyshev polynomials. This approach requires the computation of polynomial coefficients which depends on the bounds of the eigenvalues of the preconditioned system. Alternatively, some inner iterations at coarse levels can be used to stabilize the outer iterations, which lead to parameter-free AMLI methods [9, 10, 72, 85]. These methods utilize a sequence of coarse-grid problems that are obtained from repeated application of a natural (and simple) hierarchical basis transformation, which is computationally advantageous. Moreover, the underlying technique of these methods often requires only a few minor adjustments (mainly two-level hierarchical basis transformation) even if the underlying problem changes significantly.

6.1 Construction of Hierarchical Spaces

The hierarchical basis techniques result in splittings in which the angle between the coarse space and its hierarchical complement is uniformly bounded with respect to the mesh size. We recall from Section 2.6, the following two-level hierarchical basis representation for

stiffness matrix at fine level

$$\hat{A}^{(k)} = \begin{bmatrix} \hat{A}_{11}^{(k)} & \hat{A}_{12}^{(k)} \\ \hat{A}_{21}^{(k)} & \hat{A}_{22}^{(k)} \end{bmatrix} = \begin{bmatrix} \hat{A}_{11}^{(k)} & \hat{A}_{12}^{(k)} \\ \hat{A}_{21}^{(k)} & A^{(k-1)} \end{bmatrix}, \tag{6.1}$$

where $\hat{A}_{22}^{(k)}$ represents the matrix corresponding to coarse basis functions and $\hat{A}_{11}^{(k)}$ represents the matrix corresponding to its hierarchical complement, and $1 \leq k \leq L$. Recall from Section 4.2, for B-splines we have the following transformations

$$\hat{A}_{22}^{(k)} = G_k^{p,r} A^k (G_k^{p,r})^T, \tag{6.2}$$

respectively. For hierarchical complementary spaces, let $T_k^{p,r}$ be the matrix such that

$$\hat{A}_{11}^{(k)} = T_k^{p,r} A^k (T_k^{p,r})^T. \tag{6.3}$$

Here the matrix $T_k^{p,r}$ is a hierarchical complementary transfer operator, which transfers fine basis functions to a set of hierarchical complementary basis functions. The remaining two blocks of the hierarchical matrix $\hat{A}^{(k)}$ can be obtained by the following relations

$$\begin{aligned} \hat{A}_{12}^{(k)} &= T_k^{p,r} A^{(k)} (G_k^{p,r})^T, \\ \hat{A}_{21}^{(k)} &= G_k^{p,r} A^{(k)} (T_k^{p,r})^T. \end{aligned} \tag{6.4}$$

Note that similar results hold for $R_k^{p,r}$.

To construct $T_k^{p,r}$ efficiently, the following points are important.

1. The basis for hierarchical complementary space should be locally supported. In other words, the block $\hat{A}_{11}$ should be sparse and nicely structured.

2. The condition number of the $\hat{A}_{11}^{(k)}$ block should be independent of mesh size.

3. The CBS constant γ, see (6.9), should be bounded away from one, i.e. the minimum generalized eigenvalue of block $\hat{A}_{22}^{(k)}$ with respect to the Schur complement should be greater than $1/4$ for $\nu = 2$ and $1/9$ for $\nu = 3$.

The construction of $T_k^{p,r}$, based on the linear combination of fine basis functions, is not unique. Based on the above mentioned guidelines, a representation of a complementary basis function should not involve several fine basis functions because it will cause less sparse structure of matrix $T_k^{p,r}$. Based on our extensive study with different choices of linear combinations satisfying the above requirements, we present two choices for $T_k^{p,r}$, for $p = 2, 3, 4$ and for extreme cases of smoothness, namely C^{p-1} and C^0.

6.1.1 C^{p-1}-continuity

First choice of $T_k^{p,r}$:

We have the following matrix representation of hierarchical complementary space for $p = 2$ with C^{p-1}-continuity, and for $k \leq L-1$.

$$T_{k+1}^{2,p-1} = \begin{bmatrix} \boxed{\begin{matrix} 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \end{matrix}} & & & \\ & \boxed{\begin{matrix} 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \end{matrix}} & & \\ & & \ddots & \ddots \\ & & \ddots & \ddots \\ & & & \boxed{\begin{matrix} 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \end{matrix}} \end{bmatrix}.$$

The above matrix has the block structure with blocks, say $M_1^{2,p-1}$. The blocks are connected in such a way that if a block ends at ith row and jth column of $T_{k+1}^{2,p-1}$ then the next block will start at $(i+1, j-1)$th position of $T_{k+1}^{2,p-1}$. In general, for $p = 2, 3, 4$, we write the following block form of $T_{k+1}^{p,p-1}$ with blocks $M_1^{p,p-1}$

$$T_{k+1}^{p,p-1} = \begin{bmatrix} M_1^{p,p-1} & & & \\ & M_1^{p,p-1} & & \\ & & \ddots & \\ & & & M_1^{p,p-1} \end{bmatrix}, \tag{6.5}$$

where

$$M_1^{2,p-1} = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \end{bmatrix},$$

$$M_1^{3,p-1} = \begin{bmatrix} 0 & -1/2 & 3/4 & -1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/2 & 3/4 & -1/2 & 0 \end{bmatrix},$$

and

$$M_1^{4,p-1} = \begin{bmatrix} 0 & 1/2 & -1 & 1 & -1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & -1 & 1 & -1/2 & 0 \end{bmatrix},$$

respectively. The blocks are connected in such a way that if a block ends at ith row and jth column of $T_{k+1}^{p,p-1}$ then the next block will start at $(i+1, j-(p-1))$th position of $T_{k+1}^{p,p-1}$.

Second choice of $T_k^{p,r}$:

For second choice we give the following block matrix. For $k \leq L-1$

$$T_{k+1}^{p,p-1} = \begin{bmatrix} M_2^{p,p-1} & & & \\ & M_2^{p,p-1} & & \\ & & \ddots & \\ & & & M_2^{p,p-1} \end{bmatrix}, \tag{6.6}$$

where the blocks $M_2^{p,p-1}$ are given by

$$M_2^{2,p-1} = \begin{bmatrix} -1/2 & 1 & -1 & 1/2 & 0 & 0 \\ 0 & 0 & -1/2 & 1 & -1 & 1/2 \end{bmatrix},$$

$$M_2^{3,p-1} = \begin{bmatrix} 1/8 & -1/2 & 3/4 & -1/2 & 1/8 & 0 & 0 \\ 0 & 0 & 1/8 & -1/2 & 3/4 & -1/2 & 1/8 \end{bmatrix},$$

and

$$M_2^{4,p-1} = \begin{bmatrix} 1/4 & 1/2 & -1 & 1 & -1/2 & -1/4 & 0 & 0 \\ 0 & 0 & 1/4 & 1/2 & -1 & 1 & -1/2 & -1/4 \end{bmatrix},$$

respectively, and the blocks are connected in a similar way as in first choice.

6.1.2 C^0-continuity

First choice of $T_k^{p,r}$:

For C^0 continuous basis functions, we give the following matrix representation of hierarchical complementary spaces. For $p = 2$, and for $k \leq L - 1$, we have

$$T_{k+1}^{2,0} = \begin{bmatrix} \boxed{\begin{matrix} 0 & 1 & -1/4 & 0 & 0 \\ 0 & 0 & 1 & -1/4 & 0 \end{matrix}} & & & \\ & \boxed{\begin{matrix} 0 & 1 & -1/4 & 0 & 0 \\ 0 & 0 & 1 & -1/4 & 0 \end{matrix}} & & \\ & & \begin{matrix} .. & .. \\ .. & .. \end{matrix} & \\ & & & \boxed{\begin{matrix} 0 & 1 & -1/4 & 0 & 0 \\ 0 & 0 & 1 & -1/4 & 0 \end{matrix}} \end{bmatrix}.$$

The above matrix has the block structure and the blocks are connected in such a way that if a block ends at ith row and jth column of $T_{k+1}^{2,0}$ then the next block will start at $(i+1, j)$th position of $T_{k+1}^{2,0}$. In general, for $p = 2, 3, 4$, we can write the following hierarchical complementary operators

$$T_{k+1}^{p,0} = \begin{bmatrix} M_1^{p,0} & & & & \\ & M_1^{p,0} & & & \\ & & .. & & \\ & & & M_1^{p,0} & \\ & & & & M_1^{p,0} \end{bmatrix}, \tag{6.7}$$

where

$$M_1^{2,0} = \begin{bmatrix} 0 & 1 & -1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/4 & 1 & 0 \end{bmatrix},$$

$$M_1^{3,0} = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & -1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \end{bmatrix},$$

and

$$M_1^{4,0} = \begin{bmatrix} 0 & -2/3 & 5/4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2/3 & 5/4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5/4 & -2/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5/4 & -2/3 & 0 \end{bmatrix},$$

respectively.

Second choice of $T_k^{p,r}$:

Another choice for $T_{k+1}^{p,r}$ (where $k \leq L-1$) for C^0 continuous basis functions is obtained by choosing the following block matrix

$$T_{k+1}^{p,0} = \begin{bmatrix} M_2^{p,0} & & & & \\ & M_2^{p,0} & & & \\ & & .. & & \\ & & & M_2^{p,0} & \\ & & & & M_2^{p,0} \end{bmatrix}, \tag{6.8}$$

where

$$M_2^{2,0} = \begin{bmatrix} -1/4 & 1 & -1/4 & 0 & 0 \\ 0 & 0 & -1/4 & 1 & -1/4 \end{bmatrix},$$

$$M_2^{3,0} = \begin{bmatrix} 0 & -1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1/4 & 1/10 & -1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & -1/2 & 0 \end{bmatrix},$$

and

$$M_2^{4,0} = \begin{bmatrix} 0 & -5/9 & 1 & -5/9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -5/9 & 1 & -5/9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -5/9 & 1 & -5/9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -5/9 & 1 & -5/9 & 0 \end{bmatrix},$$

respectively, and blocks are connected in a similar way as in first choice

Remark 6.1. *All the above operators are defined for one-dimension. The higher dimensional operators are obtained via tensor product.*

6.2 Numerical Study of CBS Constant

The construction of optimal preconditioners in the framework of AMLI methods is based upon a theory in which the constant γ in the strengthened Cauchy-Bunyakowski-Schwarz (CBS) inequality plays a key role. The CBS constant measures the cosine of the abstract angle between the coarse space and its hierarchical complementary space. The general idea is to construct a proper splitting by means of a hierarchical basis transformation.

In the hierarchical bases context we denote by V_1 and V_2 subspaces of the space V_h . The space V_2 is spanned by the coarse-space basis functions and V_1 is the complement of V_2 in V_h , i.e., V_h is a direct sum of V_1 and V_2:

$$V_h = V_1 \oplus V_2.$$

Let $v_i \in V_i, i = 1, 2$. The CBS constant measures the strength of the off-diagonal blocks in relation to the diagonal blocks (see, (6.1)) and can be defined as the minimal γ satisfying the strengthened CBS inequality

$$|v_1^T \hat{A}_{12} v_2| \leq \gamma \left\{ (v_1^T \hat{A}_{11} v_1)(v_2^T \hat{A}_{22} v_2) \right\}^{1/2}. \tag{6.9}$$

A detailed exposition of the role of this constant can be found in [50]. The CBS constant can be estimated locally also. Let us assume that

$$\hat{A} = \sum_{E \in \mathcal{E}} R_E^T A_E R_E, \quad v = \sum_{E \in \mathcal{E}} R_E^T v_E, \tag{6.10}$$

where A_E are symmetric positive semidefinite local matrices (macro element matrices), $\mathcal{E}$ is some index set, and the summation is understood as assembling. The global splitting naturally induces the two-by-two block representation of the local matrix A_E and the related vector v_E, namely,

$$A_E = \begin{bmatrix} A_{E:11} & A_{E:12} \\ A_{E:21} & A_{E:22} \end{bmatrix}, \quad v_E = \begin{bmatrix} v_{E:1} \\ v_{E:2} \end{bmatrix}.$$

The local CBS constant corresponding to A_E satisfies the following

$$|v_{E:1}^T A_{E:12} v_{E:2}| \leq \gamma_E \left\{ (v_{E:1}^T A_{E:11} v_{E:1})(v_{E:2}^T A_{E:22} v_{E:2}) \right\}^{1/2}. \tag{6.11}$$

As it is shown in [73], the relation between global γ and local γ_E is given by

$$\gamma \leq \max_{E \in \mathcal{E}} \gamma_E < 1. \tag{6.12}$$

In the framework of isogeometric analysis, the local analysis for CBS constant for C^0-continuous basis functions can be done as in finite element analysis. However, for C^{p-1}-continuous basis functions, it is not straightforward. The extended support of B-splines (NURBS) creates dimension mismatch for macro element in the fine space and its corresponding hierarchical space, i.e., for the macro element the number of basis function in fine space is not identical with the number of basis functions in its hierarchical space. As explained for $p = 2$ in Fig. 6.1 and Fig. 6.2, for macro element the total number of basis functions in fine space are 4, while in its corresponding hierarchical space the total number of basis functions are 6. For local analysis with C^{p-1} continuous basis functions further investigations are needed.

In Tables 6.1-6.4, we provide the global γ, and the Tables 6.5-6.8 the condition number of $\hat{A}_{11}$ block (corresponding to hierarchical complementary space) is presented. In Table 6.1 and Table 6.3, for $p = 4, (C^0)$, the value of γ^2 is not less than $\frac{3}{4}$, but still we obtain optimal complexity with W-cycle. The numerical results show that the condition number of $\hat{A}_{11}$ block is independent of h. In Table 6.7 and Table 6.8, the entries marked by $*$ represent the cases where the results could not be obtained due to limitation on computational resources.

Figure 6.1: B-spline basis functions for $p = 2$ on a unit interval with 8 subdivisions. The pictures from top to bottom represent basis functions at fine level, coarse level, the hierarchical complement of coarse level, and the direct sum of coarse basis function and its hierarchical complement respectively.

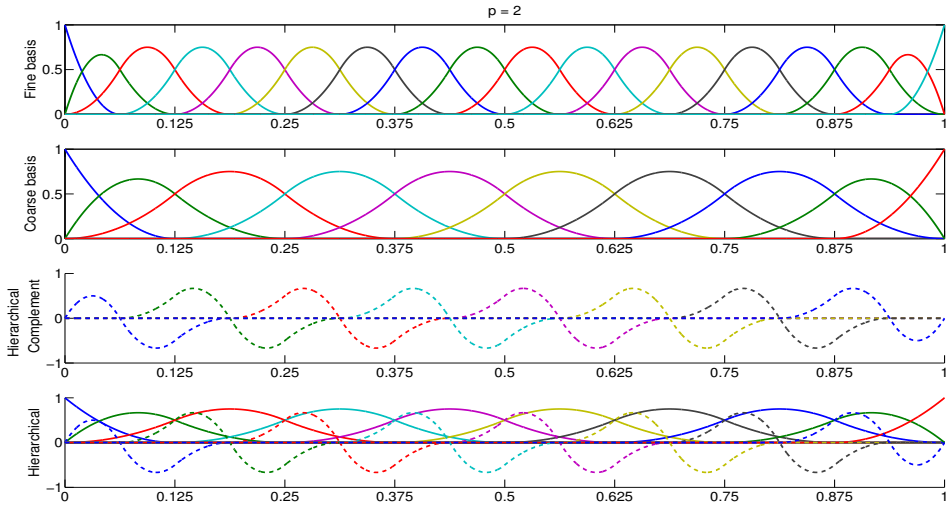


Figure 6.2: Dimension mismatch of basis function for macro element in fine space and its corresponding hierarchical space. For macro element the coarse space and its hierarchical complementary space have 3 basis functions each, which results in 6 basis function in hierarchical space. Whereas, there are 4 basis function for macro element in fine space.

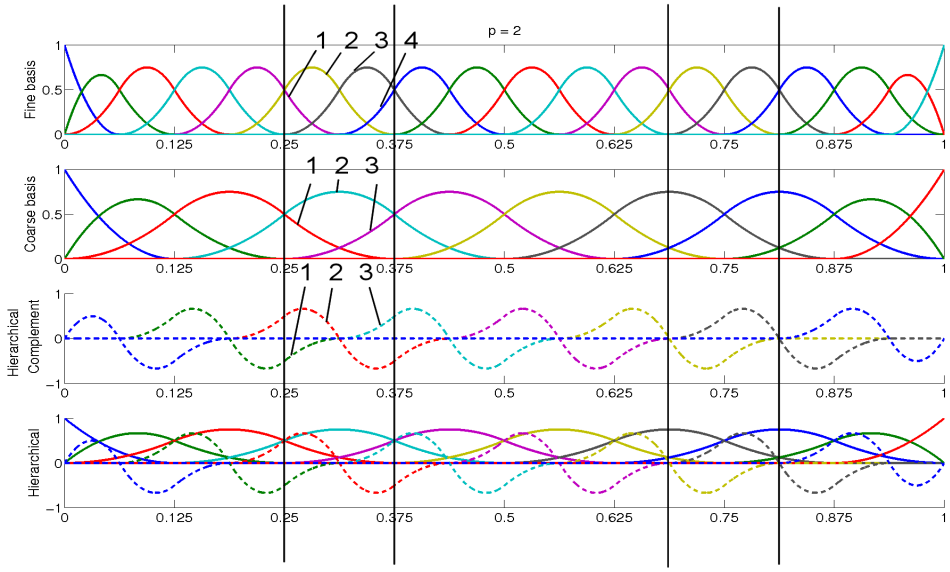


Table 6.1: γ^2 with first choice of $T_{k+1}^{p,r}$ in a square domain

$1/h$	8	16	32	64	128
$p=2,(C^0)$	0.2948	0.3169	0.3230	0.3244	0.3248
$p=2,(C^{p-1})$	0.1788	0.1872	0.1933	0.1927	0.1931
$p=3,(C^0)$	0.5580	0.5714	0.5747	0.5751	0.5757
$p=3,(C^{p-1})$	0.3585	0.3013	0.2990	0.2990	0.2990
$p=4,(C^0)$	0.7845	0.7855	0.7856	0.7857	0.7857
$p=4,(C^{p-1})$	0.5299	0.5307	0.5140	0.5138	0.5138

Table 6.2: γ^2 with second choice of $T_{k+1}^{p,r}$ in a square domain

$1/h$	8	16	32	64	128
$p=2,(C^0)$	0.2662	0.2834	0.2882	0.2894	0.2897
$p=2,(C^{p-1})$	0.0866	0.0828	0.0827	0.0827	0.0826
$p=3,(C^0)$	0.3242	0.3340	0.3363	0.3368	0.3368
$p=3,(C^{p-1})$	0.1872	0.1834	0.1791	0.1791	0.1791
$p=4,(C^0)$	0.4126	0.4198	0.4216	0.4221	0.4221
$p=4,(C^{p-1})$	0.5299	0.5307	0.5140	0.5138	0.5138

Table 6.3: γ^2 with first choice of $T_{k+1}^{p,r}$ in a quarter annulus domain

$1/h$	8	16	32	64	128
$p=2,(C^0)$	0.5236	0.5587	0.5722	0.5787	0.5823
$p=2,(C^{p-1})$	0.2803	0.2901	0.2991	0.3019	0.3048
$p=3,(C^0)$	0.6503	0.6708	0.6787	0.6825	0.6847
$p=3,(C^{p-1})$	0.4357	0.3830	0.3800	0.3779	0.3768
$p=4,(C^0)$	0.8498	0.8531	0.8549	0.8559	0.8566
$p=4,(C^{p-1})$	0.5982	0.5968	0.5792	0.5762	0.5745

Table 6.4: γ^2 with second choice of $T_{k+1}^{p,r}$ in a quarter annulus domain

$1/h$	8	16	32	64	128
$p=2,(C^0)$	0.4369	0.4670	0.4796	0.4863	0.4902
$p=2,(C^{p-1})$	0.1158	0.1128	0.1145	0.1166	0.1181
$p=3,(C^0)$	0.5211	0.5560	0.5708	0.5786	0.5830
$p=3,(C^{p-1})$	0.2910	0.2646	0.2564	0.2541	0.2530
$p=4,(C^0)$	0.5289	0.5528	0.5629	0.5681	0.5711
$p=4,(C^{p-1})$	0.5982	0.5968	0.5792	0.5762	0.5745

Table 6.5: $\kappa(\hat{A}_{11})$ with first choice of $T_{k+1}^{p,r}$ in a square domain

$1/h$	8	16	32	64	128
$p=2,(C^0)$	15.9373	16.9902	17.2737	17.3455	17.3617
$p=2,(C^{p-1})$	6.4614	6.4612	6.4589	6.4602	6.4601
$p=3,(C^0)$	49.8749	51.4403	51.8519	51.9560	51.9768
$p=3,(C^{p-1})$	24.5972	27.3123	28.4354	28.8173	28.9115
$p=4,(C^0)$	322.5053	333.7176	336.6485	336.6485	337.4931
$p=4,(C^{p-1})$	101.3294	107.7746	108.5736	110.3152	110.8763

Table 6.6: $\kappa(\hat{A}_{11})$ with first choice of $T_{k+1}^{p,r}$ in a quarter annulus domain

$1/h$	8	16	32	64	128
$p=2,(C^0)$	39.9569	45.3915	48.7833	50.9325	52.2557
$p=2,(C^{p-1})$	20.1094	22.5348	23.6238	24.1816	24.4936
$p=3,(C^0)$	143.1039	154.9739	161.5832	165.3860	167.4313
$p=3,(C^{p-1})$	57.4348	71.9216	79.5643	84.0098	86.5798
$p=4,(C^0)$	895.9249	973.3435	1007.4551	1027.7064	1041.6096
$p=4,(C^{p-1})$	220.8714	269.8018	298.3185	318.9531	331.4400

Table 6.7: $\kappa(\hat{A}_{11})$ with second choice of $T_{k+1}^{p,r}$ in a square domain

$1/h$	8	16	32	64	128
$p=2,(C^0)$	20.2427	28.8017	33.5893	34.9019	35.0668
$p=2,(C^{p-1})$	14.1704	14.9985	15.2420	15.3056	15.3183
$p=3,(C^0)$	306.1720	321.0679	325.5046	326.5145	*
$p=3,(C^{p-1})$	31.5794	42.0848	43.4486	43.6467	43.7659
$p=4,(C^0)$	1392.1584	1437.1349	1449.1122	1452.0971	*
$p=4,(C^{p-1})$	101.3294	107.6565	108.5731	110.3167	110.8768

6.3 Numerical Results for AMLI Methods

To test the performance of the AMLI methods, we consider the following test problems, whose discretizations are performed using the Matlab toolbox GeoPDEs [48, 49].

Example 6.1. Let $\Omega = (0,1)^2$. Together with $\mathcal{A} = I$, and Dirichlet boundary conditions, the right hand side function f is chosen such that the analytical solution of the problem is given by $u = e^x \sin(y)$.

Example 6.2. The domain is chosen as a quarter annulus in the first Cartesian quadrant with inner radius 1 and outer radius 2. Together with $\mathcal{A} = I$, and homogeneous Dirichlet

Table 6.8: $\kappa(\hat{A}_{11})$ with second choice of $T_{k+1}^{p,r}$ in a quarter annulus domain

$1/h$	8	16	32	64	128
$p=2,(C^0)$	39.5611	46.0474	49.6333	51.5032	52.5131
$p=2,(C^{p-1})$	43.7848	65.3521	81.3138	91.4048	98.0443
$p=3,(C^0)$	787.0425	870.8352	926.6575	965.7286	*
$p=3,(C^{p-1})$	74.8251	109.9180	127.7675	137.0648	142.2085
$p=4,(C^0)$	4161.5121	4561.4503	4751.9078	4848.1261	*
$p=4,(C^{p-1})$	220.8714	269.8018	298.3185	318.9535	331.4414

boundary conditions, the right hand side function f is chosen such that the analytic solution is given by $u = -xy^2(x^2+y^2-1)(x^2+y^2-4)$, see [48, 49].

Example 6.3. The domain is chosen as a thick quarter of a ring. Together with $\mathcal{A} = I$, and Dirichlet boundary conditions, the right hand side function f is chosen such that the analytical solution of the problem is given by $u = e^x \sin(xy)cos(z)$.

At the finest level (largest problem size), the parametric domain is divided into n equal elements in each direction. The initial guess for (iteratively) solving the linear system of equations is chosen as the zero vector. Let r_0 denote the initial residual vector and r_{it} denote the residual vector at a given PCG/FCG iteration n_{it}. The following stopping criteria is used

$$\frac{\|r_{\text{it}}\|}{\|r_0\|} \leq 10^{-8}. \tag{6.13}$$

The average convergence factor reported in the following tables is defined as $\rho = \left(\frac{\|r_{\text{it}}\|}{\|r_0\|}\right)^{1/n_{\text{it}}}$.
In the following tables, by L1, L2 and N2 we denote the linear multiplicative AMLI cycles with $\nu = 1$, $\nu = 2$ and non-linear multiplicative AMLI cycle with $\nu = 2$, respectively. By t_c, we represent the setup time, i.e., the time taken in the construction of transfer operators and generating the preconditioner for $\hat{A}_{11}$ block (for which we used the ILU(0) factorization, i.e. without any fill-in). The solver time is represented by t_s. For all the test cases we take the polynomial degree $p = 2, 3, 4$ with C^0- and C^{p-1}-continuity. Furthermore, the transfer operator $G_{k+1}^{p,r}$ is fixed and it exactly represents the coarse basis functions in the space of fine basis functions. The hierarchical complementary transfer operator $T_{k+1}^{p,r}$ are chosen in two different ways as defined in Section 6.1, see (6.5)-(6.8).
We first consider the Example 6.1 and provide t_c, t_s, n_{it} and ρ for L1-, L2-, N2- cycles with both the choices of $T_{k+1}^{p,r}$. Numerical results are presented in Tables 6.9-6.10 and Tables 6.11-6.12 for first choice and second choice of $T_{k+1}^{p,r}$, respectively .
From Tables 6.9-6.12 we observe the following:

- The number of iterations and total solution $(t_c + t_s)$ time show an h-independent convergence rates for C^{p-1}- and C^0-continuity.

Table 6.9: AMLI methods for Example 6.1, first choice of $T_{k+1}^{p,r}$ (given in (6.5)), C^{p-1} regularity

$1/h$	t_c	t_s			n_{it}			ρ		
		L1	L2	N2	L1	L2	N2	L1	L2	N2
$p=2$										
8	0.00	0.00	0.00	0.00	7	7	7	0.0641	0.0641	0.0622
16	0.00	0.00	0.01	0.01	8	7	7	0.0948	0.0966	0.0670
32	0.01	0.01	0.01	0.01	9	8	7	0.1108	0.0988	0.0672
64	0.04	0.02	0.03	0.04	9	8	7	0.1086	0.0901	0.0622
128	0.18	0.07	0.09	0.12	9	8	7	0.1166	0.0909	0.0624
256	0.72	0.25	0.30	0.41	9	8	7	0.1175	0.0879	0.0603
512	2.97	1.03	1.12	1.50	9	8	7	0.1276	0.0945	0.0620
$p=3$										
8	0.00	0.00	0.00	0.00	8	8	8	0.0901	0.0901	0.0901
16	0.01	0.01	0.01	0.01	9	9	8	0.1111	0.1129	0.0686
32	0.02	0.01	0.01	0.02	10	9	7	0.1293	0.1043	0.0577
64	0.10	0.03	0.04	0.05	10	8	7	0.1361	0.0857	0.0551
128	0.41	0.12	0.13	0.18	10	8	7	0.1369	0.0821	0.0536
256	1.76	0.48	0.46	0.63	10	8	7	0.1348	0.0794	0.0523
512	7.50	1.65	1.77	2.37	9	8	7	0.1283	0.0771	0.0511
$p=4$										
8	0.00	0.00	0.00	0.00	10	10	10	0.1139	0.1139	0.1139
16	0.01	0.01	0.01	0.01	12	12	10	0.1866	0.1882	0.1378
32	0.06	0.02	0.02	0.03	12	11	9	0.2013	0.1822	0.1100
64	0.26	0.07	0.07	0.10	12	10	9	0.2038	0.1557	0.1032
128	1.09	0.26	0.24	0.37	12	9	9	0.2028	0.1209	0.0977
256	4.57	0.98	0.88	1.21	12	9	8	0.1976	0.1182	0.0975
512	19.05	3.60	3.44	4.66	11	9	8	0.1853	0.1146	0.0930

- For C^{p-1}-continuity, the results are almost p-independent, whereas for C^0-continuity, the degree p has some effect on PCG/FCG iterations.

- For C^{p-1}-continuity, all the AMLI cycles give optimal results, and the V-cycle ($\nu = 1$) is the fastest among all. This is due to a very nice bound on γ for C^{p-1}-continuity. Therefore, in the remaining numerical computations we consider linear AMLI cycle with $\nu = 1$ and nonlinear AMLI cycle with $\nu = 2$ for C^{p-1} continuous basis functions.

- For C^0-continuity, V-cycle ($\nu = 1$) is not an optimal order method, an observation similar to standard FEM. However, for C^0-continuity, both the $\nu = 2$ cycle methods (linear and nonlinear) exhibit optimal order behavior, and nonlinear AMLI gives overall better results. Therefore, we consider only nonlinear AMLI cycle with $\nu = 2$ for

Table 6.10: AMLI methods for Example 6.1, first choice of $T_{k+1}^{p,r}$ (given in (6.7)), C^0 regularity

$1/h$	t_c	t_s			n_{it}			ρ		
		L1	L2	N2	L1	L2	N2	L1	L2	N2
$p=2$										
8	0.00	0.01	0.01	0.01	9	9	9	0.1072	0.1072	0.1072
16	0.01	0.01	0.01	0.01	11	11	9	0.1695	0.1716	0.1102
32	0.02	0.03	0.04	0.04	13	11	9	02195	0.1738	0.1110
64	0.10	0.07	0.09	0.10	14	11	9	0.2606	0.1744	0.1109
128	0.38	0.30	0.29	0.34	16	11	9	0.2973	0.1743	0.1105
256	1.65	1.25	1.04	1.23	17	11	9	0.3288	0.1736	0.1102
512	6.93	5.17	3.84	4.61	18	11	9	0.3557	0.1730	0.1100
$p=3$										
8	0.01	0.01	0.01	0.03	12	12	12	0.1999	0.1999	0.1999
16	0.02	0.03	0.04	0.03	17	17	12	0.3288	0.3305	0.2124
32	0.09	0.09	0.12	0.10	22	18	12	0.4258	0.3568	0.2129
64	0.37	0.38	0.41	0.33	27	19	12	0.5014	0.3650	0.2122
128	1.55	1.77	1.34	1.21	32	19	12	0.5581	0.3673	0.2114
256	6.73	7.87	4.88	4.51	37	19	12	0.6038	0.3670	0.2110
512	28.76	36.56	19.12	17.84	42	19	12	0.6394	0.3664	0.2108
$p=4$										
8	0.01	0.03	0.03	0.03	19	19	19	0.3631	0.3631	0.3631
16	0.05	0.07	0.10	0.09	25	26	19	0.4784	0.4827	0.3719
32	0.24	0.32	0.36	0.29	38	30	19	0.6087	0.5337	0.3720
64	1.07	1.62	1.29	1.05	52	32	19	0.6982	0.5585	0.3719
128	4.50	8.04	4.90	3.89	67	34	19	0.7585	0.5766	0.3709
256	18.73	40.11	19.41	15.25	85	35	19	0.8038	0.5827	0.3703
512	76.22	190.29	77.37	62.24	100[1]	35	19	0.8379	0.5878	0.3700

C^0 continuous basis functions in remaining numerical results.

- For $p = 4$ with C^{p-1}-continuity, we could not obtain better γ with the second choice of $T_{k+1}^{p,r}$ as compared to the first choice. Therefore, in Table 6.11, the numerical results are presented only for $p = 2, 3$ with second choice of $T_{k+1}^{p,r}$. Numerical results for $p = 4$ may be improved by choosing different operators, which demands further investigation.

- For C^{p-1}-continuity, though the number of iterations are less for second choice of $T_{k+1}^{p,r}$, the overall time $(t_c + t_s)$ is more than the first choice of $T_{k+1}^{p,r}$. This happens due to comparatively less sparse structure of second choice $T_{k+1}^{p,r}$, which results in more

[1] did not converge upto desired accuracy.

Table 6.11: AMLI methods for Example 6.1, second choice of $T_{k+1}^{p,r}$ (given in (6.6)), C^{p-1} regularity

$1/h$	t_c	t_s			n_{it}			ρ		
		L1	L2	N2	L1	L2	N2	L1	L2	N2
$p=2$										
8	0.08	0.02	0.42	0.52	5	5	5	0.0227	0.0227	0.0227
16	0.00	0.01	0.01	0.01	6	6	5	0.0304	0.0326	0.0217
32	0.02	0.01	0.01	0.01	6	6	5	0.0316	0.0311	0.0226
64	0.07	0.02	0.05	0.05	6	6	5	0.0303	0.0300	0.0224
128	0.30	0.06	0.08	0.10	6	6	5	0.0314	0.0310	0.0234
256	1.21	0.22	0.30	0.39	6	6	5	0.0301	0.0296	0.0226
512	5.18	0.88	1.05	1.62	6	6	6	0.0326	0.0321	0.0269
$p=3$										
8	0.00	0.02	0.00	0.00	7	7	7	0.0443	0.0443	0.0443
16	0.01	0.00	0.00	0.01	7	7	6	0.0560	0.0569	0.0365
32	0.04	0.01	0.01	0.02	7	7	6	0.0576	0.0494	0.0319
64	0.18	0.04	0.04	0.05	7	6	5	0.0569	0.0377	0.0216
128	0.77	0.12	0.13	0.17	7	6	5	0.0542	0.0343	0.0204
256	3.32	0.47	0.49	0.64	7	6	5	0.0502	0.0326	0.0195
512	13.99	1.60	1.89	2.44	6	6	5	0.0446	0.0311	0.0186

construction time t_c. Therefore, in the remaining numerical tests we consider only the first choice of $T_{k+1}^{p,r}$ for C^{p-1} continuous basis functions.

- For C^0-continuity, we get mixed results from both the choices of $T_{k+1}^{p,r}$. This is due to the fact that there is not much difference in number of nonzero entries in each row of $T_{k+1}^{p,r}$ for two different choices. Therefore, numerical results are provided for both the choices of $T_{k+1}^{p,r}$ for C^0 continuous basis functions.

We now consider Example 6.2 with curved boundary. The geometry for this example is represented by NURBS basis functions of order 1 in the radial direction and of order 2 in the angular direction, see [48]. Numerical results are provided for C^{p-1}-continuity with first choice of $T_{k+1}^{p,r}$ in Table 6.13, and for C^0-continuity with both the choices of $T_{k+1}^{p,r}$ in Table 6.14. All the results are qualitatively similar to that of Example 6.1 with square domain.
Finally, we consider three-dimensional problem as stated in Example 6.3. The numerical results are presented in Tables 6.15-6.16. Due to the limitation of available computer resources numerical results in three-dimensions are provided only upto $h = 1/32$. In Table 6.15, linear AMLI cycle with $\nu = 1$, and nonlinear AMLI cycle with $\nu = 2$ are given for C^{p-1} continuity with first choice of $T_{k+1}^{p,r}$. The results exhibit optimal order for both the solvers. The increased number of iterations (as compared to two-dimensional examples) can be attributed to the smaller angle between coarse space and its complementary space. For C^0-continuity the numerical results with both the choices of $T_{k+1}^{p,r}$ are given in Table 6.16.

Table 6.12: AMLI methods for Example 6.1, second choice of $T_{k+1}^{p,r}$ (given in (6.8)), C^0 regularity

$1/h$	t_c	t_s			n_{it}			ρ		
		L1	L2	N2	L1	L2	N2	L1	L2	N2
$p=2$										
8	0.00	0.00	0.00	0.00	8	8	8	0.0901	0.0901	0.0901
16	0.01	0.01	0.01	0.01	9	9	8	0.1173	0.1195	0.0918
32	0.04	0.02	0.03	0.04	10	9	8	0.1308	0.1197	0.0900
64	0.15	0.07	0.09	0.12	10	9	8	0.1430	0.1192	0.0890
128	0.62	0.27	0.32	0.41	10	9	8	0.1479	0.1191	0.0884
256	2.71	1.08	1.20	1.56	10	9	8	0.1509	0.1191	0.0880
512	11.22	4.33	4.55	6.00	10	9	8	0.1528	0.1191	0.0878
$p=3$										
8	0.01	0.01	0.03	0.03	9	9	9	0.1133	0.1133	0.1133
16	0.02	0.02	0.02	0.02	11	11	9	0.1724	0.1743	0.1191
32	0.09	0.05	0.07	0.07	13	11	9	0.2216	0.1762	0.1206
64	0.39	0.20	0.22	0.25	14	11	9	0.2627	0.1777	0.1212
128	1.62	0.88	0.79	0.91	16	11	9	0.2998	0.1782	0.1215
256	7.06	3.70	2.87	3.42	17	11	9	0.3321	0.1785	0.1216
512	29.78	16.54	11.24	13.37	19	11	9	0.3630	0.1786	0.1217
$p=4$										
8	0.01	0.02	0.02	0.02	10	10	10	0.1368	0.1368	0.1368
16	0.07	0.04	0.06	0.05	13	13	10	0.2199	0.2219	0.1419
32	0.33	0.15	0.18	0.18	15	13	10	0.2825	0.2254	0.1416
64	1.88	0.61	0.60	0.64	17	13	10	0.3259	0.2251	0.1412
128	5.92	2.51	2.20	2.41	18	13	10	0.3575	0.2247	0.1410
256	25.51	11.39	8.56	9.56	20	13	10	0.3844	0.2245	0.1408
512	104.33	49.49	35.15	39.67	21	13	10	0.4042	0.2244	0.1408

The first choice of $T_{k+1}^{p,r}$, however, does not result in an optimal order method. The optimality is restored with $\nu = 3$, which are presented in the column with $N3$. The second choice, though expensive, gives optimal order method for second order stabilization ($\nu = 2$). In Tables 6.15-6.16, The entries marked by $*$ represent the cases where the computations are performed on a machine with larger memory but shared with other users, therefore timings are not provided for these cases.

We note that for two-dimensional problems, the total time of the solvers also exhibit optimal complexity, however, for three-dimensional problem the increase in the total time $(t_c + t_s)$ for successive refinement is more than the factor of increase in number of unknowns. This is due to the construction of operators $G_{k+1}^{p,r}$ and $T_{k+1}^{p,r}$ by tensor product of matrices for one-dimensional operators (see Remark 6.1), and expensive preconditioner for $\hat{A}_{11}$ (ILU(0)).

6.4 Conclusions

We have presented AMLI methods for the linear system arising from the isogeometric discretization of the scalar second order elliptic problems. We summarize the main contribution of this chapter as follows.

1. We provide the explicit representation of B-splines as a function of mesh size h on a unit interval with uniform refinement. The explicit representation is given for C^0 and C^{p-1} continuous basis functions of polynomial degree $p = 2, 3, 4$, the most widely used cases in engineering applications. Explicit form of B-splines is important from computational point of view, as well as in forming the inter-grid transfer operators. It is intended to help the reader in writing optimized/fast computer programs.

Table 6.13: AMLI methods for Example 6.2, first choice of $T_{k+1}^{p,r}$ (given in (6.5)), C^{p-1} regularity

$1/h$	t_c	t_s		n_{it}		ρ	
		L1	N2	L1	N2	L1	N2
$p = 2$							
8	0.02	0.02	0.01	8	8	0.0802	0.0802
16	0.00	0.01	0.01	9	8	0.1201	0.0839
32	0.01	0.01	0.01	10	7	0.1499	0.0658
64	0.05	0.02	0.03	11	6	0.1838	0.0453
128	0.17	0.09	0.10	12	6	0.2048	0.0351
256	0.72	0.38	0.30	13	5	0.2211	0.0226
512	2.93	1.53	1.07	13	5	0.2374	0.0194
$p = 3$							
8	0.00	0.00	0.00	9	9	0.1201	0.1201
16	0.01	0.01	0.01	10	9	0.1560	0.1148
32	0.02	0.01	0.02	12	8	0.1839	0.0988
64	0.10	0.04	0.06	13	8	0.2104	0.0900
128	0.41	0.16	0.20	13	8	0.2363	0.0858
256	1.76	0.66	0.72	14	8	0.2514	0.0828
512	7.45	2.56	2.35	14	7	0.2644	0.0706
$p = 4$							
8	0.03	0.01	0.00	11	11	0.1686	0.1686
16	0.01	0.01	0.01	12	11	0.2073	0.1665
32	0.05	0.02	0.03	13	9	0.2419	0.1248
64	0.24	0.11	0.10	14	9	0.2549	0.1054
128	1.07	0.32	0.43	15	8	0.2688	0.0884
256	4.47	1.23	1.09	15	7	0.2924	0.0648
512	18.79	5.30	4.13	16	7	0.3061	0.0534

2. The construction of B-spline basis functions at coarse level from the linear combination of fine basis functions is provided. For $p = 2, 3, 4$, and with C^0 and C^{p-1} continuities, these transfer operators (from fine level to coarse level) are given in matrix form for a multilevel mesh. These operators can also be used to generate restriction operators in multigrid methods.

3. The transfer operators are also provided for NURBS basis functions. The formulation of NURBS operators is given in terms of B-spline operators and weights.

Table 6.14: AMLI methods for Example 6.2, C^0 regularity

with first choice of $T_{k+1}^{p,r}$ given in (6.7)					with second choice of $T_{k+1}^{p,r}$ given in (6.8)				
$1/h$	t_c	t_s	n_{it}	ρ	$1/h$	t_c	t_s	n_{it}	ρ
		N2	N2	N2			N2	N2	N2
$p = 2$					$p = 2$				
8	0.00	0.01	11	0.1744	8	0.00	0.01	10	0.1445
16	0.01	0.02	11	0.1820	16	0.01	0.02	10	0.1510
32	0.02	0.05	11	0.1791	32	0.04	0.04	10	0.1478
64	0.09	0.13	11	0.1752	64	0.15	0.15	10	0.1463
128	0.40	0.43	11	0.1730	128	0.62	0.52	10	0.1437
256	1.72	1.52	11	0.1717	256	2.65	1.93	10	0.1419
512	7.36	5.61	11	0.1704	512	11.05	7.72	10	0.1401
$p = 3$					$p = 3$				
8	0.00	0.01	13	0.2237	8	0.01	0.01	11	0.1647
16	0.02	0.04	14	0.2507	16	0.02	0.03	11	0.1780
32	0.08	0.11	14	0.2584	32	0.09	0.09	11	0.1845
64	0.34	0.39	14	0.2632	64	0.39	0.33	12	0.1883
128	1.49	1.43	14	0.2649	128	1.63	1.21	12	0.1922
256	6.35	5.37	14	0.2648	256	6.98	4.52	12	0.1938
512	27.51	20.83	14	0.2638	512	28.76	17.94	12	0.1940
$p = 4$					$p = 4$				
8	0.01	0.03	22	0.4319	8	0.01	0.02	11	0.1660
16	0.05	0.11	24	0.4516	16	0.07	0.05	11	0.1758
32	0.22	0.38	24	0.4563	32	0.32	0.19	11	0.1789
64	0.92	1.34	24	0.4591	64	1.39	0.70	11	0.1785
128	4.21	5.03	24	0.4609	128	5.99	2.64	11	0.1774
256	18.28	19.79	24	0.4639	256	25.31	10.49	11	0.1765
512	76.62	81.78	25	0.4644	512	99.22	43.15	11	0.1757

4. The construction of hierarchical spaces for B-splines (NURBS) is presented. Hierarchical spaces are constructed as direct sum of coarse spaces and corresponding hierarchical complementary spaces. We have presented matrix form of these operators.

Table 6.15: AMLI methods for Example 6.3, First choice of $T_{k+1}^{p,r}$ (given in (6.5)), C^{p-1} regularity

$1/h$	t_c	t_s		n_{it}		ρ	
		L1	N2	L1	N2	L1	N2
$p=2$							
4	0.00	0.00	0.00	8	8	0.0899	0.0899
8	0.04	0.01	0.01	12	10	0.1913	0.1438
16	0.60	0.10	0.10	13	10	0.2400	0.1484
32	7.18	1.09	0.89	15	10	0.2694	0.1346
64	*	*	*	15	9	0.2830	0.1168
$p=3$							
4	0.00	0.00	0.00	10	10	0.1415	0.1415
8	0.15	0.02	0.03	14	13	0.2492	0.2304
16	2.84	0.27	0.24	15	11	0.2923	0.1862
32	35.61	2.79	2.21	17	11	0.3215	0.1762
64	*	*	*	17	11	0.3349	0.1738
$p=4$							
4	0.01	0.01	0.01	10	10	0.1443	0.1443
8	0.52	0.06	0.07	16	16	0.3027	0.3040
16	14.81	0.82	0.85	20	17	0.3900	0.3324
32	213.74	8.82	7.55	21	15	0.4067	0.2927
64	*	*	*	21	14	0.4042	0.2546

As the choice of hierarchical complementary spaces is not unique, we have provided two different choices of these operators for each of C^0- and C^{p-1}-continuity of basis functions.

5. For a given polynomial degree p, AMLI cycles are of optimal complexity with respect to the mesh refinement. Also, for a given mesh size h, AMLI cycles are (almost) p-independent. We provided numerical results for a square domain, quarter annulus (ring), and quarter thick ring. The iteration counts, convergence factor, and timings are given for AMLI linear V-, W- and nonlinear W-cycles. Note that, for C^{p-1}-continuity the linear V-cycle also exhibits optimal convergence rates (due to very nice space splitting, which is normally not found in standard FEM). The linear and nonlinear AMLI W-cycle is optimal for all cases except for a particular case of degree $p=4$ with C^0-continuity in three-dimensional problem with first choice of $T_f^{p,r}$. For this case, the number of iterations are provided with $\nu=3$ cycle, which is optimal. The numerical results are complete for $p=2,3,4$, with C^{p-1} and C^0 continuous basis functions.

Table 6.16: AMLI methods for Example 6.3, C^0 regularity

with first choice of $T_{k+1}^{p,r}$ given in (6.7)				
$1/h$	t_c	t_s	n_{it}	ρ
		N2	N2(N3)	N2
$p=2$				
4	0.01	0.01	12 (12)	0.2124
8	0.11	0.05	15 (15)	0.2904
16	1.25	0.52	16 (15)	0.2996
32	12.06	4.51	16 (15)	0.3022
$p=3$				
4	0.07	0.04	18 (18)	0.3527
8	1.09	0.50	23 (22)	0.4408
16	12.23	5.13	26 (23)	0.4919
32	114.77	49.48	28 (23)	0.5164
$p=4$				
4	0.39	0.45	48 (48)	0.6770
8	5.84	4.36	54 (50)	0.7081
16	64.09	47.27	64 (51)	0.7497
32	*	*	73 (51)	0.7764

with second choice of $T_{k+1}^{p,r}$ given in (6.8)				
$1/h$	t_c	t_s	n_{it}	ρ
		N2	N2	N2
$p=2$				
4	0.37	0.31	11	0.1753
8	0.32	0.13	13	0.2212
16	4.29	0.76	13	0.2250
32	33.13	7.44	13	0.2261
$p=3$				
4	0.09	0.03	14	0.2663
8	1.42	0.34	16	0.3092
16	15.72	3.30	17	0.3342
32	123.05	32.24	18	0.3415
$p=4$				
4	0.98	0.23	16	0.2987
8	13.39	1.84	18	0.3465
16	144.03	17.32	18	0.3560
32	*	*	18	0.3577

Despite that the condition number of the stiffness matrix grows very rapidly with the polynomial degree, these excellent results exhibit the strength and flexibility of AMLI methods. Nevertheless, the rigorous local analysis of the CBS constant γ, particularly due to the overlapped support of B-splines, is not a straight forward task, and is still an open problem.

Chapter 7

Condition Number Study of Graph Theory Based Preconditioners for Isogeometric Discretization

We study the preconditioning of the stiffness matrix which arises from the discretization of the model problem using isogeometric method. We use graph theory based preconditioners, namely, Vaidya's preconditioners (maximum weight spanning tree) and Gremban and Miller's preconditioners (support tree). Numerical results show that these preconditioners do not perform satisfactorily for the matrices arising in isogeometric method. The purpose of this chapter is to present our study on graph theory based preconditioners towards preconditioning the linear system arising from isogeometric discretizations (2.17).

7.1 Preliminaries of Support Graph Theory

Support graph theory gives us a new class of preconditioners, called support graph preconditioners. The main idea is to use a subgraph of the graph of the coefficient matrix A as a preconditioner. These graphs are connected graphs, and the preconditioner graph edges are the subset of the actual graph edges. Moreover, the preconditioner graph edges support the actual graph edges.
We first present some definitions which are related to the graph theory.

Walk: A walk is an alternating sequence of vertices and edges that begins and end with a vertex, such that any edge in the sequence connects the vertex preceding it to the vertex following it.

Path: A path is a walk in which all the vertices are distinct.

Graph embedding: Let G and H be two graphs. An embedding of H into G is a mapping of vertices of H onto vertices of G and edges of H onto paths in G.

Dilation: The dilation of an edge e of H is the number of edges in its support path (length of support path) in G. The dilation of the embedding is the length of the longest path in G onto which an edge of H is mapped.

Congestion: The congestion of an edge e in G is the number of paths of the embedding that contain e. The congestion of the embedding is the maximum congestion of the edges in G.

Support: The support $\sigma(A, B)$ of matrix B for matrix A is

$$\min\{\tau : \tau B - A \text{ is positive semi-definite (PSD)}\}.$$

The following results [58] are of fundamental importance for graph-theory based preconditioners.

Lemma 7.1 (Support lemma)**.** *Let A and B be SPD matrices. If $\tau B - A$ is PSD, where τ is a positive number, then $\lambda_{\max}(B^{-1}A) \leq \tau$.*

The support lemma shows that the support of a matrix pair (A, B) bounds its eigenvalue, i.e.,

$$\lambda_{\max}(B^{-1}A) \leq \sigma(A, B).$$

Note that

$$1/\lambda_{\min}(B^{-1}A) = \lambda_{\max}(A^{-1}B) \leq \sigma(B, A),$$

which implies

$$\kappa(B^{-1}A) \leq \sigma(A, B).\sigma(B, A).$$

Lemma 7.2 (Splitting lemma)**.** *If $A = A_1 + A_2 + ... + A_m$, where $A_1, A_2, ..., A_m$ are all PSD matrices, then A is a PSD matrix.*

Furthermore, it can be proved that

$$\sigma(A, B) \leq \max\{\sigma(A_i, B_i)\},$$

where

$$A = \sum A_i, \quad \text{and} \quad B = \sum B_i,$$

satisfies the splitting lemma.
In other words, let $A = A_1 + A_2 + \ldots + A_m$, and $B = B_1 + B_2 + \ldots + B_m$. Assume that we have a set $\{\tau_1, \tau_2, \ldots, \tau_m\}$, such that $\tau_i B_i - A_i$ is PSD for all i. Let $\tau^* = \max\{\tau_1, \tau_2, \ldots, \tau_m\}$ then $\tau^* B_i - A_i$ is PSD for all i.

7.2 Graph Preconditioning Techniques

7.2.1 Maximum Weight Spanning Tree Preconditioners

Graph preconditioners, introduced by Vaidya [98] in early nineties, use maximum weight spanning tree (MWST) preconditioners to bound the condition number of a preconditioned system. Vaidya's method constructs a preconditioner V whose underlying graph G_V is a subgraph of G_A (graph of A). The graph G_V of the preconditioner has the same set of vertices as G_A, and a subset of the edges of G_A. The methodology of Vaidya's preconditioners is described in Algorithm 7.1. Vaidya proposed two classes of preconditioners. The first

Algorithm 7.1 Vaidya's Preconditioner

Require: G_A (Graph of A)

Ensure: Preconditioner V

1. *Find a maximum weight spanning tree T in G_A.*
2. *Decompose it into a set of k connected subgraphs $T_1, ..., T_k$, such that each T_i has roughly same number of vertices.*
3. *Form G_V by adding to T the heaviest edge between T_i and T_j for all i and j.*
4. *Add nothing if there are no edges between T_i and T_j or the heaviest edge between T_i and T_j is already in T.*
5. *The preconditioner V is the matrix whose graph is G_V.*

class, MWST preconditioners, guarantees a condition-number bound of $O(n^2)$ for any $n \times n$ sparse diagonally dominant symmetric matrix [26]. The second class of preconditioners is based on MWST augmented with a few extra edges. The cost of factoring this second class of preconditioners depends on how many edges are added to the tree. Vaidya proposed that the factorization cost can be balanced with the iteration costs, and he provided balancing guidelines for some classes of matrices. This class of preconditioners guarantees that the work in the linear solver is bounded by $O(n^{1.75})$ for any sparse diagonally dominant matrix, and by $O(n^{1.2})$ for sparse diagonally dominant matrices whose underlying graphs are planar. We construct MWST by simply dropping some off-diagonal non-zeros from A and modifying the diagonal elements to maintain certain row-sum property. We construct the preconditioners in the following ways.

1. Algorithm 7.2 constructs MWST with maximum value (not absolute maximum) and makes the matrix of MWST diagonally dominant. Note that the preconditioner matrix is diagonally dominant but not an M-matrix.

2. Algorithm 7.3 first transforms the A matrix into an M-matrix, and then finds the MWST. So the resulting preconditioner is diagonally dominant as well as M-matrix.

3. Algorithm 7.4 first drops all the positive off-diagonal entries, and then constructs MWST of modified stiffness matrix. To maintain the row-sum property it performs a weighted distribution of the sum of dropped positive off-diagonal entries and dropped

negative off-diagonal entries (while forming MWST) on the negative off-diagonal entries of MWST. In a variant of Algorithm 7.4 we make the following changes. If the sum of dropped positive off-diagonal entries and dropped negative off-diagonal entries (while forming MWST) is positive then it is added to the diagonal entry, otherwise the algorithm performs a weighted distribution of the sum on the negative off-diagonal entries of MWST. Note that in the variant of Algorithm 7.4 the preconditioned matrix is not symmetric.

4. Algorithm 7.5 first constructs MWST with maximum value (not absolute maximum), and calculates the sum of the dropped positive and negative off-diagonal entries. If this sum is positive it performs a weighted distribution of this sum on the negative off-diagonal entries of the tree, and if this sum is negative then it adds the sum to the diagonal entry of MWST.

Algorithm 7.2 MWST preconditioner (Diagonally Dominant)

Require: Stiffness Matrix A

Ensure: Preconditioner B_1

1. *Find maximum weight spanning tree B_{mwst} with maximum value (NOT absolute maximum).*
2. ***if*** *B_{mwst} is diagonally dominant* ***then***
3. $B_1 = B_{mwst}$ *(output matrix).*
4. ***else***
5. *find $s_i = \sum_{i \neq j} b_{ij}$ for each row i of B_{mwst}.*
6. ***if*** $b_{ii} < s_i$ ***then***
7. $b_{ii} = s_i$.
8. *The resulting matrix is B_1.*

Algorithm 7.3 MWST preconditioner (Diagonally dominant M-matrix)

Require: Stiffness Matrix A

Ensure: Preconditioner B_2

1. ***for*** $i = 1$ *to # of rows in A* ***do***
2. ***for*** $j = 1$ *to # of columns in A* ***do***
3. ***if*** $a_{ij} > 0$ ***then***
4. $a_{ii} = a_{ii} + a_{ij}$
5. $a_{ij} = 0$
6. *From step (1), we get an M-matrix, say A_m.*
7. *Find maximum weight spanning tree B_{mwst} with absolute maximum value of A_m.*
8. *The resulting matrix B_{mwst} is our preconditioner matrix B_2.*

Some numerical results using Vaidya's preconditioners and support tree preconditioners (see Section 7.2.2) are given in Table 7.1 using linear basis functions. Here M_w denotes the MWST preconditioner matrix, P_V denotes the preconditioner from Vaidya's approach, k

Algorithm 7.4 Approximation by M-matrices (a), note the comments for the variant

Require: Stiffness matrix A
Ensure: Matrix B_3

1. ***for*** $i = 1$ *to* $\#$ *of rows in* A ***do***
2. $\quad s_p(i) = 0$
3. $\quad$ ***for*** $j = 1$ *to* $\#$ *of columns in* A ***do***
4. $\quad\quad$ ***if*** $i \neq j$ ***then***
5. $\quad\quad\quad$ ***if*** $a_{ij} > 0$ ***then***
6. $\quad\quad\quad\quad s_p(i) = s_p(i) + a_{ij}$
7. $\quad\quad\quad\quad a_{ij} = 0$
8. *The resulting matrix is M-matrix, say* A_m.
9. *Find MWST of* A_m.
10. ***for*** $i = 1$ *to* $\#$ *of rows in* A_m ***do***
11. $\quad s_n(i) =$ *sum of row-wise dropped negative entries while finding MWST* A_m
12. $\quad s_{n(A_m)}(i) =$ *sum of off-diagonal entries of* i^{th} *row of* A_m
13. $\quad S(i) = s_p(i) + s_n(i)$
14. $\quad$ ***if*** $S(i) > 0$ *{*< 0 *for the variant}* ***then***
15. $\quad\quad$ ***for*** $k = 1$ *to* $\#$ *of rows in* A_m ***do***
16. $\quad\quad\quad$ ***if*** $i \neq k$ ***then***
17. $\quad\quad\quad\quad a^m_{ik} = a^m_{ik} + \{S(i) \times |\frac{a^m_{ik}}{s_{n(A_m)(i)}}|\}$
18. $\quad$ ***else***
19. $\quad\quad a^m_{ii} = a^m_{ii} + S(i)$
20. *The output matrix is* B_3 *{*B^v_3 *for the variant}*

Algorithm 7.5 Approximation by M-matrices (b)

Require: Stiffness matrix A
Ensure: Matrix B_4

1. *Find MWST of* A, *say* A_T.
2. ***for*** $i = 1$ *to* $\#$ *of rows in* A_T ***do***
3. $\quad s_p(i) =$ *sum of row-wise dropped positive entries while finding MWST* A_T
4. $\quad s_n(i) =$ *sum of row-wise dropped negative entries while finding MWST* A_T
5. $\quad s_{n(A_T)}(i) =$ *sum of off-diagonal entries of* i^{th} *row of* A_T
6. $\quad S(i) = s_p(i) + s_n(i)$
7. $\quad$ ***if*** $S(i) > 0$ ***then***
8. $\quad\quad$ ***for*** $k = 1$ *to* $\#$ *of rows in* A_T ***do***
9. $\quad\quad\quad$ ***if*** $i \neq k$ ***then***
10. $\quad\quad\quad\quad a^T_{ik} = a^T_{ik} + \{S(i) \times |\frac{a^T_{ik}}{s_{n(A_T)(i)}}|\}$
11. $\quad$ ***else***
12. $\quad\quad a^T_{ii} = a^T_{ii} + S(i)$
13. *The output matrix is* B_4

denotes the number of added extra edges, and P_T denotes the preconditioner from support tree approach.

Table 7.1: Vaidya's preconditioning of A (linear basis function)

mesh	4×4	8×8	16×16	32×32	64×64
$\kappa(A)$	3.15	12.82	51.71	207.34	829.85
$\kappa(M_w^{-1}A)$	2.75	10.73	45.80	184.08	737.79
$\kappa(P_V^{-1}A), k = (n/log_{10}n)$	1.00	5.54	13.43	53.15	158.17
$\kappa(P_V^{-1}A), k = (n/2)$	1.91	5.54	11.71	20.47	46.23
$\kappa(P_T^{-1}A)$	3.04	12.77	51.70	207.33	829.85

7.2.2 Support Tree Preconditioners

Gremban and Miller extended Vaidya's work, and introduced support tree preconditioners. A support tree preconditioner is a preconditioning matrix whose associated graph is a tree on a superset of the vertices. The tree is constructed to provide good support for coefficient matrix. Each vertex in the graph of the matrix becomes a leaf of the support tree, and the internal vertices of the tree are added vertices. The algorithmic way of constructing support tree preconditioners is given in Algorithm 7.6.

Algorithm 7.6 Support Tree Preconditioner

Require: G_A (Graph of A)
Ensure: Preconditioner V

1. *Find a small edge separator that gives a balanced cut.*
2. *Assign the root of the tree to this separator.*
3. *Recursively build support trees for the components left after removing the separator.*
4. *Add edges from the root of the subtrees of the components to the root of the tree.*

The construction of support tree is based on hierarchical decomposition of the graph G_A. The algorithm removes from G_A a set of edges, known as a separator, that breaks it into a small number of subgraphs $G_1, G_2, \ldots, G_k$. The algorithm then recursively partitions each G_i until the graph is decomposed into single vertices. The resulting support tree T of the graph G_A can not be used as preconditioner because number of vertices in T are more than the number of vertices in G_A. Now we perform Gaussian elimination on

$$T = \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix}$$

to eliminate all the internal vertices, referred by subscript 2, here block B_{11} is a square matrix of the same order as A, say n, block B_{22} is a square matrix whose order is the number of extra added (internal) vertices in the support tree, say m, and block B_{12} is a rectangular

matrix of order $n \times m$. Eliminating the internal vertices of T produces a matrix T' defined as

$$T' = \begin{bmatrix} B & 0 \\ 0 & D \end{bmatrix},$$

where B is the Schur compliment $B_{11} - B_{12}B_{22}^{-1}B_{12}^{T}$, and D is a diagonal matrix. Gremban and Miller used this matrix B as a preconditioner of A. The condition number of support tree preconditioned matrix is of $O(n \log n)$. Unfortunately, involvement of matrix B_{22}^{-1} seriously limits the advantages of these preconditioners, and does not offer a significant gain in the condition number of preconditioned system $(B^{-1}A)$, see last row of Table 7.1.

7.3 Local Preconditioning

To study the effect of local preconditioning (element level), we now consider the symmetric positive semi-definite (SPSD) stiffness matrix A_e of one element (without imposing the boundary condition) but with increasing p. We study the effective condition number κ_{eff} of the generalized eigenvalue problem

$$\tilde{A}_e x = \lambda \tilde{B}_e x \tag{7.1}$$

where $\tilde{A}_e = P_e^T A_e P_e$, $\tilde{B}_e = P_e^T B_e P_e$, B_e is some matrix obtained from A_e, and the matrix $P_e \in \mathbb{R}^{n\times(n-1)}$, defined as

$$P_e = \begin{pmatrix} 1 & & & & & & \\ -1 & 1 & & & & & \\ & -1 & 1 & & & & \\ & & \cdots & \cdots & \cdots & & \\ & & & & -1 & 1 & \\ & & & & & -1 & \end{pmatrix}, \tag{7.2}$$

is the transformation matrix to eliminate the zero eigenvalue.

7.3.1 M-matrix Approach

We approximate the SPSD matrix A_e by an M-matrix B_e which is obtained from A_e by the following approaches.

1. Algorithm 7.7 drops all positive off-diagonal entries. Note that the resulting preconditioned matrix in this approach is not positive definite.

2. Algorithm 7.8 adds all positive off-diagonal entries to the diagonal and replaces them by zero.

3. Algorithm 7.9 drops all positive off-diagonal entries and performs a weighted distribution of the sum of dropped positive off-diagonal entries on the negative off-diagonal entries (row-wise). Note that the resulting preconditioned matrix is not symmetric.

Algorithm 7.7 Approximation by M-matrices (c)

Require: Element stiffness matrix A_e
Ensure: Matrix B_e
1. ***for*** $i = 1$ *to* $\#$ *of rows in* A_e ***do***
2. ***for*** $j = 1$ *to* $\#$ *of columns in* A_e ***do***
3. ***if*** $i \neq j$ ***then***
4. ***if*** $a_{ij}^e > 0$ ***then***
5. $a_{ij}^e = 0$
6. *The output matrix is* B_e.

Algorithm 7.8 Approximation by M-matrices (d)

Require: Element stiffness matrix A_e
Ensure: Matrix B_e
1. ***for*** $i = 1$ *to* $\#$ *of rows in* A_e ***do***
2. ***for*** $j = 1$ *to* $\#$ *of columns in* A_e ***do***
3. ***if*** $a_{ij}^e > 0$ ***then***
4. $a_{ii}^e = a_{ii}^e + a_{ij}^e$
5. $a_{ij}^e = 0$
6. *The output matrix is* B_e.

Algorithm 7.9 Approximation by M-matrices (e)

Require: Element stiffness matrix A_e
Ensure: Matrix B_e
1. ***for*** $i = 1$ *to* $\#$ *of rows in* A_e ***do***
2. $s_p(i) = 0, s_n(i) = 0$
3. ***for*** $j = 1$ *to* $\#$ *of columns in* A_e ***do***
4. ***if*** $i \neq j$ ***then***
5. ***if*** $a_{ij}^e > 0$ ***then***
6. $s_p(i) = s_p(i) + a_{ij}^e$
7. $a_{ij}^e = 0$
8. ***if*** $a_{ij}^e < 0$ ***then***
9. $s_n(i) = s_n(i) + a_{ij}^e$
10. ***for*** $k = 1$ *to* $\#$ *of rows in* A_e ***do***
11. ***if*** $i \neq k$ ***then***
12. $a_{ik} = a_{ik} + \{s_p(i) \times |\frac{a_{ik}}{s_n(i)}|\}$
13. *The output matrix is* B_e.

7.3.2 Two-colors Approach

Given a mesh, multi-coloring consists of assigning a color to each point so that the couplings between two points of the same color are eliminated in the discretization matrix. We color the graph of stiffness matrix by two colors such that each color forms a disjoint set. This can also be done by reordering the matrix in such a way that the diagonal blocks of the matrix form two disjoint sets of nodes. Suppose we divide the nodes of the graph of stiffness matrix into two disjoint sets, say S_1 and S_2. Then

$$A_e = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

where block A_{11} is the matrix of connections within S_1, block A_{22} is the matrix of connections within S_2, block A_{12} is the matrix of connections of S_1 to S_2, and block A_{21} is the matrix of connections of S_2 to S_1. We denote the modified matrix by B_e such that

$$B_e = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix},$$

where B_{11}, B_{12}, B_{21}, and B_{22} are the modified versions of A_{11}, A_{12}, A_{21}, and A_{22}, respectively.
The following three algorithms modify A_{11} to a diagonal matrix. The other three blocks A_{12}, A_{21} and A_{22} are returned unchanged.

1. Algorithm 7.10 modifies A_{11} by simply dropping all off-diagonal entries.
2. Algorithm 7.11 drops all off-diagonal entries and the diagonal entries are the row-wise sum of A_{11}.
3. Algorithm 7.12 drops all off-diagonal entries and the diagonal entries are the row-wise absolute sum of A_{11}.

Algorithm 7.10 Two Colors Approaches (a)

Require: Element stiffness matrix A_e
Ensure: Matrix B_e
1. ***for*** $i = 1$ *to* $\#$ *of rows in* A_{11} ***do***
2. ***for*** $j = 1$ *to* $\#$ *of columns in* A_{11} ***do***
3. $B_{11}(i, i) = A_{11}(i, i)$
4. *The other blocks of* B_e *are same as the blocks of* A_e *respectively*
5. *The output matrix is* B_e

In the following three approaches the algorithms modify the blocks A_{11} and A_{12}, and return the remaining two blocks A_{21} and A_{22} unchanged.

1. Algorithm 7.13 drops all off-diagonal entries of A_{11}, and performs a row-wise weighted distribution of the sum of dropped entries of A_{11} in A_{12}.

Algorithm 7.11 Two Colors Approaches (b)

Require: Element stiffness matrix A_e
Ensure: Matrix B_e
1. ***for*** $i = 1$ *to* $\#$ *of rows in* A_{11} ***do***
2. $\quad s(i) = 0$
3. $\quad$ ***for*** $j = 1$ *to* $\#$ *of columns in* A_{11} ***do***
4. $\quad\quad s(i) = A_{11}(i, j) + s(i)$
5. $\quad B_{11}(i, i) = s(i)$
6. *The other blocks of* B_e *are same as the blocks of* A_e *respectively*
7. *The output matrix is* B_e

Algorithm 7.12 Two Colors Approaches (c)

Require: Element stiffness matrix A_e
Ensure: Matrix B_e
1. ***for*** $i = 1$ *to* $\#$ *of rows in* A_{11} ***do***
2. $\quad s(i) = 0$
3. $\quad$ ***for*** $j = 1$ *to* $\#$ *of columns in* A_{11} ***do***
4. $\quad\quad s(i) = abs(A_{11}(i, j)) + s(i)$
5. $\quad B_{11}(i, i) = s(i)$
6. *The other blocks of* B_e *are same as the blocks of* A_e *respectively*
7. *The output matrix is* B_e

2. Algorithm 7.14 drops all off-diagonal entries of A_{11}, adds the row-wise sum of positive off-diagonal entries of A_{11} to its diagonal, and performs a row-wise weighted distribution of the sum of the remaining negative entries of A_{11} in A_{12}. We make the following changes in a variant of Algorithm 7.14, drop all off-diagonal entries of A_{11}, adds the row-wise sum of negative off-diagonal entries of A_{11} to its diagonal, and performs a row-wise weighted distribution of the sum of the remaining positive entries of A_{11} in A_{12}.

The last two approaches are as follows.

- In Algorithms 7.10–7.14 the block B_{11} is a diagonal matrix. Algorithm 7.15 returns a non-diagonal block B_{11} by adding the sum of positive entries to the diagonal for each row in A_{11}, and keeping the negative entries untouched. The resulting matrix B_{11} is positive definite.
- In Algorithms 7.13–7.15 the preconditioning matrix B is non-symmetric. Algorithm 7.16 returns a symmetric matrix by performing the following steps:
  - The absolute row-sum of A_{11} forms B_{11}.
  - To maintain row-sum property, the added quantity is distributed (in a weighted sense) to the block A_{12} to get B_{12}

Algorithm 7.13 Two Colors Approaches (d)

Require: Element stiffness matrix A_e
Ensure: Matrix B_e
1. ***for*** $i = 1$ *to* $\#$ *of rows in* A_{11} ***do***
2. $\quad s_1(i) = 0$
3. $\quad$ ***for*** $j = 1$ *to* $\#$ *of columns in* A_{11} ***do***
4. $\quad\quad$ ***if*** $i \neq j$ ***then***
5. $\quad\quad\quad s_1(i) = A_{11}(i,j) + s_1(i)$
6. $\quad\quad B_{11}(i,i) = A_{11}(i,i)$
7. ***for*** $i = 1$ *to* $\#$ *of rows in* A_{12} ***do***
8. $\quad s_2(i) = 0$
9. $\quad$ ***for*** $j = 1$ *to* $\#$ *of columns in* A_{12} ***do***
10. $\quad\quad$ ***if*** $i \neq j$ ***then***
11. $\quad\quad\quad s_2(i) = A_{12}(i,j) + s_2(i)$
12. $\quad$ ***for*** $k = 1$ *to* $\#$ *of columns in* A_{12} ***do***
13. $\quad\quad B_{12}(i,k) = A_{12}(i,k) + \{s_1(i) \times |\frac{A_{12}(i,k)}{s_2(i)}|\}$
14. *The other two blocks* B_{21}, B_{22} *are same as the blocks of* A_e *respectively*
15. *The output matrix is* B_e

Algorithm 7.14 Two Colors Approaches (e), note the comments for the variant

Require: Element stiffness matrix A_e
Ensure: Matrix B_e
1. ***for*** $i = 1$ *to* $\#$ *of rows in* A_{11} ***do***
2. $\quad s_1(i) = 0, s_2(i) = 0$
3. $\quad$ ***for*** $j = 1$ *to* $\#$ *of columns in* A_{11} ***do***
4. $\quad\quad$ ***if*** $i \neq j$ ***then***
5. $\quad\quad\quad$ ***if*** $A_{11}(i,j) > 0$ ***then***
6. $\quad\quad\quad\quad s_1(i) = A_{11}(i,j) + s_1(i)$
7. $\quad\quad\quad$ ***else***
8. $\quad\quad\quad\quad s_2(i) = A_{11}(i,j) + s_2(i)$
9. $\quad\quad B_{11}(i,i) = A_{11}(i,i) + s_1(i)$ *{*$s_2(i)$ *for the variant}*
10. ***for*** $i = 1$ *to* $\#$ *of rows in* A_{12} ***do***
11. $\quad s_3(i) = 0$
12. $\quad$ ***for*** $j = 1$ *to* $\#$ *of columns in* A_{12} ***do***
13. $\quad\quad s_3(i) = A_{12}(i,j) + s_3(i)$
14. $\quad$ ***for*** $k = 1$ *to* $\#$ *of columns in* A_{12} ***do***
15. $\quad\quad B_{12}(i,k) = A_{12}(i,k) + [s_2(i)$ *{*$s_1(i)$ *for the variant}* $\times |\frac{A_{12}(i,k)}{s_3(i)}|]$
16. B_{21} *and* B_{22} *are same as* A_{21} *and* A_{22} *respectively*
17. *The output matrix is* B_e

– Transpose of B_{12} forms B_{21}. Again, to maintain row-sum property, the added quantity in B_{21} is added to the diagonal entry of A_{22} to get B_{22}.

Algorithm 7.15 Two Colors Approaches (f)

Require: Element stiffness matrix A_e
Ensure: Matrix B_e
1. ***for*** $i = 1$ *to* # *of rows in* A_{11} ***do***
2. $s(i) = 0$
3. ***for*** $j = 1$ *to* # *of columns in* A_{11} ***do***
4. ***if*** $i \neq j$ ***then***
5. ***if*** $A_{11}(i, j) > 0$ ***then***
6. $s(i) = A_{11}(i, j) + s(i)$
7. $A_{11}(i, j) = 0$
8. $B_{11}(i, j) = A_{11}(i, j)$
9. $B_{11}(i, j) = B_{11}(i, j) + s(i)$
10. *The other blocks of* B_e *are same as the blocks of* A_e *respectively*
11. *The output matrix is* B_e

7.4 Numerical Results

For our numerical tests we consider the problem (2.10) with domain $\Omega = (0, 1)^2$, $u = 0$ on boundary, and $\mathcal{A} = \mathcal{I}$. In Figure 7.1, we first depict the sparsity pattern of the coefficient matrices using fourth degree NURBS polynomials, with C^0 and C^{p-1} continuity, with uniform knot vectors in a single patch of 4×4 mesh and 16×16 mesh. Clearly, the block structure is reduced for higher continuity and the band is denser, however, the nonzero size is reduced due to the reduced problem size.
The condition number of the coefficient matrix with different mesh sizes is shown in Table 7.2 for C^0- and C^{p-1}-continuity. Note that, in the following tables $\times$ signifies that the particular case could not be computed (due to limitations of the corresponding algorithm), and $-$ signifies that the corresponding entry is not computed because the preceding results did not show any improvement.

7.4.1 Preconditioning of the Stiffness Matrix

Since the qualitative behavior of the numerical results does not essentially change for $p \geq 5$, to avoid proliferation we omit the preconditioning results for $p \geq 5$ in all cases. In Table 7.3 we present the results for the condition number of the preconditioned system $P^{-1}A$, where P is the preconditioning matrix of A. The preconditioner P is obtained from various approaches listed in the algorithms, and the Jacobi preconditioner D (i.e. the diagonal of the matrix A). Since the preconditioned system of Algorithm 7.5 is singular, its results are not shown here. In Table 7.4 the condition number of the preconditioned system $P^{-1}A$ are

Algorithm 7.16 Two Colors Approaches (g)

Require: Element stiffness matrix A_e
Ensure: Matrix B_e

for $i = 1$ *to* $\#$ *of rows in* A_{11} ***do***
  $s_1(i) = 0, s_2(i) = 0$
  for $j = 1$ *to* $\#$ *of columns in* A_{11} ***do***
    $s_1(i) = s_1(i) + abs(A_{11}(i,j))$
    if $A_{11}(i,j) < 0$ ***then***
      $s_2(i) = -2 \times abs(A_{11}(i,j)) + s_2(i)$
  $B_{11}(i,i) = s_1(i)$
for $i = 1$ *to* $\#$ *of rows in* A_{12} ***do***
  $s_3(i) = 0$
  for $j = 1$ *to* $\#$ *of columns in* A_{12} ***do***
    $s_3(i) = A_{12}(i,j) + s_3(i)$
  for $k = 1$ *to* $\#$ *of columns in* A_{12} ***do***
    $B_{12}(i,k) = A_{12}(i,k) + \{s_2(i) \times |\frac{A_{12}(i,k)}{s_3(i)}|\}$
$B_{21} = B_{12}^T$
Let the resulting matrix after modifying the blocks $A_{11}, A_{12},$ *and* A_{21} *be* E
for $i = (\#$ *of rows in* $B_{11} + 1)$ *to* $\#$ *of rows in* E ***do***
  $s_4(i) = 0$
  for $j = 1$ *to* $\#$ *of columns in* E ***do***
    $s_4(i) = s_4(i) + E(i,j)$
for $i = 1$ *to* $\#$ *of rows in* A_{22} ***do***
  $B_{22}(i,i) = A_{22}(i,i) - s_4(i + \#$ *of rows in* $B_{11})$
The output matrix is B_e

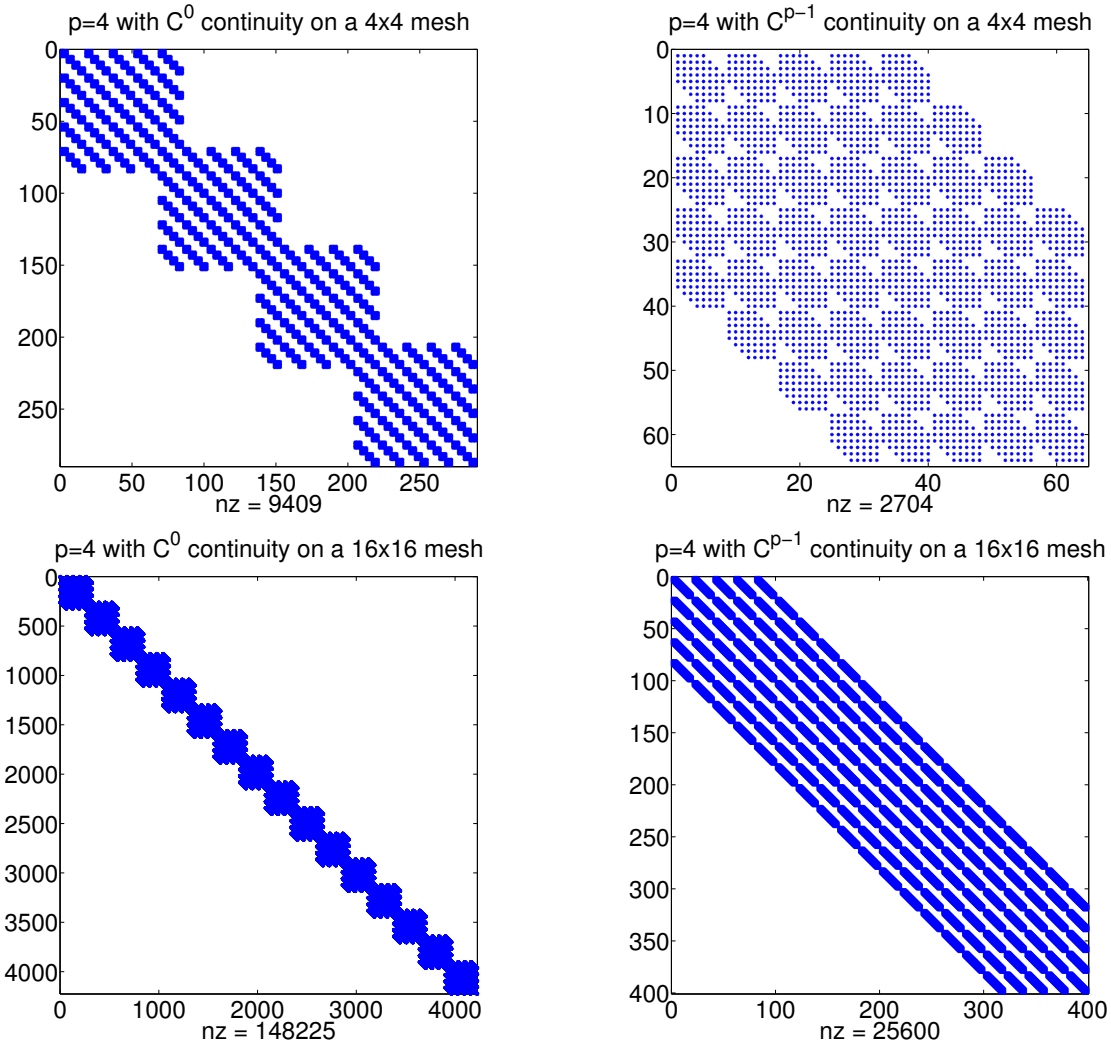
shown for C^{p-1} continuity.

7.4.2 Local Preconditioning

We now present the effective condition number of the generalized eigenvalue problems (7.1) in Table 7.5, where B_e is computed from Algorithm 7.8. Recall that the preconditioned matrix in Algorithm 7.7 is not positive definite, and in Algorithm 7.9 it is not symmetric, therefore, corresponding results are not shown here.
The observation from the two-colors approach discussed in Section 7.3.2 are as follows.

- The resulting preconditioned system from Algorithm 7.11 is indefinite because in the matrix A_{11} the negative off-diagonal entries are dominating the diagonal entry.
- The resulting preconditioned system from Algorithms 7.12 and 7.15 are positive definite but the effective condition number of the preconditioned system is very large.
- The Algorithms 7.10, 7.13 and 7.14 give indefinite preconditioned systems.

Figure 7.1: Sparsity pattern of A



- The resulting preconditioning system in Algorithm 7.16 is SPD but the effective condition number of the preconditioned system is very large.

7.4.3 Preconditioning of the Schur Complement Matrix

In this section we present the condition number of Schur complement S (obtained by eliminating all degrees of freedom inside the elements) and the preconditioned Schur complement system.
The condition number of the Schur Complement for C^0-continuity is shown in Table 7.6. In Table 7.7, results are shown for the condition number of the preconditioned system $P_S^{-1}S$, where P_S is the preconditioning matrix obtained from various approaches listed in the algorithms, and the diagonal matrix obtained from S.
The eigenvalues of the generalized eigenvalue problems $\tilde{S}_e x = \lambda \tilde{Bs}_e x$ are calculated, and

Table 7.2: Condition number of A

p \ mesh	2×2	4×4	8×8	16×16	32×32
C^0-continuity					
2	7	13	36	140	555
3	75	107	120	270	1075
4	881	1100	1189	1215	1762
5	11094	12951	13680	13886	13940
6	146296	162488	168707	170451	170900
C^{p-1}-continuity					
2	4	4	5	20	78
3	31	29	29	29	82
4	340	269	240	223	215
5	4177	3220	2148	1812	1700
6	54385	46606	20819	15133	–

the effective condition number κ_{eff} of (S_e, Bs_e) is shown in Table 7.8, where Bs_e is computed from Algorithm 7.8. Algorithms 7.4 and 7.5 have also been tried but results are not favorable and not shown here.

Table 7.3: Condition number of $P^{-1}A$ for C^0-continuity

p \ mesh	2×2	4×4	8×8	16×16	32×32
$P = D$					
2	2	6	22	87	346
3	20	30	48	192	766
4	181	249	275	331	1323
$P = B_1$, see Algorithm 7.2					
2	2	6	23	92	366
3	5	12	48	190	761
4	50	60	127	524	–
$P = B_2$, see Algorithm 7.3					
2	×	4	16	66	280
3	20	28	32	98	–
4	220	299	325	336	–
$P = B_3$, see Algorithm 7.4					
2	×	16	155	979	9409
3	18	388	4525	22757	–
4	171	3259	41763	–	–

Table 7.4: Condition number of $P^{-1}A$ for C^{p-1}-continuity

mesh / p	2×2	4×4	8×8	16×16	32×32
$P = B_1$, see Algorithm 7.2					
2	2	3	6	21	84
3	9	9	13	19	80
4	89	45	105	257	1056
$P = B_2$, see Algorithm 7.3					
2	×	4	4	17	68
3	×	22	20	20	53
4	228	185	123	116	113

Table 7.5: Effective condition number of (A_e, B_e) for $p = 2, \ldots, 6$

B_e / p	2	3	4	5	6
Alg. 7.8	2.4	25.8	246.5	2336	23999

Table 7.6: Condition number of S

mesh / p	2×2	4×4	8×8	16×16	32×32
2	3	9	28	106	417
3	9	14	37	147	585
4	29	40	48	178	706
5	100	123	130	205	815
6	355	410	430	436	951

7.5 Conclusion

We have numerically studied the behavior of condition numbers of graph theory based preconditioners for NURBS-based isogeometric discretization of Poisson equation. It can be seen that the condition number of preconditioned system, based on both the types of Vaidya's preconditioners, as well as support tree preconditioners of Gremban and Miller, is not independent of mesh-size. It is important to note that the support graph preconditioners are sensitive to the nonzero structure of the coefficient matrix, and not to the values of its entries. However, these methods are limited to the class of Symmetric Diagonally Dominant (SDD) matrices. Unfortunately, except for $p = 1$ when we get the stiffness matrix as an M-matrix, the matrices obtained from isogeometric method for $p \geq 2$ can not be guaranteed to be SDD. We tried a few approaches (discussed in earlier sections) for higher p using support graph preconditioners but in the current form these approaches do not give condition num-

Table 7.7: Condition number of $P_S^{-1}S$

p \ mesh	2×2	4×4	8×8	16×16	32×32
$P_S = D_S$					
2	2	5	19	73	291
3	7	10	33	132	529
4	20	29	47	181	720
$P_S = B_{S_1}$, see Algorithm 7.2					
2	2	5	20	77	307
3	3	8	29	115	457
4	5	12	45	176	703
$P_S = B_{S_2}$, see Algorithm 7.3					
2	×	4	14	59	241
3	5	6	19	76	303
4	18	24	26	99	398

Table 7.8: Effective condition number of (S_e, B_{S_e}) for $p = 2, \ldots, 6$

B_{S_e} \ p	2	3	4	5	6
Alg. 7.8	1.3	5.8	20.1	64.2	203.2

bers independent of mesh-size. Therefore, it is not possible to develop optimal order solver with these preconditioners.

Chapter 8

Conclusions

There are two main topics covered in this thesis. The first one deals with the condition number estimates of the matrices arising in the isogeometric discretization of the elliptic partial differential equations. The second one is the fast iterative solvers for the resulting linear system of equations. Our approaches to solve linear system are based on multigrid and multilevel methods. We also discuss the preconditioning techniques based upon graph theory. We can summarize the contribution of this thesis as follows.

Estimates of Eigenvalues

The estimates of maximum and minimum eigenvalues for the stiffness matrix and the mass matrix are given for h-refinement and p-refinement. For h-refinement, upper bounds and lower bounds for maximum and minimum eigenvalues are presented for the stiffness matrix and the mass matrix. For p-refinement, we present the upper and lower bounds for maximum eigenvalue of the stiffness and the mass matrix. For minimum eigenvalue of the stiffness matrix and the mass matrix, we provided only lower bounds.

Estimates of Condition Number

The condition number of the stiffness matrix with respect to mesh size h grows as $\mathcal{O}(h^{-2})$. For the mass matrix the condition number is independent of the mesh size h and scales as $\mathcal{O}(1)$. We have proved that in the p-version of the isogeometric method, the condition number of the stiffness matrix and the mass matrix exponentially grows with p and is bounded above by $p^{2d}4^{pd}$ and $p^{2(d-1)}4^{pd}$ for the stiffness matrix and the mass matrix respectively.

Explicit Representation of B-splines

The recursive form of B-splines may not be efficient from the computation cost and memory point of view. Moreover, the recursive form offer little insight in devising intergrid transfer operators. Therefore, we presented explicit formulas for the B-spline basis functions for a uniform mesh on a unit interval. The explicit representation of B-spline basis functions for a fixed mesh size h is given for $p = 2, 3, 4$, and for C^0- and C^{p-1}-continuity.

Multilevel B-splines and NURBS

The multilevel transfer operators from one level to another are at the heart of the multigrid or multilevel methods. Therefore, the construction of B-spline basis functions at coarse level from the linear combination of fine basis functions is provided. For $p = 2, 3, 4$, and with C^0 and C^{p-1} continuities, these transfer operators (from fine level to coarse level) are given in matrix form for a multilevel mesh. The transfer operators are also provided for NURBS basis functions. The formulation of NURBS operators is given in terms of B-spline operators and weights.

Multigrid Methods

The multigrid methods are presented for isogeometric discretization of scalar second order elliptic problems. The smoothing property of the relaxation method, and the approximation property of the intergrid transfer operators are analyzed. These properties, when used in the framework of classical multigrid theory, imply uniform convergence of two-grid and multigrid methods. For two dimensions, numerical results include the problems with variable coefficients, simple multi-patch geometry, a quarter annulus, and the dependence of convergence behavior on refinement levels L, whereas for three dimensions, only the constant coefficient problem in a unit cube is considered.

AMLI Methods

We have numerically studied AMLI methods for isogeometric discretizations. For B-splines (NURBS), hierarchical spaces are constructed as direct sum of coarse spaces and corresponding hierarchical complementary spaces. We have presented matrix form of these operators. As the choice of hierarchical complementary spaces is not unique, we have provided two different choices of these operators for each of C^0- and C^{p-1}-continuity of basis functions. The numerical study of CBS constant and the condition number of $\hat{A}_{11}$ block, corresponding to hierarchical complementary basis functions, are presented. For a given polynomial degree p, AMLI cycles are of optimal complexity with respect to the mesh refinement. For a given mesh size h, AMLI cycles are p-independent for C^{p-1} continuous basis function. Numerical results indicate almost p-independent convergence rates for C^0 continuous basis functions.

Graph Theory Based Preconditioning

The condition number of preconditioned system using Vaidya's preconditioner (maximum weight spaning tree) and Gremban and Miller preconditioner (support tree) is analyzed. It can be seen that the condition number of preconditioned system, based on both the types of Vaidya's preconditioners, as well as support tree preconditioners of Gremban and Miller, is not independent of mesh-size.

Future Directions

We present here some possible future direction.

1. The estimates for the minimum eigenvalues of the stiffness matrix and the mass matrix depend on the stability constant of B-Splines. Using the stability constant of B-Splines we could prove the lower bound of the minimum eigenvalue. The estimates for upper bound of minimum eigenvalue of the stiffness matrix and the mass matrix is still an open problem.

2. A general formula for the explicit representation of B-splines basis functions, for any mesh size h and for any polynomial degree p with uniform and non-uniform intervals, is not available. This will be advantageous from computation and memory point of views.

3. The iterative solvers which are fully robust with respect to the polynomial degree p are not yet available.

4. Due to larger support (in $p+1$ knot spans) of B-splines, local analysis of CBS constant is still an open problem.

Bibliography

[1] G.P. Astrachancev. An iterative method for solving elliptic net problems. *USSR Computational Math. Math. Phys.* **11(2)**, 171–182, 1971.

[2] F. Auricchio, L. Beirão da Veiga, A. Buffa, C. Lovadina, A. Reali and G. Sangalli. A fully "locking-free" isogeometric approach for plane linear elasticity problems: A stream function formulation. *Comput. Methods Appl. Mech. Engrg.* **197**, 160-172, 2007.

[3] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali and G. Sangalli. Isogeometric collocation methods. Math. Models Methods Appl. Sci., 20 (11) (2010), pp. 2075–2107.

[4] O. Axelsson. Iterative solution methods, *Cambridge University Press*, 1994.

[5] O. Axelsson and V. A. Barker. *Finite Element Solution of Boundary Value Problems: Theory and Computation.* Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2001.

[6] O. Axelsson and I. Gustafsson. Preconditioning and two-level multigrid methods of arbitrary degree of approximations. *Math. Comp.*, 1983; 40:219–242.

[7] O. Axelsson and P.S. Vassilevski. Algebraic multilevel preconditioning methods I. *Numer. Math.*, 1989; 56:157–177.

[8] O. Axelsson and P.S. Vassilevski. Algebraic multilevel preconditioning methods II. *SIAM J. Numer. Anal.*, 1990; 27:1569–1590.

[9] O. Axelsson and P.S. Vassilevski. A black box generalized conjugate gradient solver with inner iterations and variable-step preconditioning. *SIAM J. Matrix Anal. Appl.*, 1991; 12(4):625–644.

[10] O. Axelsson and P.S. Vassilevski. Variable-step multilevel preconditioning methods, I: self-adjoint and positive definite elliptic problems. *Numer. Lin. Alg. Appl.*, 1994; 1:75–101.

[11] I. Babuska, J.E. Osborn. Finite element-Galerkin approximation of the eigenvalues and eigenvectors of selfadjoint problems *Mathematics of computation, Volume 52, No. 186*. 275-297, 1989.

[12] I. Babuska, B.A. Szabo, and I.N. Katz. The p-version of the Finite Element Method. *SIAM Journal on Numerical Analysis, Volume 18, No. 3*. 515-545, 1981.

[13] N.S. Bakhvalov. On the convergence of a relaxation method with natural constraints on the elliptic operator. *USSR Computational Math. Math. Phys.* **6(5)**, 101–135, 1966.

[14] Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. Math. Models Methods Appl. Sci., 16 (2006), pp. 1–60.

[15] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. Comput. Methods Appl. Mech. Engrg., 199 (5–8) (2010), pp. 229–263.

[16] Y. Bazilevs, V.M. Calo, T.J.R. Hughes and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. Comput. Mech., 43 (1) (2008), pp. 3–37.

[17] Y. Bazilevs, V.M. Calo, Y. Zhang and T.J.R. Hughes. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Comput. Mech.* **38**, 310–322, 2006.

[18] Y. Bazilevs and T.J.R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers and Fluids* **36**, 12–26, 2007.

[19] Y. Bazilevs, C. Michler, V.M. Calo and T.J.R. Hughes. Weak Dirichlet boundary conditions for wall-bounded turbulent flows. *Comput. Methods Appl. Mech. Engrg.* **196**, 4853–4862, 2007.

[20] L. Beirão da Veiga, A. Buffa, J. Rivas and G. Sangalli. Some estimates for h-p-k-refinement in isogeometric analysis. Numer. Math., 118 (2) (2011), pp. 271–305.

[21] L. Beirão da Veiga, D. Cho, L.F. Pavarino, and S. Scacchi. Overlapping Schwarz methods for isogeometric analysis. *SIAM J. Numer. Anal.*, 50 (3) (2012), pp. 1394–1416.

[22] L. Beirão da Veiga, D. Cho, L.F. Pavarino, and S. Scacchi. BDDC preconditioners for isogeometric analysis,. *Math. Models Methods Appl. Sci.*, 2012, in press, http://dx.doi.org/10.1142/S0218202513500048.

[23] L. Beirão da Veiga, D. Cho, L.F. Pavarino, and S. Scacchi. Isogeometric Schwarz preconditioners for linear elasticity systems. *Comput. Methods Appl. Mech. Engrg.*, 253 (2013), Pages 439–454.

[24] L. Beirão da Veiga, D. Cho and G. Sangalli. Anisotropic NURBS approximation in isogeometric analysis. Comput. Methods Appl. Mech. Engrg., 209–212 (2012), pp. 1–11.

[25] R. Bellman. A note on an inequality of E. Schmidt. *Amer. Math. Soc.,* **50**, 734-736, 1944.

[26] M. Bern, J. Gilbert, B. Hendrickson, N. Nguyen, and S. Toledo. Support Graph Preconditioners. *SIAM J. Matrix Anal. Appl.* 27 (2006), 930–951.

[27] E. Boman, D. Chen, B. Hendrickson and S. Toledo. Maximum-weight-basis preconditioners. *Numer. Linear Algebra Appl.* 11 (2004), 695–721.

[28] E. Boman and B. Hendrickson. Support theory for preconditioning. *SIAM J. Matrix Anal. Appl.* 25 (2003), 694–717.

[29] D. Braess. *Finite Elements: Theory, Fast Solvers and Applications in Solid Mechanics.* Cambridge University Press, 2007.

[30] J.H. Bramble. *Multigrid methods.* Pitman Research Notes in Mathematics Series 294. Longman Scientific & Technical, Harlow, 1993.

[31] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.* **31**, 333–390, 1977.

[32] A. Brandt, S. McCormick and J. Ruge. Algebraic multigrid (AMG) for automatic multigrid solutions with applications to geodetic computations. *Tech. Report*, Institute for Computational Studies, Fort Collins, Colorado, 1982.

[33] S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods.* Springer, 2008.

[34] A. Buffa, J. Rivas, G. Sangalli and R. Vazquez. Isogeometric discrete differential forms in three dimensions. SIAM J. Numer. Anal., 49 (2) (2011), pp. 818–844.

[35] A. Buffa, G. Sangalli and R. Vazquez. Isogeometric analysis in electromagnetics: B-splines approximation. Comput. Methods Appl. Mech. Engrg., 199 (17–20) (2010), pp. 1143–1152.

[36] D. Chen and S. Toledo. Vaidya's Preconditioners: Implementation and Experimental Study. *ETNA* 16 (2003), 30–49.

[37] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North Holland Publishing Company, 1978.

[38] N. Collier, D. Pardo, L. Dalcin, M. Paszynski and V.M. Calo. The cost of continuity: a study of the performance of isogeometric finite elements using direct solvers. *Comput. Methods Appl. Mech. Engrg.*, 213–216 (2012), pp. 353–361.

[39] J.A. Cottrell, T.J.R. Hughes and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009.

[40] J.A. Cottrell, T.J.R. Hughes and A. Reali. Studies of refinement and continuity in isogeometric structural analysis.*Comput. Methods Appl. Mech. Engrg.* **196**, 4160–4183, 2007.

[41] J.A. Cottrell, A. Reali, Y. Bazilevs and T.J.R. Hughes. Isogeometric analysis of structural vibrations. Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 5257–5296.

[42] C. de Boor. On Calculating with B-Splines. *Journal of Approximation Theory* **6**, pp. 50-62, 1972.

[43] C. de Boor. On local linear functionals which vanish at all B-splines but one. *Theory of Approximation with Applications, A. G. Law and N. B. Sahney (eds.), Academic Press (New York)*. 120-145, 1976.

[44] C. de Boor. Splines as linear combinations of B-Splines, a survey. In *Approximation Theory II, G. G. Lorentz, C. K. Chui and L. L. Schumaker, (Eds.), Academic Press (New York)*, pp. 1-47, 1976.

[45] C. de Boor. *A Practical Guide to Splines.* Springer-Verlag, New York, 1978.

[46] C. de Boor. The exact condition of the B-Spline basis may be hard to determine. *Journal of Approximation theory.* **60**, 344-359, 1990.

[47] C. de Boor and J.W. Daniel. Splines with Nonnegative B-Spline Coefficients. *Mathematics of Computation* **28(126)**, pp. 565-568, 1974.

[48] C. de Falco, A. Reali and R. Vazquez. GeoPDEs: A research tool for Isogeometric Analysis of PDEs. *Adv. Eng. Softw.* 42, 1020–1034, 2011.

[49] C. de Falco, A. Reali and R. Vazquez. GeoPDEs webpage. http://geopdes.sourceforge.net

[50] V. Eijkhout, P.S. Vassilevski. The role of the strengthened Cauchy-Bunyakowski-Schwarz inequality in multilevel methods. *SIAM Review*, 33 (1991), pp. 405–419.

[51] A. Ern, J.-L. Guermond. Evaluation of the condition number in linear systems arising in finite element approximations. *ESAIM Math. Mod. Numer. Anal.* **40(1)**, 29-48, 2006.

[52] R.P. Fedorenko. A relaxation method for solving elliptic difference equations. *USSR Computational Math. Math. Phys.* **1(5)**, 1092–1096, 1961.

[53] R.P. Fedorenko. The rate of convergence of an iterative process. *USSR Computational Math. Math. Phys.* **4(3)**, 227–235, 1964.

[54] M.S. Floater. Evaluation and properties of the derivative of a NURBS curve. *Mathematical Methods in CAGD, T. Lyche and L.L. Schumaker (eds.), Academic Press, Boston,* **2**, pp. 261-274, 1992.

[55] K.P.S. Gahalaut, J.K. Kraus and S.K. Tomar. Multigrid methods for isogeometric discretization. *Comput. Methods Appl. Mech. Engrg.*, 253 (2013), pp. 413–425.

[56] K.P.S. Gahalaut and S.K. Tomar. Condition number estimates for matrices arising in the isogeometric discretizations. *RICAM report*, 23–2012.

[57] H. Gomez, T.J.R. Hughes, X. Nogueira and V.M. Calo. Isogeometric analysis of the isothermal Navier Stokes Korteweg equations. Comput. Methods Appl. Mech. Engrg., 199 (2528) (2010), pp. 1828–1840.

[58] K. Gremban. Combinatorial preconditioners for sparse, symmetric, diagonally dominant linear systems. PhD thesis, School of Computer Science, Cargenei Mellon University, Pittsburgh, (1996).

[59] W. Hackbusch. A multi-grid method applied to a boundary-value problem with variable coefficients in a rectangle. *Report 77-17*, Institut für Angewandte Mathematik, Universität Köln, 1977.

[60] W. Hackbusch. *Multigrid Methods and Applications*. Springer, 1985.

[61] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*. Springer, 1994.

[62] M.R. Hestenes and E.L. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, 49 (1952), 409–436.

[63] K. Hoellig. Multivariate Splines. *SIAM J. Numer. Anal.* **19(5)**, pp. 1013-1031, 1982.

[64] K. Hoellig. Stability of B-Spline basis via knot insertion. *Computer Added Geometric Design.* **17**, pp. 447-450, 2000.

[65] K. Hoellig, U. Reif and J. Wipper. Weighted Extended B-Spline Approximation of Dirichlet Problems. *SIAM J. Numer. Anal.* **39(2)**, pp. 442-462, 2002.

[66] N. Hu, X-Z Guo, and I.N. Katz. Bounds for eigenvalues and condition numbers in the p-version of the finite element method. *Mathematics of computation.* **67/224**, 1423-1450, 1998.

[67] T.J.R. Hughes, J.A. Cottrell and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Engrg.* 194 (2005), 4135–4195.

[68] T.J.R. Hughes, A. Reali and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. Comput. Methods Appl. Mech. Engrg., 199 (2010), pp. 301–313.

[69] C. Johnson. *Numerical solution of partial differential equations by the finite element method.* Cambridge University Press, Cambridge, New York, 1987.

[70] S.K. Kleiss, C. Pechstein, B. Jüttler, and S.K. Tomar. IETI - Isogeometric Tearing and Interconnecting. *Comput. Methods Appl. Mech. Engrg.*, 247–248 (2012), pp. 201–215.

[71] V.G. Korneev. *Finite element schemes of higher order of accuracy (in Russian)*, Leningrad University Press, 1977.

[72] J. Kraus. An algebraic preconditioning method for M-matrices: linear versus nonlinear multilevel iteration. *Numer. Lin. Alg. Appl.*, 2002; 9:599–618.

[73] J. Kraus and S. Margenov. *Robust Algebraic Multilevel Methods and Algorithms*. *Radon Series on Computational and Applied Mathematics*, 5, de Gruyter, Berlin, New York, 2009. ISBN 978-3-11-019365-7.

[74] J. Kraus, P. Vassilevski and L. Zikatanov. Polynomial of best uniform approximation to $1/x$ and smoothing in two-level methods. *Comput. Methods Appl. Math.*, 2012; 12(4): 448–468.

[75] V. Lakshmikantham and S. Leela. The Origin of Mathematics. *University Press of America, Inc., Lanham*, 2000, 104pp.

[76] C. Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Nat. Bur. Stand.*, 49 (1952), 33–53.

[77] T. Lyche, K. Scherer. On the sup-norm condition number of the multivariate triangular Bernstein basis. *Multivariate Approximation and Splines, G. Nuernberger, J. W. Schmidt, and G. Walz (eds.), ISNM.125, Birkhäuser Verlag, Basel.* 141-151, 1997.

[78] T. Lyche, K. Scherer. On the p-norm condition number of the multivariate triangular Bernstein basis *Journal of Computational and Applied Mathematics.* **119**, 259–273, 2000.

[79] J.F. Maitre and O. Pourquier. About the conditioning of matrices in the p-version of the finite element method for second order elliptic problems. *Journal of Computational and Applied Mathematics* **63**, 341–348, 1995.

[80] J.F. Maitre and O. Pourquier. Condition number and diagonal preconditioning: Comparison of the p-version and the spectral element methods. *Numer. Math.* **74**, pp. 69-84, 1996.

[81] J.M. Melenk. On condition numbers in hp-FEM with Gauss-Lobatto-based shape functions. *Journal of Computational and Applied Mathematics. Volume 139, Issue 1*. 21-48, 2002.

[82] J.M. Melenk and I. Babuska. The Partition of Unity Finite Element Method: Basic Theory and Applications. *Comput. Methods Appl. Mech. Engrg.***139**, pp. 289-314, 1996.

[83] B. Mössner and U. Reif. Stability of tensor product B-Splines on domains. *J. Approx. Theory* **154**, 1-19, 2008.

[84] Y. Notay. Flexible conjugate gradients. *SIAM J. Sci. Comput.*, 2000; 22(4):1444–1460.

[85] Y. Notay. Robust parameter-free algebraic multilevel preconditioning. *Numer. Lin. Alg. Appl.*, 2002; 9:409–428.

[86] E.T. Olsen, J. Douglas, Jr. Bounds on spectral condition numbers of matrices arising in the p-version of the finite element method. *Numerische Mathematik.* **69**, 333-352, 1995.

[87] J.M. Peña. B-Splines and Optimal Stability. *Mathematics of Computation* **66**, pp. 1555-1560, 1997.

[88] L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication)*, Second ed., Springer-Verlag, 1997.

[89] A. Reusken. A new lemma in multigrid convergence theory. *RANA* **91-07**, Eindhoven, 1991.

[90] D.F. Rogers. *An Introduction to NURBS With Historical Perspective.* Academic Press, 2001.

[91] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, Boston, 1996.

[92] K. Scherer, A.Y. Shadrin. New upper bound for the B-Spline basis condition number, I. *East J. Approx.* **2**, 331-342, 1996.

[93] K. Scherer, A.Y. Shadrin. New upper bound for the B-Spline basis condition number, II. A proof of de Boor's 2^k-conjecture. *Journal of Approximation theory.* **99**, 217-229, 1999.

[94] L.L. Schumaker. *Spline Functions: Basic Theory.* Cambridge University Press, 2007.

[95] G. Strang and G.J. Fix. *An Analysis of the Finite Element Method.* Prentice Hall, 1973.

[96] B. Szabo and I. Babuska. *Finite Element Analysis.* John Wiley & Sons, 1991.

[97] U. Trottenberg, C.W. Oosterlee and A. Schüller. *Multigrid.* Academic Press, 2001.

[98] P. Vaidya. Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners. An unpublished manuscript. A talk based on the manuscript was presented at the IMA Workshop on Graph Theory and Sparse Matrix Computations, Minneapolis, (1991).

[99] R.S. Varga. *Matrix Iterative Analysis.* Prentice Hall, New Jersey, 1965.

[100] P. Vassilevski. Multilevel block factorization preconditioners. *Springer*, New York, 2008.

Eidesstattliche Erklärung

I hereby declare under oath that the submitted doctoral dissertation has been written solely by me without any outside assistance, information other than provided sources or aids have not been used and those used have been fully documented. The dissertation here present is identical to the electronically transmitted text document.

Linz, May 2013

_____

Krishan Pratap Singh Gahalaut

Curriculum Vitae

Name: Krishan Pratap Singh Gahalaut

Nationality: India

Year of Birth: 1983

Place of Birth: Kukargawan, Hathras, UP, India

Education:

1999–2002	Bachelor of Science, Meerut University, Meerut, India
2002–2004	MSc (Mathematics), Banaras hindu University, Varanasi, India
2007–2009	MTech (Industrial Mathematics and Scientific Computings), Indian Institute of Technology Madras, Chennai, India

Special Activities:

2010 - 2013	Research Scientist at RICAM, Austria
2008 - 2009	DAAD scholarship at TU Darmstadt, Germany
2009	Lecturer at GGU Bilaspur, CG, India