Adaptive numerical modeling of engineering problems using hierarchical Fup basis functions and Control Colume IsoGeometric Analysis

Kamber, Grgo

UNIVERSITY OF SPLIT

FACULTY OF CIVIL ENGINEERING, ARCHITECTURE AND GEODESY

Grgo Kamber, mag.ing.aedif.

Adaptive numerical modeling of engineering problems using hierarchical Fup basis functions and Control Volume IsoGeometric Analysis

Doctoral dissertation

Split, 2021.

Grgo Kamber, mag.ing.aedif.
Serial number: 051

This doctoral dissertation has been submitted for evaluation to Faculty of Civil Engineering, Architecture and Geodesy of University of Split, for the purpose of obtaining the doctoral degree in the field of Technical sciences, scientific field of Civil engineering

Supervisor: prof. dr.sc.Vedrana Kozulić
Co-supervisor : prof. dr.sc. Hrvoje Gotovac

The committee for evaluation:

doc. dr. sc. Mijo Nikolić
prof. dr. sc. Željan Lozina
doc. dr. sc. Nino Krvavica

The defense committee:

doc. dr. sc. Mijo Nikolić_____

prof. dr. sc. Željan Lozina _____

doc. dr. sc. Nino Krvavica _____

The doctoral dissertation has beed befended on February 8th, 2021.

Secretary:

Saša Delić, dipl. iur.

This dissertation contains:

115 pages
63 figures
2 tables
73 references

To my family

Acknowledgments

This thesis has been supported by the Croatian Science Foundation (in Croatian Hrvatska zaklada za znanost - HRZZ) through grant number: UIP-2013-11-8103, and Faculty of Civil Engineering, Architecture and Geodesy (FGAG) at University of Split.

I would like to express my gratitude to my supervisors, prof. Vedrana Kozulić and prof. Hrvoje Gotovac who guided me throughout this project with their support, guidance and friendship. They encouraged me to be professional and do the right thing even when the road got though. The amount of time and energy they invested in our work is really special. They were always at a disposal and ready to help, both in professional and personal sense.

I am very thankful to all my colleagues and friends who supported me and offered deep insight into the study. Special thanks goes to Luka Malenica who was always there at disposal for my problems. I would like to thank committee members for their useful comments and suggestions which improved the thesis.

A very special gratitude goes to my girlfriend Kristina who had demanding role of being incredibly understanding and supportive throughout all working days and nights.

Finally, special thanks goes to my parents Ana and Ante, sisters Marina and Gabi and brother Mihael. I owe them so much, for their continuous support and understanding.

UDC 517.58:519.65]:624.04](043.3)

Grgo Kamber, mag.ing.aedif.

Adaptive numerical modeling of engineering problems using hierarchical Fup basis functions and Control Volume IsoGeometric Analysis

Abstract:

The main objective of this thesis is to utilize the powerful approximation properties of Fup basis functions for numerical solutions of engineering problems with highly localized steep gradients while controlling spurious numerical osicllations and describing different spatial scales.

The concept of isogeometric analysis (IGA) is presented as a unified framework for multiscale representation of the geometry and solution. This fundamentally high-order approach enables the description of all fields as continuous and smooth functions by using a linear combination of spline basis functions. Classical IGA usually employs Galerkin or collocation approach using B-splines or NURBS as basis functions. However, in this thesis, a third concept in the form of control volume isogeometric analysis (CV-IGA) is used with Fup basis functions which represent infinitely smooth splines. Novel hierarchical Fup (HF) basis functions is constructed, enabling a local *hp*-refinement such that they can replace certain basis functions at one resolution level with new basis functions at the next resolution level that have a smaller length of the compact support (*h*-refinement), but also higher order (*p*-refinement). This *hp*-refinement property enables spectral convergence which is significant improvement in comparison to the hierarchical truncated B-splines which enable *h*-refinement and polynomial convergence. Thus, in domain zones with larger gradients, the algorithm uses smaller local spatial scales, while in other region, larger spatial scales are used, controlling the numerical error by the prescribed accuracy. The efficiency and accuracy of the adaptive algorithm is verified with some classic 1D and 2D benchmark test cases with application to the engineering problems with highly localized steep gradients and advection-dominated problems.

Keywords: control volume, isogeometric analysis, spline basis functions, hierarchical Fup basis functions, adaptive numerical modeling, advection-dominated problems, local hp-refinement

UDK 517.58:519.65]:624.04](043.3)

Grgo Kamber, mag.ing.aedif.

Adaptivno numeričko modeliranje inženjerskih problema koristeći hijerarhijske Fup bazne funkcije i IzoGeometrijsku Analizu Kontrolnih Volumena

Sažetak:

Glavni cilj ove teze je iskoristiti moćna aproksimacijska svojstva Fup baznih funkcija za numeričko modeliranje inženjerskih problema s izrazito lokaliziranim gradijentima uz kontrolu numeričkih oscilacija.

Predstavljen je koncept izogeometrijske analize (IGA) kao cjeloviti pristup za višerezolucijsko modeliranje geometrije i rješenja. Pristup fundamentalno višeg reda omogućuje opis različitih polja kao kontinuiranih i glatkih funkcija koristeći linearnu kombinaciju spline baznih funkcija. Klasična IGA obično koristi Galerkin-ov ili kolokacijski pristup koristeći B-spline ili NURBS kao bazne funkcije. Međutim, u ovoj tezi se koristi treći koncept u obliku izogeometrijske analize kontrolnih volumena (CV-IGA) uz primjenu Fup baznih funkcija koje predstavljaju beskonačno glatke spline funkcije. Konstruirane su nove hijerarhijske Fup (HF) bazne funkcije koje omogućuju lokalno *hp*-poboljšanje tako da zamjenjuju određene bazne funkcije na jednom nivou novim baznim funkcijama na višem nivou gdje imaju manju duljinu kompaktnog nosača (*h*-poboljšanje), ali i viši red (*p*-poboljšanje). Svojstvo *hp* poboljšanja je da omogućuje spektralnu konvergenciju što predstavlja značajan doprinos u odnosu na modificirane hijerarhijske B-splineove koji omogućavaju *h*-poboljšanje i polinomsku konvergenciju. Dakle, u dijelovima domene gdje su rješenja najzahtjevnija, algoritam koristi bazne funkcije višeg reda s manjim kompaktnim nosačem, dok u ostalim dijelovima koristi rjeđi raspored baznih funkcija nižeg reda kontrolirajući numeričku pogrešku s definiranom točnošću. Učinkovitost i točnost adaptivnog algoritma provjerena je na nekim od klasičnih 1D i 2D referentnih numeričkih testova.

Ključne riječi: kontrolni volumeni, izogeometrijska analiza, spline bazne funkcije, hijerarhijske Fup bazne funkcije, adaptivno numeričko modeliranje, advektivno-dominantni problemi, lokalno hp-poboljšanje

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview and motivation

Many industrial and real applicative problems in computational mechanics have been solved by numerical simulations that require large computational resources including parallel processing and the use of CPU/GPU clusters. Therefore, it is of great importance that computer resources are used as efficiently as possible. In parallel with the technological development and use of powerful computers in solving various engineering problems, there has been an intensive development of realistic mathematical models in science. Such models are often expressed as boundary (or initial-boundary) problems defined by a particular set of partial differential equations (PDEs). Since exact solution of PDE is most often not known, to predict various physical phenomena it is necessary to approximate the problem numerically.

Numerical modeling of such processes typically faces many difficulties, especially in modeling abrupt localized solution changes using appropriate numerical control. The implementation of these models requires an efficient numerical analysis and description of all spatial/temporal scales that describe the solution. Many different numerical approaches and methods have been proposed in recent decades. In general, each method has its advantages, but also disadvantages, and none can be singled out as the best for all problems. The best known and most flexible methods are finite element method (FEM), finite difference method (FDM) and finite volume method (FVM) [1]–[9]. There are various other methods such as the spectral element method (SEM), boundary element method (BEM), discrete element method (DEM) which, together with various collocation, meshfree and other hybrid approaches are only practical for limited classes of problems.

Physical laws for space and time dependent problems are usually described by partial differential equations (PDEs). For most problems and geometric shapes of the domain, the corresponding PDE cannot be solved by analytical methods. Instead, approximate solutions are determined, which are usually based on different types of discretization. Discretization methods approximate PDEs by a set of equations of numerical models that can be solved with numerical methods. The solution of the numerical equations of the model represents an approximation of the real solution of the PDE.

First, it is important to understand the different forms of PDEs. PDEs

can be classified into elliptical, hyperbolic and parabolic types. When solving these differential equations, it is necessary to specify boundary and/or initial conditions. The required input parameters can be estimated based on the type of PDE. Examples of PDEs in each category include the Poisson equation (elliptical), the wave equation (hyperbolic), and Fourier's law (parabolic). Regarding the interpolation profiles for the discretization of the governing PDEs, two main approaches have historically dominated in the field of computational mechanics.

The first and oldest procedure is the finite difference method (FDM). The starting point of the FDM is to cover the domain with a (mostly uniform) grid. At each point of the grid, the differential form of PDE is discretized by approximating the derivatives with expressions of finite differences. These expressions are generally derived by using Taylor series expansion or polynomial fitting through a certain number of specific neighboring grid points. Since each point of the grid gives one discretized (algebraic) equation with several unknowns, the equations of all points must be combined in the system and solved simultaneously. The final result of the FDM are the solution values at the grid points. However, there is no explicit reference how a solution behaves between grid points. In this sense, a FDM can be thought of as akin to a laboratory experiment, in which a set of instrument readings enables us to establish the distribution of the measured quantity in the domain under investigation (Patankar [10]). Apart from the fact that the conservative properties of the equations are generally not preserved, the main drawback due to which FDM does not have wide application is its limitation to simple geometries.

The finite volume method (FVM) can be considered a natural improvement of the FDM. In general, the solution domain is divided into an finite number of non-overlapping control volumes (CVs), and a conservative form of PDE is integrated over each CV. The volume integrals over the CVs are converted into surface integrals by the divergence theorem, and interpolation is used to express values at the boundaries of the CV in terms of nodal (center of CV) values. The FVM is conservative by its construction and its main advantages are the direct physical meaning of the discretized equations and a formulation that is also suitable for complex geometries.

The FDM and FVM do not use interpolation profiles in terms of a defined interpolation space, such as trial basis functions space. Rather, they use local approximation profiles. This local interpolation profiles are not necessarily the same for all terms in the governing equations. Even though this approach permits complete freedom, as mentioned before, the solution is not uniquely defined throughout the domain, except for discrete nodal (grid) points. Moreover, the consequence of using interpolation profiles that are mostly one-dimensional is usually reduced accuracy for multidimensional problems. This is particularly true when first-order interpolation is used on meshes oblique to flow gradients [10].

The second widely used approach to interpolation profiles is the one adopted in finite element method (FEM). First, the trial vector space of basis functions is defined, then a linear combination of these functions is used for

both interpolation and differentiation during the discretization process. One of the benefits of using FEM is that it offers great freedom in the selection of discretization, both in the elements that may be used to discretize space and the basis functions. However, the most widely used basis functions among the FEM community are Lagrangian polynomials.

The FEM finds its applications in almost all fields of computational mechanics. Good approximation properties, ability to handle arbitrary geometries, straightforward construction of higher order approximations on unstructured grids and strong mathematical background (e.g., Zienkiewicz *et al.* [11] and Bathe [12]) are some of the main strengths of the FEM [13]. Moreover, FEM is considered to have best approximation properties when applied on problems governed by symmetric (self-adjoin) differential operators. However, presence of convection/advection operators in the governing equations for fluid flow renders the system of equation to be non-symmetric and the best approximation property in energy norm, which gave FEM a success in structural mechanics, is lost. Furthermore, this non-symmetric character of fluid flow is the main reason for stability issues so special stabilization techniques are needed when FEM is used for simulation of complex fluid flows. Moreover, absence of the local conservation properties and flux discontinuities between finite elements are additional weaknesses of FEM.

The numerical solutions produced by FEM are continuous and smooth inside a particular element. However, usage of classical Lagrangian basis functions ensures only C^0 continuity on the element boundaries. Moreover, the gap between computer-aided design (CAD) for the geometry description on the one hand and finite element analysis (FEA) for the solution description on the other hand has been long evident. It is estimated that about 80% of overall analysis time is devoted to mesh generation in the automotive, aerospace, and ship building industries. In the automotive industry, a mesh for an entire vehicle takes about four months to create [14]. Furthermore, once a mesh is constructed, during each mesh refinement a communication with the CAD system is necessary and since this link is often unavailable explains why adaptive refinement is still primarily an academic endeavor rather than an industrial technology.

This gap between CAD systems and FEA (as well as overall numerical analysis) is due mostly to differences in the used interpolation (basis) functions. Whereas classical polynomials have dominated in the field of numerical analysis, spline-based basis functions (e.g., B-splines, nonuniform rational *B*-splines (NURBS) [14], T-splines [15], hierarchical B-splines (HB) [16] etc.) play a crucial role in the field of computational geometry. True popularity of spline functions for numerical analysis was achieved by the introduction of the concept of isogeometric analysis (Hughes *et al.* [14] and Cottrell *et al.* [17]). The main idea of isogeometric analysis (IGA) is to bridge the gap between FEA to describe a numerical solution and a CAD system to describe geometry using the same type of spline basis functions for both systems. The core of IGA is the isoparametric concept (widely used in classical FEM), where the basis functions used to approximate the solution field

are also used to describe the geometry. IGA turns this idea in a different direction and selects a basis used to describe geometry in CAD systems as the basis for numerical approximating of unknown fields. The key difference is that IGA allows accurate representation of geometry in CAD terms in contrast to classical FEA where geometry is only approximated.

IGA is closely related to the meshless or mesh-free methodologies due to its use of spline basis functions. Application of spline basis functions enables some properties not seen in FEM, such as exact geometry description, usage of higher-order basis functions, higher continuity of solution and geometry, more efficient refinement adaptive procedures and multiresolution approach. Efficient numerical modeling using spline functions does not always have to be associated exclusively with IGA involving geometry transformations, because everything can only be performed in the physical domain. Furthermore, the geometry constraints and boundary conditions can be satisfied exactly using the Rvachev solution structure method (see Rvachev *et al.* [18], Höllig *et al.* [19], and Kozulić and Gotovac [20]).

The development of adaptive methods [21]–[24] for local refinement and coarsening became one of the most researched topics within IGA. Since a fundamental limitation of traditional NURBS is the lack of potential for local refinement, several solutions have been derived, such as T-splines [15], [25]–[30], hierarchical B-splines (HB) [16], truncated hierarchical B-splines (THB) [31]–[35] and locally refined B-splines (LR) [36]. Furthermore, linear independence, stability and partition of unity as well as local refinement and adaptation became center topics for these adaptive solutions.

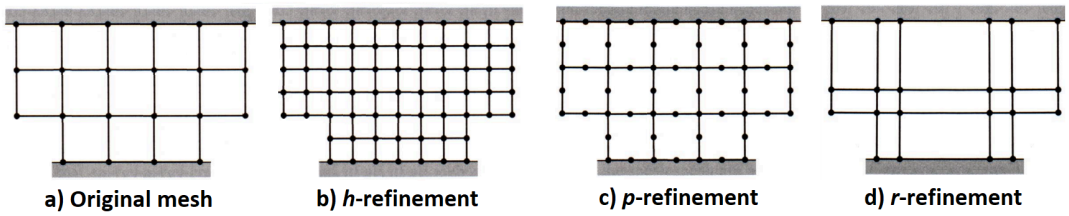


FIGURE 1.1: Refinement procedures

Adaptive isogeometric methods attract a lot of attention and are a very active field of research. Improvement procedures are h-refinement (Figure 1.1b; spline functions of the same order but smaller knot intervals, i.e. higher frequencies; changing element size), p-refinement (Figure 1.1c; higher degree of basis functions), r-refinement (Figure 1.1d; redesigning the mesh without changing the basis order; keep the number of nodes constant and adjust their positions) and their combination. Even though B-splines and NURBS are most commonly used spline technologies in the isogeometric settings, due to their tensor product structure, they are not well suited to treat localized phenomena. Hierarchical B-splines (HB) constitute one of the most promising solutions to easily define adaptive spline base which preserve the non-negativity of standard B-splines and enables the possibility to properly deal with local problems. However, since the hierarchical B-spline basis functions in non-rational form do not satisfy partition of unity, it may produce

ill-shaped control mashes at the refined level. To overcome this deficiency, the truncated mechanism was first developed by Giannelli *et al.* [16] for the hierarchical B-spline basis functions to form a partition of unity and to decrease the overlapping of basis functions for better numerical conditioning.

In addition to spline functions, relatively lesser-known atomic or R_{bf} basis functions have been used in recent times (Rvachev's basis functions, see Rvachev and Rvachev [37] and Gotovac [38]). R_{bf} or atomic basis functions can be placed between classical polynomials and spline functions. However, in practice, their use as basis functions is closer to splines or wavelets (see Beylkin and Keiser [39]). Gotovac [38] systematizes the existing knowledge about atomic basis functions and transforms them into a numerically appropriate form. Kozulić [40] and Gotovac and Kozulić [41] showed the basic possibilities of using these functions in structural mechanics and numerical analysis. The use of Fup basis functions, as the most commonly used atomic basis functions, has been shown to solve the problem of signal processing (see Kravchenko *et al.* [42]), the initial problem (see Gotovac and Kozulić [43]) and the boundary problems using the non-adaptive Fup collocation method (see Kozulić and Gotovac [44] and Gotovac *et al.* [45]).

Gotovac *et al.* [46] presented a true multiresolution approach based on the Adaptive Fup Collocation Method (AFCM). The heart of the AFCM methodology lies in the Fup basis functions in conjunction with the collocation procedure. However, the main drawback was the lack of global and local mass balance due to the properties of the collocation framework, computationally expensive head solution to obtain an accurate velocity field without numerical oscillations for high heterogeneity cases and inability to describe the general irregular geometry. Applications of the AFCM have been shown for the analysis of the flow and transport in heterogeneous porous media relating to the travel time statistics Gotovac *et al.* [47]. Brajcic Kurbasa [48] presented atomic basis functions of the exponential type. For the first time, the properties of exponential ABFs were investigated in detail and expressions for calculating the value of the function and all necessary derivatives at an arbitrary point of the domain were derived, as well as some features necessary for their practical application in a form suitable for numerical analysis. Malenica [49] used Fup basis functions as representative members of the spline family in the development of a novel numerical model for groundwater flow in karst aquifers. Futhermore, it also presents the development of a full space-time adaptive collocation algorithm with particular application to advection-dominated problems while Kamber *et al.* [50] set foundation for adaptive spatial procedure using Fup basis functions with control volume formulation.

1.2 Hypothesis

The main objective of this thesis is to demonstrate the capabilities of spline basis functions through the development of novel numerical method with specific application to problems with highly localized steep gradients while controlling spurious numerical oscillations. Emphasis is given to Fup basis

functions (Rvachev and Rvachev [37] and Gotovac and Kozulić [41]) as representative member of the spline family. The Fup function can be obtained by a convolution procedure using contracted B-splines and infinitely derivable up atomic function. In this way, Fup is closely related to B-spline. However, Fup basis functions have better approximation properties compared to the B-splines due to the convolution with the up function containing all orders of polynomials by parts and infinite continuity. Due to this property, adaptive procedure using Fup basis functions should allow much more accurate solutions compared to adaptive solutions obtained by B-spline basis functions. The particular contribution of this thesis lies in the property of hierarchical Fup basis functions, in relation to B-splines, enabling local hp-refinement, which means that higher resolution levels have basis functions not only of smaller length of the compact support (higher frequencies; h-refinement) but also contain basis functions of higher order (p-refinement). Thus, the basic hypothesis behind the proposed method is to obtain spectral convergence for problems with highly localized steep gradients, while the existing hierarchical B-splines (HB) and truncated hierarchical B-splines (THB) achieve convergence determined by the degree of the polynomial of the basis functions.

This work can be considered as an upgrade of the previously developed adaptive Fup collocation method (AFCM; [51]). The main idea of the AFCM is to dynamically adapt the computational grid during the simulation so that the algorithm uses more collocation points (i.e., higher resolutions) only in areas where the solution changes are demanding (e.g., localized step gradients or discontinuities). Furthermore, formulation of control volumes was used in numerical modeling of groundwater flow in karst aquifers Malenica [49] and Malenica *et al.* [52], and was shown to contain several interesting properties such as: local and global conservation properties, direct physical meaning of the discretized equations, very close to Gallerkin's solution for much lower computational costs, and increased accuracy and stability in relation to the aforementioned collocation method. Therefore, in this thesis for the first time the hierarchical Fup basis functions (one-dimensional and two-dimensional) with control volume formulation will be presented.

Due to certain similar properties with classical IGA [14], [17], used method is called control volume isogeometric analysis (CV-IGA) [49], [52]. However, since classical isogeometric analysis mostly uses B-spline or NURBS basis functions in conjunction with the Galerkin [8] or collocation formulation, here, the proposed model will be based on Fup basis functions and the control volumes formulation enabling local and global mass conservation, as well as approximate solutions of higher smoothness. The application of CV-IGA to real application problems will be presented in the latter part of the thesis.

The main goal of this thesis is to develop hierarchical Fup basis functions (HF) and to use good approximation properties of these functions in creating new adaptive technique within control volume isogeometric analysis (CV-IGA) for numerical solutions of engineering problems arising in the field of

structural mechanics and fluid mechanics. The formulation of control volumes would allow for local and global conservation properties, with computational costs that are between Galerkin (high CPU price) and collocation (low CPU price).

The developed adaptive algorithm will be applied in solving demanding engineering problems. First, an analysis of the 1-D problem will be performed starting with the approximation of the known function to show some basic parameters of the adaptive algorithm. Then, on the example of the advective-dispersion equation (ADE), the efficiency of new adaptive method will be presented through numerical solving of the differential equation when in most cases we do not know the analytical solution. In particular, a comparison with analogous *h*-adaptive procedures based on IGA and hierarchically modified B-splines will be shown. Furthermore, the method is extended to 2-D analysis and verified on problems such as mass and energy conduction (heat) generally shown with the diffusion and advection-dispersion equation, the elasticity problem, and in solving Poisson equation. A comparison of numerical results obtained with hierarchical B-splines and the achievement of spectral convergence obtained with the novel numerical procedure will be shown.

1.3 Outline

Chapter 2 provides the background of the classical numerical methods such as finite difference method, finite volume method and finite element method.

In Chapter 3 the mathematical background of the spline basis functions is provided with addition of the less known Fup basis functions that belong to the class of atomic functions and can be regarded as infinitely differentiable B-splines. Furthermore, a detailed description of Fup basis functions and their relationship with B-splines is given.

Chapter 4 addresses the isogeometric (IGA) approach with addition of the three numerical formulations: Galerkin, Collocation and Control volume. Two of those formulations can be regarded as classical IGA formulations, more specifically Galerkin and Collocation formulations are usually used for spatial discretization. However, here a novel control volume IGA formulation is presented in addition to two classical IGA formulations. The concept of IGA is presented as a unified framework for a multi-scale description of the geometry and solution fields, where all approximated fields are represented as continuous and smooth functions.

In Chapter 5, hierarchical spline functions, i.e. hierarchical B-spline basis functions, are presented with its refinement procedures. Since the hierarchical B-spline basis functions in non-rational form do not satisfy partition of unity, the truncated mechanism was introduced to overcome that deficiency. In addition to hierarchical B-splines, more efficient truncated hierarchical B-splines procedure is presented.

Chapter 6 presents the development of the novel adaptive algorithm that is based on Fup basis functions that belong to the class of atomic functions. Hierarchical Fup (HF) basis functions that have the option of local

hp-refinement such that they can replace certain basis functions at one resolution level with new basis functions at the next resolution level that have a smaller length of the compact support (*h*-refinement) but also higher order (*p*-refinement) is presented in detail. Satisfying the boundary conditions, adaptive criteria and various stabilization methods are also described.

Chapter 7 presents the numerical results confirming efficiency of the developed method (described in Chapter 6) applied to some of the classical numerical problems such as advection-dispersion problem, heat conduction problem, linear elasticity problem, etc.

Finally, Chapter 8 summarizes the most important findings of the thesis and provides suggestions for future research.

Chapter 2

Classical numerical methods

This chapter serves as review of some classical numerical methods prior introduction of the isogeometric analysis. Furthermore, the first examples of adaptive numerical modeling were focused upon using classical numerical methods such as finite difference (FDM), finite element (FEM) and finite volume methods (FVM) [1]–[9].

2.1 Finite difference method

The finite difference approximations for derivatives can be considered as one of the simplest and of the oldest methods to solve differential equations. Their development was stimulated by the emergence of computers that offered a convenient framework for dealing with complex problems of science and technology. The main idea of FDM is focused on approximating differentials. The domain is partitioned in time and in space and approximations of the solution are computed at those points (time and space points). The difference between the exact solution and the numerical solution is determined by the error that is committed by going from a differential operator to a difference operator. This error is called the discretization error or truncation error, reflecting the fact that a finite part of a Taylor series is used in the approximation. In contrast to this, weighted residual methods evaluate the integral of a differential equation while optimizing an approximation such that the integrals of the approximated solutions and the correct solution match on a given domain.

Let us consider one-dimensional case for simplicity. The main concept behind any finite difference scheme is connected to the definition of the derivative of a smooth function u at a point $x \in \mathbb{R}$:

$$u'(x) = \lim_{\Delta x \to 0} \frac{u(x + \Delta x) - u(x)}{\Delta x} \tag{2.1}$$

and to the fact that the quotient on the right-hand side provides a "good" approximation of the derivative when Δx tends to the 0. In other words, to get a good approximation Δx needs to be sufficiently small. Moreover, the approximation is good when the error committed in this approximation tends towards zero when Δx tends to zero. If the function u is sufficiently smooth in the neighborhood of x, it is possible to quantify this error using a Taylor expansion. The most common equations are:

- Backward finite difference schemes for approximating first derivatives

$$\frac{du}{dx} = \frac{u(x) - u(x - \Delta x)}{\Delta x} + O(\Delta x) \tag{2.2}$$

- Central finite difference schemes for approximating first derivatives

$$\frac{du}{dx} = \frac{u(x + \Delta x) - u(x - \Delta x)}{2\Delta x} + O\left(\Delta x^2\right) \tag{2.3}$$

- Forward finite difference schemes for approximating first derivatives

$$\frac{du}{dx} = \frac{u(x + \Delta x) - u(x)}{\Delta x} + O(\Delta x) \tag{2.4}$$

- Central finite difference schemes for approximating second derivatives

$$\frac{d^2u}{dx^2} = \frac{u(x + \Delta x) - 2u(x) + u(x - \Delta x)}{\Delta x^2} + O\left(\Delta x^2\right) \tag{2.5}$$

where the term $O(\Delta x)$ in (2.2) and (2.4) indicates that error of the approximation is proportional to Δx and analogously the term $O\left(\Delta x^2\right)$ in (2.3) and (2.5) indicates that error of the approximation is proportional to Δx^2.

The finite change of the solution is approximated on a very small finite interval using one of these equations. All of these equations are linear which means that the solution is linearly approximated.

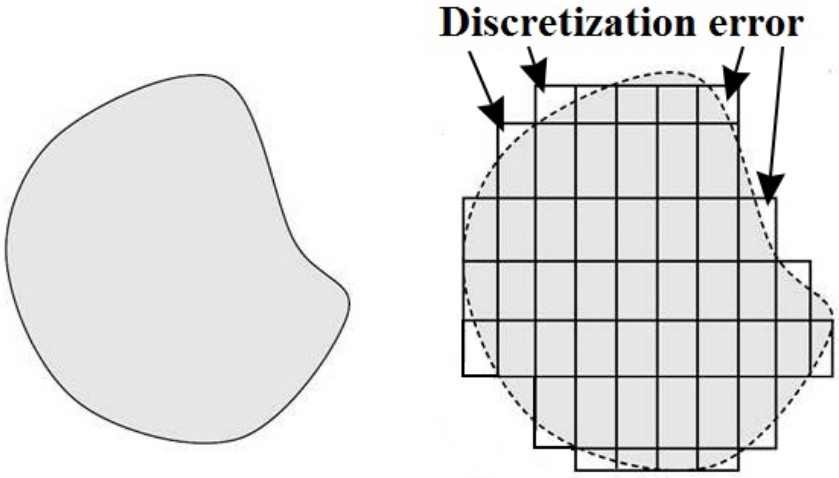


FIGURE 2.1: Treatment of complex geometries with FDM

The main advantage of FDM is the fact that it is a very exact method with solutions usually significantly closer to the solution, in comparison to results obtained from, e.g., weighted residual methods. However, there are several disadvantages. The most severe disadvantage is the requirement for structured grids because it usually does not cope very well with complex (Figure 2.1) or multi-scale geometries and is computationally very expensive.

Furthermore, FEM and FVM have significantly less strict requirements on grid structuring and allow locally adapting the grid to suit the local geometry which gives the main advantages of both methods in comparison with FDM.

However, many differential equations involve time-dependency, which gives rise to a time-dependent differential, finite difference schemes becomes essential for both FEM and FVM. Many numerical solvers will use FEM and FVM to solve the space-dependent terms of a differential equation and then use one of the previously mention schemes (backward finite, central finite or forward finite) to step in time. Obviously, the main solution of the differential equation is then obtained using a combination of FVM or FEM for the space-dependent terms and FDM for the time-dependent terms.

2.2 Finite volume method

The next method that will be presented is the finite volume method that can be considered as a natural improvement of FDM. FVM can be used on all differential equations, which can be written in divergence form (the equation is written using divergence operators).

Generally, the calculation domain is subdivided into an finite number of nonoverlapping control volumes (CVs) such that there is one CV surrounding each node (see Figure 2.2), and the conservative form of the governing PDEs is integrated over each CV. Gauss theorem is usually applied to convert the volume integral over the divergence into a surface integral across the boundaries. Thus, the integral is therefore turned from integrating the differential of the dependent variable inside of the cells into surface integrals of the fluxes of the dependent variable across the boundary of the cells. This substantially simplifies the differential equation. The interpolation is used to express the values at the CV boundaries in terms of nodal values. Again, like in FDM, the final results are the solution values at the nodal points. Since different interpolation profiles can be used for different expressions that occur in the governing equations, there is no explicit reference for the variation of the solution between the nodal points.

Approach used within FVM is based on the concept that all mass that would "diverge" out of the CV must inherently pass the boundary of the CV at some time, and if this flux is integrated over time, the total change of mass in the CV can be derived. Moreover, a conservation approach of the dependent variable is obtained by monitoring the fluxes of the dependent variable across the boundary of the cells.

Example Consider one-dimensional heat conduction equation governed by

$$\frac{\partial T}{\partial t} - c^2 \Delta T = 0 \tag{2.6}$$

where c^2 is a positive constant (thermal conductivity) which can be thought of as a "diffusion coefficient" for heat and T describes the temperature function along a one-dimensional domain.

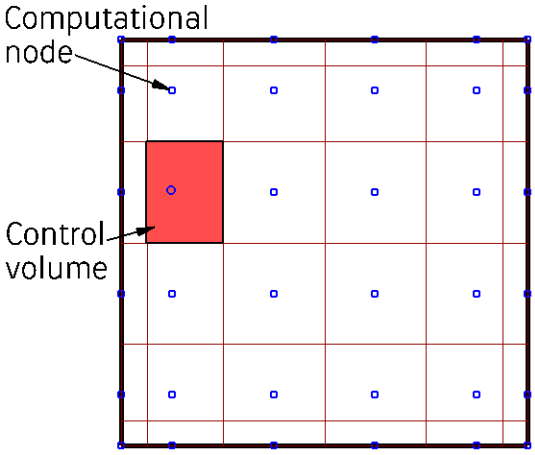


FIGURE 2.2: 2D computational domain subdivided into an finite number of nonoverlapping CVs

In order to obtain the conservative form of (2.6) we need to integrate this equation over the control volume resulting in

$$\int_{CV} \frac{\partial T}{\partial t} dCV - \int_{CV} c^2 \frac{\partial^2 T}{\partial x^2} dCV = 0 \tag{2.7}$$

Now we can exchange the differential and the integral of the first term on the left-hand side, because our control volume is fixed in space and not dependent on time. Moreover, this allows us to convert the partial into a regular differential. Second, we can use Gauss theorem to convert the volume integrals into surface integrals. Doing this, we can rewrite (2.7) to

$$\frac{d}{dt} \int_{CV} T dCV - \int_{\partial CV} c^2 \frac{\partial T}{\partial x} \mathbf{n} d\partial CV = 0 \tag{2.8}$$

where ∂CV represents boundary of the domain and $\overrightarrow{n}$ is the outward unit normal to the boundary.

In general, after writing governing equation in differential notation it becomes valid for every point in the computational domain. Moreover, if we work on the differential form, we would obtain an FDM problem that can be solved if we take into account CV of infinitesimally small lateral size. However, as presented in section 2.1, FDM is computationally very expensive because it would require a large number of CVs with very small lateral size in order to obtain exact results. It would be desirable to increase the control volumes to larger volumes and that is exactly what is being done in FVM. The reason the FVM allows work with a larger CV is that it does not approximate the changes of the dependent variable along the control volume, but performs a conservation of the dependent variable for each CV. This allows the calculation of the average value for the dependent variable within the cell after which the CV is replaced by one value. It is obvious that if we take a CV of larger dimensions, the more "averaged" solution will become. For

example, finer features of the solution such as spikes in concentration will not be visible anymore because they are blurred by the averaging process. Because of that averaging process, FVM is surprisingly stable against large changes and discontinuities in the dependent variable. However these tend to be problematic for FDM because at discontinuities a function usually has no derivative, so FDM in these locations cannot approximate values.

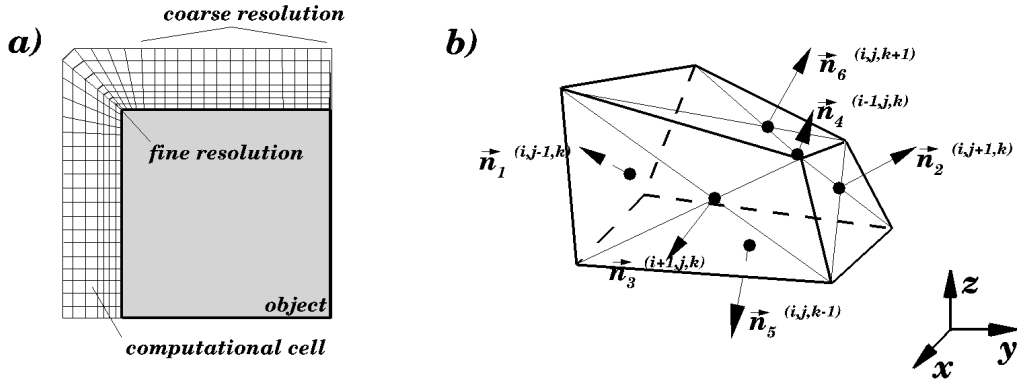


FIGURE 2.3: Cells used in FVM. a) A two-dimensional computational domain showing the typical quadrangles. At locations where large changes in the dependent variables are expected, the resolution is higher. b) A three-dimensional uneven hexaeder with the unit normal vectors facing the respective neighboring cell.

The most important implication for the volume discretization of our computational domain, because FVM requires the dependent variable to be in conservative form, is the fact that the discreet volumes must remain fixed in space. These volumes are referred as cells in contrast to the term "elements", which is the term used in finite element method. In two-dimensional domain, cells are quadrangles (not not necessarily squares; see Figure 2.3a), and in three-dimensional space they are six-face hexaeder (see Figure 2.3b). These cells i.e., shapes are adapted to the discretization of the volume and should be designed in such manner that they allow a good resolution of the computational domain. This is important especially in areas which are expected to show large changes of the dependent variables.

Another important point to consider is the fact that the shape of the cells can be adapted to suit the geometry of the environment. This is in contrast to FDM, which usually requires the control volumes to be equally shaped [53]. Furthermore, the most attractive feature is that the resulting solution implies that the integral conservation of quantities such as mass, momentum, and energy is exactly satisfied over any group of used control volumes (local conservation) and, of course, over the whole calculation domain (global conservation). This property is valid for any number of grid points [10].

2.3 Finite element method

Third method is by far one of the most commonly used method in numerical analysis. Finite element method (FEM) was originally developed for solving problems in solid-state mechanics, but it has since found wide application in all areas of computational physics and engineering, as well as in computational fluid dynamics (CFD). The main difference compared to the previous two methods (FDM and FVM) is that FEM is governed by the principle of minimization of energy i.e., it uses variational formulation of the problem. In other words, when a particular boundary condition (like force or displacement) is applied to a body, leading to a several configurations for that body, only that configuration where the total energy is minimum is the one that is achieved.

The basic concept underlying the FEM is relatively simple. It can be thought of as splitting the computational domain into very small but finite-sized elements of geometrically simple shapes (see Figure 2.4). The word "finite" is used to describe the limited, or finite, number of degrees of freedom used to model the behaviour of each element. The elements are assumed to be connected to one another, but only at interconnected joints, known as nodes. It is important to note that the elements are small regions, not separate entities like bricks, and there are no cracks or surfaces between them. The next step is to solve system of equations, mathematically represented by governing PDEs that describe the physics of the problem, and formulate these equations for each finite-sized element. This is solved by approximating the fields within each element as a simple function with a finite number of degrees of freedom (DOF). By stitching the individual solutions i.e., the contribution from all elements are assembled and a large sparse matrix equation system is solved, a global solution can be obtained.

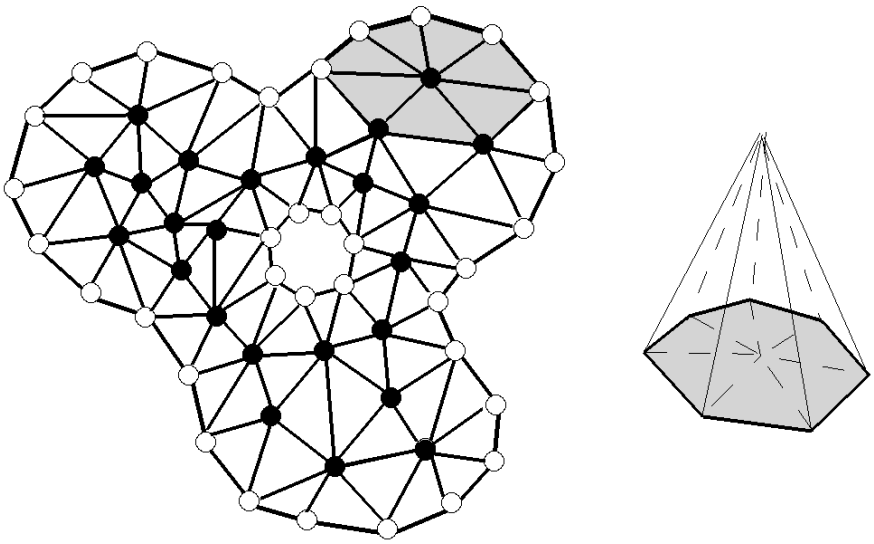


FIGURE 2.4: Hat-function on a triangulation.

The first step of FEM is the discretization i.e., the process of representing a component as an assemblage of finite elements of the computational domain. In FEM the complete set, or assemblage of elements, is usually referred to as the mesh and the support points of the solution are referred to as nodes. Moreover, because the two-dimensional FEM prefers triangles (see Figure 2.4) and the tree-dimensional FEM prefers tetrahedra as the smallest unit in the mesh for the most applications, since they can be adapted more easily to complicated boundaries, this procedure is also called triangulation. In general, the choice of nodes should match the complexity of the computational domain in a sense that the curvatures of the domain are approximated as closely as possible. Furthermore, regions where the expected solution will have steeper gradients i.e, more localized solution should be more finely resolved than regions where small changes are expected.

Another important property of FEM is that it uses linear combination of basis (shape) functions for both interpolation and differentiation during the discretization process. Because of that, the final solution is uniquely defined throughout the whole computational domain, unlike in FDM and FVM where the solution is not uniquely defined throughout the domain, except for discrete nodal (grid) points. The most widely used basis functions among the FEM community are Lagrangian polynomials. The same functions that are commonly used to construct test (weighted) functions are also used for geometry mapping through the isoparametric concept. In general, solution $\tilde{u}$ in FEM is represented by linear combination of Lagrangian basis functions in form

$$\tilde{u} = \sum_{j=1}^{N} \alpha_j \varphi_j \tag{2.9}$$

where α_j are unknown coefficients and φ_j are basis functions. In case of multidimensional basis functions they are simply constructed using the tensor product of one-dimensional basis functions.

2.4 Collocation method

The forth method that will be presented is collocation method. It is used for solving integral and differential equations in which the approximate solution is determined from the condition that the equation is satisfied at certain given points (collocation points). It belongs to one of the two special cases of the method of weighted residuals, the second one is Galerkin method.

Method of weighted residual is based on finding undetermined coefficients by minimizing residual while approximating the desired function u. The desired function u is replaced by a finite series approximation

$$u \approx \tilde{u} = \sum_{j=1}^{N} \alpha_j \varphi_j \tag{2.10}$$

where the set of functions $\varphi_j, j = 1, 2, ..., N$ can be defined over both the time and space domain, and $\alpha_j, j = 1, 2, ..., N$ are unknown i.e., undetermined

coefficients. In the FEM, the functions φ_j are selected to be polynomials that satisfy certain boundary conditions set by the problem, and are variously denoted, depending upon the area in which the method is applied, as shape functions, basis functions or interpolation functions.

After substituting the assumed solution $\tilde{u}$ into the PDE of the form

$$Lu - f = 0 \tag{2.11}$$

where f is a known function, and L denotes differential operator involving spatial derivative of u, results in

$$L\tilde{u} - f = R \tag{2.12}$$

where R is a measure of error commonly referred to as the residual. Since the assumed solution is only approximate, in general it does not satisfy the differential equation that results in an error or what is usually called "residual". The residual is then made to vanish in some average sense over the entire solution domain to produce a system of algebraic equations. The objective is to find the unknown coefficients α_j such that residual R is minimized. A straightforward scheme would be to set the integral of residual to zero. But before that, we must introduce weight functions $w_i, i = 1, 2, ..., N$ so that integral

$$\int_D R(x) w_i(x) dD = 0, \quad i = 1, 2, ..., N \tag{2.13}$$

could be solved over the domain D for the N unknown coefficients. Equation (2.13) is the general equation describing the method of weighted residual. This paved the way for a multitude of schemes that emerged from this one expression through the definition of the weighting functions $w_i(x)$. Three schemes that are most commonly encountered in engineering practice are the Galerkin, subdomain and collocation. The Galerkin method uses the weighting functions to be the same as the basis functions, as defined in (2.10). In the subdomain method, the weighting function is set to be unity in the subdomain D_i and zero elsewhere, i.e.,

$$w_i(x) = \begin{cases} 1 & \text{x } \in \Omega_i \\ 0 & \text{x } \notin \Omega_i \end{cases}, \Omega_i \in \Omega. \tag{2.14}$$

However, the collocation method is the simplest to implement. Here, the weighting function $w_i(x)$ is chosen to be the Dirac delta. The Dirac delta function belongs to the class of generalized functions. It can be set in different ways, and the simplest and physically justified way is to interpret Dirac function as an impulse function.

Figure 2.5 shows the impulse function I_a, which has the property of being intense on the interval $[-a, a]$ with an intensity value of $\frac{1}{2a}$, and outside the interval it is equal to zero. Moreover, the value of the integral can be

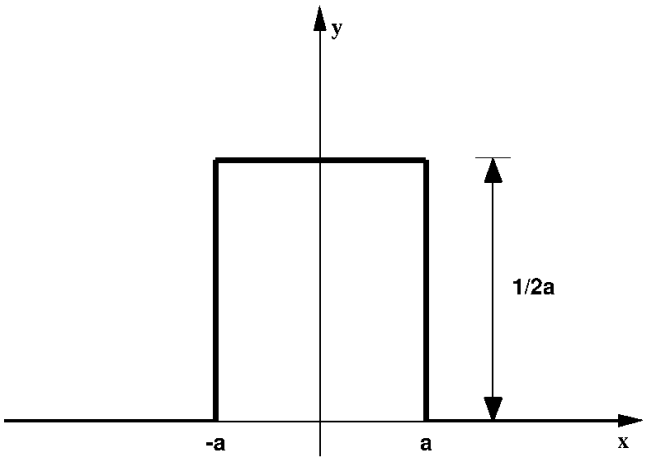


FIGURE 2.5: Impulse function

calculated as

$$\int_{-\infty}^{\infty} I_a(x)dx = 1 \tag{2.15}$$

Dirac delta function is special case of impulse function,

$$\delta(x) = \lim_{a \to 0} I_a(x) \tag{2.16}$$

with properties,

$$\delta(x) = \begin{cases} 0 & \text{x } \neq 0 \\ \infty & \text{x} = 0 \end{cases} \tag{2.17}$$

$$\int_{-\infty}^{\infty} \delta(x)dx = 1 \tag{2.18}$$

and a particularly important property

$$\int_{-\infty}^{\infty} f(x)\delta(x - x_i)dx = f(x_i) \tag{2.19}$$

Following this it can be concluded that the collocation method corresponds to the choice of the weighting functions as Dirac functions at the collocation points

$$w_i = \delta_i = \delta(x - x_i) \tag{2.20}$$

Collocation method requires no integration in the numerical procedure and generates the N equations required to evaluate the undetermined unknown coefficients α_j. As might be expected, the accuracy of this scheme depends heavily on the location of the collocation points.

2.5 Other methods

Previously mentioned methods are by far most used numerical methods with strong mathematical background and a wide range of applications in engineering problems.

However, of the aforementioned methods, FEM and FVM can be thought as the most popular and versatile discretization techniques. Methods such as FDM, discrete element method (DEM), boundary element method (BEM) which together with various meshfree, collocation and other hybrid approaches are practical only for limited types of problems. There are various other methods such as the spectral methods, belongs to the variational methods, that are especially adapted to the approximation of smooth solutions but are limited to simple geometries and methods using wavelets basis.

A DEM, also called a distinct element method, is an explicit numerical model which approximates the mechanical behavior of an assembly of arbitrary shaped particles. Simulation of millions of particles on a single processor became possible due to the advances in computing power and numerical algorithms for nearest neighbor. Because of that DEM is becoming widely accepted as effective method for solving problems in granular flows, rock mechanics, discontinuous materials (see [54]). One of the main problem of the DEM is that it is relatively computationally intensive, which limits either the number of particles or the length of a simulation. As such, the DEM has become accepted and widely used to model the mechanical behaviour and flow of particulate geomaterials.

Idea behind the BEM is that we can approximate the solution of the PDE by looking at the solution to the PDE on the boundary and then use that information to find the solution inside the computational domain. In other words, the approximate solution obtained by BEM of the PDE is an exact solution of the differential equation in the domain and is parametrized by a finite set of parameters living on the boundary. In order to develop the boundary element method however, it is necessary first to formulate an equivalent boundary integral equation to the governing equation [55]. It is very useful when domains are very large where for example a FEM approximation would have too many elements to be practical. There are several advantages but also disadvantages of using BEM over other numerical methods like FEM or FDM. Only the boundary of the domain needs to be discretized, especially in two dimensions where the boundary is just a curve. The solution in the inside part of the computational domain is approximated with a rather high convergence rate. The physically relevant data, in some applications, are given by the boundary values of the solution or its derivatives and not by the solution in the interior of the domain. However, boundary integral equations require the explicit knowledge of a fundamental solutions of the differential equation which is not always available, only for linear PDE with constant or some specifically variable coefficients. Furthermore, if the boundary is not smooth but has corner and edges, or if the boundary conditions are discontinuous, e.g. in mixed boundary value problems, the solutions of the boundary value problem will have singularities at the boundary.

Chapter 3

Spline basis functions

This chapter provides the mathematical background for two representative members of spline functions used in this work. The chapter starts with description of B-splines and NURBS followed by a description of up and Fup basis functions. Here, the procedure using the convolution theorem is presented for construction of the Fup functions that clearly demonstrates their close relationship to B-splines.

3.1 B-spline

Piecewise polynomial approximations are fundamental to many applications, but it is not straightforward to join the polynomial segments smoothly while keeping local flexibility. However, B-spline basis functions handle the smoothness constraints in a very elegant fashion and provide the base with great numerical properties.

The B-spline basis functions parametric space is local to "patches" rather than elements. In one dimensional domain, a knot vector is a set of non-decreasing real numbers representing coordinates in the parametric space of the curve

$$\Xi = \left\{\xi_1, \xi_2, ..., \xi_{n+p+1}\right\} \tag{3.1}$$

where ξ_i is the i-th knot, i is the knot index, $i = 1, 2, ..., n + p + 1$, p is the polynomial order of the B-spline, and n is the number of basis functions which comprise the B-spline. The interval $\left[\xi_1, \xi_{n+p+1}\right]$ is called a patch. If knots are equally-spaced in the parametric space, they are said to be uniform, and non-uniform otherwise. More than one knot can be located at the same coordinate in the parametric space, and are referred to as repeated knots. A knot vector is said to be open if its first and last knots appear $p + 1$ times.

The simplest example of an algebraic B-spline is the B-spline of the zero order $B_0(\xi)$:

$$B_0(\xi) = \begin{cases} 1 & \xi \in [-1/2, 1/2] \\ 0 & elsewhere \end{cases} \tag{3.2}$$

whose Fourier transform (FT) can be obtained in the following manner:

$$f_0(t) = \int_{-\infty}^{+\infty} B_0(\xi) \cdot e^{it\xi}\, d\xi = \int_{-1/2}^{+1/2} 1 \cdot cos(t \cdot \xi)\ dt = \frac{sin(t/2)}{t/2} \tag{3.3}$$

Since $B_0(\xi)$ (in the form (3.2)) has discontinuities in the points $\xi = \pm 1/2$, for practical application it can be expressed as a continuous function by the inverse FT. Applying the integral operator $\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-it\xi}dt$ on the right-hand side of the expression (3.3), FT is transformed into the function itself, i.e. into algebraic spline $B_0(\xi)$:

$$B_0(\xi) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{sin\,(t/2)}{t/2} \cdot e^{-it\xi}dt \tag{3.4}$$

Considering uniform distribution of the knots, B-splines of the n-th order according to the law - $\xi_k = k - (n+1)/2, k = 0, 1, ..., n+1$, can be presented as:

$$B_n(\xi) = \frac{1}{n!}\sum_{k=0}^{n+1}(-1)^k \cdot C_{n+1}^k \cdot \left(\xi + \frac{n+1}{2} - k\right)_+^n \tag{3.5}$$

where C_n^k are binomial coefficients:

$$C_n^k = \binom{n}{k} = \frac{(n)!}{(n-k)! \cdot k!} \tag{3.6}$$

For example, B-splines up to the second order can be presented according to (3.5) as follows:

$$\begin{aligned} B_0(\xi) &= (\xi + 1/2)_+^0 - (\xi - 1/2)_+^0 \\ B_1(\xi) &= (\xi + 1)_+^1 - 2(\xi)_+^1 + (\xi - 1)_+^1 \\ B_2(\xi) &= \left[(2\xi + 3)_+^2 - 3(2\xi + 1)_+^2 + 3(2\xi - 1)_+^2 - (2\xi - 3)_+^2\right]/8 \end{aligned} \tag{3.7}$$

Figure 3.1 shows that the compact support of $B_n(\xi)$ consists of $(n+2)$ knots and $(n+1)$ unit characteristic intervals. Moreover, $B_n(\xi)$ is presented by the local polynomial of the n-th order on each interval $[\xi_k, \xi_{k+1}]$. For instance, $B_2(\xi)$ has four knots and three characteristic intervals. Furthermore, by increasing the B-spline order, the length of its compact support also increases, and when $n \to \infty$, the length goes to infinity. The coordinate ξ_T is called the vertex of the basis function (point with maximum function value) and serves as the origin for the shifting of the basis functions along the ξ axis by the length of the characteristic interval. Figure 3.2 presents the cubic B-spline $B_3(\xi)$ with the first three derivatives.

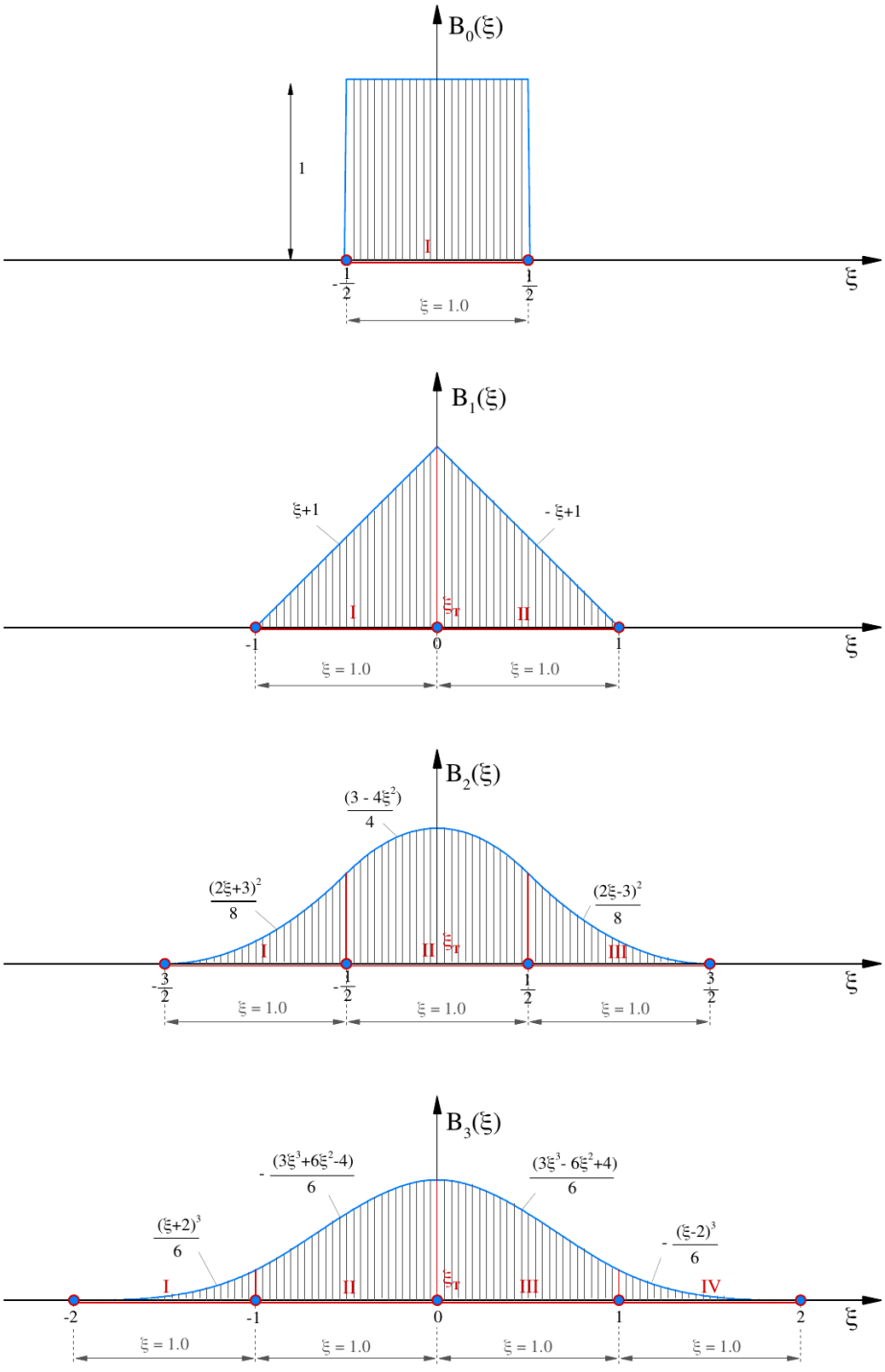


FIGURE 3.1: B-splines: $B_n(\xi)$, $n = 0, 1, 2, 3$.

Figures 3.1 - 3.3 presents the connection between B-splines and their derivatives. The first derivative of $B_3(\xi)$ can be presented as linear combination of contracted and shifted $B_2(\xi)$. Furthermore, the second derivative of $B_3(\xi)$ can be presented as a linear combination of contracted and shifted $B_1(\xi)$, and so on. Finally, each i-th derivative of $B_n(\xi)$ is a linear combination of the contracted and shifted $B_{n-i}(\xi)$.

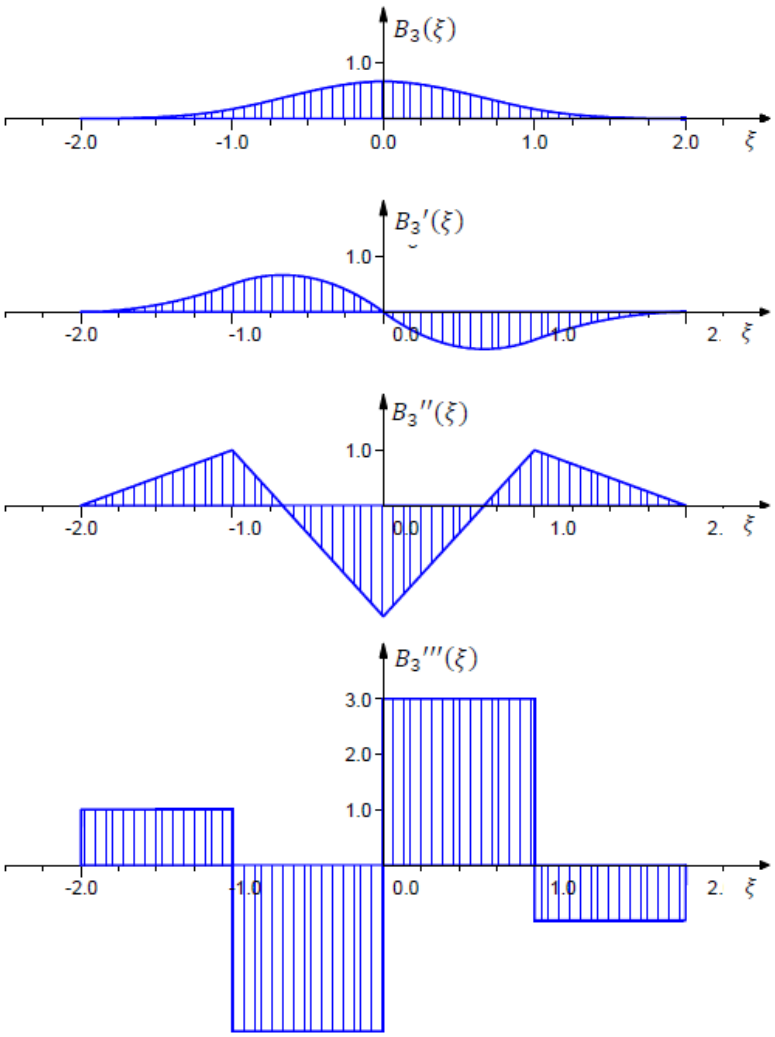


FIGURE 3.2: $B_3(\xi)$ with its first three derivatives.

Following the mentioned properties, $B_n(\xi)$ can be presented by convolution in the following form:

$$B_n(\xi) = \int_{-\infty}^{\infty} B_{n-1}(\xi - t)\, B_0(t)\; dt \tag{3.8}$$

or:

$$B_n(\xi) = B_{n-1}(\xi) * B_0(\xi) = \underbrace{B_0(\xi) * \ldots * B_0(\xi)}_{(n+1)\ \ times} \tag{3.9}$$

where n is the order of the B-spline and $B_0(\xi)$ is given by (3.2). The convolution theorem states that the Fourier transform (FT) of $B_n(\xi)$ can be expressed as a product of $(n+1)$ particular FT's of $B_0(\xi)$ according to (3.9):

$$f_n(t) = \left(\frac{sint/2}{t/2}\right)^{n+1} \tag{3.10}$$

so the inverse FT of $B_n(\xi)$ analogous to the expression 3.4 is defined by:

$$B_n(\xi) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left(\frac{sin\,(t/2)}{t/2} \right)^{n+1} \cdot e^{-it\xi} dt \tag{3.11}$$

Equation (3.9) implies that the support of $B_n\,(\xi)$ is the union of the $(n+1)$ characteristic intervals $\Delta\xi$. Figure 3.3 shows generation of B-splines according to (3.9) and the convolution theorem.

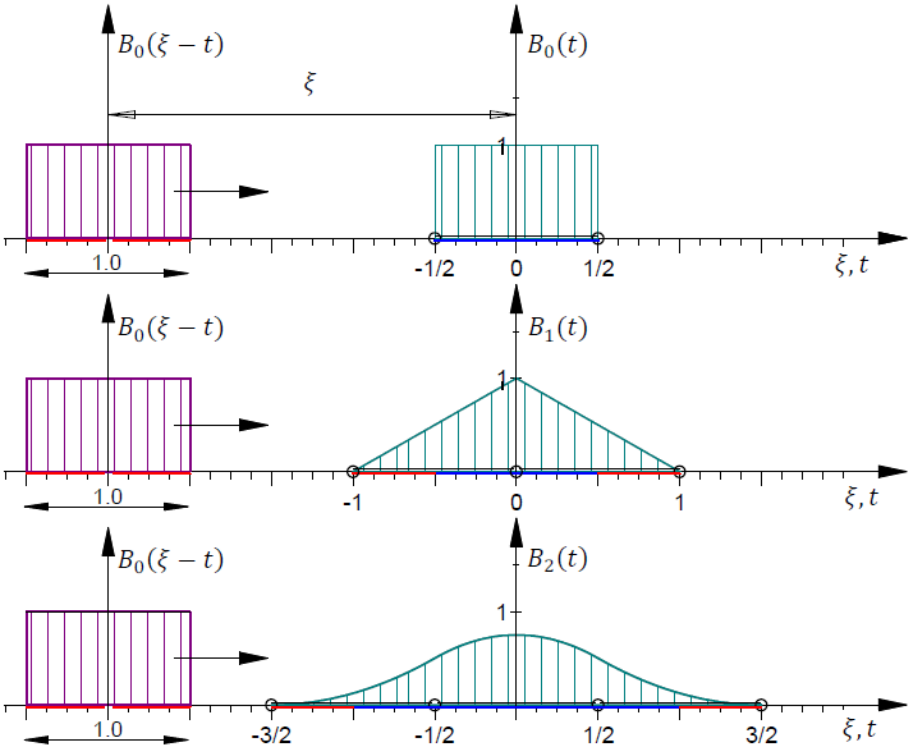


FIGURE 3.3: Generation of $B_2(\xi)$ using the convolution theorem.

Furthermore, this is not the only way to express B-splines. Having in mind knot vector, defined in the beginning of this section, B-spline basis functions are defined recursively (see Cottrell *et al.* [17]) starting with piecewise constants $(n = 0)$:

$$B_{i,0}(\xi) = \begin{cases} 1 & \xi_i \leq \xi < \xi_{i+1} \\ 0 & elsewhere \end{cases} \tag{3.12}$$

and for $n > 0$, B-splines are defined by

$$B_{i,n}(\xi) = \frac{\xi - \xi_i}{\xi_{i+n} - \xi_i} B_{i,n-1}(\xi) + \frac{\xi_{i+n+1} - \xi}{\xi_{i+n+1} - \xi_{i+1}} B_{i+1,n-1}(\xi). \tag{3.13}$$

Figure 3.4 presents B-spline basis functions for $n = 0, 1, 2$ on a uniform knot vector. An interesting fact is that standard piecewise constant and linear

finite element functions are the same for $n = 0, 1$. However, for higher-orders of B-spline basis functions they differ from their FEA counterparts.

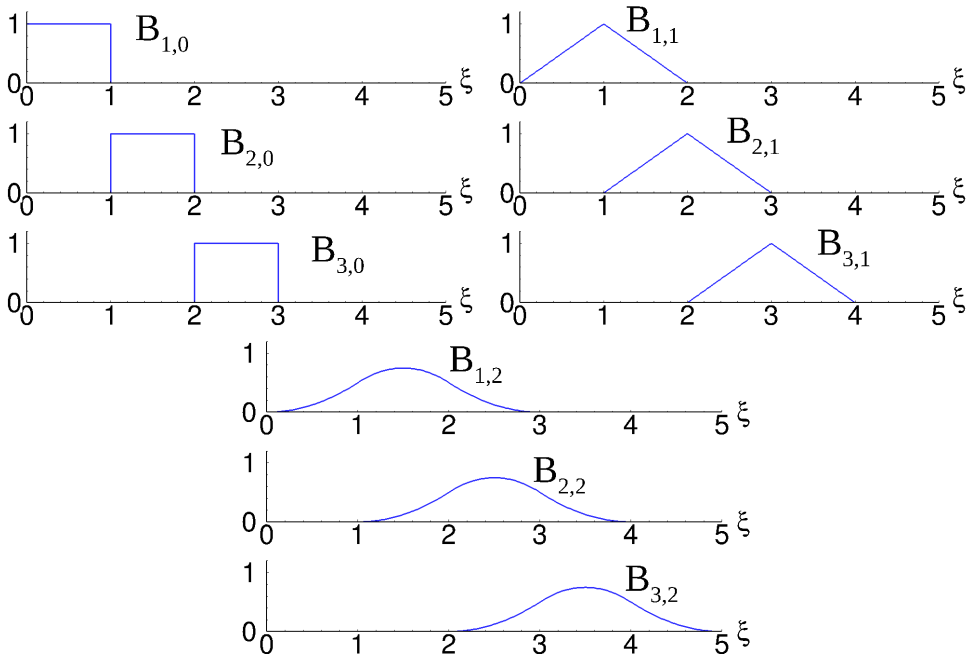


FIGURE 3.4: Basis functions of order 0, 1, and 2 for uniform knot vector $\Xi = \{0, 1, 2, ...\}$.

Finally, we can summarize the properties of the B-splines basis functions as follows:

1. B_n-spline is positive on $n + 1$ characteristic intervals and vanishes outside this interval i.e., B-splines have compact support where they have strictly positive non-zero values; elsewhere, they are zero, implying localized approximation properties.

2. B_n-spline is $(n - 1)$-times continuously differentiable with discontinuities of the n-th derivative.

3. A linear combination of shifted B_n-splines by a characteristic interval describes algebraic polynomials up to the n-th order.

4. A linear combination of m shifted B-splines by a characteristic interval describes a unit constant function ("partition of unity"), that is

$$\sum_{i=1}^{m} B_{i,n}(\xi) = 1 \tag{3.14}$$

5. B_n-splines can be presented by a linear combination of the shifted B-splines of the same order but using two-times-smaller support. This implies that B-splines support multiresolution analysis and efficient adaptive numerical procedures (e.g., [21]–[25], [31]–[34]). This can be

highlighted as the most interesting, since a hierarchical Fup basis function (following the properties of Fup basis functions) will be developed based on this property.

3.2 NURBS

Just like in classical finite element analysis (FEA) B-spline curves in $\mathbb{R}^d$ are constructed by taking a linear combination of B-spline basis functions. Many properties of B-spline curves follow directly from the properties of their basis functions. Here, the basis functions vector-valued coefficients are referred to as control points. These control points are somewhat identical to nodal coordinates in FEA by being the coefficients of the basis functions but the non-interpolatory nature of the base does not lead to a concrete interpretation of the values of the control points. Piecewise-polynomial B-spline curve can be described using m basis functions $B_{i,n}, i = 1, 2, ..., m$, and corresponding control points $P_i \in \mathbb{R}^d, i = 1, 2, ..., m$ by

$$C(\xi) = \sum_{i=1}^{m} B_{i,n}(\xi) P_i \tag{3.15}$$

where index i in P_i is not a reference to one of its d components, rather serves to identify the control point. Hence, a B-spline curve in essence is a mapping from a one dimensional parametric space to physical space. Control polygon is given by piecewise linear interpolation of the control points. Interesting property of the B-splines is the ability to intuitively change their shape by adjusting the control points. This power is used within Non-Uniform Rational B-splines (NURBS), by constructing a basis for the NURBS space from knot vectors and to build curves, surfaces and solids from linear combinations of basis functions and control points. In that view, everything that has been defined i.e., that applies to B-splines, also applies to NURBS. NURBS can be thought as extension of B-splines by associating a weight with each control point. For designers, NURBS are interesting for obtaining more control of the represented curve without increasing the number of control points or increasing the degree. Moreover, it also represent exactly some curves with conic sections, such as circles and ellipses.

NURBS are mathematical representations of 3-D geometry that can accurately describe any shape from simple 2-D line, arc, or curve to the most complex 3-D organic free-form surface or solid. Because of their generality, flexibility, accuracy, i.e., excellent properties NURBS models are the most popular representation method in CAD.

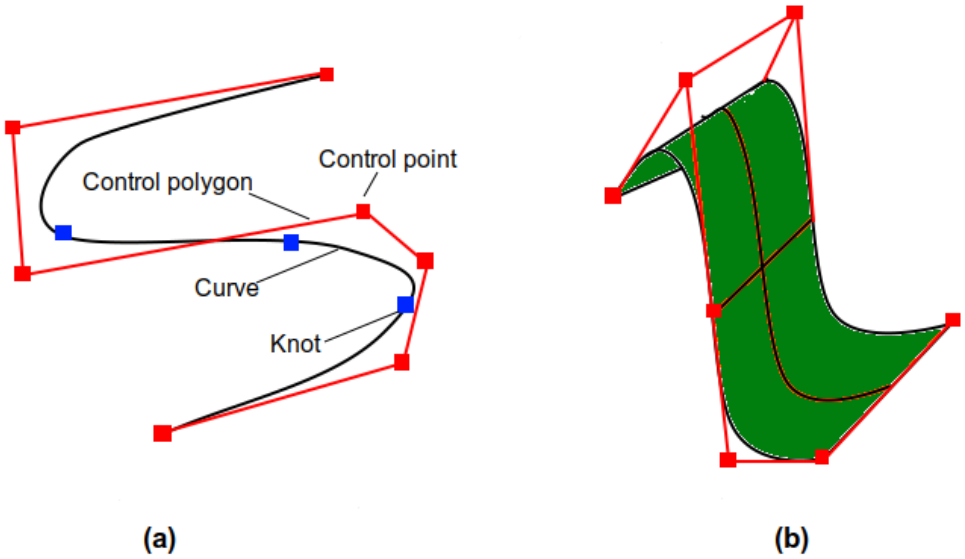


FIGURE 3.5: NURBS geometries. (a) NURBS curve; (b) NURBS surface.

NURBS curves and surfaces (see Figure 3.5) are generalizations of both B-splines and Bézier curves and surfaces, the primary difference being the weighting of the control points (red squares in Figure 3.5), which makes NURBS curves rational.

NURBS basis is given by

$$R_i^n(\xi) = \frac{B_{i,n}(\xi)w_i}{W(\xi)} = \frac{B_{i,n}(\xi)w_i}{\sum_{j=1}^{m} B_{j,n}(\xi)w_j} \tag{3.16}$$

where $W(\xi)$ is weighting function, $B_{i,n}(\xi)$ is the standard B-spline basis function and w_i is referred to as the weight factor. Using (3.16) in conjuction with the NURBS curve control points

$$(P_i)_j = \frac{(P_i^w)_j}{w_i}, \quad j = 1, ..., d \tag{3.17}$$

leads to an equation for a NURBS curve,

$$C(\xi) = \sum_{i=1}^{m} R_i^n(\xi)P_i \tag{3.18}$$

where $(P_i)_j$ is the j^{th} component of the vector B_i. Note that (3.18) is identical form to that for B-splines.

Weights play an important role in defining the basis, but they are divorced from any explicit geometric interpretation in this setting, and we are free to choose control points independently from their associated weights. Also note that if the weights are all equal, then $R_i^n(\xi) = B_{i,n}(\xi)$ and the curve is again a polynomial. Thus, B-splines are a special case of NURBS [17].

Some of the great qualities that makes NURBS ideal choice for computer-aided modeling are as follows:

1. have a well-known and a precise definition (one common mathematical form) that can accurately represent both standard geometric objects like lines, arcs, ellipses, etc., and free-form geometry like car bodies and human bodies.

2. the amount of information required for a NURBS representation of a piece of geometry is much smaller than the amount of information required by simpler methods

3. can be evaluated reasonably quickly by numerically accurate and stable algorithms.

3.3 Atomic basis functions

Fup basis functions belong to the class of atomic functions (see [37],[41]) and span vector space of algebraic polynomials, while their properties are closely related to the B-splines, as will be explained in the sequel.

Atomic basis functions (ABF) are infinitely derivative solutions of functionally differential equations in the form

$$Ly(x) = \lambda \sum_{k=1}^{M} C_k y(ax - b_k) \tag{3.19}$$

where L is a linear differential operator with constant coefficients, λ is a nonzero scalar value, C_k are coefficients of the solutions, $a > 0$ is the support length parameter of the finite function, b_k are the coefficients that determine the shifts of the finite basis functions. The type of the finite function from the class of atomic basis functions is determined by the choice of the operator L in (3.19). Thus, we distinguish atomic basis functions of the algebraic, exponential and trigonometric type.

One way to obtain $up(\xi)$ function, the simplest and basic atomic function, is by contracting $B_0(\xi)$ (see Eq. 3.2) to half the length of the compact support $(h_0/2)$ and thus obtaining another member of the convolution, then contracting it again to half the length of the compact support $(h_0/4)$ obtaining the third member of the convolution and so on, i.e. following convolution procedure:

$$up\,(\xi) = B_0\,(\xi) * B_0\,(2\xi) * \cdots * B_0\left(2^k\xi\right) * \cdots * B_0\,(2^\infty\xi) \tag{3.20}$$

Applying Paley-Wiener theorem (see [56]) in form $\int_{-\infty}^{+\infty} B_0(2^k\xi)d\xi = 1$, it follows that the ordinates of each subsequent member in (3.20) are doubled:

$$B_0(2^k\xi) = \begin{cases} 2^k & \xi \in [-2^{-k-1}, 2^{-k-1}] \\ 0 & elsewhere \end{cases}, \quad k = 0, 1, ..., \infty. \tag{3.21}$$

The function $up(\xi)$ is obtained by an infinite number of convolutions of the contracted $B_0(\xi)$ with compact support 2^{-k} and vertex value 2^k (see Eq. 3.20), as shown in Figure 3.6. According to (3.20), the compact support of $up(\xi)$ is the union of an infinite number of finite intervals. However, its compact support is finite:

$$h_{up} = \sum_{k=0}^{\infty} \frac{1}{2^k} = 2 \quad \rightarrow \quad \text{supp } up(\xi) = [-1,\ 1] \tag{3.22}$$

Rvachev and Rvachev [37] proved that the length of the compact support (3.22) can be presented as a distance metric of the set of binary-rational points 2^{-k}, while all other points as $\pm 1/3$, $\pm 4/7$, $\pm\sqrt{2}/2$, $\pm\pi/8$ contain zero metric length. The convolution procedure (3.20) causes $up(\xi)$ to contain all polynomial orders by parts of its compact support.

Fourier transform of the basic atomic function $up(\xi)$ corresponds to the product of an infinite number of Fourier transformations of the convolution factor

$$F_0(t) = \prod_{j=1}^{\infty} \frac{\sin(t/2^j)}{t/2^j}. \tag{3.23}$$

Due to its infinite number of continuous and non-zero derivatives, function $up(\xi)$ can be regarded as a perfect spline. However, it is still not an analytic function at any point of its support. Moreover, its finiteness is higher than that of B-splines. The mother atomic function $up(\xi)$ retains the good localized property of B-splines but also possesses the property of universality, as trigonometric or algebraic polynomials. Universality means that adding new basis functions in some approximation can only improve or at least not change the previous approximation which means that the new approximation contains old approximation as a vector subspace.

The values of $up(\xi)$ and its derivatives can be found exactly in the form of rational numbers in the binary-rational points. Those binary-rational points are defined as:

$$\xi_{br} = -1 + k \cdot 2^{-m}, \quad m \in N, \quad k = 1,\ \ldots, 2^{m+1}. \tag{3.24}$$

At all other points of the compact support calculation of $up(\xi)$ can be done only approximately, but up to the computer accuracy. Even though the calculation of $up(\xi)$ can use FT (3.23), Gotovac and Kozulić [41] found a more appropriate expression for the calculation of $up(\xi)$ values in binary-rational points:

$$up(\xi_{br}) = \frac{2^{-m(m+1)/2}}{m!} \sum_{j=1}^{k} \delta_j \sum_{\ell=0}^{[m/2]} C_m^{2\ell} \cdot (2(k-j)+1)^{m-2\ell} \cdot a_{2\ell} \tag{3.25}$$

where $C_m^{2\ell}$ are binomial coefficients, δ_j are coefficients that have sign according to the following recursive formulas:

$$\delta_{2k-1} = \delta_k, \quad \delta_{2k} = -\delta_k, \quad k \in N, \quad \delta_1 = 1 \tag{3.26}$$

and expression $[m/2]$ represents the maximum integer of the fraction within the brackets, and $a_{2\ell}$ are even moments of $up(\xi)$ defined by the following recursive formulas:

$$a_{2k} = \frac{(2k)!}{2^{2k}-1} \sum_{\ell=1}^{k} \frac{a_{2k-2\ell}}{(2k-2\ell)!(2\ell+1)!}, \qquad k \in N \;\; ; \;\; a_0 = 1 \tag{3.27}$$

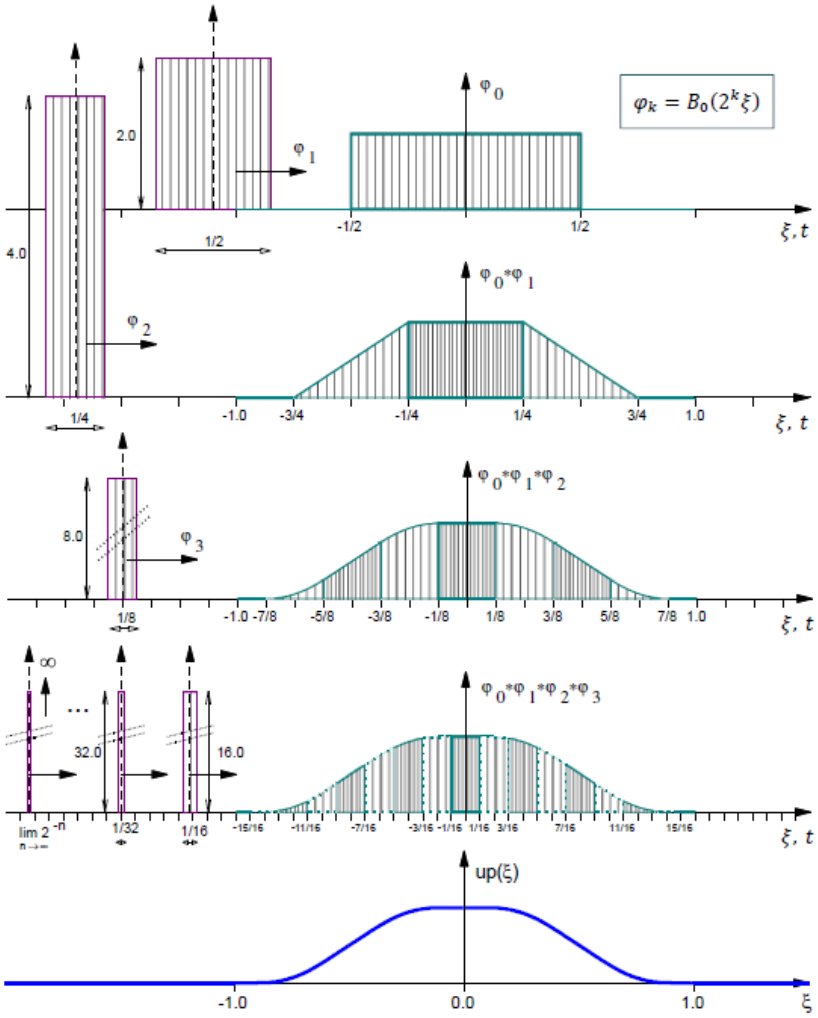


FIGURE 3.6: Generation of the function $up(\xi)$ using the convolution theorem (see Eq. 3.20).

For the calculation of $up(\xi)$ values at arbitrary points, Gotovac and Kozulić [41] suggested a special series based on Taylor series of the $up(\xi)$ function at the binary-rational points ξ_{br} (because it is then a polynomial of the n-th order). Values of the even function $up(\xi)$ in arbitrary point $\xi \in [0,1]$ can be presented as follows:

$$up\,(\xi) = 1 - up(\xi - 1) = 1 - \sum_{k=1}^{\infty} \; (-1)^{1+p_1+\cdots+p_k} \; p_k \sum_{j=0}^{k} C_{jk} \cdot \Delta_k{}^{j} \tag{3.28}$$

where the coefficients C_{jk} are rational numbers containing values of $up(\xi)$ at the binary-rational points $\xi_k = -1 + 1/2^m$ [41]:

$$C_{jk} = \frac{1}{j!} 2^{j(j+1)/2} up(-1 + 2^{-(k-j)}) \; ; \quad j = 0\,,\, 1\,,\, \ldots\,,\, k \; ; \; k = 1\,,\, 2\,,\, \ldots\,,\, \infty \tag{3.29}$$

Factor Δ_k in (3.28) presents the difference between the real value of coordinate ξ and its binary presentation with k bytes, where $p_1 \ldots p_k$ are the digits 0 or 1:

$$\Delta_k = \xi - \sum_{i=1}^{k} p_i \cdot \frac{1}{2^i} \tag{3.30}$$

This means that the obtained accuracy of the coordinate ξ reflects on the accuracy of $up(\xi)$ values at arbitrary points, which in turn depends on the computer accuracy. For a chosen m, the calculation error of $up(\xi)$ values at the arbitrary points ξ (equal to the residual of series (3.28) when k= 1 , ... , m), does not exceed the value of $up\,(-1+2^{-m})$. In this work, 2^{16} binary-rational points (*m=16*) were used, which means that the calculation error is of the order of $up\,\left(-1+2^{-16}\right) = 0.117 \cdot 10^{-51}$, which is significantly smaller than the computer accuracy. In practice, for all numerical calculations, it is sufficient to use (3.25) and 2^{16} binary-rational points (*m=16*) because its density enables interpolation of the value at any arbitrary point up to the computer accuracy.

For an exact description of polynomials up to the n-th order on the interval $\Delta\xi_n = 2^{-n}$, it is necessary to use 2^{n+1} basis functions obtained by shifting $up(\xi)$ for $\Delta\xi_n$. Such a relatively large number of basis functions implies poor approximation properties of $up\,(\xi)$. This is the main reason why application of $up(\xi)$ in numerical analysis for practical purposes is quite limited.

3.4 Fup basis functions

$Fup_n(\xi)$ are another class of atomic basis functions, also belonging to the polynomial types of basis functions, which require only (n+2) basis functions to exactly describe polynomials up to the n-th order on interval $\Delta\xi_n = 2^{-n}$. For instance, for the development of a 4-th order polynomial, only 6 or (n+2) $Fup_4(\xi)$ are needed in comparison to 32 $up(\xi)$ basis functions. The compact support of $Fup_n(\xi)$ contains n+2 characteristic intervals $\Delta\xi_n = 2^{-n}$:

$$supp \;\; Fup_n(\xi) = \left[-\,(n{+}2)\cdot 2^{-n-1},\; (n{+}2)\cdot 2^{-n-1} \right] \tag{3.31}$$

For $n = 0$, the following holds:

$$Fup_0(\xi) = up(\xi) \tag{3.32}$$

Function $Fup_n(\xi)$ can be defined as finite solution of differentially functional equations, in a similar way as a function $up(\xi)$. By analogous procedure as for function $up(\xi)$, general expression for the Fourier transform $F_n(t)$

of the function $Fup_n(\xi)$ is obtained by

$$F_n(t) = \left(\frac{sin(t/2^{-n-1})}{t/2^{-n-1}}\right) \prod_{j=n+2}^{\infty} \frac{\sin(t/2^j)}{t/2^j}. \tag{3.33}$$

Furthermore, function $Fup_n(\xi)$ can be obtained by a convolution procedure using the contracted B_n and up basis function:

$$Fup_n(\xi) = B_n(2^n\,\xi) * up\,(2^{n+1}\,\xi) \tag{3.34}$$

This means that $Fup_n(\xi)$ is closely related to $B_n(\xi)$ and that they together share all the mentioned properties. However, $Fup_n(\xi)$ has better approximation properties than $B_n(\xi)$ due to the convolution with the up function containing all orders of polynomials by parts and infinite continuity. Moreover, they share the same convergence properties because it is directly linked by the polynomial order which can be exactly described by linear combination of these functions. Additionally, the $Fup_n(\xi)$ has better approximation properties which are paid by one more characteristic interval for the same n-th order of basis functions. Equation (3.34) is not numerically favorable for calculating the value of the function $Fup_n(\xi)$.

Atomic basis functions have a "deeper" mathematical background, and they are generally solutions of differential-functional equations, which for $Fup_n(\xi)$ take the following form:

$$Fup_n'(\xi) = 2\sum_{k=0}^{n+2}\left(C_n^k - C_n^{k-2}\right)\cdot Fup_n\left(2\xi - \frac{k}{2^n} + \frac{n+2}{2^{n+1}}\right) \tag{3.35}$$

where C_n^k are binomial coefficients defined as

$$C_n^k = \binom{n}{k} = \frac{(n)!}{(n-k)!\cdot k!} \tag{3.36}$$

The relationship between $Fup_n(\xi)$ and the shifted $Fup_{n+1}(\xi)$ can be presented in the general form:

$$Fup_n(\xi) = \frac{1}{2^{n+1}}\sum_{k=0}^{n+1} C_{n+1}^k \cdot Fup_{n+1}\left(\xi - \frac{k}{2^{n+1}} + \frac{n+1}{2^{n+2}}\right) \tag{3.37}$$

Equations (3.35-3.37) present the atomic structure of these basis functions because a function and its derivatives are decomposed by a linear combination of these same functions (Rvachev and Rvachev [37]). $Fup_n(\xi)$ can be calculated by a linear combination of $up(\xi)$ mutually shifted by the characteristic interval 2^{-n}:

$$Fup_n(\xi) = \sum_{k=0}^{\infty} C_k(n)\cdot up\left(\xi - 1 - \frac{k}{2^n} + \frac{n+2}{2^{n+1}}\right) \tag{3.38}$$

The zero coefficient in (3.38) is:

$$C_0(\mathrm{n}) = 2^{C^2_{n+1}} = 2^{n(n+1)/2} \tag{3.39}$$

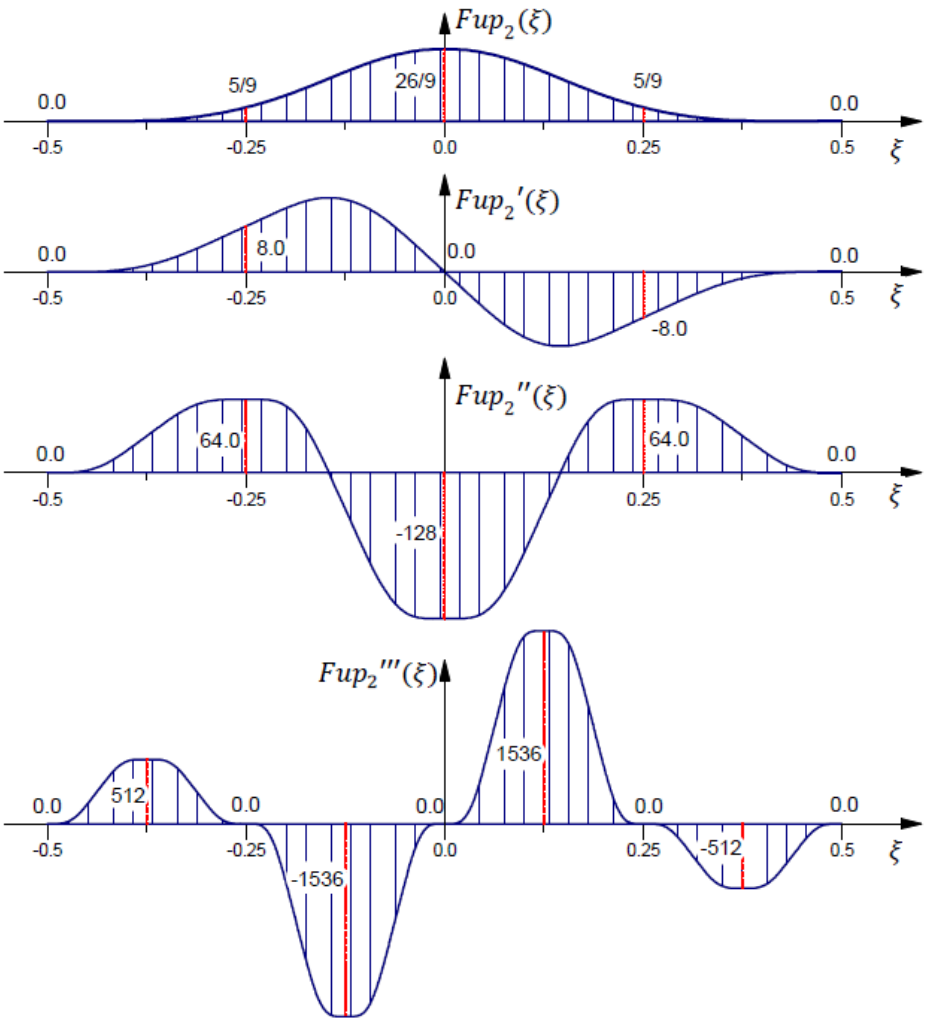


FIGURE 3.7: Function $Fup_2(\xi)$ with its first three derivatives.

Other coefficients are calculated in the form $C_k(n) = C_0(n) \cdot C_k'(n)$, where the coefficients $C_k'(n)$ are obtained using the following recursive formulas:

$$\begin{aligned} C_0'(n) &= 1 \\ C_k'(n) &= (-1)^k C^k_{n+1} - \sum_{j=1}^{min\left\{k\,;\,2^{n+1}-1\right\}} C_{k-j}'(n) \cdot \delta_{j+1} \end{aligned} \tag{3.40}$$

For example, Figure 3.8 shows function $Fup_4(\xi)$ and shifted $up(\xi)$ functions from which a linear combination yields $Fup_4(\xi)$ following (3.38). The compact support for function $Fup_4(\xi)$ according to (3.31) is $\left[-\frac{3}{16}, \frac{3}{16}\right]$, with zero coefficient $C_0 = 2^{C^2_5} = 1024$ and the rest of the coefficients (C_0') are calculated using (3.40).

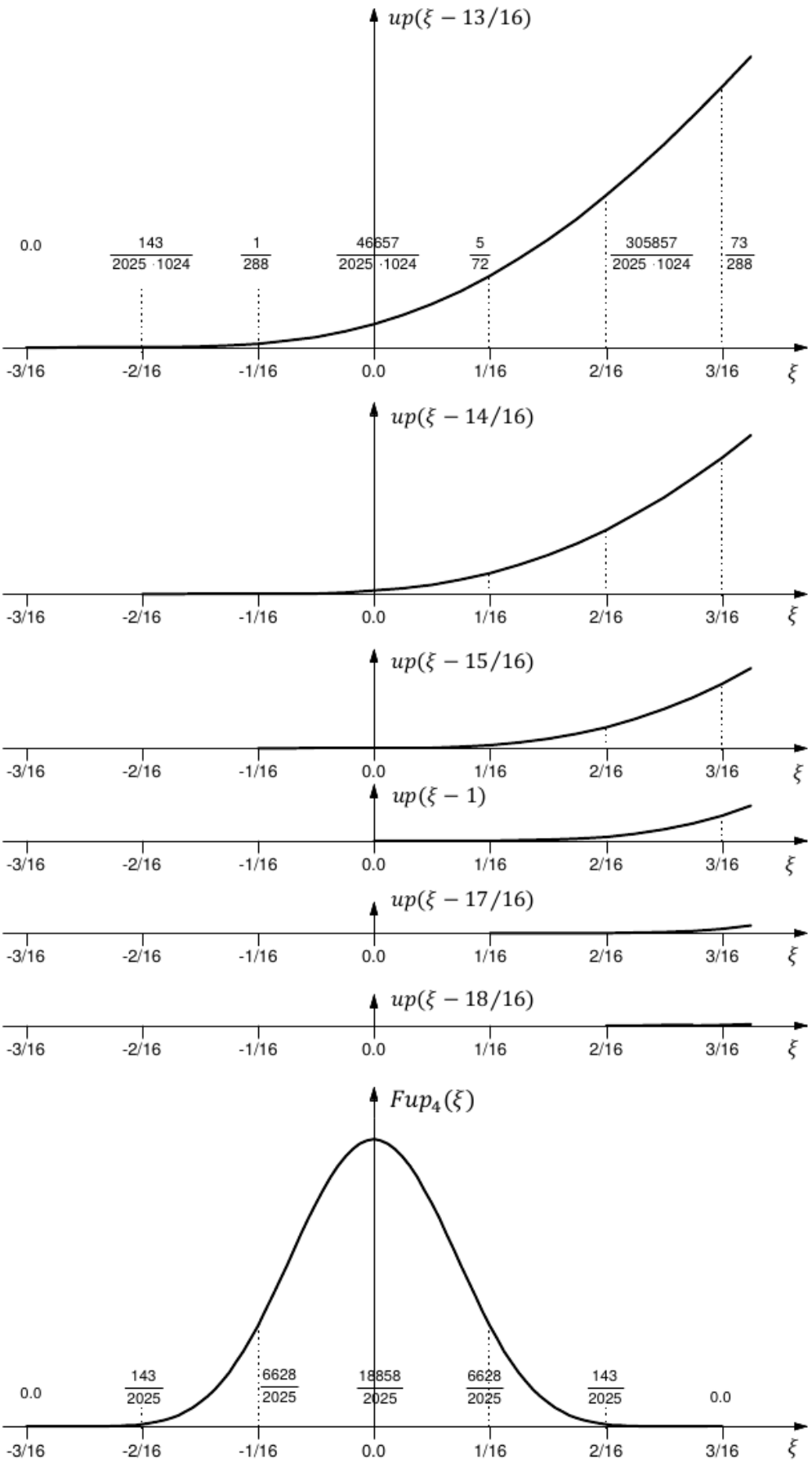


FIGURE 3.8: Function $Fup_4(\xi)$ as a linear combination of the shifted $up(\xi)$ functions.

Derivatives of $Fup_n(\xi)$ are calculated from (3.35) and (3.38). Figure 3.7 presents $Fup_2(\xi)$ with its first three derivatives, which has the same number of characteristic intervals as $B_3(\xi)$. However, the third derivative has no discontinuities due to the enhanced continuity of $Fup_2(\xi)$.

In the numerical modeling of boundary value problems, there is a need to modify boundary basis functions in order to keep the same approximation properties as inside the domain. The concept of boundary basis functions refers to the linear combination of basis functions whose compact supports are at least partially located inside the domain. For simpler notation, modified boundary Fup_n basis functions are designated as $\varphi_{n,j}$, $j = -[(n+1)/2], ..., [n/2]$ on the left domain boundary ξ_A, and $j = N - [n/2], ..., N + [(n+1)/2]$ on the right domain boundary ξ_B (N is the number of characteristic intervals $\Delta\xi_n$).

Modified boundary basis functions $\varphi_{n,j}$ are presented in the form of a linear combination of the original Fup_n basis functions, which are hereinafter referred to as $y_{n,i}(x)$, $i = -[(n+1)/2], ..., j$ on the left boundary and $i = j, ..., N + [(n+1)/2]$ on the right boundary.

The boundary basis functions $\varphi_{n,j}$ on the left domain boundary are modified so that i-th derivation is satisfied in a manner

$$\begin{aligned} &\varphi_{n,j}^{(i)}(\xi_A) \neq 0 \quad for \quad j + [(n+1)/2] \leqslant i \leqslant n \\ &\varphi_{n,j}^{(i)}(\xi_A) = 0 \quad otherwise; \qquad\qquad i \in \mathbb{N} \end{aligned} \tag{3.41}$$

Modification of the right boundary basis functions are achieved by translating and mirroring the left modified boundary basis functions.

In the vector space of mutually displaced Fup_n basis functions, it is necessary to modify the $(n+1)$ basis functions on the left and right boundary domain, whose compact supports are partially outside of the domain (Figure 3.9).

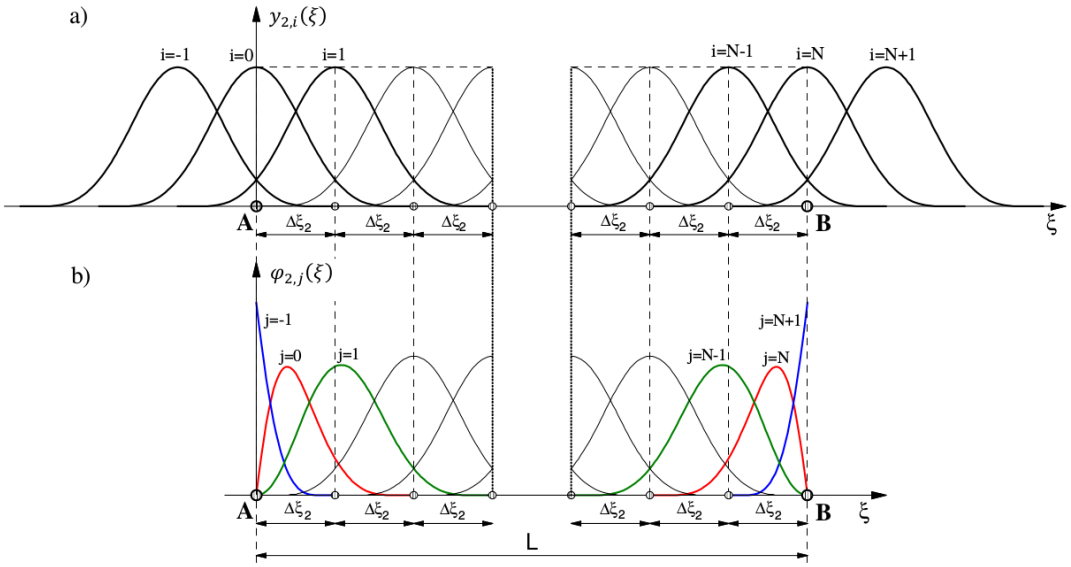


FIGURE 3.9: a) Linear combination of mutually displaced $y_{2,i}(\xi)$ basis functions and b) Modified boundary $\varphi_{2,j}(\xi)$ basis functions on the left and right boundary domain.

In Figure 3.9a a linear combination of basis functions $y_{2,i}(\xi)$ is shown, while Figure 3.9b shows the modified boundary Fup_2 basis functions at the left and right boundaries of the domain. Modified boundary Fup_2 basis functions on the left boundary are defined as

$$\begin{aligned}
\varphi_{2,-1}(\xi) &= \tfrac{36}{5} y_{2,-1}(\xi) \\
\varphi_{2,0}(\xi) &= -\tfrac{36}{5} y_{2,-1}(\xi) + \tfrac{18}{13} y_{2,0}(\xi) \\
\varphi_{2,1}(\xi) &= y_{2,-1}(\xi) - \tfrac{5}{13} y_{2,0}(\xi) + y_{2,1}(\xi)
\end{aligned} \tag{3.42}$$

while modified boundary Fup_2 basis functions on the right boundary are obtained by translating the left boundary basis functions for length $L = N \cdot \Delta\xi_2$ and their mirror symmetry around the right boundary, i.e. $\varphi_{2,j}(\xi) = \varphi_{2,j}(-\xi - L)$

$$\begin{aligned}
\varphi_{2,N-1}(\xi) &= y_{2,N+1}(\xi) - \tfrac{5}{13} y_{2,N}(\xi) + y_{2,N-1}(\xi) \\
\varphi_{2,N}(\xi) &= -\tfrac{36}{5} y_{2,N-1}(\xi) + \tfrac{18}{13} y_{2,N}(\xi) \\
\varphi_{2,N+1}(\xi) &= \tfrac{36}{5} y_{2,N+1}(\xi)
\end{aligned} \tag{3.43}$$

Modified Fup_n boundary basis functions of other orders are modified in the same way as presented Fup_2 (Figure 3.9), so that i-th derivation is satisfied following (3.41). Figure 3.10 shows $(n+1)$ modified basis functions with one in the middle (black curve) that is not modified because its whole compact support is inside the domain.

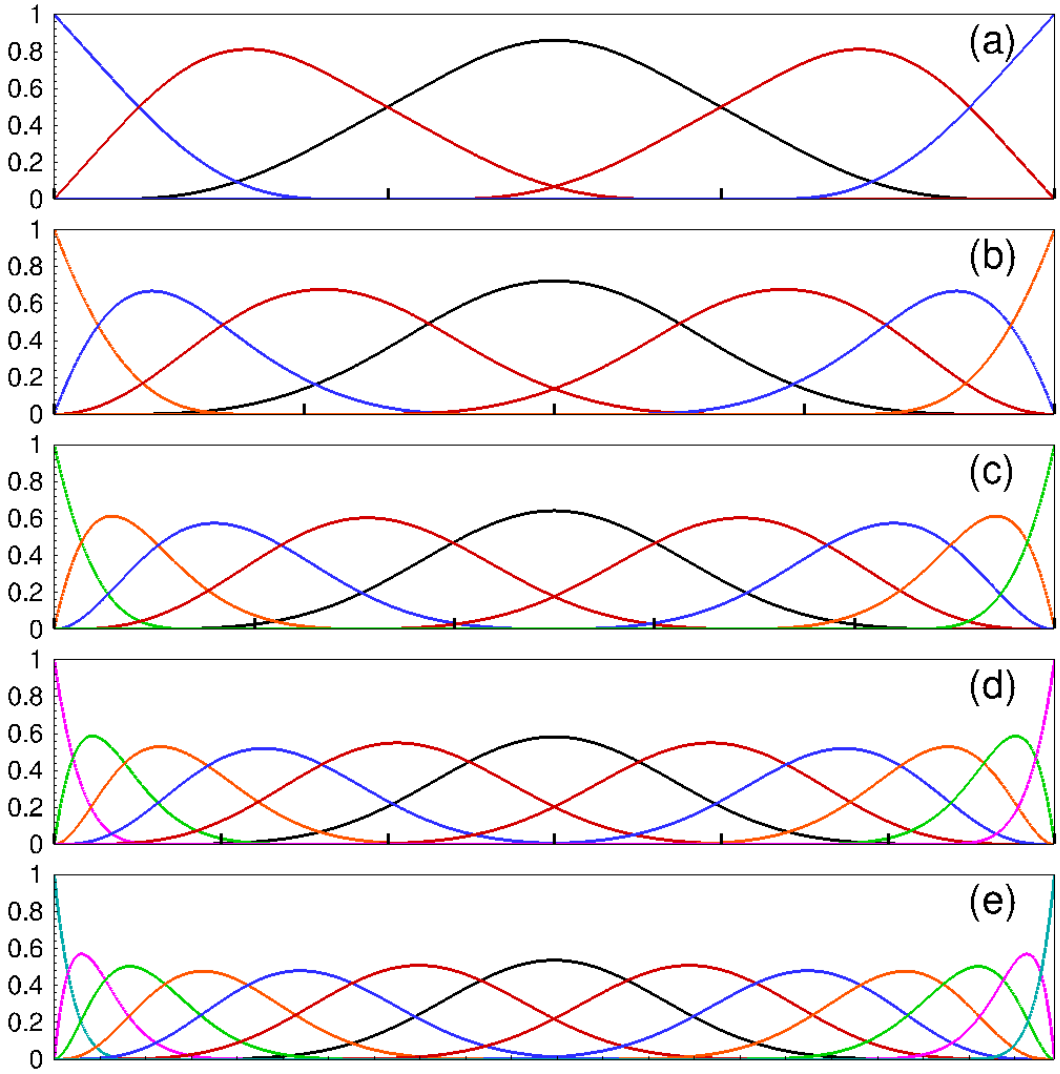


FIGURE 3.10: Modified $n+1$ Fup$_n$ boundary functions on the right and left boundary domain. (a) n=1; (b) n=2; (c) n=3; (d) n=4; (e) n=5.

Chapter 4

Isogeometric analysis (IGA)

This chapter serves as a short introduction to classical isogeometric analysis (IGA), followed by a description of three different numerical formulations (Galerkin, collocation and control volume) used for spatial discretization of a simple steady-state advection-dispersion problem. Galerkin and collocation IGA formulations are considered as a classical IGA formulations. The concept of IGA is presented as a unified framework for a multiscale description of the geometry and solution fields.

4.1 Introduction

Relatively recent ago, designers worked at drawing boards and designs were drawn with pencils and then passed to stress analysis. Ever since computers were introduced, times changed. Now designers generate CAD (Computed Aided Design) files which must be translated into analysis-suitable geometries, meshed and input to large-scale finite element analysis (FEA) codes. This is not trivial and for complex engineering designs it is estimated to take over 80% of the overall analysis time [14]. Design of sophisticated engineering systems is based on a wide range of simulation methods and computational analysis, such as fluid dynamics, structural mechanics, electromagnetics, acoustics, heat transfer, etc. The integration of FEA and CAD has proven to be a considerable problem and some fundamental changes have taken place to fully integrate analysis processes and engineering design.

Obviously, the way to break down the barriers between engineering design and analysis is to reconstitute the whole process, while at the same time maintaining compatibility with existing practice. The basic step is to focus on one and only one geometric model, which can be used directly as an analysis model or from which geometrically precise analysis models can be built automatically. This will require a change from classical FEA to an analysis procedure based on CAD representations. This concept is introduced by Hughes *et al.* [14] (see also Cottrell *et al.* [17]) and is referred to as Isogeometric analysis (IGA).

The main idea of IGA is to bridge the gap between finite element analysis (FEA) to describe a numerical solution and a CAD system to describe geometry using the same type of smooth and higher-order (spline) basis functions for both systems. The core of IGA is the isoparametric concept, widely used in classical FEM, where the basis functions used to approximate the solution

fields are also used to describe the geometry. IGA turns this idea in a different direction and selects the basis used to describe geometry in CAD systems as a basis for numerical approximation of the unknown solution fields. The main difference is that, in contrast to classical FEA, where the geometry is only approximated, IGA allows an accurate representation of the geometry in a CAD sense. Furthermore, in addition to clear advantages for describing geometry, spline basis functions (hierarchical splines) allows for adaptive local improvement of the solution.

4.2 Geometry description

There are a number of computational geometry spline candidates that may be used in isogeometric analysis. The most widely used in engineering design are B-splines and NURBS. The greatest strengths of NURBS are that they can exactly represent all conic sections, and therefore circles, cylinders, spheres, etc., they are convenient for free-form surface modeling, and there exist great number of numerically stable algorithms that generate NURBS objects. Moreover, they posses useful mathematical properties such as C^{n-1}-continuity for nth order NURBS, ability to be refined through knot insertion and convex hull properties. Besides NURBS and B-splines, there are also T-splines (see Sederberg *et al.* [15]). They extend NURBS to permit local refinement and coarsening, and are very robust in their ability to efficiently sew together adjacent patches. Moreover, T-splines are used to generate analysis-suitable models for arbitrary topological complexity [57], can be locally refined [58] and are capable of significantly reducing the number of the superfluous control points.

It should be noted that in FEA there is one notion of a mesh and one notion of an element, but also that one element has two representations, one in the parent domain and one in physical space. Degrees of freedom of the elements are usually the values of the basis functions at the nodes, and elements are usually defined by their nodal coordinates. Finite element basis functions, often referred to as "interpolation functions" or "shape functions", are typically interpolatory and may take on positive and negative values. However, for example in NURBS, the basis functions are usually not interpolatory and there are two notions of meshes, the control mesh and the physical mesh. The control points (see Figure 4.1) define the control mesh and the control mesh interpolates the control points. The control points enables the designer to create a wide range of desired objects, for instance, in the aviation or car industry. The control mesh consists of multilinear elements and does not conform to the actual geometry. Instead, it can be described like a scaffold, that controls the geometry. Control variables that defines the control mesh are the degrees of freedom that are located at the control points (red circles on the Figure 4.1). Just like the control elements may be degenerated to more primitive shapes (triangles in 2D or tetrahedra in 3D), the control mesh can also be distorted and even, to some extent, inverted, while at the same time the physical geometry my still remain valid for enough smooth NURBS.

The physical mesh, i.e., decomposition of the actual geometry, consists of two types of elements, the patch and the knot span (see Figure 4.1). The patch may be thought of as a macro-element or subdomain. While there are multiple patches in FEM (one element one patch) in IGA most geometries, for academic test cases, can be modeled with a single patch. Each patch has two representations, one in physical space and one in a parent domain. Patches in two-dimensional topologies are rectangles (see Figure 4.1), and in three dimensions are a cuboid in the parent domain representation. Patches can be decomposed into knot spans which are bounded by knots which are points, lines and surfaces in 1D, 2D, and 3D topologies, respectively.

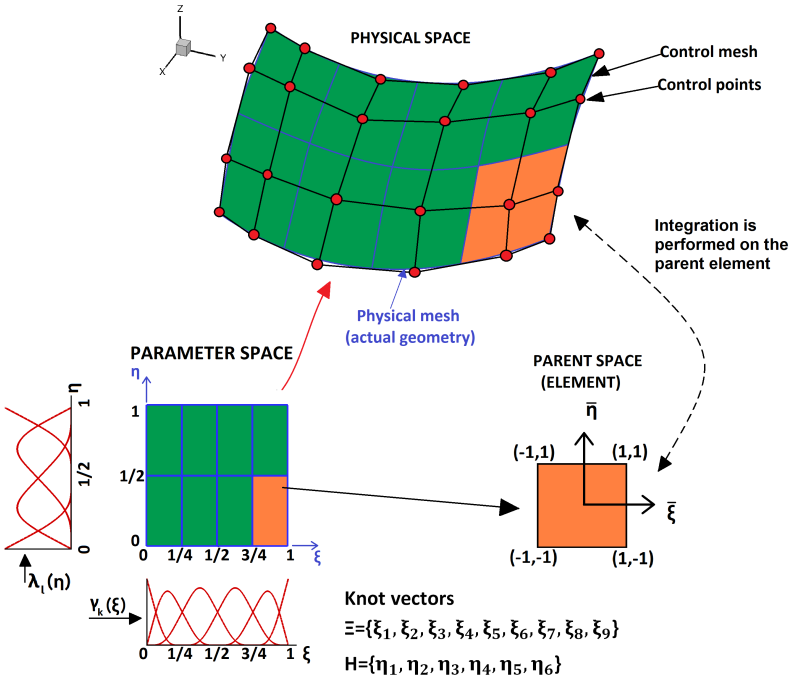


FIGURE 4.1: Schematic illustration of isogeometric analysis (IGA): physical space with control points and control mesh, parameter space with spline basis functions and related parent elements, knot vectors, and index space.

Figure 4.1 shows schematic illustration of IGA how one 2D subdomain or patch is transformed from the parameter (virtual) space to the physical (real) space using following spline representation

$$x(\xi,\eta) = \sum_{j=1} x_j \phi_j(\xi,\eta); \quad y(\xi,\eta) = \sum_{j=1} y_j \phi_j(\xi,\eta) \tag{4.1}$$

where x_j and y_j are the coordinates of the control points $\mathbf{B}(x_j, y_j)$ in the physical space, while ξ and η represents the coordinates in the parameter space. However, the main part of (4.1) are spline basis functions ϕ_j which in classic IGA are B-splines and NURBS. It is clear from (4.1) that IGA operates only with basis functions in the virtual regular domain since transformations from

the virtual to real, and vice versa are defined by the Jacobian

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \sum_{j=1}^{N} \begin{bmatrix} \frac{\partial \phi_j}{\partial \xi} x_j & \frac{\partial \phi_j}{\partial \xi} y_j \\ \frac{\partial \phi_j}{\partial \eta} x_j & \frac{\partial \phi_j}{\partial \eta} y_j \end{bmatrix} \tag{4.2}$$

and its inverse

$$J^{-1} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix} = \frac{1}{\det J} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix}, \tag{4.3}$$

as in classic FEM. However, the main difference is that IGA considers the transformation of each patch, which can be thought of as a macro-element or a subdomain, while the FEM performs transformations for each element [17].

The numerical solution in the virtual domain is also described by independent set of spline basis functions

$$u(\xi, \eta) = \sum_{j=1} \alpha_j \varphi_j(\xi, \eta) \tag{4.4}$$

It should be noted that number and order of the basis functions in the (4.1) and (4.4) may not be the same.

4.3 Three IGA formulations

In the following, the discretization process will be presented by considering a simple steady-state advection-dispersion equation (ADE) in the form:

$$\nabla \cdot (D \nabla u(\mathbf{x})) - \nabla \cdot (v u(\mathbf{x})) = 0 \quad in \quad \Omega \tag{4.5}$$

with appropriate boundary conditions:

$$u(\mathbf{x}) = u_D \quad on \quad \Gamma_D \tag{4.6}$$

$$(D \nabla u(\mathbf{x})) \cdot \mathbf{n} = q_N(\mathbf{x}) \quad on \quad \Gamma_N \tag{4.7}$$

where $u(\mathbf{x})$ represents the dependent variable, while the first (D) and second (v) term in Equation (4.5) represent influence of the dispersive (diffusive) and advective (convective) flux, respectively, which in general may be function of time, space and/or an unknown solution. Domain boundaries under the Dirichlet and Neumann boundary conditions are Γ_D and Γ_N, respectively, and $\mathbf{n}$ is the outward normal vector.

Method of weighted residuals can be thought as a general approach for deriving the different numerical formulations. The main idea is to integrate differential equation (4.5) over the domain of interest and multiply it by a finite number of weighting (test) functions $w_i(\mathbf{x})$:

$$\int_\Omega \nabla \cdot (D \nabla u(\mathbf{x})) w_i(\mathbf{x}) d\Omega - \int_\Omega \nabla \cdot (v u(\mathbf{x})) w_i(\mathbf{x}) d\Omega = 0 \tag{4.8}$$

where the number of test functions (w_i) is generally the same as the number of basis functions, and because of that the number of equations is equal to the number of unknowns.

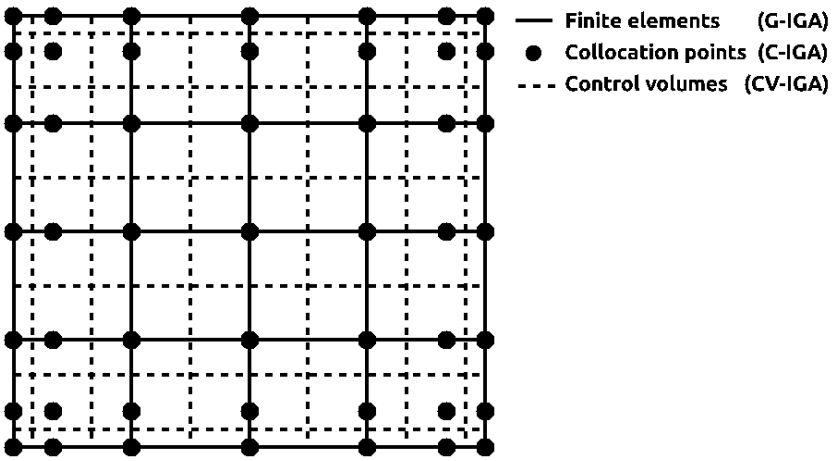


FIGURE 4.2: Discretization of 2D domain with three different IGA formulations.

Two most used formulations in IGA [14], [17] are Galerkin and collocation formulations using B-splines or NURBS as basis and test functions. However in this work besides those two methods, new formulation control volume within IGA (CV-IGA) will be introduced. Furthermore, Fup basis functions and the CV formulation are used, and referring to this method as control volume isogeometric analysis (CV-IGA), first developed using only low order Fup_1 and uniform non-adaptive discretization in [52] and applied to multiphysics model of the flow in karst aquifers. The main idea of CV-IGA is to utilize the powerful approximation and adaptive properties of Fup basis functions for numerical solutions of engineering problems that arise in the field of fluid mechanic with the conservation properties of the CV formulation (as in control volume finite element method; [59]–[61]). The CV formulation enables local and global conservation properties, with the computational cost between Galerkin (high-cost) and collocation (low-cost) (see [49]). One way to derive the three formulations, which will be presented in this paper, is to use different weighting (test) functions.

4.3.1 Galerkin formulation (G-IGA)

By using the same functions to construct both the trial (basis) functions and weight (test) functions spaces, the classical Galerkin formulation is obtained. In G-IGA, the spline basis functions φ_i, usually B-splines or NURBS, used as the weighting functions, are also used for the solution and geometry description. By setting $w_i(\mathbf{x}) = \varphi_i(\mathbf{x})$ and substituting it into equation (4.8) gives

$$\int_\Omega \nabla \cdot (D\nabla u(\mathbf{x}))\varphi_i(\mathbf{x})d\Omega - \int_\Omega \nabla \cdot (vu(\mathbf{x}))\varphi_i(\mathbf{x})d\Omega = 0, \tag{4.9}$$

and by applying integration by parts and divergence theorem (Green-Gauss-Ostrogradski theorem), the weak form of (4.9) is obtained:

$$\begin{aligned}&\int_{\Omega}(D\nabla u(\mathbf{x}))\nabla\varphi_i(\mathbf{x})d\Omega-\int_{\Omega}(vu(\mathbf{x}))\nabla\varphi_i(\mathbf{x})d\Omega=\\&\int_{\Gamma_D}(D\nabla u(\mathbf{x})-vu(\mathbf{x}))\mathbf{n}w_i(\mathbf{x})d\Gamma+\int_{\Gamma_N}q_N\varphi_i(\mathbf{x})d\Gamma\end{aligned}\tag{4.10}$$

Weak formulation (4.10) assumes that Neumann conditions can be directly imposed. However, test functions need to be zero on Dirichlet boundary so that Dirichlet boundary integral disappear at the right side in (4.10). In the classical FEM, the Dirichlet boundary conditions are satisfied by imposing solution values in the finite element nodes. It is possible because, classical FEM uses Lagrangian basis functions which are collocated in finite element nodes. Since in IGA multidimensional domains, higher-order spline basis functions have no such Kronecker property on Dirichlet boundary, special treatment of essential boundary conditions is needed.

By expressing solution $u(\mathbf{x})$ as a linear combination of spline basis functions ($u(\mathbf{x}) = \alpha_j\varphi_j(\mathbf{x})$), the weak form (4.10) can be obtained in the final discrete form. The maximum number of nonzero basis functions (φ_j) for a given discretized equation is determined by the polynomial order (n) of the basis functions. The main difference from the FEM is that the control points are not necessarily located at the element corners (see Figure 4.2) and the fact that the control variables, i.e. coefficients of the linear combination α_j cannot be interpreted as nodal values due to the non-interpolatory nature of spline basis functions.

4.3.2 Collocation formulation (C-IGA)

The collocation approach is different from other two mentioned methods because it is carried by using the Dirac delta function (see Eq. (2.20)), producing:

$$\int_{\Omega}\nabla\cdot(D\nabla u(\mathbf{x}))\delta(\mathbf{x}-\mathbf{x}_i)d\Omega-\int_{\Omega}\nabla\cdot(vu(\mathbf{x}))\delta(\mathbf{x}-\mathbf{x}_i)d\Omega=0\tag{4.11}$$

Due to the properties of the Dirac delta function, demanding integration procedure is eliminated. Because of that, at each i-th internal collocation point the governing PDE (4.5) is satisfied in the strong differential form:

$$\nabla\cdot(D\nabla u(\mathbf{x}_i))-\nabla\cdot(vu(\mathbf{x}_i))=0.\tag{4.12}$$

By differentiating equation (4.12) and obtaining,

$$D\nabla^2u(\mathbf{x}_i)-v\nabla u(\mathbf{x}_i)=0\tag{4.13}$$

it is obvious that strong collocation formulation (4.11) requires at least C^2 continuity of the basis functions, whereas the Galerkin form (4.10) required only C^1 continuity.

Collocation greatest strength comes from the fact that it avoids costly numerical integration and the fact that the number of nonzero elements is significantly reduced compared with the Galerkin formulation. Because of that C-IGA can be competitive with G-IGA in terms of computational cost for a given accuracy, especially for higher-order spline basis functions, which is proven by Schillinger *et al.* [62]. Another difference from Galerkin formulation is that collocation formulation satisfies the boundary conditions in the strong sense, i.e., both the Dirichlet and Neumann conditions are satisfied exactly.

The choice of the collocation points in not an easy task. It is of the most important components for the success of the collocation method. In recent years, considerable progress has been made on this topic especially within C-IGA and various sets of collocation points. Most notably being Greville points (abscissae), knot maxima and Demko points. Because of their simple definition and stable procedure, the Greville points have been widely adopted as the default choice (e.g., [62]).

In case of the B-splines of order n, the Greville points are defined to be the mean location of $n-1$ consecutive knots in the knot vector for each basis spline function of order n [63]. Since Fup basis functions have one more characteristic interval for the same order, the grid points of the Greville abscissae calculated for the B_n correspond to the Greville abscissae grid points of the Fup_{n-1}. The Greville abscissa (Figure 4.2 - black circles) for the Fup_n basis functions can easily be computed from a knot vector $\Xi = \{\xi_1, \xi_2, ..., \xi_{m+n+2}\}$

$$\hat{\xi}_i = \frac{1}{n+1}(\xi_{i+1} + ... + \xi_{i+n+1}), \quad i = 1, ..., m \tag{4.14}$$

where n is the order of the basis functions, and m is the number of basis functions.

4.3.3 Control volume formulation (CV-IGA)

The control volume formulation is performed by firstly dividing the domain of interest by m control volumes (see Figure 4.2) $(\Omega_i; i = 1, ..., m)$. CV formulation [10] uses test functions defined in the following form:

$$w_i(x) = \begin{cases} 1 & \mathrm{x} \in \Omega_i \\ 0 & \mathrm{x} \notin \Omega_i \end{cases}, \Omega_i \in \Omega. \tag{4.15}$$

Substituting (4.15) in (4.8), the direct integral form of the governing equation is recovered:

$$\int_{\Omega_i} \nabla \cdot (D \nabla u(\mathbf{x})) d\Omega - \int_{\Omega_i} \nabla \cdot (v u(\mathbf{x})) d\Omega = 0 \tag{4.16}$$

It should be noted that the integration is performed only over the i-th control volume (CV) due to the properties of the test functions (4.15). The surface

integrals at left side over the control volume are transformed into a line integrals across Ω_i boundaries Γ_i using Gauss's theorem,

$$\int_{\Gamma_i} \big(D\nabla u(\mathbf{x})\big)\boldsymbol{n}d\Gamma - \int_{\Gamma_i} \big(vu(\mathbf{x})\big)\boldsymbol{n}d\Gamma = 0 \tag{4.17}$$

where $\boldsymbol{n}$ is outward normal vector, thus obtaining the conservative form.

Finally, weak formulation (4.17) is defined on each control volume using spline basis functions and test functions (4.15) in order to get fully discretized control volume formulation:

$$\alpha_j \left[\int_{\Gamma_i} \big(D\nabla \varphi_j(\mathbf{x})\big)\boldsymbol{n}d\Gamma_i - \int_{\Gamma_i} \big(v\varphi_j(\mathbf{x})\big)\boldsymbol{n}d\Gamma_i\right] = \int_{\Gamma_{N_i}} q_N d\Gamma_N \tag{4.18}$$

where i is the subscript which denotes index of control volumes and row of stiffness matrix, while j is the subscript which denotes index of spline basis functions and column of the stiffness matrix. It is valid for all internal and boundary control volumes with Neumann boundary conditions. However, as in G-IGA, Dirichlet boundary control volumes requires special treatment of essential boundary conditions.

Conservation is an interesting feature of the control volume formulation. The conservation is exactly satisfied over any control volume (local conservation), as well as over the whole computational domain (global conservation). Furthermore, even the coarse-mesh solution exhibits an exact integral balance [10].

CV-IGA requires cheaper numerical integrations then G-IGA because control volume formulation (4.17) requires only integration over CV boundaries Γ_i, while Galerkin formulation (see Eq. (4.9)) requires (full) integration over the part of the domain where the particular test function is defined. Furthermore, the number of nonzero basis functions for each discretized equation in CV-IGA is lower then in G-IGA, thus the cost for the solution of the system of equations is generally lower then that for G-IGA. For comparison, the number of nonzero basis functions for CV-IGA for each discretized equation is $(n+2)^{dim}$ for odd order of basis functions and $(n+3)^{dim}$ for even, whereas for G-IGA this number is defined by $(2n+3)^{dim}$, where dim denotes the dimensionality of the problem.

Chapter 5

Adaptive techniques based on B-spline and IGA

This chapter introduces hierarchical B-splines (HB) as well as truncated hierarchical B-splines (THB) as adaptive technique used within IGA for local refinement. HB consists of defining a suitable set of basis functions on different hierarchical levels. However, this strategy can be improved using truncation operation to recover partition of unity and to have superior stability properties, giving rise to the THB. Giannelli *et al.* [16] introduced THB-splines with aim to reduce the overlap of basis functions within HB by choosing a spline basis with smaller supports. The reduced overlap yields a sparser stiffness matrix which results with less time required for its assembly, less memory to store it, and potentially less time for solving the discrete problem.

The analogue of *h*-refinement is knot insertion, whose insertion does not change a curve geometrically or parametrically. Consider a knot vector, $\Xi = \{\xi_1, \xi_2, ..., \xi_{m+n+1}\}$, and let $\overline{\xi} \in [\xi_k, \xi_{k+1}]$ be a desired new knot. By inserting new knot, the new $m+1$ basis functions are formed recursively following (3.12) and (3.13) with new knot vector $\Xi = \{\xi_1, \xi_2, ..., \xi_k, \overline{\xi}, \xi_{k+1}, ..., \xi_{m+n+1}\}$ (see Figure 5.1a). It should be noted that each unique internal knot value may apper no more then n times otherwise the curve becomes discontinuous.

Figure 5.1a presents an example of a knot insertion. Starting knot vector for the quadratic B-spline basis functions is $\Xi^0 = \{0,0,0,1,2,3,3,3\}$. A new knots are inserted at $\overline{\xi_1} = 0.5, \overline{\xi_2} = 1.5$ and $\overline{\xi_3} = 2.5$, which gives 3 new basis functions in the new layout (see Figure 5.1a). After inserting new knots $\overline{\xi_i}$, $i = 1,2,3$, knot vector is changed, thus giving a new knot vector $\Xi^1 = \{0,0,0,\frac{1}{2},1,\frac{3}{2},2,\frac{5}{2},3,3,3\}$. This process of enriching the solution space can be repeated by adding more basis functions of the same order. This subdivision strategy is seen to be analogous to the classical *h*-refinement strategy in finite element analysis.

In previous refinement procedure, the polynomial order of the basis functions did not increase while inserting a knot(s). However, there is a strategy of order elevation (*p*-refinement) where the polynomial order of basis functions may be increased without changing the geometry or parameterization. Furthermore it should be noted that each unique knot value in Ξ must be repeated to preserve discontinuities in the nth derivative of the curve that is being elevated. This strategy of order elevation is an analogue of *p*-refinement in the FEA.

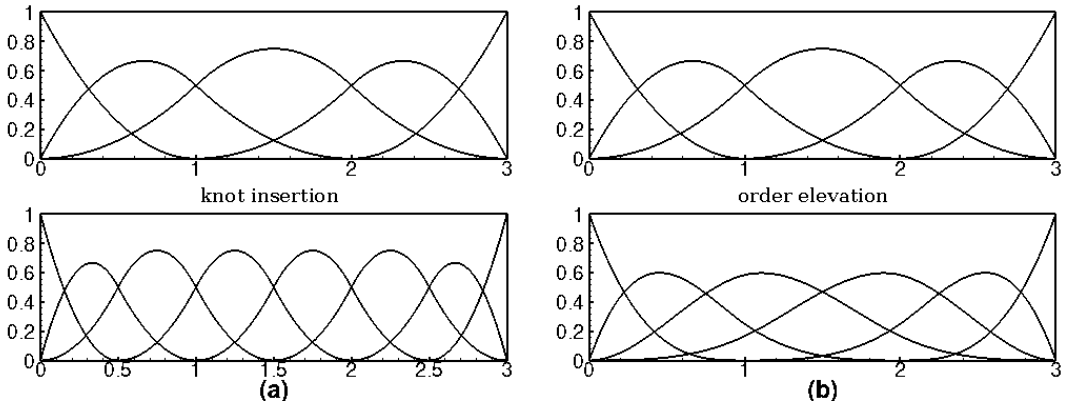


FIGURE 5.1: Two different strategies of refinement within IGA. Original knot vector for upper left and right domain is $\Xi^0 = \{0,0,0,1,2,3,3,3\}$. (a) Knot insertion - spline functions of the same order but smaller knot intervals, i.e. higher frequencies; changing element size. Knot vector after knots insertion $\Xi^1 = \{0,0,0,\frac{1}{2},1,\frac{3}{2},2,\frac{5}{2},3,3,3\}$. (b) Order elevation - higher degree of basis functions; element size remains the same. Knot vector after order elevation $\Xi^1 = \{0,0,0,0,1,2,3,3,3,3\}$.

An example of order elevation is shown in Figure 5.1b with starting knot vector $\Xi^0 = \{0,0,0,1,2,3,3,3\}$ for the quadratic B-spline basis functions. This time the multiplicity of the knots is increased by one giving one more control point and basis function. New knot vector for four cubic basis functions is $\Xi^1 = \{0,0,0,0,1,2,3,3,3,3\}$ and are presented in Figure 5.1b (right down).

5.1 Hierarchical B-spline basis functions

Hierarchical B-splines aim to construct base that is combined from coarse level B-splines and fine level B-splines at different regions of the domain. The fine level B-splines are generated from coarse level B-splines by global *h*-refinement operations. Only a subset of the fine level functions are included based on a selected refinement region. Refinement is procedure of gaining finer control over a spline curve or surface. For curves, refinement is a local process that permits the change of control vertices in one region of the curve while leaving control vertices in other regions unaffected.

Uniform B-spline basis functions $\mathcal{B}_{i,n}(\xi)$ $(i = 1,2,...,m)$ of a given order n are defined on a knot vector $\Xi = \{\xi_1, \xi_2, ..., \xi_{m+n+1}\}$, where $\xi_i \in \mathbb{R}$ is the i-th knot and m is the number of basis functions. $\mathcal{B}_{i,n}$ is defined on local support $[\xi_i, ..., \xi_{i+n+1}]$. B-spline basis functions are refinable, which enables the construction of HB and its truncated variant THB a modified version of HB with the only difference that the basis functions whose support overlaps finer levels are truncated [31], [33], [34]. Assume Ξ^0 is an initial uniform knot vector; $\Xi^l (l = 1,2,...)$ can be obtained by l subdivisions of Ξ^0, while l also presents the resolution level, where for $l = 0$ mesh is always uniform.

Hierarchical B-spline basis is constructed by excluding coarse level functions whose supports are contained within the selected refinement area and replacing them by the fine level functions whose supports are contained within the same area as the excluded coarse functions (see Figure 5.2).

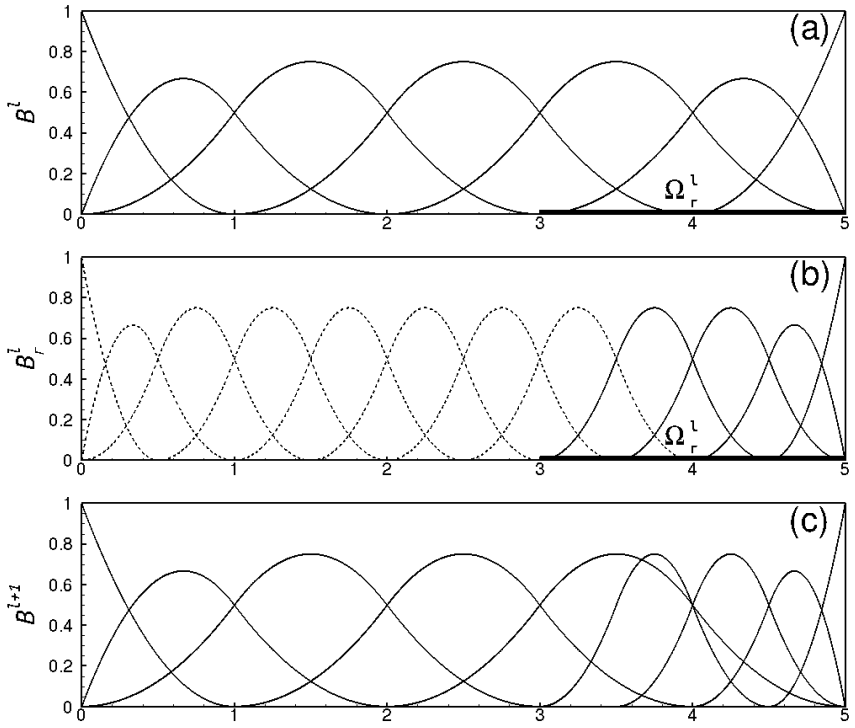


FIGURE 5.2: Local hierarchical h-refinement of the B_2-spline basis with knot vector $\Xi^l = \{0,0,0,1,2,3,4,5,5,5\}$ and refinement area $\Omega^l_r = (3,5)$. $\mathcal{B}^l$ are hierarchical B-splines on refinement level l, $\mathcal{B}^{l+1}_r$ are refinement B-splines used to construct finer resolution level, and $\mathcal{B}^{l+1}$ are B-splines combined from coarse and finer resolution level.

Figure 5.2 shows a hierarchically h-refined basis in three rows (pictures). The B-splines $\mathcal{B}^l$ from resolution level l (Figure 5.2a) are plotted in the first row and the h-refined B-splines $\mathcal{B}^{l+1}_r$ of the $\mathcal{B}^l$ are plotted in the second row (Figure 5.2b). The marked refinement area is set as $\Omega^l_r = (3,5)$. Functions contained within $(3,5)$ from $\mathcal{B}^l$ are marked passive and excluded from further process while the rest are marked active. Functions within $(3,5)$ from $\mathcal{B}^{l+1}_r$ are marked active (solid black curve) and included in further process. To construct $\mathcal{B}^{l+1}$ (Figure 5.2c), only active basis functions from both level are included, $(\mathcal{B}^{l+1} = \mathcal{B}^l_a \cup \mathcal{B}^{l+1}_{r,a})$.

To guarantee refinement, Ω^{l+1}_r must at least contain the support of a single basis function from the higher-level basis $\mathcal{B}^{l+1}$. Since higher level basis functions have smaller supports, having this as a minimal selection criterion could lead to the addition of fine (higher) level functions without excluding any coarse level functions, leading to excessive overlap of functions from different levels. Figure 5.2 shows how this choice has some disadvantages with

respect to e.g. possibility of having a partition of unity basis [24] and the bandwidth of the resulting stiffness matrices.

Generally, the hierarchical B-spline basis functions in non-rational form do not satisfy partition of unity which is evident from Figure 5.2. In last row (Figure 5.2c), basis function from coarse resolution level $\mathcal{B}^l$ overlaps with first two (active) basis functions from finer resolution level $\mathcal{B}_r^{l+1}$, which can lead to poor numerical conditioning.

5.2 Truncated hierarchical B-spline basis functions

Truncated hierarchical B-splines (THB) were introduced and analysed in [16], [64]. THB-splines can be considered as an upgrade for hierarchical B-splines (HB) i.e., an alternative basis for the space of hierarchical splines, that regains the partition of unity property and reduces the support of the basis functions, therefore reducing the interaction between them. In the classical hierarchical construction, coarse basis functions of a certain level l whose support is completely covered by finer basis functions of level $l+1$ are replaced. However for THB, the replacement is done as in the hierarchical case with addition that coarse basis functions whose support has a non-empty overlap with Ω^{l+1} are truncated (see Figure 5.4).

THB refinability (see [16], [31]) indicates that a basis function B_n^l defined on Ξ^l can be represented as a linear combination of $n+2$ B_n^{l+1} basis functions defined on Ξ^{l+1} as,

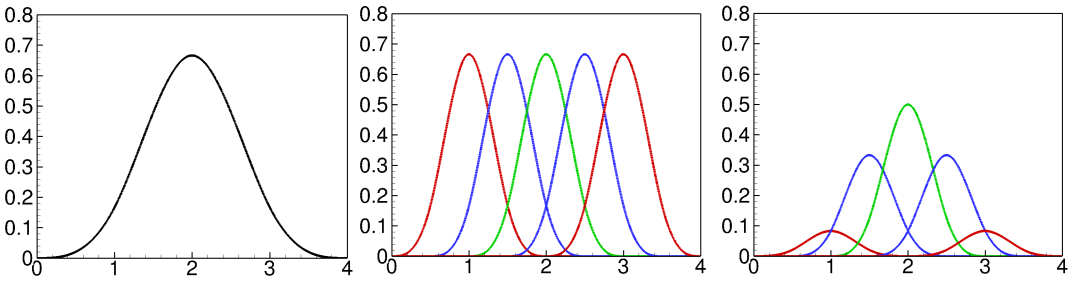
$$B_{i,n}^l(\xi) = \sum_{k=0}^{n+1} c_{i,k}^n B_{2i+k,n}^{l+1}(\xi) \quad \text{with} \quad c_{i,k}^n = \frac{1}{2^n}\binom{n+1}{k}, i = 0, ..., m^l - 1 \quad (5.1)$$

where $c_{i,k}^n$ are the refinement coefficients (see table 5.1) and m^l is the number of basis functions defined on Ξ^l. This procedure enables h-adaptive methods because each next resolution level has basis functions with two times smaller compact support (h-refinement). The $n+2$ basis functions $B_{2i+k,n}^{l+1}$ on the next level are called the children of $B_{i,n}^l(\xi)$ i.e., denoted as,

$$chdB_{i,n}^l(\xi) = \left\{ B_{2i+k,n}^{l+1}(\xi) | k = 0, 1, ..., n+1 \right\}. \quad (5.2)$$

Figure 5.3 shows how a uniform univariate cubic B-spline basis function $B_{0,3}^l$ can be represented by a weighted summation of its five children $B_{k,3}^{l+1}(k = 0, 1, 2, 3, 4)$. Each color represents one refinement coefficients and if basis functions have same refinement coefficients then they are colored with same color (for example, red is assigned for $\mathrm{B}_{0,3}^{l+1}(\xi)$ and $\mathrm{B}_{4,3}^{l+1}(\xi)$ because $c_{0,0}^3 = c_{0,4}^3 = 1/8$).

Figure 5.3b presents 5 B_3 spline basis functions after h-refinement where each *children* have two times smaller compact support then starting B_3 spline



(a) $B^l_{0,3}(\xi)$ (b) $B^{l+1}_{k,3}(\xi); (k = 0, ..., 4)$ (c) $c^3_{0,k} B^{l+1}_{k,3}(\xi); (k = 0, ..., 4)$

FIGURE 5.3: Refinability of a $B_3(\xi)$ spline basis function. (a) $B^l_{0,3}(\xi)$ is defined on the knot vector $\Xi^l = \{0, 1, 2, 3, 4\}$; (b) $B^{l+1}_{k,3}(\xi); (k = 0, 1, 2, 3, 4)$ are defined on a knot vector $\Xi^{l+1} = \{0, \frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4\}$; and (c) $B^{l+1}_{k,3}(\xi)$ is weighted with $c^3_{0,k} = \frac{1}{8}\binom{4}{k}$ for $k = 0, 1, 2, 3, 4$.

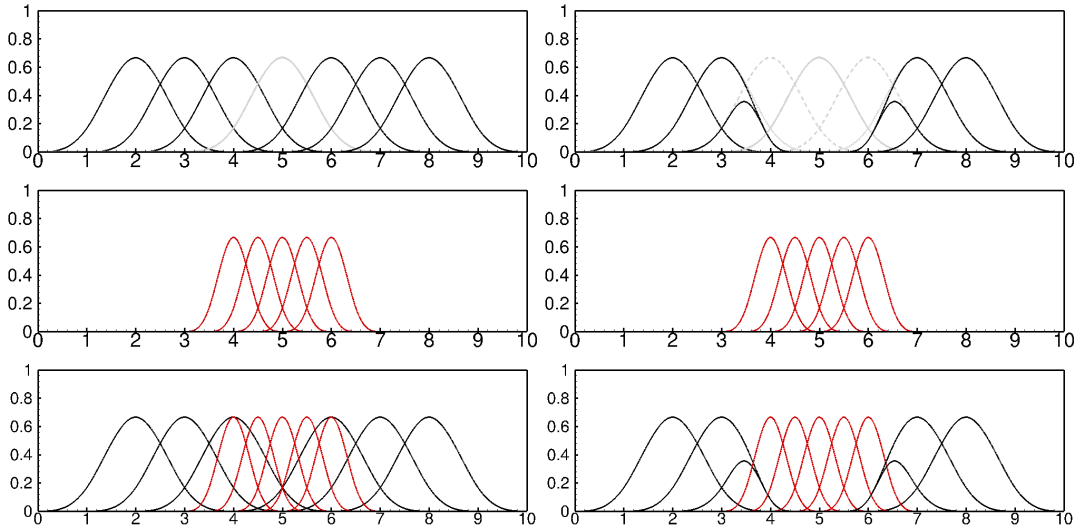
basis function (Figure 5.3a). After including refinement coefficients (see Table 5.1) for each basis functions refined B_3 spline basis functions are obtained, as presented in Figure 5.3c, and according to (5.1) $B^l_{0,3}$ (Figure 5.3a) can be represented by summation of its five children (refined basis functions; Figure 5.3c).

n \ $c^n_{i,k}$	$c^n_{i,0}$	$c^n_{i,1}$	$c^n_{i,2}$	$c^n_{i,3}$	$c^n_{i,4}$	$c^n_{i,5}$	$c^n_{i,6}$
1	1/2	1	1/2	-	-	-	-
2	1/4	3/4	3/4	1/4	-	-	-
3	1/8	1/2	3/4	1/2	1/8	-	-
4	1/16	5/16	5/8	5/8	5/16	1/16	-
5	1/32	3/16	15/32	5/8	15/32	3/16	1/32

TABLE 5.1: Refinement coefficients for B_n-spline; n=1,2,3,4,5.

In the following, construction of only two consecutive levels with basis functions from level l and $l+1$ will be shown, where $l \geq 0$. Starting from the initial parametric domain Ω^l with equally spaced knots $\Xi^l = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $\mathcal{B}^l$ set of B-spline basis functions are defined on a level l (see Figure 5.4). The supports of all the basis functions $\mathcal{B}^l$ from initial level l covers Ω^l i.e., $\Omega^l = supp\ B^l$. According to [65], the function space spanned by $\mathcal{B}^l$ can be enlarged by replacing the certain B-spline basis functions with their children, which indicates a local refinement of basis functions. Figure 5.4 shows a construction process for univariate cubic THB but also for HB in three steps:

- Identify a set of basis functions $\mathcal{B}^l_p \subseteq \mathcal{B}^l$ to be refined at level l (gray solid curve) and designate them as *passive* while the remaining basis functions in $\mathcal{B}^l$ are designated as *active* ($\mathcal{B}^l_a = \mathcal{B}^l \setminus \mathcal{B}^l_p$).



(a) HB-splines on level l and $l+1$ (b) THB-splines on level l and $l+1$

FIGURE 5.4: Comparison of univariate cubic HB- and THB-splines. (a) Three steps to construct univariate cubic HB-spline basis function without truncation and (b) tree steps to construct univariate cubic HB-spline basis function with truncation (THB).

- Obtain the children at level $l+1$ (red solid curves) only for the *passive* $\mathcal{B}_p^l$ and define them as *active*; $\mathcal{B}_a^{l+1} = chd\mathcal{B}_p^l$.
- Merge all of the basis functions that are *active* from levels l and $l+1$ to obtain the hierarchical B-spline basis functions on the new level,

$$\mathcal{B}_{hbf}^{l+1} = \mathcal{B}^{l+1} = \mathcal{B}_a^l \cup \mathcal{B}_a^{l+1}. \tag{5.3}$$

Eq. (5.3) refers to the global selection of all active basis functions, where the active basis functions are updated in each recursive step described above. Hierarchical B-spline basis functions in nonrational form do not satisfy partition of unity. To overcome that problem and to decrease the overlapping of basis functions for better numerical conditioning, a truncated mechanism for hierarchical B-splines was developed [16], [31]. Figure 5.4 shows how in the classical hierarchical construction, coarse basis functions from level l whose support is completely covered by finer B-splines of level $l+1$ are replaced. THB-splines refinement (replacement) works as in the hierarchical case with addition of active coarse basis functions $\mathcal{B}_a^l$ whose supports have a non-empty overlaps with Ω^{l+1}. These functions need to be modified or truncated as follows.

Definition. *Given a set of (passive) basis functions $\mathcal{B}_p^l$ to be refined, refinement area is defined as $\Omega^{l+1} = supp\mathcal{B}_p^l$. Provided that $B_i^l \notin B_p^l$ is refinable and following equation* (5.1) *for its refinability gives,*

$$B_i^l(\xi) = \sum_{supp B_j^{l+1} \subseteq supp B_i^l} c_{i,j} B_j^{l+1}(\xi), \tag{5.4}$$

where $c_{i,j} \in \mathbb{R}$ *are refinement coefficients from mid-knot insertions (see table 5.1), and* $\mathcal{B}_j^{l+1}(\xi) \in chdB_i^l(\xi)$. *The truncated basis function* $\mathcal{B}_i^l$ *is defined as*

$$trunB_i^l(\xi) = \sum_{suppB_j^{l+1} \not\subseteq \Omega^{l+1}} c_{i,j} B_j^{l+1}(\xi) \tag{5.5}$$

with respect to $\mathcal{B}_p^l$ [31].

Equation (5.5) indicates that only children of $\mathcal{B}_i^l$ whose supports are fully contained in Ω_{l+1} are discarded while constructing the truncated basis function $trunB_i^l$. In Figure 5.4, the gray solid line represents the basis function to be refined $\mathcal{B}_p^l$ which is also set as passive, and refinement area is $\Omega^{l+1} = [3, 7]$. In case for univariate cubic hierarchical B-splines, each basis function from level l has five children on level $l + 1$, and four basis functions surrounding $\mathcal{B}_p^l$ (2 on the left and 2 on the right; gray dashed curve) need to be truncated because they have children with supports fully contained in Ω^{l+1}. For the two basis functions adjacent to $\mathcal{B}_p^l$ three children are discarded, and for the other two basis functions, only one children is discarded. Basis functions that are far away from refinement area Ω^{l+1} i.e., they do not have children within that area, are not truncated. After truncating all designated basis functions, new level is constructed by combining active functions from level l (black solid curve and gray dashed curve; non-truncated and truncated) with active basis functions from level $l + 1$ (red solid curve; $\mathcal{B}_a^{l+1} = chd\mathcal{B}_p^l$).

The hierarchical B-spline basis with truncation has been proven to form a partition of unity and therefore achieves strong stability [64]. It gives a sparser connectivity among basis functions at different levels, and it can preserve geometry when local refinement is performed [31].

Chapter 6

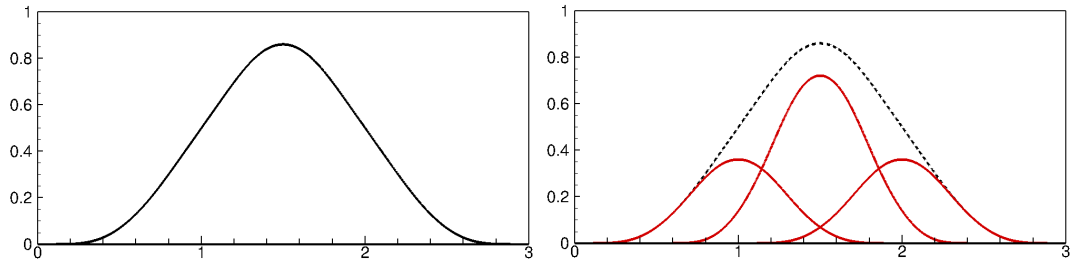
Adaptive modeling based on hierarchical Fup basis functions and CV-IGA

Throughout the rest of this thesis strategy for the adaptation of Fup to non-uniform meshes and their local refinement is investigated. A novel adaptive algorithm that is based on control volume IGA formulation and novel hierarchical Fup basis functions [37], [38], [41] is presented. Because of its similarity to the concept of isogeometric analysis (IGA) it is called control volume isogeometric analysis (CV-IGA). In this chapter process of constructing 1D and 2D hierarchical Fup (HF) basis functions is shown which are closely related to the HB and THB. HF have the option of local hp-refinement such that they can replace certain basis functions at one resolution level with new basis functions at the next resolution level that have a smaller length of the compact support (h-refinement) but also higher order (p-refinement). This feature provides spectral convergence and presents a substantial improvement in comparison to THB that enable polynomial convergence. Furthermore, additional stabilization methods are included in adaptation procedure to achieve even greater results and to deal with spurious numerical oscillations as is the case for advection-dominated problems [66], [67].

6.1 Hierarchical Fup basis functions

B-spline and Fup basis functions are refinable, which enables the construction of THB and HF basis functions [16], [31], [33]. As shown in previous chapter, THB refinability indicates that a basis function B_n^l defined on Ξ^l can be represented as a linear combination of $n+2$ B_n^{l+1} basis functions defined on Ξ^{l+1} following Eq. (5.1) where n represents basis function order and Ξ^l is a knot vector for resolution level l. This procedure enables h-adaptive methods because each next resolution level has basis functions with two times smaller compact support (h-refinement).

However, Fup basis functions refinement is done in a different way. Basis function Fup_n^l defined on Ξ^l can be represented as a linear combination of



(a) $\text{Fup}_1^l(\xi)$ (b) $C_2^k\text{Fup}_2^{l+1}(\xi - \frac{k}{2^2} + \frac{2}{2^3}); (k = 0, 1, 2)$

FIGURE 6.1: Refinability of a $\text{Fup}_1(\xi)$ basis function. (a) $\text{Fup}_1^l(\xi)$ is defined on the knot vector $\Xi^l = \{0, 1, 2, 3\}$; and (b) $\text{Fup}_2^{l+1}(\xi - \frac{k}{2^2} + \frac{2}{2^3})$ are defined on a knot vector $\Xi^{l+1} = \{0, \frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, 3\}$ with $C_2^k = \frac{1}{2^2}\binom{2}{k}$ for $k = 0, 1, 2$.

$n + 2$ Fup_{n+1}^{l+1} basis functions defined on Ξ^{l+1},

$$Fup_n^l(\xi) = \sum_{k=0}^{n+1} C_{n+1}^k \cdot Fup_{n+1}^{l+1}\left(\xi - \frac{k}{2^{n+1}} + \frac{n+1}{2^{n+2}}\right), \tag{6.1}$$

where C_{n+1}^k are the refinement coefficients (see Table 6.1)

$$C_{n+1}^k = \frac{1}{2^{n+1}}\binom{n+1}{k} \tag{6.2}$$

The $n + 2$ basis functions Fup_{n+1}^{l+1} are called the *children* of Fup_n^l, denoted as

$$chdFup_n^l(\xi) = \left\{ Fup_{n+1}^{l+1}\left(\xi - \frac{k}{2^{n+1}} + \frac{n+1}{2^{n+2}}\right) \middle| k = 0, 1, ..., n+1 \right\} \tag{6.3}$$

Figure 6.1a shows a basis function Fup_1^l defined on a knot vector $\Xi^l = \{0, 1, 2, 3\}$, and Figure 6.1b shows its three children $\text{Fup}_2^{l+1}(\xi - \frac{k}{2^2} + \frac{2}{2^3})$ defined on a knot vector $\Xi^{l+1} = \left\{0, \frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, 3\right\}$, where $k = 0, 1, 2$. According to Eq. (6.1), Fup_1^l (Figure 6.1b - black dashed curve) can be represented by a weighted summation of its three children Fup_2^{l+1} (Figure 6.1b - red solid curve). In contrast to THB, hierarchical Fup basis functions (HF) enable *hp*-adaptive methods because each next resolution level not only decreases compact support but also increases the order of the basis functions (*hp*-refinement).

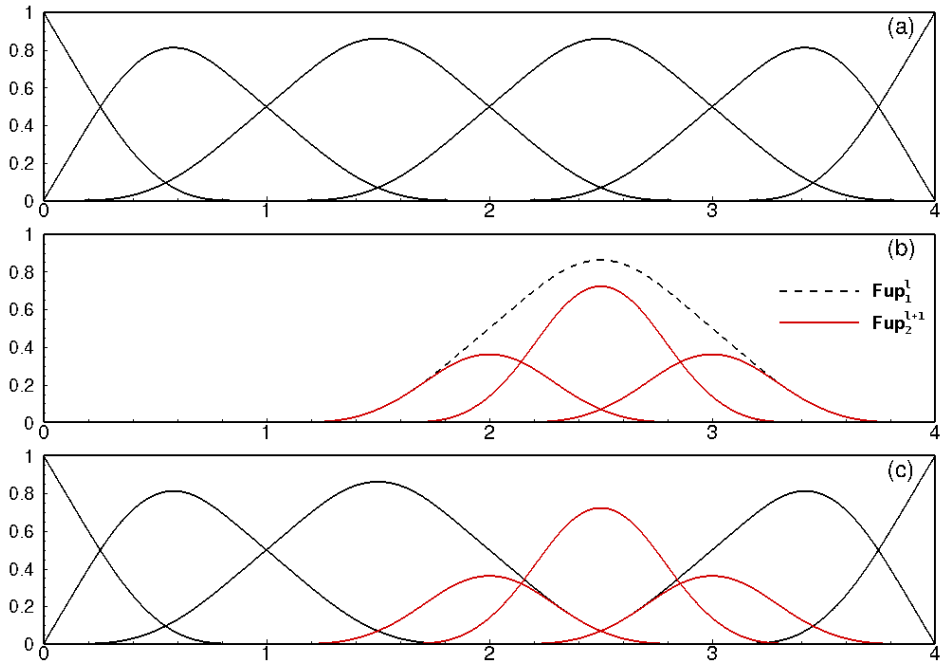


FIGURE 6.2: The three steps to construct hierarchical Fup basis functions. (a) In level l, basis functions $\mathcal{F}_p^l$ that need to be refined are determined (black dashed curve $|Fup_1^l$) and they are defined as *passive*, while remaining basis functions are defined as *active*; (b) In level $l+1$, three children (red solid curves $|Fup_2^{l+1}$) are designated as *active*; and (c) all *active* basis functions from levels l and $l+1$ are summed and form the hierarchical Fup basis functions $\mathcal{F}_{hbf}^{l+1}$.

n \ C_{n+1}^k	C_{n+1}^0	C_{n+1}^1	C_{n+1}^2	C_{n+1}^3	C_{n+1}^4	C_{n+1}^5	C_{n+1}^6
1	1/4	1/2	1/4	-	-	-	-
2	1/8	3/8	3/8	1/8	-	-	-
3	1/16	1/4	3/8	1/4	1/16	-	-
4	1/32	5/32	5/16	5/16	5/32	1/32	-
5	1/64	3/32	15/64	5/16	15/64	3/32	1/64

TABLE 6.1: Refinement coefficients for Fup$_n$ basis functions; n=1,2,3,4,5.

At the zero coarsest level, we can define a set of uniformly distributed Fup basis functions $\mathcal{F}^0$. The initial domain is covered with the compact supports of all the Fup basis functions in $\mathcal{F}^0$ i.e., $\Omega^0 = supp\mathcal{F}^0$. Since Fup basis functions are refinable, it indicates that the function space spanned by $\mathcal{F}^0$ can be enlarged by replacing the selected Fup basis functions with their children (see Eq. (6.1)) [41]. In the following, we will show only two consecutive levels and construct level $l+1$ from the level l.

Figure 6.2 illustrates the construction process of hierarchical Fup basis functions in three steps:

- Identify a set of basis functions $\mathcal{F}_p^l \subseteq \mathcal{F}^l$ to be refined at level l (black dashed curve) and designate them as *passive* while the remaining basis functions in $\mathcal{F}^l$ (black solid curves) are designated as *active* ($\mathcal{F}_a^l = \mathcal{F}^l \setminus \mathcal{F}_p^l$).
- Obtain the children at level $l+1$ (red solid curves) only for the *passive* Fup_n^l and define them as *active*; $\mathcal{F}_a^{l+1} = chd\mathcal{F}_p^l$.
- Merge all of the basis functions that are *active* from levels l and $l+1$ to obtain the hierarchical Fup basis functions,

$$\mathcal{F}_{hbf}^{l+1} = \mathcal{F}^{l+1} = \mathcal{F}_a^l \cup \mathcal{F}_a^{l+1}. \tag{6.4}$$

As mentioned in previous section, hierarchical B-spline basis functions in nonrational form do not satisfy partition of unity. To overcome that problem and to decrease the overlapping of basis functions for better numerical conditioning, a truncated mechanism for hierarchical B-splines was developed [16], [31]. However, hierarchical Fup basis functions satisfy partition of unity such that every Fup_n basis function on the zero coarsest level is multiplied with constant 2^{-n}. Since every Fup_n^l basis function defined on the level l can be represented as a linear combination of $n+2$ Fup_{n+1}^{l+1} basis functions defined on the level $l+1$ (see Eq. (6.1)), it entails that all of the Fup basis functions that are created at higher resolution levels also satisfy partition of unity.

6.2 1-D adaptive algorithm

6.2.1 Basic adaptive algorithm parameters

The trial function space of uniformly distributed $\text{Fup}_n(\xi)$ basis functions on the resolution level l of a given order n are defined using the knot vector. A knot vector composed of $m+n+2$ knot values will generate m linearly independent basis functions of degree n. The number of basis functions on the first resolution level $(m^l; l=0)$ is defined as input parameter, thus the length of the characteristic interval $(\Delta\xi)$ for any resolution level l is calculated as

$$\Delta\xi^l = \frac{\xi_{m+n+2} - \xi_1}{(m^0 - n - 1)2^l} \tag{6.5}$$

where m^0 is the number of basis functions on the first resolution level, and ξ_1 and ξ_{m+n+2} are first and last members of the knot vector on the first resolution level $(l=0)$, respectively.

Basis functions whose compact supports are at least partially located outside the domain are modified so that the i-th derivation is satisfied following equation (3.41). In the vector space of mutually displaced Fup_n basis functions, it is necessary to modify the $(n+1)$ basis functions on the left and right boundary of the domain. Figure 6.3 shows a uniform set of $\text{Fup}_{1,i}(\xi)$ $(i = 1, ..., m;\ m = 6)$ basis functions defined on the knot

vector $\Xi = \{0,0,0,1,2,3,4,4,4\}$ with modified basis functions on the left ($\text{Fup}_{1,1}$,$\text{Fup}_{1,2}$) and right ($\text{Fup}_{1,5}$,$\text{Fup}_{1,6}$) boundary of the domain.

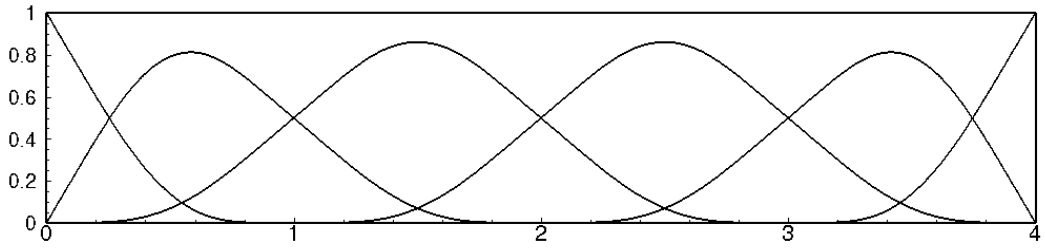


FIGURE 6.3: $Fup_1(\xi)$ basis function trial space

The vertex of the basis function, i.e., the coordinate ξ_T, is the point with the maximum function value. The vertex serves as the origin for the shifting of the basis functions along the ξ axis by the length of the characteristic interval ($\Delta\xi$). However, not all vertices are uniformly spaced according to the length of the characteristic interval. Vertices of the modified boundary basis functions (see section 3.4) are shifted and located inside the domain area. Their exact location can be calculated. One possibility is to calculate Greville abscissae grid points following Eq. (4.14). From this point, when basis function vertex is mentioned, it is referred to the real coordinate of the vertex, except for modified boundary basis functions whose real coordinate of the vertex coordinate is represented by the Greville point.

6.2.2 Control volume distribution by levels

The classical FVM can be applied on all differential equations which can be written in the divergence form, but it is in common use for discretizing computational fluid dynamics equations [68]. For clarity, steady state ADE (4.5) with appropriate boundary conditions (4.6) and (4.7) is considered.

First, the domain is divided by m CVs (Figure 6.4; $\Omega_i; i = 1, ..., m$). Then ADE is integrated over each CV and multiplied with a finite number of weighting (test) functions w_i (see Eq. (4.15)).

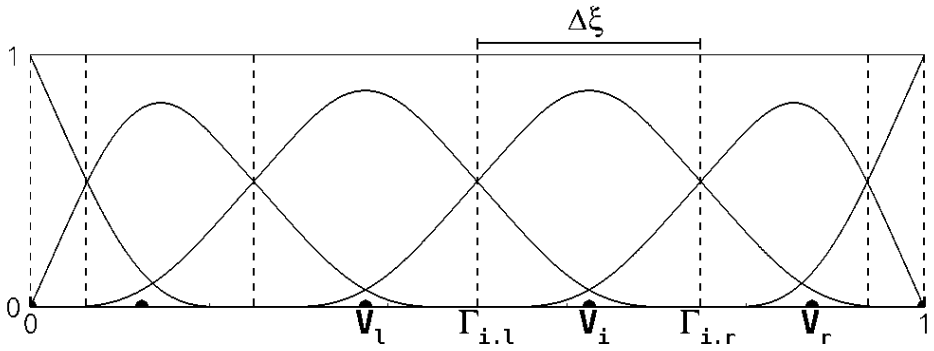


FIGURE 6.4: Nonoverlapping control volume scheme for one-dimensional case.

To derive CV-IGA, the 1D parameter space (from 0 to 1) defined by uniform open knots is shown on Figure 6.4. The observed domain is subdivided

into a set of nonoverlapping CVs such that each CV surrounds one corresponding vertex (Figure 6.4 - black dots) of the Fup basis functions. End points of CV are defined at the half of the distance between neighboring vertices (Greville abscissae for the modified basis functions; basis function vertices for other basis functions). Note that the number of basis functions is the same as the number of CVs.

In Figure 6.4, the focus is on the grid point that represents vertex V_i, which has the grid points V_l on the left and V_r on the right side as its neighbors. The dashed lines ($\Gamma_{i,l}$ and $\Gamma_{i,r}$) surrounding vertex V_i represent faces of the CV in a manner such that each CV boundary lies in the middle between two vertices.

In the following, three consecutive levels with control volume distribution after replacing certain basis functions for adaptation on each level will be shown. Main focus will be on CV distribution i.e., how each CV is being set for every new resolution level, without going in details why certain basis functions are being replaced from current resolution level. Main difference in higher resolution levels is overlapping of CVs. Adaptive criteria will be discussed in the following sections.

Figure 6.5a shows uniformly distributed Fup basis functions on the knot vector $\Xi^0 = \{0, 0, 0, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, 1, 1, 1\}$ with the positions of the vertices (black dots) and corresponding CVs where each ith CV boundary is represented with dashed line. Figure 6.5b shows second resolution level after replacing $\text{Fup}_{1,5}$ basis function (black dashed curve) from first resolution level (assigned as passive) with three Fup_2 basis functions (assigned as active) and defined on knot vector $\Xi^1 = \{\frac{2}{8}, \frac{5}{16}, \frac{3}{8}, \frac{7}{16}, \frac{4}{8}, \frac{9}{16}, \frac{5}{8}\}$. After assembling new setup of Fup basis functions on the new resolution level ($\mathcal{F}^1 = \mathcal{F}_a^0 \cup \mathcal{F}_a^1$), new CV distribution is set following the same instructions as before but having in mind that there are 3 new vertices (1 vertex with old position as replaced basis function and 2 with new vertices positions). It should be noted that size of CVs that are in contact with lower resolution level are enlarged to solve problem of linearly dependent equations that would occur on higher resolution levels. Problems would occur on levels that are $l \geq 4$, because two or more test functions from higher resolution level $(l+1)$ would overlap exactly with test function from lower level (l), thus giving linearly dependent equations. That problem is solved by enlarging test functions (CVs) from higher resolution level $(l+1)$ that are in contact with lower resolution level (l). Enlargement can be chosen anywhere from interval $\delta \in \langle 0, \frac{1}{2} \rangle$. This means that the CVs boundaries are distanced from Greville points by the length of the compact support plus δ of the compact support length (see Figure 6.5b). Here, δ is chosen as $\delta = \frac{1}{4}$, thus first and last CVs in one continuous sequence (considering only higher resolution level) have wider area then rest of the control volumes (see Eq. (6.5)) in that sequence (Figure 6.5b-c).

Figure 6.5c shows the third resolution level after replacing one Fup_2 (red dashed curve) basis function from $\mathcal{F}^{l+1}$ with four Fup_3 basis functions defined on knot vector $\Xi^2 = \{\frac{5}{16}, \frac{11}{32}, \frac{3}{8}, \frac{13}{32}, \frac{7}{16}, \frac{15}{32}, \frac{4}{8}, \frac{17}{32}, \frac{9}{16}\}$. Again after assembling new setup of Fup basis functions, new CV distribution is set as before. However, now all four vertices are in new positions. In the case when replaced Fup basis function is even order, new Fup basis functions (order

of basis functions increase because of Eq. (6.1)) became odd order (Figure 6.5 b-c) and positions of all new vertices have new coordinates compared to the replaced one. Otherwise, when replaced Fup is odd, then new Fup ba-

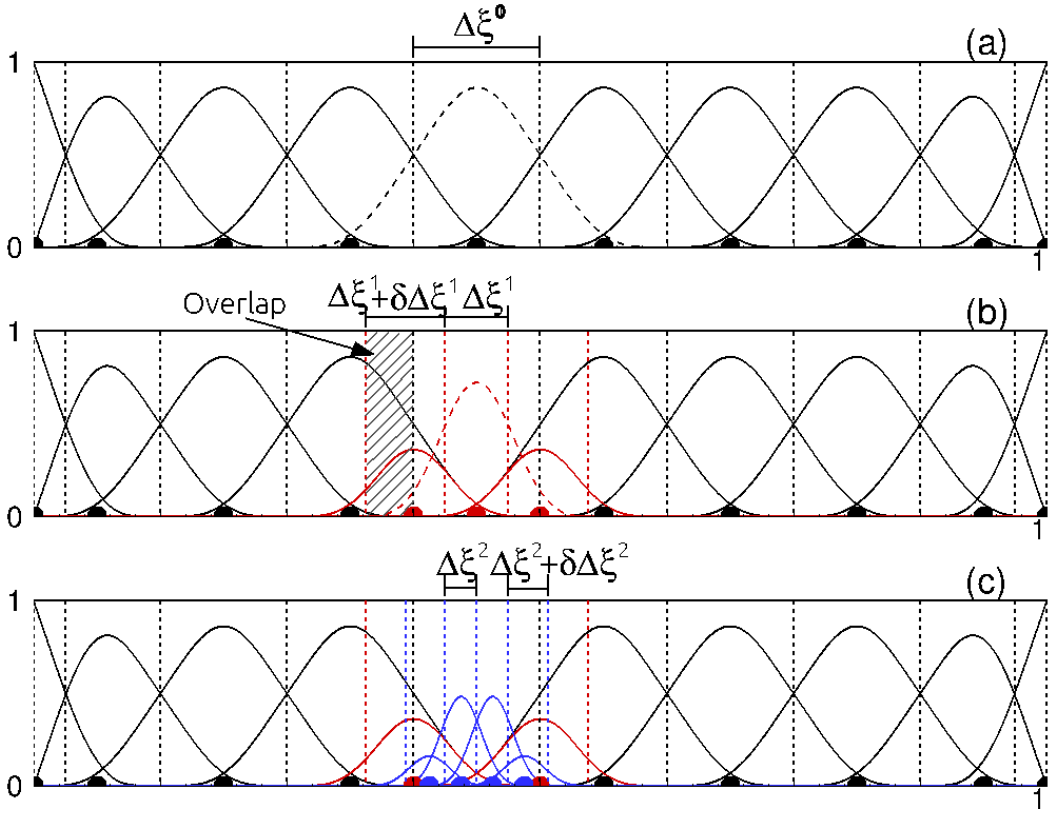


FIGURE 6.5: CV distribution for 3 consecutive levels. (a) uniform nonoverlapping CV distribution for first resolution level with knot vector $\Xi^0 = \{0,0,0,\frac{1}{8},\frac{2}{8},\frac{3}{8},\frac{4}{8},\frac{5}{8},\frac{6}{8},\frac{7}{8},1,1,1\}$; (b) CV distribution on second resolution level with active Fup basis functions from $\mathcal{F}^0$ and $\mathcal{F}^1$ with overlapping CVs; (c) CV distribution on third resolution level with active Fup basis functions from $\mathcal{F}^0$, $\mathcal{F}^1$ and $\mathcal{F}^2$ and overlapping CVs.

sis functions became even order (Figure 6.5 a-b) and one of the new basis functions vertex (middle one) coincides with the replaced (origin) Fup basis function vertex.

6.2.3 Adaptive algorithm for function approximation and BVP

The adaptive spatial strategy used in this work is a novel approach based on the control volume IGA formulation and hierarchical Fup basis functions (*hp*-refinement). In the following part, adaptive scheme for approximating known function is presented. It is used for easier understanding of whole adaptive process and serves as introduction for BVPs. In later part, adaptive strategy for solving BVP with its differences but also similarities with

approximation of a known function is presented. The main idea is to represent the known function (f) in an adaptive manner so that coarse control volumes and lower order of Fup basis functions are used in regions where the solution is smooth, and fine control volumes and a higher order of Fup basis functions are used in those areas where the solution varies strongly.

The approximation $\tilde{f}(x)$ of the known function $f(x) : \Omega \rightarrow \mathbb{R}$ is presented in the form of the linear combination (2.9). The difference between the known function $f(x)$ and its numerical approximation $\tilde{f}(x)$ yields the numerical error:

$$\varepsilon(x) = f(x) - \tilde{f}(x) = f(x) - \sum_{j=1}^{m} \alpha_j \varphi_j(x) \tag{6.6}$$

The meaning of the approximation is to minimize the error $\varepsilon(x)$. If the control volume formulation (see 4.3.3) is applied, the unknown coefficients α_j are obtained from the following system of equations

$$\sum_{j=1}^{m} \alpha_j \int_{\Omega} \varphi_j(x) w_i d\Omega = \int_{\Omega} f(x) w_i d\Omega; \quad i, j = 1, 2, ..., m \tag{6.7}$$

which, after calculating the integrals, has a reduced matrix form

$$a_{ij}\alpha_j = b_i; \quad i, j = 1, 2, ..., m \tag{6.8}$$

where

$$a_{ij} = \int_{\Omega_i} \varphi_j(x) d\Omega; \quad b_i = \int_{\Omega_i} f(x) d\Omega. \tag{6.9}$$

The adaptive procedure using the hierarchical Fup basis functions is presented in Figure 6.6.

If we consider a spatial domain $\Omega = [0, 8]$ with $m^0 = 10$ Fup$_n$ basis functions with order $n = 1$, we can define Fup basis functions on the zero coarsest level $\mathcal{F}^0$, with the initial uniform knot vector $\Xi^0 = \{0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 8, 8, 8\}$. The characteristic interval for the zero coarsest level is defined via Eq. (6.10). Since HF compact support on the next level decreases the length two times with respect to the previous level, the characteristic interval for any resolution level can be defined as

$$\Delta\xi^l = \frac{8 - 0}{(10 - 1 - 1)2^l}; \quad l = 0, 1, 2, ... \tag{6.10}$$

where l represents the resolution level. Last, we need to define an adaptive criteria (ε_A), which defines whether Fup basis functions are kept (marked as *active*) or replaced (marked as *passive*). First, in order to define the control volume formulation at the zero level, the domain must be subdivided into m nonoverlapping control volumes where each CV edge lies in the middle between two Greville points. In Figure 6.6, the corresponding control volumes are presented above basis functions for better visualization. Then, the

coefficients α_j are calculated from the system of equations (6.8) so that the Fup approximation $(\tilde{f})$ satisfies the mean function values over all control volumes (CVs).

The adaptive criteria for the function approximation are defined as

$$\int_{\Omega_A} \frac{1}{\Omega_A} \left(|f(x) - \tilde{f}(x)|\right) dx < \varepsilon_A; \quad i = 1, 2, ..., m \tag{6.11}$$

where ε_A represents the defined threshold and Ω_A is the integration area. Since the numerical approximation $(\tilde{f})$ satisfies the average function value of the known function (f) over every CV_i $(i = 1, 2, ..., m)$ on the current (zero) resolution level, the main idea for enabling an adaptation is to test how close the numerical approximation $(\tilde{f})$ is with respect to the known function (f) on the each half of the volume (see Figure 6.7). If all CVs satisfy adaptive criteria, the adaptive procedure stops. However, if one or more CVs did not satisfy Eq. (6.11), than those CVs are marked as refinable (e.g., Fig. 6.6a - $\mathrm{CV}_{5,6}$).

Furthermore, all corresponding $\mathrm{Fup}^0_{1,i}$ basis functions that are at least partially located inside refinable CVs are marked as *passive*-$\mathcal{F}^0_p$ (Fig. 6.6a-black dashed curve with local knot vectors, $\Xi^0_4 = \{1, 2, 3, 4\}$, $\Xi^0_5 = \{2, 3, 4, 5\}$, $\Xi^0_6 = \{3, 4, 5, 6\}$, $\Xi^0_7 = \{4, 5, 6, 7\}$).

Other $\mathrm{Fup}^0_{1,i}$ basis functions are marked as *active*, i.e., $\mathcal{F}^0_a$, and they are kept in the next level. For the *passive* $Fup^0_{1,i}$ basis functions, the algorithm introduces their $\mathrm{Fup}^1_{2,i}$ children (Fig. 6.6b-right) (see Eq. (6.1)). It is possible that certain *children* of two neighboring basis functions mutually overlap (see 6.2.2), and then from these basis functions of the next level a "new" one basis function is formed (Fig. 6.6b-left). Now, all $\mathrm{Fup}^1_{2,i}$ basis functions $\Big($Fig. 6.6b-red curves with local knot vectors: $\Xi^1_4 = \left\{1, \frac{3}{2}, 2, \frac{5}{2}, 3\right\}$, $\Xi^1_5 = \left\{\frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}\right\}$, $\Xi^1_6 = \left\{2, \frac{5}{2}, 3, \frac{7}{2}, 4\right\}$, $\Xi^1_7 = \left\{\frac{5}{2}, 3, \frac{7}{2}, 4, \frac{9}{2}\right\}$, $\Xi^1_8 = \left\{3, \frac{7}{2}, 4, \frac{9}{2}, 5\right\}$, $\Xi^1_9 = \left\{\frac{7}{2}, 4, \frac{9}{2}, 5, \frac{10}{2}\right\}$, $\Xi^1_{10} = \left\{4, \frac{9}{2}, 5, \frac{10}{2}, 6\right\}$, $\Xi^1_{11} = \left\{\frac{9}{2}, 5, \frac{10}{2}, 6, \frac{13}{2}\right\}$, $\Xi^1_{12} = \left\{5, \frac{10}{2}, 6, \frac{13}{2}, 7\right\}\Big)$ are marked as *active* and they are merged with all *active* basis functions from the level 0 ($\mathcal{F}^0_a$).

Fig. 6.6b represents the new setup of Fup basis functions on the level $(l = 1)$ with their CVs consisting of *active* Fup_1 and Fup_2 basis functions. The procedure now calculates new Fup coefficients (α_j) from the new system of equations (6.8) so that the Fup approximation $(\tilde{f})$ satisfies the function values over all control volumes (CVs) from the level 1.

The numerical procedure is repeated if one or more CVs from level 1 did not satisfy Eq. (6.11), and those CVs are marked as refinable (e.g., Fig. 6.6b - $\mathrm{CV}_{8,9}$). Analogously, corresponding basis functions that are at least partially located inside refinable CVs are marked as *passive*-$\mathcal{F}^1_p$ (Fig. 6.6b-red dashed curve). For the new *passive* basis functions on level 1, we introduce

their children (Fig. 6.6c-right) and merge them with other *active* basis functions on the level 1 to obtain the new setup of Fup basis functions on level 2 with their CVs (Fig. 6.6c). If one or more of the $n+1$ modified boundary Fup_n basis functions is selected for refinement on the left and/or right boundary, the algorithm automatically marks all $n+1$ boundary basis functions on the left and/or right boundary as *passive*. This is necessary because the presented modified boundary basis functions are calculated in the form of a linear combination of the original Fup_n basis functions (see (3.41)). The adaptive procedure is repeated until the prescribed accuracy is not achieved at all CVs or the maximum user-defined level is not reached.

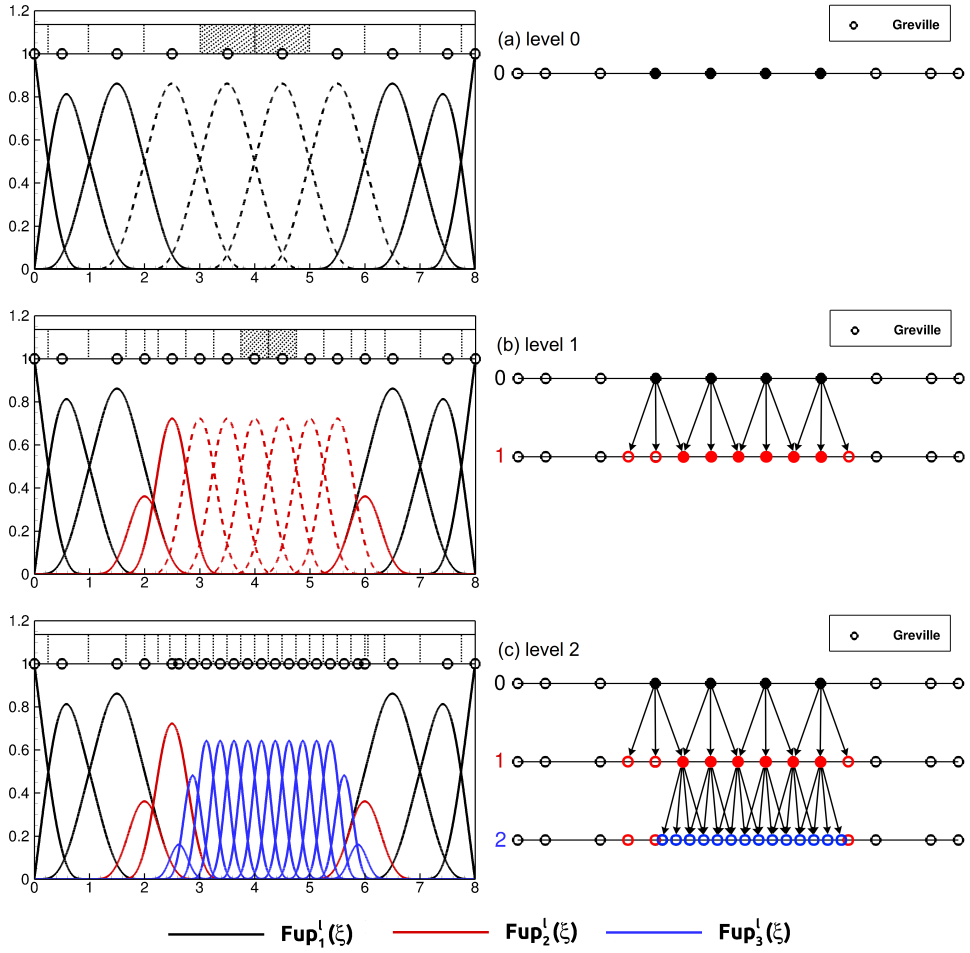


FIGURE 6.6: Adaptive procedure with Fup_n basis functions within 3 levels. Left: basis functions with corresponding CVs; Right: tree structure of Fup basis functions on the zero coarsest level and its children on higher levels.

The adaptive spatial strategy used for the boundary value problem (BVP) is in some sense similar to the one used for function approximation. In the following, focus will be on the main differences between these two strategies. The major differences are adaptive criteria and adaptation of boundary conditions.

In the function approximation, a known function is approximated, while in BVP, we usually do not know the solution of the differential equation.

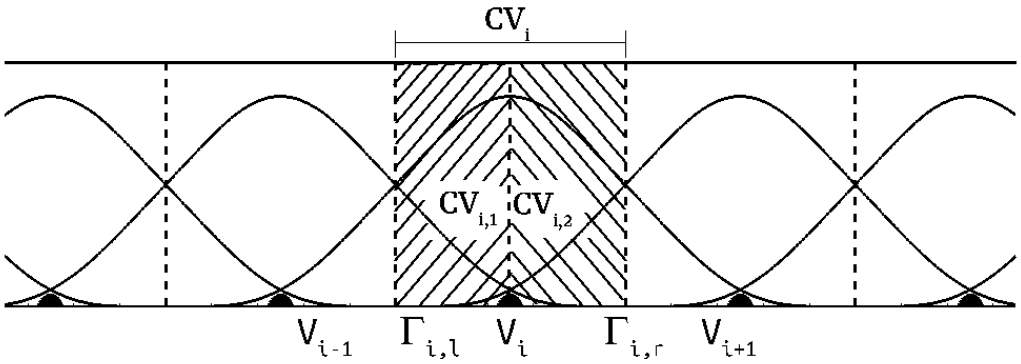


FIGURE 6.7: Dividing i-th CV into two parts ($CV_{i,1}$ and $CV_{i,2}$) while testing adaptive criteria on i-th CV.

The question is how to solve (approximate) the BVP. One of the possible approaches is shown considering the ADE (see Eq. (4.5)). In that case, solving ADE is reduced to the flux conservation over all CVs (see Eq. (4.18)). Since the CV formulation exactly satisfies Eq. (4.18) (i.e., the weak integral form of the conservation law) over each CV on the current resolution level, the adaptive criteria is used to check the conservation error for each half of the particular i-th CV (see Figure 6.7). All CVs where at least one of the half ($CV_{i,1}$ and/or $CV_{i,2}$; see Figure 6.7) has the conservation error greater than the prescribed threshold are marked as refinable, and the adaptive procedure refines selected basis functions in the next level in the same way presented for the function approximation (see Figure 6.6). Neumann boundary conditions on their respective CVs are satisfied in same sense as all internal CVs by checking conservation error for each half of the particular i-th CV on the boundary of the domain, i.e. Neumann boundary conditions are weakly imposed by incorporating the known value (4.7) to the weak formulation (4.18). However, Dirichlet boundary conditions are satisfied in the strong sense by directly satisfying the boundary conditions value (4.6).

6.2.4 Adaptive criteria

Adaptive criteria i.e., ε_A defines whether Fup basis functions are kept or replaced while refining resolution level l. For i-th control volume (CV), boundary is defined via $\Gamma_{i,l}$ and $\Gamma_{i,r}$, where subscript letter l (left) and r (right) represents side faces of the CV in a manner such that each CV boundary lies in the middle between two adjacent vertices.

The adaptive criteria can be defined in many ways. They do not have to be the same for every problem. In the function approximation, adaptive criteria is set to be related to the function accuracy Equation (6.11), while in BVP (ADE (4.5) or weak formulation (4.18)), the criteria is set to be the mass conservation error. There are many other meaningful numerical and physical choices. For example, for function approximation, the function derivatives can be an ideal option in some cases. Furthermore, if someone is using collocation formulation, then checking numerical error via Eq. (6.6) can be thought of as a natural step. In addition, for BVP, the solution error between

two resolution levels can be defined as criteria [51]. Moreover, satisfaction of the Peclet number can be very valuable for ADE problems (see [67]). Finally, any combination of these criteria can also be new obtained criteria. For ADE advection-dominated problems, inclusion of the stabilization technique in the few first resolution levels can also be a valuable part of an adaptive strategy, which will be shown in verification examples.

6.3 2-D adaptive algorithm

6.3.1 2-D hierarchical basis functions

Multidimensional Fup basis functions are obtained as tensor products of the one-dimensional basis functions defined for each coordinate direction. For example, the two-dimensional Fup basis functions are defined as,

$$Fup_n(\xi,\eta) = Fup_n(\xi) \cdot Fup_n(\eta) \tag{6.12}$$

where $Fup_n(\xi)$ and $Fup_n(\eta)$ are nth order Fup basis functions that are defined in the ξ- and η- parametric directions, respectively. Figure 6.8 shows two-dimensional $\text{Fup}_1(\xi,\eta)$ basis function and one of its partial derivatives.

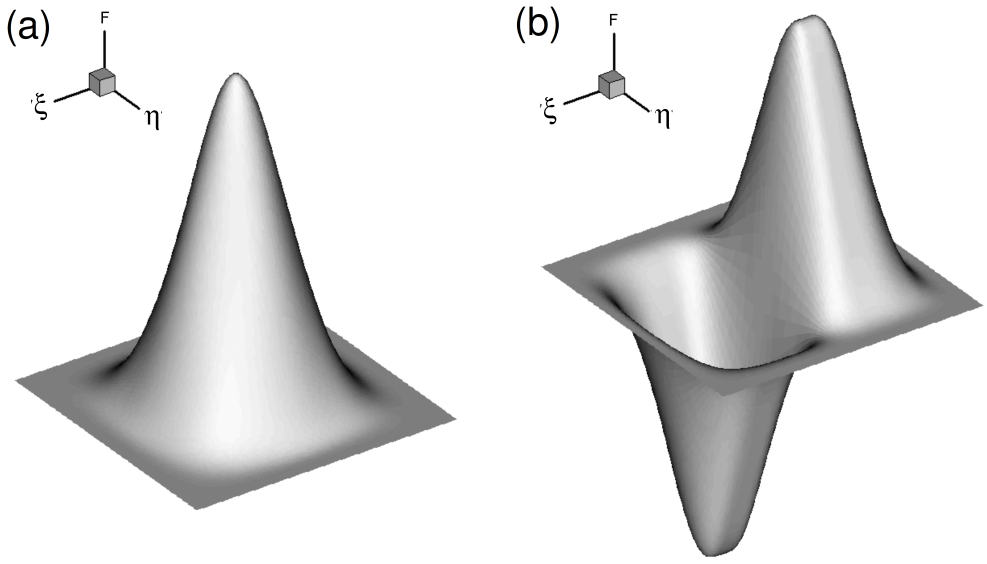


FIGURE 6.8: 2D Fup basis functions. F=$\text{Fup}_1(\xi,\eta)$; a)F and b)$\frac{\partial F}{\partial \xi}$.

For 1D Fup basis functions, Fup_n^l defined on Ξ^l can be represented as a linear combination of $n+2$ Fup_{n+1}^{l+1} defined on Ξ^{l+1} (see Eq. (6.1)). However, since 2D Fup basis functions are made as tensor product of the 1D Fup basis functions, $\text{Fup}_n^l(\xi,\eta)$ defined on the level l can be represented as a linear

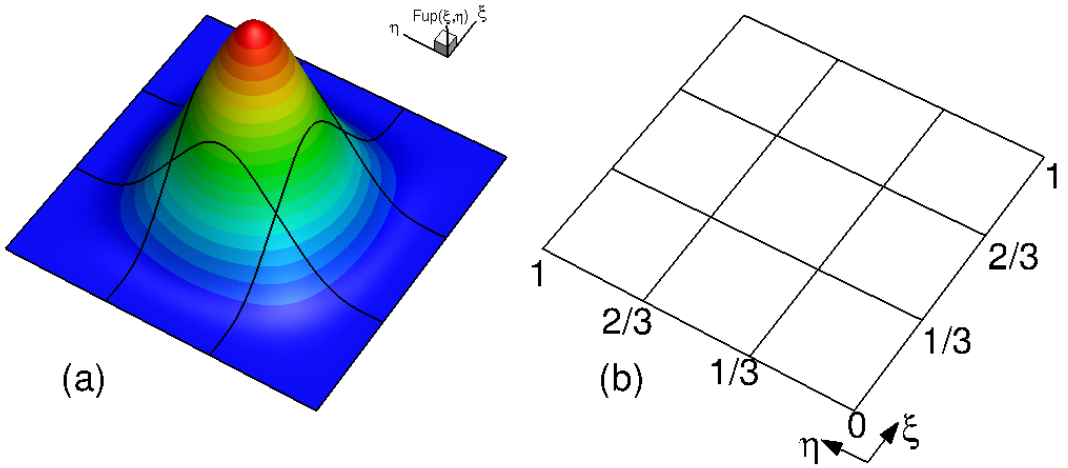


FIGURE 6.9: (a) $\text{Fup}_1^l(\xi,\eta)$ defined on the knot vector (b) $\Xi^l = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$ and $H^l = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$ in ξ- and η- directions, respectively.

combination of $(n+2)$x$(n+2)$ i.e., $(n+2)^d$ Fup_{n+1}^{l+1} defined on the level $l+1$,

$$Fup_n^l(\xi,\eta) = \sum_{i=0}^{n+1}\sum_{j=0}^{n+1} C_{n+1}^i C_{n+1}^j Fup_{n+1}^{l+1}\left(\xi - \frac{i}{2^{n+1}} + \frac{n+1}{2^{n+2}}\right) Fup_{n+1}^{l+1}\left(\eta - \frac{j}{2^{n+1}} + \frac{n+1}{2^{n+2}}\right) \tag{6.13}$$

where C_{n+1}^i and C_{n+1}^j are refinement coefficients (see Eq. (6.2) and Table 6.1), and d represents dimensional (in this case $d = 2$).

Figure 6.9 shows Fup_1^l defined on the knot vectors $\Xi^l = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$ and $H^l = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$, and Figure 6.10 shows its nine children (see Eq. (6.3)) defined on a knot vectors $\Xi^{l+1} = \{0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}, 1\}$ and $H^{l+1} = \{0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}, 1\}$. According to Eq. (6.13), Fup_1^l (Figure 6.9a) can be represented by a weighted summation of its nine children Fup_2^{l+1} (Figure 6.10a-i).

Refinement of $\text{Fup}_n(\xi,\eta)$ basis functions is again done in three steps. First, identify a set of basis functions $\mathcal{F}_p^l \subseteq \mathcal{F}^l$ to be refined at level l and designate them as *passive*. Remaining basis functions are designated as *active* ($\mathcal{F}_a^l = \mathcal{F}^l \setminus \mathcal{F}_p^l$). After obtaining the children at level $l+1$ (only for *passive*; $\mathcal{F}_p^l$) and defining them as *active* new setup of Fup basis functions at level $l+1$ is obtained by merging all of the basis functions that are *active* from levels l and $l+1$, thus obtaining the hierarchical Fup basis functions.

6.3.2 Basic adaptive algorithm parameters

Since multidimensional Fup basis functions are obtained (constructed) as tensor product of the one-dimensional Fup basis functions, basic parameters that were defined for 1D case will apply here with small changes.

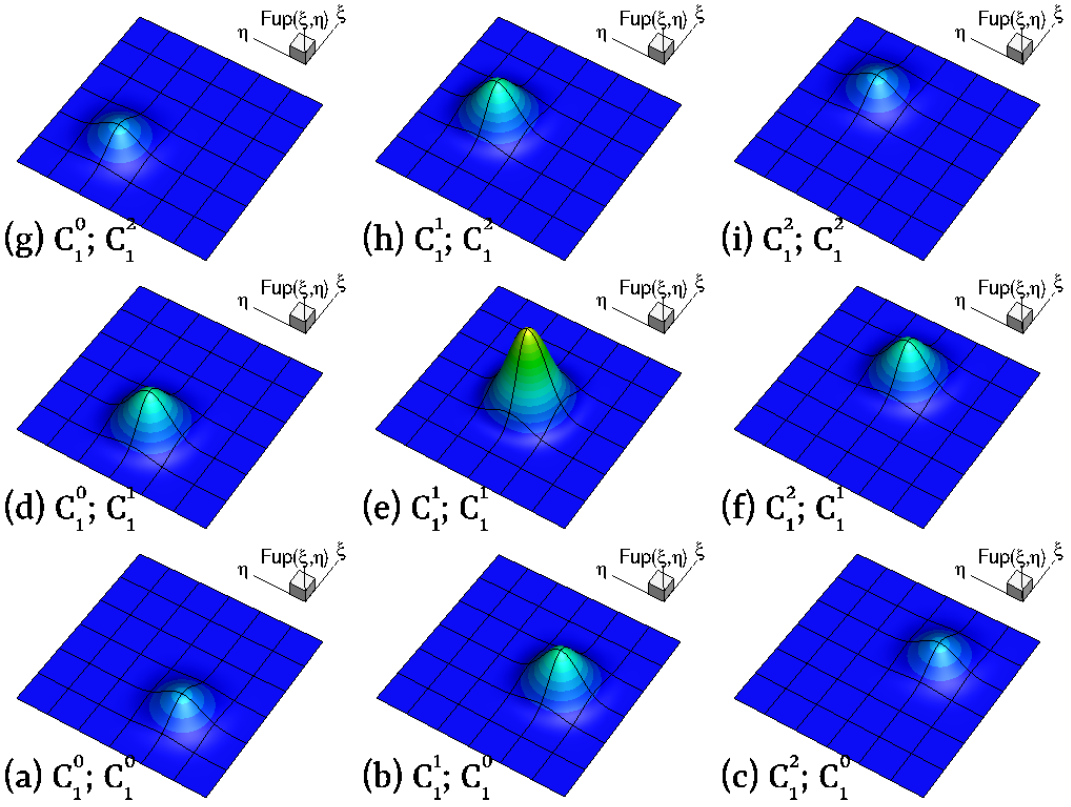


FIGURE 6.10: $\mathrm{Fup}_1^l(\xi,\eta)$ presented in the Figure 6.9 can be represented as linear combination of 9 $\mathrm{Fup}_2^{l+1}(\xi,\eta)$ with refinement coefficients C_{n+1}^i and C_{n+1}^j $(i=0,...,n+1; j=0,...,n+1)$.

The trial function space of uniformly distributed $\mathrm{Fup}_n(\xi,\eta)$ basis functions on the resolution level l and given order n are defined over the knot vectors in the form $\Xi=\{\xi_1,\xi_2,...,\xi_{m^\xi}\}$ and $H=\{\eta_1,\eta_2,...,\eta_{m^\eta}\}$ (see Figure 6.9b), where m^ξ and m^η represents number of basis functions in ξ- and η-directions, respectively. The number of basis functions on the first resolution level $m^{l,\xi}, m^{l,\eta}; l=0$ are defined as input parameters.

Length of the characteristic intervals $(\Delta\xi,\Delta\eta)$ are calculated as

$$\Delta\xi^l=\frac{\xi_{m^\xi+n+2}-\xi_1}{(m^{l,\xi}-n-1)2^l};\quad \Delta\eta^l=\frac{\eta_{m^\eta+n+2}-\eta_1}{(m^{l,\eta}-n-1)2^l}, \tag{6.14}$$

where ξ_1 and $\xi_{m^\xi+n+2}$ are first and last members of the knot vector in ξ-direction and η_1 and $\eta_{m^\eta+n+2}$ are first and last member of the knot vector in η- direction on the first resolution level $(l=0)$.

Basis functions whose compact support is at least partially located outside the domain are modified by satisfying ith derivation (see equation (3.41)). In the vector of mutually displaced Fup_n basis functions in 2D, it is necessary to modify the $(n+2)$ basis functions in ξ and/or η direction if they are near boundary of the domain. Figure 6.11 shows three cases of modification of boundary Fup basis functions defined on the knot vectors $\Xi=\left\{0,0,0,\frac{1}{3},\frac{2}{3},1,1,1\right\}$ and $H=\left\{0,0,0,\frac{1}{3},\frac{2}{3},1,1,1\right\}$. Figure 6.11b) shows

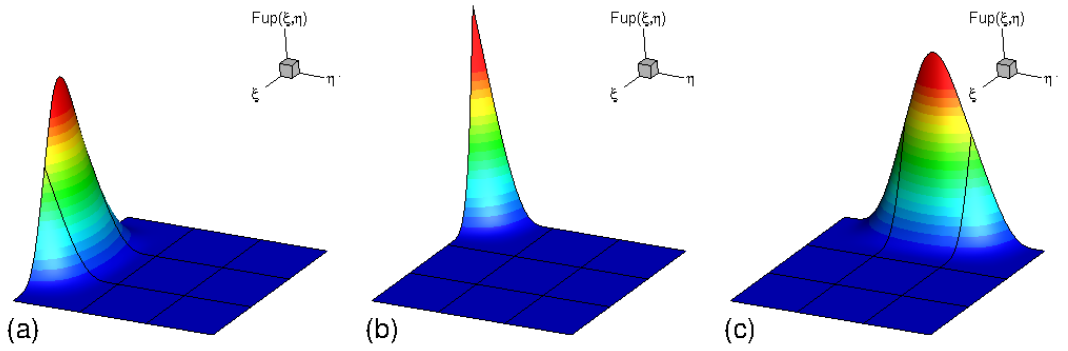


FIGURE 6.11: 3 cases of modified boundary $\text{Fup}_1^l(\xi,\eta)$ basis function. (a) Modified only in η- direction while in ξ- direction basis function is without modification. (b) Modified in both ξ- and η- direction. (c) Modified only in ξ- direction while in η- direction basis function is without modification.

$\text{Fup}_1(\xi,\eta)$ with modification in both directions (ξ- and η-) because compact supports are located outside of the domain in each direction. Furthermore, Figure 6.11(a,c) shows modification in only one direction since compact support is at least partially located outside the domain in η direction (Figure 6.11a) or ξ direction (Figure 6.11c).

Adaptive criteria (ε_A) defines whether Fup basis functions are kept or replaced while refining resolution level l. For ith control volume (CV), boundary is defined via $\Gamma_{i,l}$, $\Gamma_{i,r}$, $\Gamma_{i,u}$ and $\Gamma_{i,d}$, where subscript letter l (left), r (right), u (up) and d (down) represents side faces of the CV in a manner such that each CV boundary lies in the middle between two adjacent Greville points.

6.3.3 Control volume distribution by levels

To derive CV-IGA, the 2D parameter space defined by uniform (open) knot vector is shown on Figure 6.12. Domain is subdivided into a set of CVs such that each CV surrounds one corresponding vertex (Figure 6.12 - black circles) of basis functions and CVs boundary is defined at half distance between neighboring vertices. Note that like before (one-dimensional case) the number of basis functions is the same as number of CVs.

In Figure 6.12, the focus is on the grid point that represents vertex $V_{i,j}$, which has the grid points V_l, V_r, V_u and V_d as its neighbors and $CV_{i,j}$ boundaries (Figure 6.12 - red line) are marked as $\Gamma_{i,j,l}, \Gamma_{i,j,r}, \Gamma_{i,j,d}, \Gamma_{i,j,u}$.

Figure 6.13 shows a nested sequence of CVs domain, together with the corresponding vertices for each resolution level l, where each CV is linked with only one vertices i.e., the number of basis functions corresponds to the number of CVs.

CVs on the higher level ($l \geq 1$) are build in a slightly different way then on starting level where all CVs are nonoverlapping. Each CVs boundary on the higher level (see Figure 6.13) are positioned exactly half the length of the characteristic intervals $\Delta\xi$, $\Delta\eta$ (see Eq. (6.14)) from the corresponding Fup basis function vertex, thus higher levels (CVs) are overlapping with lower

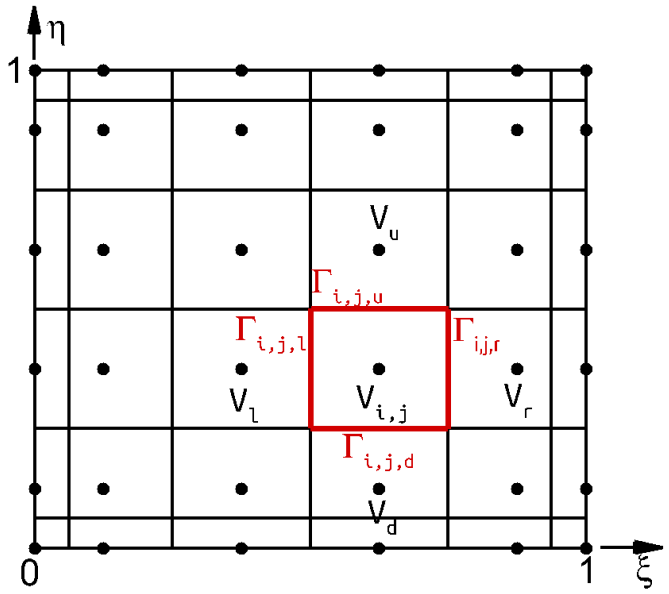


FIGURE 6.12: Control volume scheme for two-dimensional case.

levels (CVs). Overlapping process makes this algorithm even more robust, but at the cost of more expensive numerical integration.

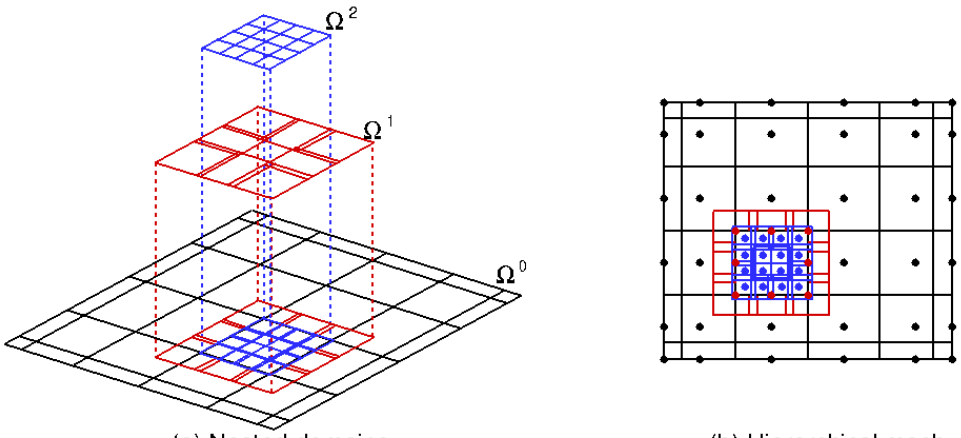


FIGURE 6.13: A nested sequence of CV domains for the construction of the Fup hierarchy according to relation $\Omega^l \supseteq \Omega^{l+1}$ for $l = 0, 1, 2$ for two-dimensional case.

Also, it should be emphasized that the CVs overlap is in order to obtain the simplest possible algorithm when compiling new levels and to simplify the numerical integration across each control volume. A more complex process which would avoid overlapping of CVs but would make integration process more complex is replacing each control volume with voronoi cells.

Figure 6.13b) shows uniformly distributed Fup_1 basis functions on the knot vectors Ξ^0 and H^0 with the position of vertices (black dots) and corresponding CV where each ith CV boundary is represented with solid line.

Furthermore, after replacing one Fup_1 (assigned as passive) basis function from the first resolution level, 9 Fup_2 basis functions (red dots, assigned as active) are introduced to the second resolution level (Ω^1). After assembling all active basis functions (assembles active Fup_1 and Fup_2 basis functions; $\mathcal{F}^1 = \mathcal{F}_a^0 \cup \mathcal{F}_a^1$), new CV distribution is set by placing new boundaries for higher order of Fup basis functions. CVs boundaries are set like in one-dimensional case. New level that is in contact with lower level l has increased CV area to avoid problem of linearly dependent equations. Enlargement is set as $\delta = \frac{1}{4}$. Since this is in two-dimensional domain, CVs in contact with the lower resolution level l form a ring of CVs that have an increased surface area due to the contact with the lower resolution level l by δ factor (see Figure 6.13). Refinement procedure continues and replaces one Fup_2 (assigning it as passive) basis function that is in the middle of previously inserted Fup_2 basis functions with 16 Fup_3 basis functions (blue dots) that are assigned as active. After assembling all active basis functions, new CV distribution is set for last resolution level, thus final CV setup is finished and is shown in Figure 6.13b). Figure 6.13a) represents a nested sequence of CVs after two refinement level and shows how with every new level, CVs act like a "patch" for lower level because it covers the part lost by removing CVs that are connected with *passive* Fup basis functions. Overlapping of the new CVs with lower level makes this algorithm even more robust but main advantage is easier process of constructing test (weight) functions in two-dimensional domains.

6.4 Stabilization methods

In addition to the adaptive algorithm (*hp*-refinement with Fup basis functions), an additional stabilization technique for controlling numerical oscillations is included, especially at lower resolution levels. The stabilization is based upon the modified Streamline Upwind/Petrov-Galerkin (SUPG) method (see [66], [67]) inside proposed CV-IGA. Essentially, the perturbation function is added to the test function,

$$p = \gamma \frac{\Delta\xi}{2} \frac{d\varphi}{d\xi} \tag{6.15}$$

where γ is a coefficient to control the magnitude of the perturbation function, $\Delta\xi$ is the element size (CV) and φ is the interpolation function, i.e., we are allowing more weight to the node in the upstream direction and reducing the weight to the node in the downstream direction. Applying this perturbation function leads to the addition of the artificial dispersion to the ADE (4.5). Basically, an artificial dispersion,

$$\tilde{D} = \gamma \frac{\Delta\xi}{2} v \tag{6.16}$$

which only has an effect in the streamline direction, is introduced to Eq. (4.18) to reduce oscillations. This is based on the argument that the oscillations

in the numerical solutions are generated by the high Peclet number in the streamline direction.

The global Peclet number is defined as,

$$Pe_g = \frac{v \cdot L}{D}. \tag{6.17}$$

where L represents the domain length, and D and v are dispersion and velocity, respectively. While the Peclet number is high, $Pe > 2$, the numerical solution of the advection-dominated problems will generate oscillations in the solutions. One way to stabilize oscillations is by changing the grid size. Since every new resolution level changes the compact support size, the grid Peclet number is defined as,

$$Pe_e = \frac{v \cdot \Delta\xi}{D}. \tag{6.18}$$

where $\Delta\xi$ represents the grid size. The magnitude of the coefficient γ,

$$\gamma = coth\left(\frac{Pe_e}{2}\right) - \frac{2}{Pe_e} \tag{6.19}$$

defines quantity of the added artificial dispersion which depends on the element Peclet number.

For two dimensional cases, the idea of upwinding can not be easily applied. Various methods have been proposed to implement the basic idea of upwinding to two dimensional analyses. In this thesis, upwinding method that extends to their use in two dimensional analyses with control volume procedure will be presented. The procedure of upwinding schemes for the two dimensional analysis is analogous to that for the one dimensional analysis. Oscillations in numerical solutions to advection-dispersion problems for the two dimensional cases are more complicated than for the one dimensional cases. In the two dimensional case, most of the upwinding schemes provide numerical solutions that still contain oscillations and a defect that is usually called the crosswind diffusion effect. This effect usually occurs when the velocity field is unaligned with the grid lines. The crosswind diffusion in numerical solutions is a spurious diffusion in the direction perpendicular to the streamline resulting in the deterioration of the numerical results.

The starting point is to consider the weighted residual formulation

$$\int_\Omega (w - p)\left[\nabla \cdot (\mathrm{v}u) - \nabla \cdot (D\nabla u)\right] d\Omega = 0 \tag{6.20}$$

where the perturbation function p is defined as

$$p = \frac{zC_{2r}h_e}{2||v_e||}\nabla \cdot (vw) = A\nabla \cdot (vw) \tag{6.21}$$

where v_e represents average element velocity ($||v_e|| = \sqrt{v_x^2 + v_y^2}$ in two dimensions), h_e represents element characteristic length and rest are defined

as

$$z = \coth(Pe) - \frac{1}{Pe} \approx min\left[1, \frac{Pe}{3}\right], \tag{6.22}$$

where Pe represents discrete Peclet number defined as

$$Pe = \frac{||v_e||he}{2D_e} \tag{6.23}$$

and last

$$C_{2r} = \begin{cases} 1 & \text{for steady-state problems} \\ \frac{2}{\sqrt{15}} & \text{for transient problems (phase optimization)} \\ C_{\Delta t} & \text{for transient problems} \end{cases} \tag{6.24}$$

with $C_{\Delta t}$ representing algorithmic Courant number $\left(C_{\Delta t} = \frac{||v_e||\Delta t}{h_e}\right)$. Since CVs that are used for two-dimensional problems are rectangular-shaped elements, the element characteristic length can be calculated as

$$h_e = \frac{1}{||v_e||}\left(|v_x|\Delta x - |v_y|\Delta y\right). \tag{6.25}$$

For other element shapes and more details see Yu and Heinrich [69].

The minus sign for perturbation function p in Equation (6.20) is made to correct the direction for the artificial diffusion term for the control volume method. After integrating Equation (6.20) by parts to reduce order of the continuity requirement of both the weighting function and the discrete variable gives,

$$\begin{aligned} -\sum \int_{\Omega_{CV}} \nabla w \cdot (vu - D\nabla r) d\Omega_{CV} + \int_{\Gamma_2} wvu \cdot \mathbf{n} d\Gamma_2 - \int_{\Gamma_2} whd\Gamma_2 \\ + \sum \int_{\Gamma_{CV}} \left[w\left(vu - D\nabla u\right) \cdot \mathbf{n}\right]_{-}^{+} d\Gamma_{CV} \\ + \int_{\Omega} \int_{\Omega} Awv \cdot \nabla\left[\nabla \cdot (vu) - \nabla \cdot (D\nabla u)\right] d\Omega \\ - \sum \int_{\Gamma_{CV}} Av \cdot \mathbf{n}\{w\left[\nabla \cdot (vu) - \nabla \cdot (D\nabla u)\right]\}_{-}^{+} d\Gamma_{CV} = 0. \end{aligned} \tag{6.26}$$

In Equation (6.26), the fifth integral represents a weighted residual formation of the equation

$$v \cdot \nabla\left[\nabla \cdot (vu) - \nabla \cdot (D\nabla u)\right] = 0. \tag{6.27}$$

Weighted residual formulation of Equation (6.27) is identically zero because any solution to the general conservation equation (see Equation (4.5)) subject to appropriate boundary conditions (see (4.6) and (4.7)) is also a solution of Equation (6.27). By setting w to be equal to the control volume test

(weight) function (see (4.15) makes the first integral in Equation (6.26) go to zero. Moreover, according to the Brooks and Hughes [70], the upwind contributions from the diffusion term in sixth integral in Equation (6.26) can be neglected, thus giving final weak form which can be written as

$$\begin{aligned} &\sum \int_{\Gamma_{CV}} (vu - D\nabla u) \cdot \mathbf{n} d\Gamma_{CV} + \int_{\Gamma_2} vu \cdot \mathbf{n} d\Gamma_2 \\ &- \int_{\Gamma_2} h d\Gamma_2 - \sum \int_{\Gamma_{CV}} Av \cdot \mathbf{n} \left[\nabla \cdot (vu)\right] d\Gamma_{CV} = 0. \end{aligned} \tag{6.28}$$

Last integral in Equation (6.28) arises as a result of the perturbation weighting and can be interpreted as an artificial diffusion term acting in the streamline direction. Furthermore, in case of limiting case when Peclet number equals to zero ($Pe = 0$), A equals to zero and the formulation reduces to standard control volume formulation.

Chapter 7

Numerical verification of adaptive algorithm on 1-D and 2-D problems

This chapter provides verification examples that illustrates the efficiency of the HF for function approximation and for modeling BVPs. Problems are presented in two parts, as one-dimensional and two-dimensional problems. The first example is an approximation of a known function that has a steep front in the middle of the domain. This example demonstrates HF's ability to capture sharp fronts by introducing new levels into a portion of the domain where it is needed. The second example describes the solution of the Poisson's equation, which has an exact solution (for one-dimensional and two-dimensional problem). Furthermore, third example presents advection-dominated advection-dispersion problem. For one-dimensional problem a steep from on the right boundary of the domain arises from the dominated advection and has exact solution. However, two-dimensional problem does not have an exact solution. Moreover, for both one-dimensional and two-dimensional problems, stabilization is included for better numerical solutions. Last example presents classical L-shape problem i.e., stationary heat conduction problem with exact solution. All of the examples illustrate the ability of HFs to efficiently and accurately describe different spatial scales.

7.1 Approximation of a known function

7.1.1 1D Approximation

The selected test function for 1-D problem $(f(x) : \Omega \to \mathbb{R})$ for the function approximation is

$$f(x) = -\tanh\left(\frac{x - 2/3}{0.008}\right) \tag{7.1}$$

with chosen numerical parameters at the zero level $n = 1$, $m^0 = 18$ and the domain defined as $\Omega = [0, 2]$. The error threshold is set as $\varepsilon_s = 10^{-5}$, which implies that the residual (see Eq. (6.6)) between the Fup approximation and the given function over all half CVs on every level must be less than this prescribed threshold. Figure 7.1 shows the evolution of the adaptive procedure using HF at six consecutive resolution levels starting with uniform Fup_1 basis functions.

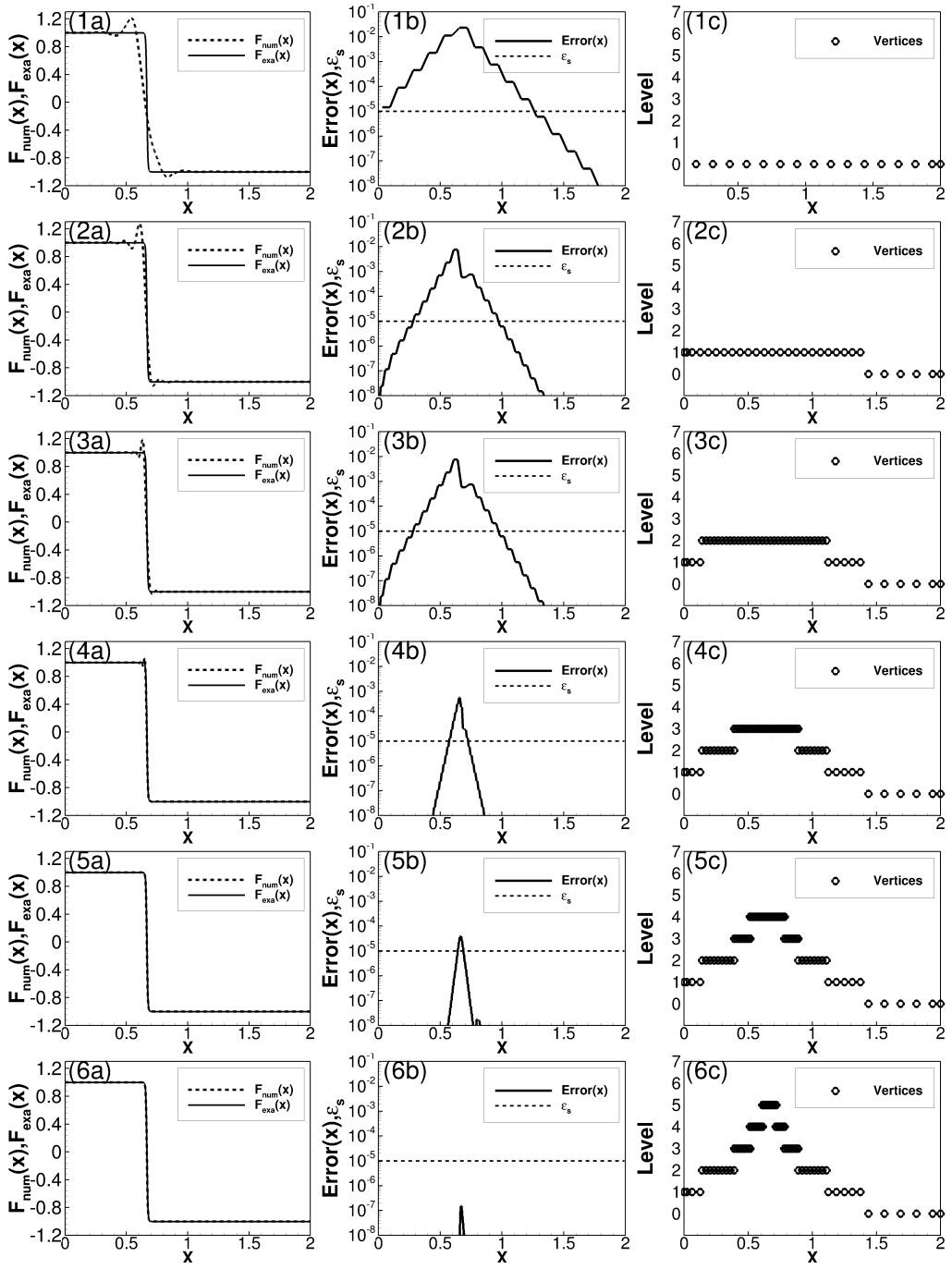


FIGURE 7.1: HF approximation of the function (7.1). (a) the given function (black solid curve) and its HF approximation (black dashed curve), (b) the absolute difference between the numerical and exact solution (black solid line) and the error threshold (black dashed line), and (c) the adaptive grid on different resolution levels (black circle represents Fup basis function vertices).

Function approximations (black dashed curve) of the given function (black solid curve) for each level are shown in Figure 7.1 (1a-6a). The error measure between the numerical approximation and the given function (presented as Error(x) in semi-log scale) is calculated as the integral difference

between those two functions on all half CVs, as shown in Figure 7.1 (1b-6b).

The basis functions used for the numerical approximation, i.e., active basis functions, at each level are represented by their vertices (Fig. 7.1, 1c-6c, black circles).

Although CVs are not directly shown, they can be visualized with the help of basis functions' vertices (see 6.2.2), since every CV's edge is placed between the vertices of the two adjacent functions.

The adaptive procedure (hp-refinement) is repeated until all residuals are less than the prescribed threshold, as shown in Figure 7.1. Note that fine CVs with a higher order of Fup basis functions are obtained only around the front, while in other regions, the adaptive grid uses only the lower order of Fup basis functions and coarse CVs. This helps to reduce the computational cost and increase efficiency.

For HF, the main idea is to control the numerical error through the adaptive algorithm over all CVs by employing the error threshold. Hence, it is interesting to test how the numerical approximation accuracy behaves when changing the adaptive threshold. Figure 7.2 shows a demonstration of the efficiency in terms of the L_2 error norm as a function of the total degrees of freedom (DOFs), and the slopes of the simulated lines represent the convergence rate (p). The total number of DOFs represents all basis functions needed to obtain the solution (m). Application of the control volume formulation with uniform Fup_n basis functions achieves the maximum theoretical convergence of order $p = n + 1$. However, the control volume formulation with adaptive hierarchical Fup_n basis functions yields spectral convergence (solid line with filled squares), as shown in Figure 7.2. Moreover, it should be noted that the adaptive procedure achieves a higher accuracy than the prescribed threshold (dashed line with filled squares, ε_A), clearly proving the control of the numerical error. It means that real numerical error of the function approximation is strictly less than the prescribed threshold.

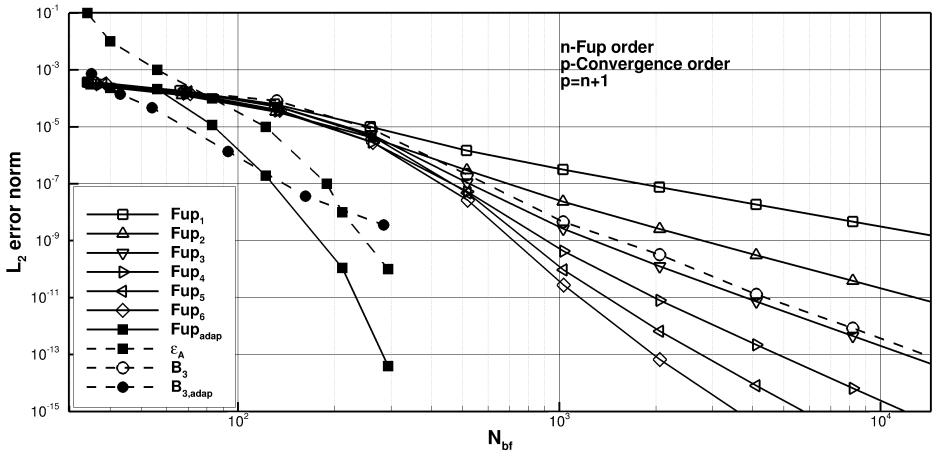


FIGURE 7.2: Convergence analysis obtained with uniform and adaptive Fup_n and B_3 basis functions.

Uniform B-spline basis functions such as B_3 achieve a maximum convergence order of $p = n + 1$ (dashed line with empty circles) just as the uniform

Fup$_n$ basis functions, as shown in Figure 7.2. Uniform Fup basis always gives more accurate solution than uniform B-splines due to its enhanced continuity, but the convergence rate is related to the basis functions order in both cases. However, the adaptive THB$_3$ (dashed line with filled circles) procedure does not achieve spectral convergence such as the adaptive procedure with HF basis functions. It reaches a higher accuracy then the uniform procedure but keeps the maximum theoretical convergence of order $p = n + 1$, as shown in Figure 7.2.

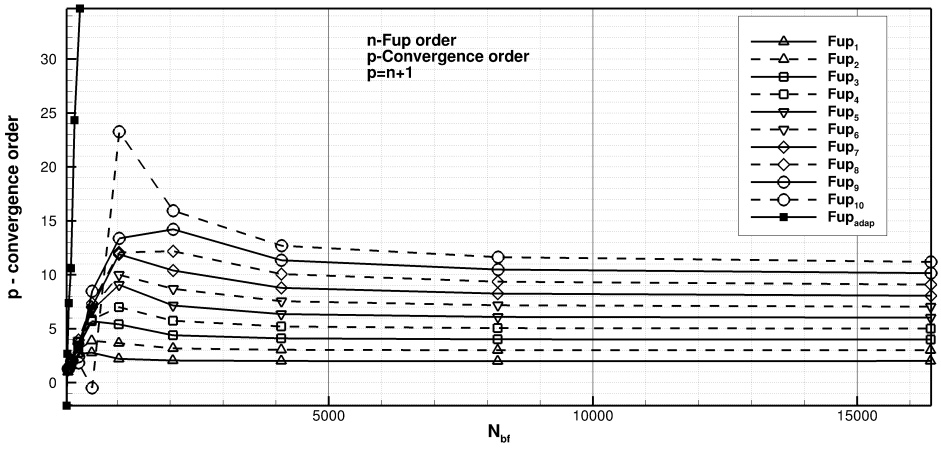


FIGURE 7.3: Analysis of the convergence order and number of basis functions: uniform Fup basis functions vs adaptive algorithm.

Next, the efficiency of the algorithm using hierarchical Fup basis functions is demonstrated in comparison with the results obtained with the uniform grid, by monitoring the number of basis functions required to achieve the maximum convergence order (Figure 7.3). At approximately $m = N_{bf} = 4096$ (DOF) uniform Fup_n $(n = 1, 2, ..., 10)$ basis functions achieve the maximum (theoretically expected, i.e., $p = n + 1$) convergence order. However, the adaptive procedure does not follow the same "rule", since it achieves spectral convergence with a significantly higher convergence order, as shown in Figure 7.3. It should be noted that the convergence order makes sense only when we are on fine grids and achieve a monotone convergence (above 4000 DOF for the uniform procedure).

7.1.2 2D Approximation

For the 2-D problem selected test function is

$$f(x, y) = \arctan\left(50\left(-0.25 + \sqrt{x^2 + y^2}\right)\right) \tag{7.2}$$

with chosen numerical parameters at the zero level $n = 1$, $m_x^0 = 10$, $m_y^0 = 10$ and the domain defined as $\Omega = [0, 1]^2$. The error threshold is set as $\varepsilon_s = 10^{-7}$,

which implies that the residual (see Eq. (6.6)) between the Fup approximation and the given function (7.2) over all half CVs on every level must be less than this prescribed threshold. Figure 7.1 shows the evolution of the adaptive procedure using HF at five consecutive resolution levels starting with uniform $\mathrm{Fup}_1(x, y)$ basis functions.

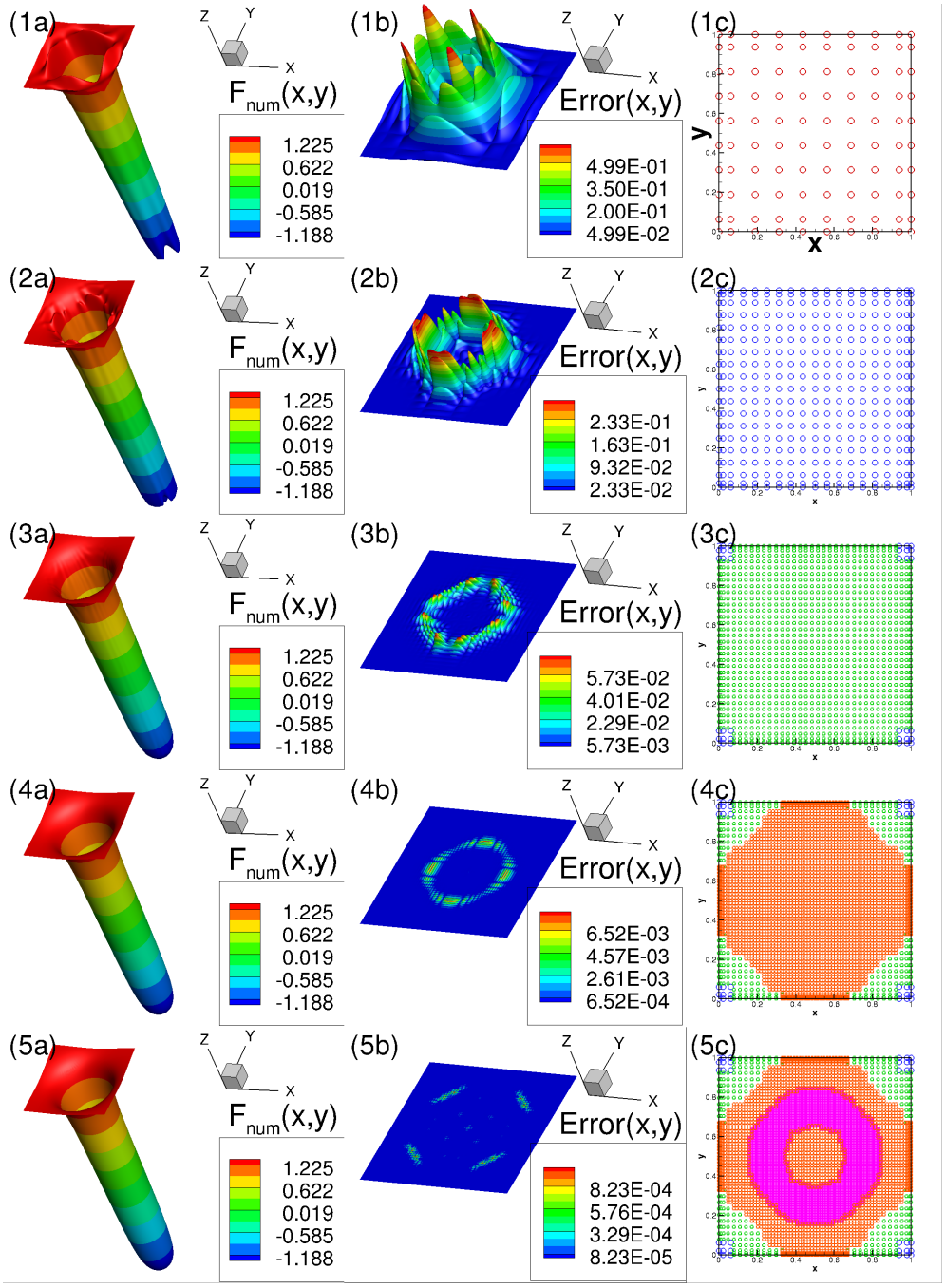


FIGURE 7.4: HF approximation of the function (7.2). (a) HF approximations of the given function, (b) the absolute difference between the numerical and exact solution and (c) the adaptive grid on different resolution levels where each color represents Fup basis function vertices on different level.

Function approximations ($Fun_{num}(x, y)$) of the given function (7.2) for each level are shown in Figure 7.4(1a-5a). The error measure between the numerical approximation ($Error(x, y)$) is calculated as the integral difference between those two functions on all quarters of the CVs (the 1-D error is tested on the halves of the CVs), as shown in Figure 7.4(1b-5b). Figure 7.4(1c-5c) shows active basis functions used for the numerical approximation and are represented by their vertices (each color represents one level, i.e., active basis functions on that level). Again, CVs are not directly shown but can be visualized with the help of the basis functions vertices.

The adaptive procedure is repeated until all residuals are less then the prescribed threshold. For given function (7.2) and adaptive threshold set as $\varepsilon_s = 10^{-7}$, adaptive procedure needs 5 levels to approximate given problem, as shown in Figure 7.4. Note that fine CVs with a higher order of Fup basis functions are obtained only around the "well" edges since this area is hardest to approximate. Moreover, in other regions the adaptive grid uses lower order of Fup basis functions and coarse CVs which helps reducing the computational cost and increases efficiency.

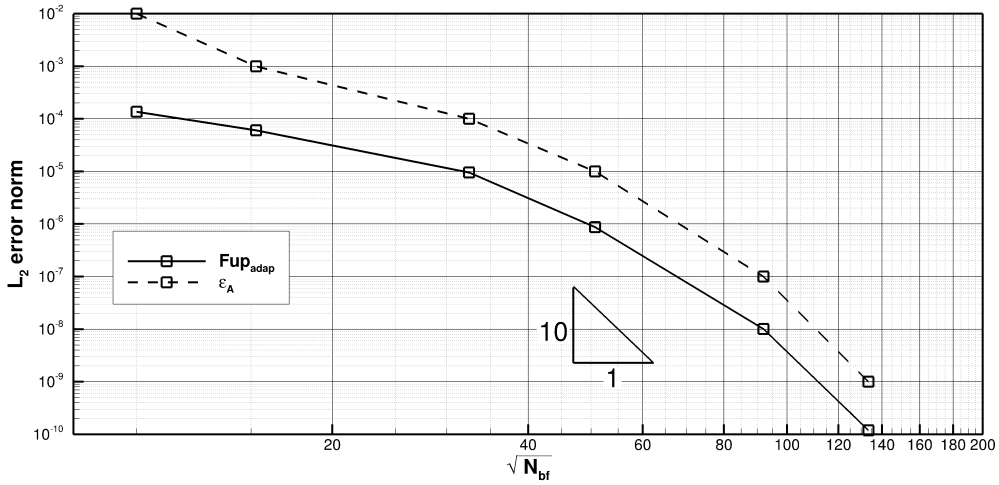


FIGURE 7.5: Convergence analysis obtained with adaptive Fup$_n$ basis functions.

Figure 7.5 shows efficiency in terms of the L_2 error norm as a function of the total DOFs with slope representing the convergence rate (p). Adaptive procedure just like in one-dimensional case yields spectral convergence (solid line with empty squares), as shown in Figure 7.5. Furthermore, adaptive procedure achieves a higher accuracy then the prescribed threshold (dashed line with empty squares, ε_A), thus proving the control of the numerical error. This means that the real numerical error of the function approximation is strictly less then the prescribed threshold.

7.2 Poisson equation

7.2.1 1D Numerical solution

For one-dimensional problem, reduced case of Eq. (4.5) leads to diffusion type of problem in order to describe flow in a heterogeneous porous media by setting advection term to zero

$$\nabla \cdot (K(x)\nabla h(x)) = 0 \tag{7.3}$$

where h is the hydraulic (piezometric) head and $K(x)$ is the hydraulic conductivity function that can be derived from Eq. (4.5). In the case of a homogeneous aquifer, the groundwater flow problem (7.3) reduces to the Laplace equation.

Heterogeneity is defined as

$$K(x) = e^{10sin(8x+1)-10} \tag{7.4}$$

with appropriate boundary conditions:

$$h(0) = 0 \tag{7.5}$$

$$h(L) = 1 \tag{7.6}$$

The exact solution of the presented problem is given by

$$h(x) = h(0) + [h(L) - h(0)]\frac{\int_0^x 1/(e^{10sin(8x+1)-10})dx}{\int_0^L 1/(e^{10sin(8x+1)-10})dx}. \tag{7.7}$$

The input parameters for numerical analysis are set as $n = 1$, $m^0 = 18$ with the domain size set as $L = 1(m)$. The error threshold is set as $\varepsilon_s = 5 \cdot 10^{-6}$, which implies that the mass conservation error over all half CVs on every level must be less than this prescribed threshold.

Figure 7.6 presents the numerical solution of the flow in heterogeneous porous media defined by Eq. (7.3). The adaptive grid captures the front, as shown in Figure 7.6(1b-4b). The convergence analyses are shown in Fig. 7.7 and Fig. 7.8 for the head solution for the uniform and adaptive procedure. The convergence analysis in IGA have been performed and tested using the collocation (C-IGA) or Galerkin (G-IGA) method, unlike the control volume method (CV-IGA) [62]. Figure 7.7 depicts a demonstration of the efficiency in the terms of the L_2 error norm as a function of DOF, and shows that the convergence rate for CV-IGA is the optimal $(p = n + 1)$ for odd and the suboptimal $(p = n)$ for even order (n) of basis functions. Malenica [49] performed an analysis with three methods (C-IGA, CV-IGA and G-IGA) and concluded that G-IGA (Galerkin) yields the optimal convergence rate for the Poisson problem for all orders of Fup basis functions (i.e., $p = n + 1$), while C-IGA yields suboptimal convergence rates of $p = n - 1$ for odd basis functions and $p = n$ for even basis functions. Figure 7.7 illustrates that CV-IGA yields the optimal convergence $p = n + 1$ for odd basis functions, but the suboptimal

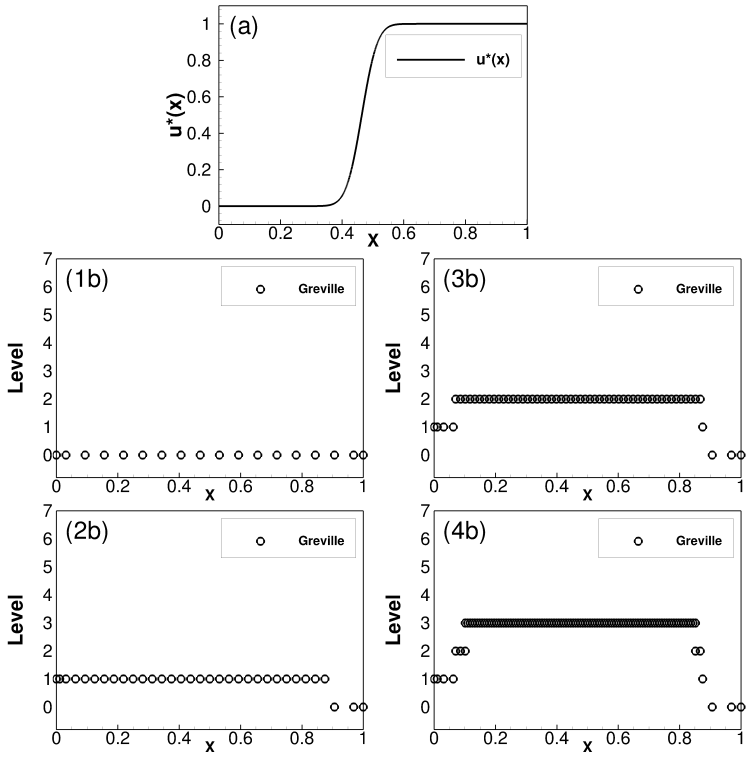


FIGURE 7.6: Numerical solution of the flow in heterogeneous porous media defined by Eq. (7.3). (a) HF approximation (black solid curve) on the last resolution level, (1b-4b) adaptive grid on different resolution levels (black circle represents Fup basis functions' vertices).

rate $p = n$ for even basis functions (it is still mystery in IGA community why it is case for C-IGA; the same situation appears for CV-IGA and even basis functions). It is important to emphasize that CV-IGA is a less expensive numerical procedure then G-IGA due to the reduced number of nonzero matrix entries and inexpensive numerical integrations. CV-IGA lies between G-IGA and C-IGA in terms of efficiency and computational cost. Once again, the adaptive procedure for the diffusive-like boundary value problem exhibits spectral convergence (black solid line with filled triangles).

Furthermore, the efficiencies of the uniform and adaptive algorithms are compared (Figure 7.7). The "mystery" [62] of even basis functions having a reduced convergence rate for the uniform procedure is also presented in Figure 7.8, where Fup_1 and Fup_2 have the same convergence order of $p = 2$, Fup_3 and Fup_4 have the same convergence order of $p = 4$, etc. However, the adaptive procedure (Fig. 7.7 - Fup_{adap}) again shows a significant deviation from the uniform distribution of basis functions with its spectral convergence order.

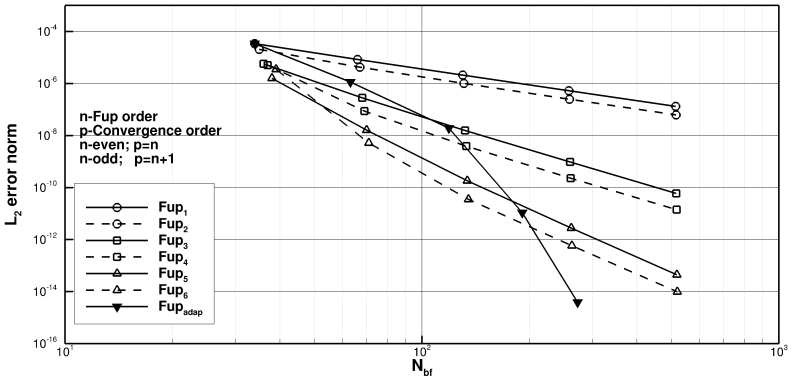


FIGURE 7.7: Convergence analysis of the head field for the uniform and adaptive procedure.

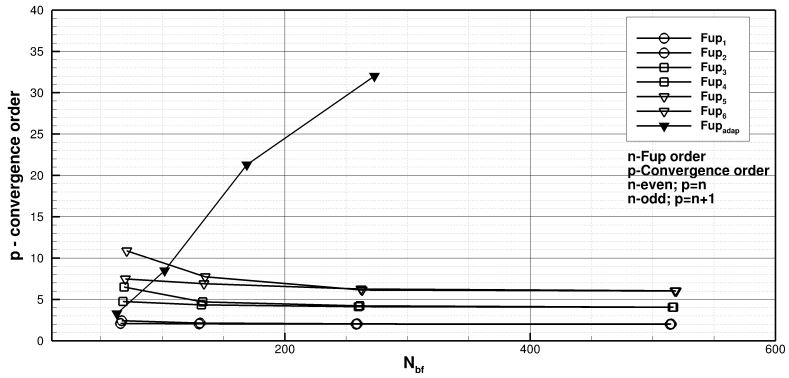


FIGURE 7.8: Analysis of the convergence order and number of basis functions: uniform Fup basis functions vs adaptive algorithm.

7.2.2 2D Numerical solution

For 2D Poissson equation verification problem, so called wavefront well problem is considered. It is commonly used example for testing adaptive refinement algorithms because of a steep wave front in the interior of the domain [7], [71], [72]. Parameters determine the steepness and location of the wave front. With the arctangent wave front that has exact solution that is similar to the function (7.2), there is a mild singularity at the center of the circle. However, for this test center of the circle is outside the domain, thus performance on the wave front is examined, not the singularity.

Problem is defined in the form

$$\nabla \cdot (-\kappa \nabla u(x,y)) = f(x,y) \quad (x,y) \in \Omega \tag{7.8}$$

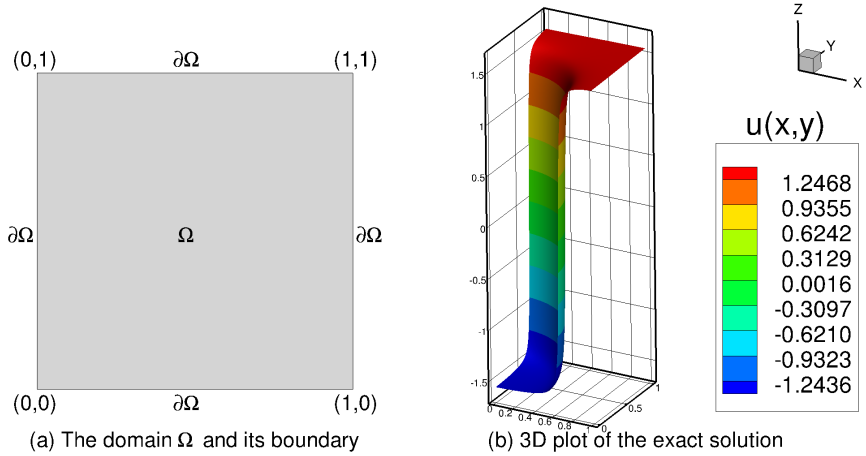
with boundary conditions

$$u(x,y) = u_D(x,y) \quad (x,y) \in \partial\Omega \tag{7.9}$$

The numerical simulation domain is defined by a square area $\Omega = [0,1]x[0,1]$ where the boundaries are $\Gamma_D = \partial\Omega$ and $\Gamma_N = \emptyset$ (see Figure 7.9a). The exact analytical solution for the pressure field is given by:

$$u(x,y) = \arctan\left(\alpha\left(r - r_0\right)\right) \quad \text{where} \quad r = \sqrt{(x-x_c)^2 + (y-y_c)^2} \tag{7.10}$$

where x_c and y_c represents center of the circular wave front, r_0 is the distance from the wave front to the center of the circle, and α gives the steepness of the wave front.



(a) The domain Ω and its boundary (b) 3D plot of the exact solution

FIGURE 7.9: Numerical solution domain and exact solution plot of the wave well problem defined by Eq. (7.8).

It should be noted that the right hand side $f(x,y)$ is generated by taking the Laplacian (∇^2) of the exact solution given in Equation (7.10). The exact solution depicted in Figure 7.9b displays a "front"-type of behavior where the solution is rapidly changing across a circular band (a quarter of a circle) inside the domain. For the conductivity matrix κ only isotropic case is considered, and for simplicity in deriving the source function, the conductivity equivalent coefficient is set equal to

$$\kappa = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7.11}$$

The adaptive simulation shown in Figures 7.10 and 7.11 is performed with starting polynomial degree $n = 1$. Number of basis functions on uniform level is defined as $m_x^0 = 18$, $m_y^0 = 18$, center of circular wave front is set at $x_c = y_c = -0.05$ with $r_0 = 0.7$ and $\alpha = 100$. The error threshold is set as $\varepsilon_s = 1 \cdot 10^{-4}$, which implies that the mass conservation error over all half CVs on every level must be less than this prescribed threshold.

Figures 7.10 and 7.11 presents the numerical solution of the wave front given by Eq. (7.8). With every new level, numerical solution becomes closer to the real solution (see Figures 7.10 and 7.11 1b-6b). Even though, difference between numerical solution and exact solution is presented in Figures 7.10 and 7.11 (1b-6b) because exact solution is known, it was not the adaptive criteria used for testing like in approximating function (7.1) and (7.2). Here,

adaptive criteria is used to check conservation error for each quarters of the particular i-th CV on the current resolution level. Quarters of the CVs are used because CV formulation exactly satisfies governing equation (i.e., the weak integral form of the conservation law), over each CV on the current resolution level. The adaptive grid captures the front (see Figures 7.10 and 7.11 1c-6c) and repeats adaptive procedure until conservation error is less then the prescribed threshold at each quarter of the CVs. For given parameters and using HF, six levels are needed in order to find numerical solution that has conservation error less then prescribed error threshold on all quarters of the CVs.

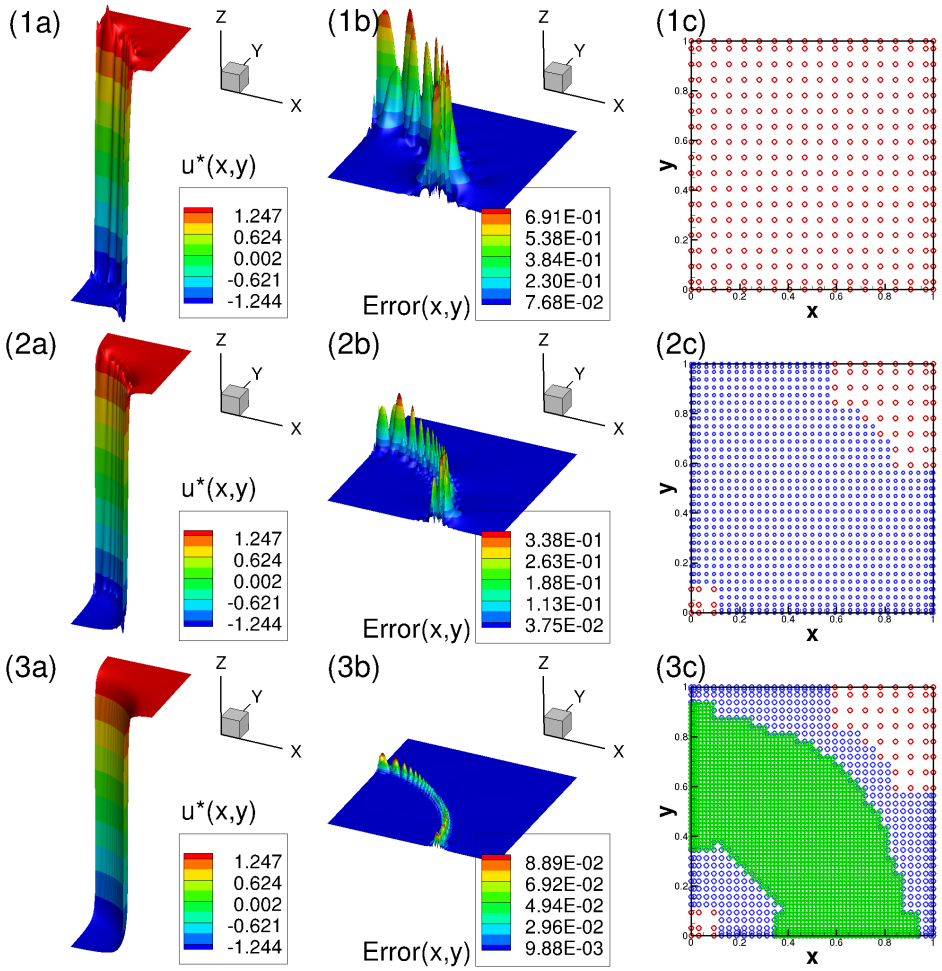


FIGURE 7.10: Numerical solution of the wave front well problem defined by Eq. (7.8) (first part, from first level up to third). (a) HF approximation, (b) the absolute difference between the numerical and exact solution and (c) the adaptive grid on different resolution levels where each color represents Fup basis function vertices on different level.

The convergence analysis for the uniform and adaptive procedure is shown in Figure 7.12. Figure 7.12 depicts a demonstration of the efficiency in the terms of the L_2 error norm as a function of DOF, and like in one-dimensional case shows that the convergence rate for CV-IGA is the optimal

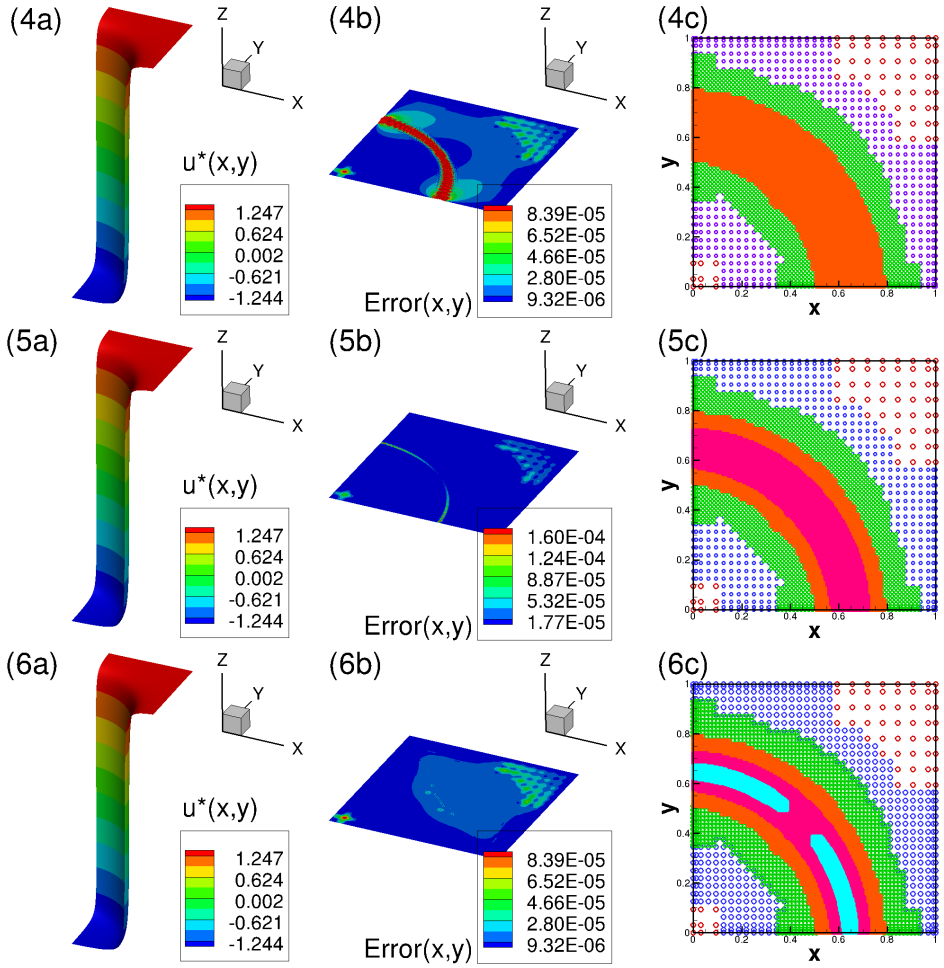


FIGURE 7.11: Numerical solution of the wave front well problem defined by Eq. (7.8) (from forth level up to sixth level). (a) HF approximation, (b) the absolute difference between the numerical and exact solution and (c) the adaptive grid on different resolution levels where each color represents Fup basis function vertices on different level.

($p = n + 1$) for odd and suboptimal ($p = n$) for even order of basis functions. It is important to emphasize that once again, the adaptive procedure for the diffusive-like boundary value problem exhibits spectral convergence (Figure 7.12 black solid line with filled squares).

7.3 Advection-dispersion equation

7.3.1 1D Numerical solution

The third example uses the advection-dispersion equation described by Eq. (4.5), where D $[m^2/s]$ and v $[m/s]$ are constant with appropriate boundary conditions:

$$u(0) = 0 \tag{7.12}$$

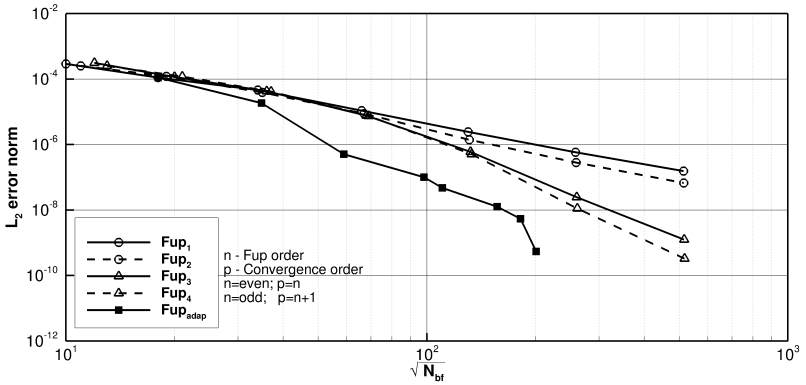


FIGURE 7.12: Convergence analysis of the wave front problem given in the form (7.8) for the uniform and adaptive procedure.

$$u(L) = 1 \tag{7.13}$$

and in this example, u[-] can be physically interpreted as concentration. The numerical input parameters are set as $n = 1$, $m^0 = 18$ and the domain size, dispersion and velocity are determined by the following parameters:

$$L = 1(m); \quad D = 1 \cdot 10^{-6}(m^2/s); \quad v = 1 \cdot 10^{-3}(m/s) \tag{7.14}$$

which defines the global Peclet number as $Pe_g = 1000$. The error threshold is set as $\varepsilon_s = 5 \cdot 10^{-6}$, which implies that the mass conservation error over all half CVs on every level must be less than this prescribed threshold. The exact solution of the presented steady 1D advection-dispersion problem is given by

$$u(x) = \frac{e^{(x \cdot v)/D} - 1}{e^{v/D} - 1}. \tag{7.15}$$

Figure 7.13 presents the evolution of the sharp boundary layer and corresponding adaptive spatial grid at five consecutive resolution levels. The results show that in the case of advection-dominated transport problem, the adaptive grid (Fig. 7.13, 1c-5c) accurately captures the front and controls the numerical oscillations and numerical error with higher resolution levels (Fig. 7.13, 1a-5a). The first level presents the significant onset of instabilities due to the uniform grid and high corresponding element (grid) Peclet number (Pe= 62.5). Instabilities are reduced at the third resolution level.

Additionally, the numerical error is reduced with each resolution level (Fig. 7.13, 1b-5b), i.e., introducing a higher order of Fup basis functions and fine control volumes helps to obtain a better numerical solution.

The stabilization technique (described in 6.4) is not added on all resolution levels but only for the first three levels ($l = 0, 1, 2$). Since the Peclet number is higher at the initial resolution levels, it means it is more relevant to apply stabilization only to those resolution levels.

Figure 7.14 presents the evolution of the numerical solution and corresponding adaptive spatial grid at five consecutive resolution levels with the

stabilization method applied to the adaptive algorithm. Significant improvement can be seen on the first two levels (e.g., Fig. 7.13-1a vs. Fig. 7.14-1a). At the same time, the computational cost is reduced since fewer basis functions (Fig. 7.13-5c 108 Fup basis functions vs. Fig. 7.14-5c 44 Fup basis functions) are needed to achieve the same mass conservation error on all half CVs (Fig. 7.13-5b vs. Fig. 7.14-5b).

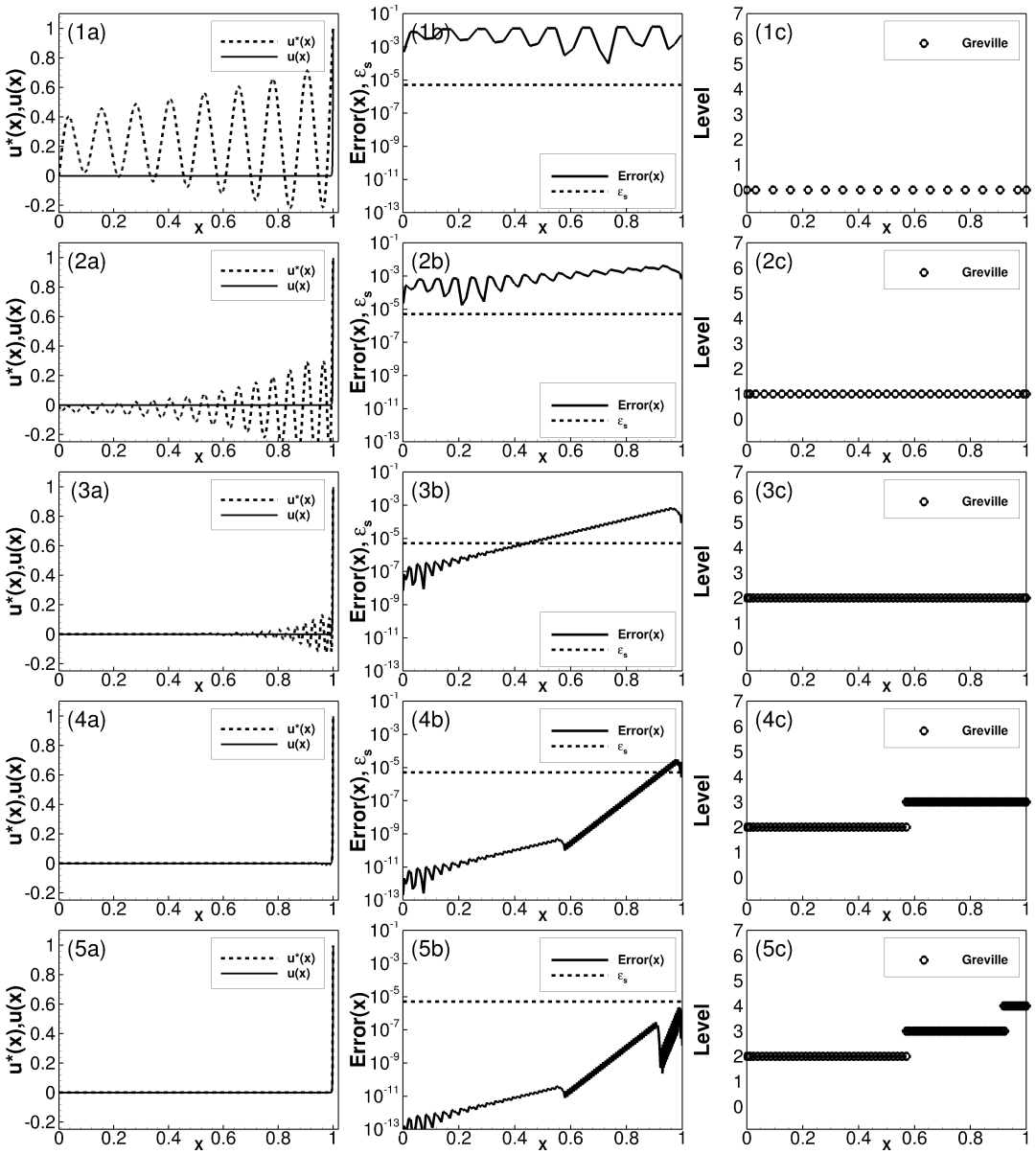


FIGURE 7.13: Numerical solution of the ADE (4.5) at different resolution levels (without stabilization procedure); (1a-5a) analytical solution (black solid curve) and its HF approximations (black dashed curve) (1b-5b) mass conservation error over all half CVs (black solid line) and thresholds (black dashed line) and (1c-5c) adaptive grid on different resolution levels (black circle represents Fup basis functions' vertices).

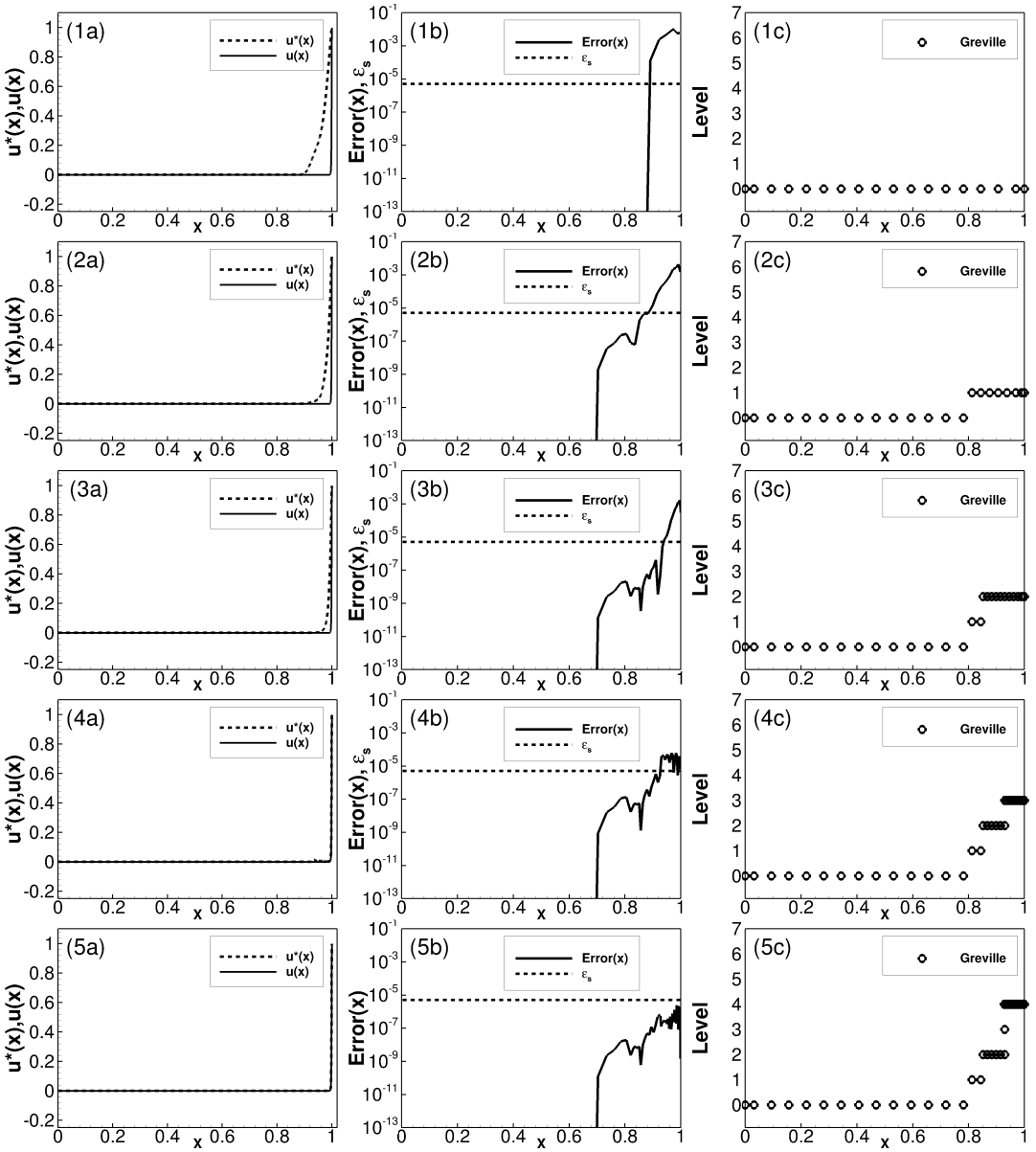


FIGURE 7.14: Numerical solution of the ADE (4.5) with stabilization procedure at the first three levels; (1a-5a) analytic solution (black solid curve) and its HF approximation (black dashed curve),(1b-5b) mass conservation error over all half CVs (black dashed line) and (1c-5c) adaptive grid on different resolution levels (black circle represents Fup basis functions' vertices).

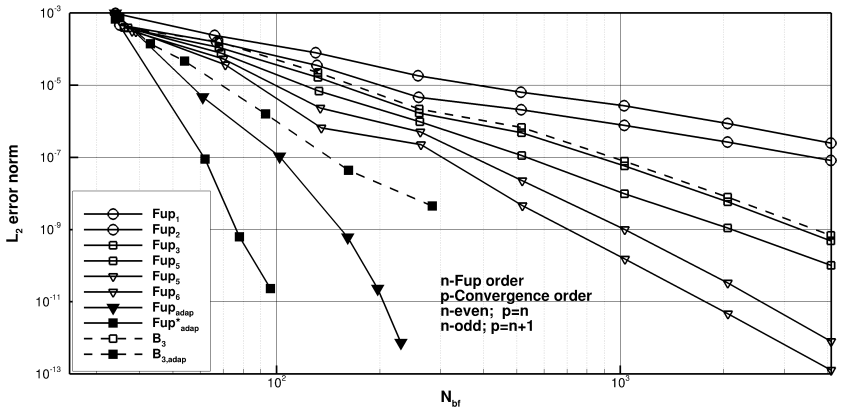


FIGURE 7.15: Convergence analysis for uniform and adaptive method

Furthermore, the efficiencies of the uniform and adaptive algorithms are compared (Figure 7.15). Again, the "mystery" of even basis functions having a reduced convergence rate for the uniform procedure is also presented. Fup_1 and Fup_2 have the same convergence order of $p = 2$, Fup_3 and Fup_4 have the same convergence order of $p = 4$, etc., as shown also in Figure 7.16.

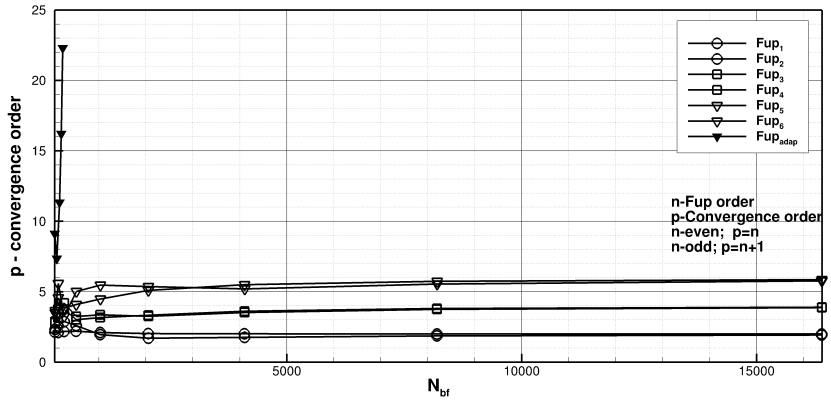


FIGURE 7.16: Analysis of the convergence order and number of basis functions: uniform Fup basis functions vs adaptive algorithm

However, the adaptive procedure with HF (Fig. 7.15- solid line with filled triangles, Fup_{adap}) again shows a significant improvement compared to the adaptive procedure with THB (Fig. 7.15- dashed line with filled squares, $\text{B}_{3,adap}$) and the uniform distribution of basis functions (Fup or B-spline), with its spectral convergence order. It is important to point out that the adaptive technique with the stabilization method (Fig. 7.15- solid line with filled squares, Fup^*_{adap}) at the first three levels achieves even better efficiency. Since the Peclet number is lower at the higher resolution levels, there is no need for stabilization on those levels.

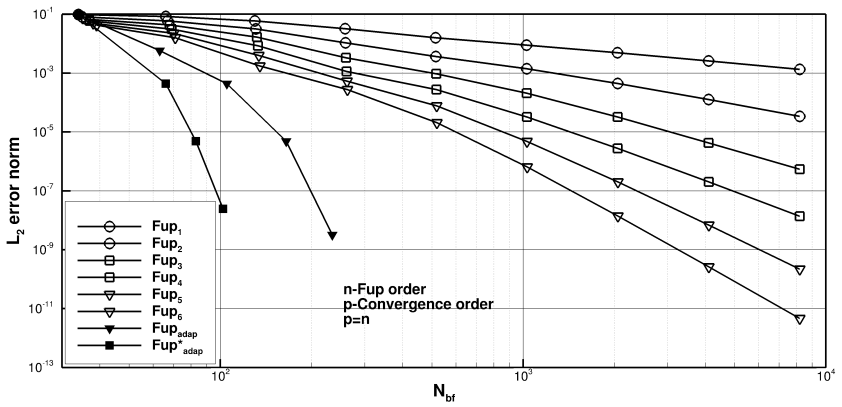


FIGURE 7.17: Convergence analysis for the uniform and adaptive method (first derivative of the solution).

Figure 7.17 presents the convergence analysis of the first derivative of the solution (sum of advective and dispersive flux). Now, CV-IGA yields the optimal convergence rate $(p = n)$ for all orders of uniform Fup basis functions. Since calculation of fluxes is usually more important than basic variable, CV-IGA can be regarded as optimal IGA methodology satisfying local/global conservation conditions as well as obtaining accuracy very close to the optimal Galerkin solution. The adaptive procedure with stabilization (Fig. 7.17- solid line with filled squares, Fup^*_{adap}) again shows a significant improvement compared to the uniform distribution of basis functions as well as the adaptive technique without the stabilization method (Fig. 7.17- solid line with filled triangles, Fup_{adap}).

7.3.2 2D Numerical solution

Two-dimensional benchmark example is taken from [14], [17], [24] which consists of solving the advection-dispersion equation

$$D\Delta u - v \cdot \nabla u = 0 \tag{7.16}$$

on the unit-square with discontinuous Dirichlet boundary conditions (see Figure 7.18). The dispersion D coefficient is chosen extremely small ($D = 8 \cdot 10^{-4}$) compared to the advection velocity $v = (\sin\theta, \cos\theta)^T$, thus very sharp layers arise that start at the discontinuity of the boundary condition. Sharp interior and boundary layers require stable numerical techniques in addition to increased resolution to be accurately captured. Adaptation with Fup basis functions (hp-refinement) gives very accurate numerical results but still needs large number of basis functions (unlike uniform basis layout) so SUPG stabilization (see 6.4) is employed as additional factor to adaptive procedure.

Adaptive resolution of the internal and boundary layers are investigated with the present HF procedure starting from $m_x^0 = 18$, $m_y^0 = 18$ Fup basis functions on the starting (uniform) resolution level. The error threshold is set as $\varepsilon_s = 1 \cdot 10^{-4}$, which implies that the mass conservation error over all

quarters CVs on every level must be less then this prescribed threshold. The exact solution of the presented problem is not known.

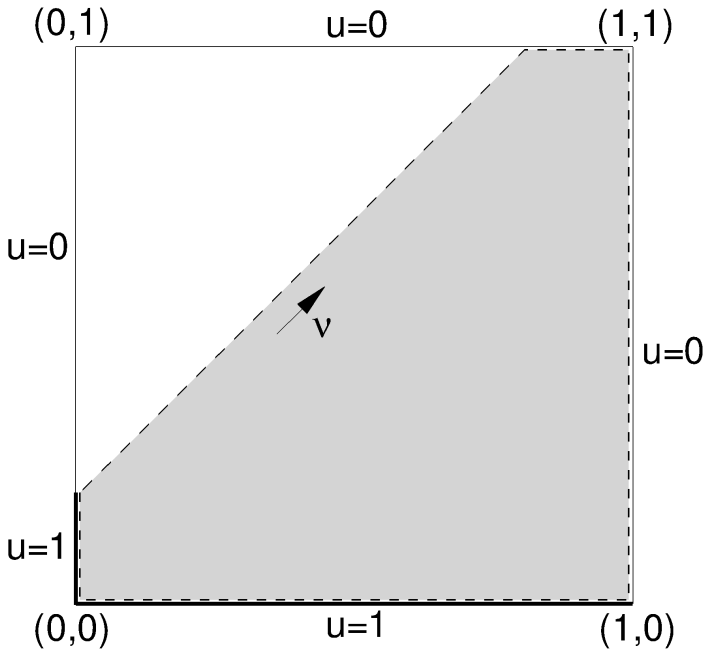


FIGURE 7.18: Numerical solution domain with discontinuous Dirichlet boundary conditions for the Advection-dispersion problem.

Figure 7.19 presents the evolution of the sharp boundary layer and corresponding adaptive spatial grids at five consecutive resolution levels in two-dimensional domain. It can be observed that the refinement captures the location of the internal and the boundary layers very well. Despite the high Peclet number no stability or robustness issues in the adaptive algorithm were encountered. There are some under- and overshooting of the first (uniform) level along the internal layer. These nonphysical oscillations are a result of the discretization of the first order spatial derivative in the convective term when the convective term dominates the other terms in the governing equation. Moreover, five adaptive HF refinement levels are required to get control over the undershooting close to the jump at the inflow boundary. Mass conservation error detects the internal layer as well as the boundary layer. However, the refined patches are not only around the boundary layer and the internal layer, solely because adaptive algorithm or mass conservation error is not good enough solution for this form of the problem. But, just like in one-dimensional case, here, in two-dimensional case for advection-dominated problem, stabilization process is also included (see Figure 7.20).

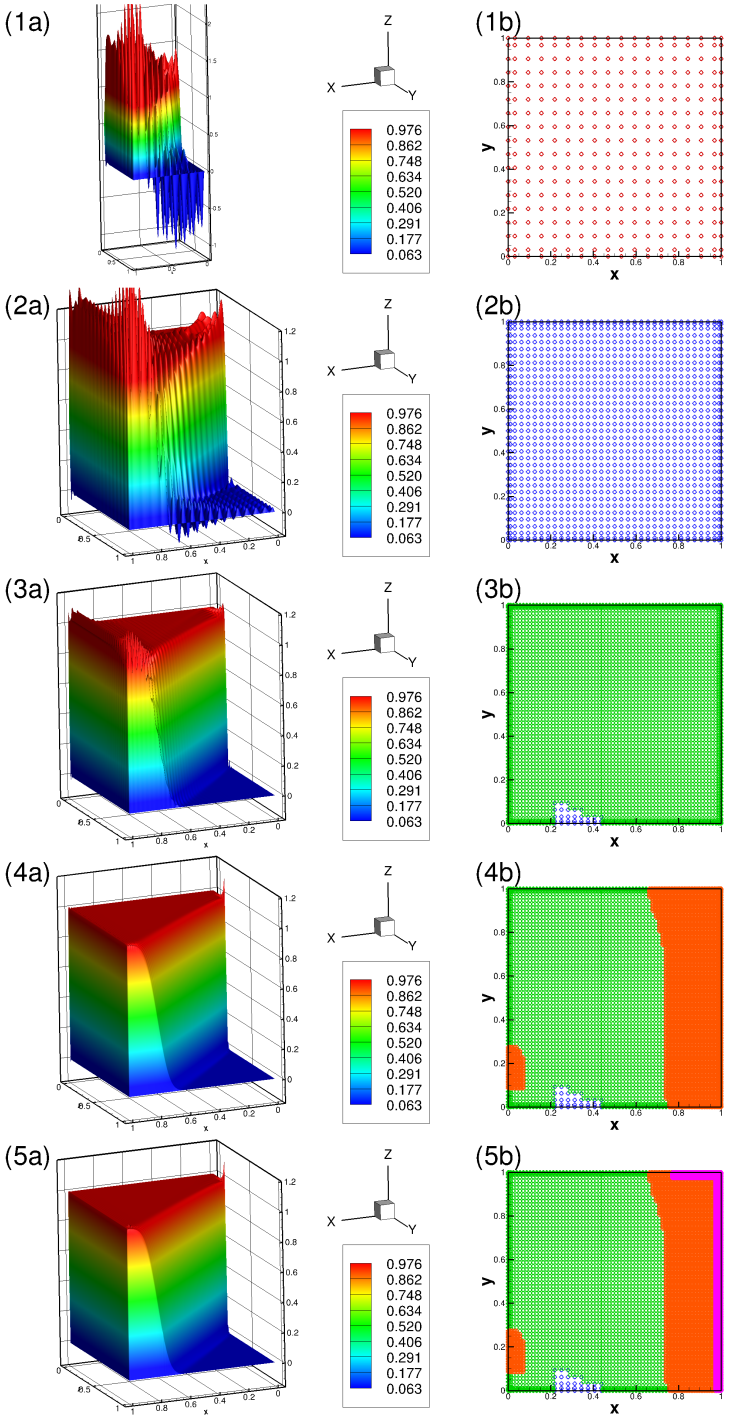


FIGURE 7.19: Numerical solution of the ADE (7.16) at different resolution levels (without stabilization); (1a-5a) HF approximation

It is more relevant to apply stabilization only to the first couple of levels (in this case, for the first three levels, $l = 0, 1, 2$) because the Peclet number is higher at the initial resolution levels. Figure 7.20 presents the evolution of the numerical solution and corresponding adaptive spatial grids at four consecutive resolution levels with the stabilization method (see 6.4) applied

to the adaptive algorithm. Comparing solutions with (Figure 7.20) and without (Figure 7.19) stabilization method yields significant improvement on the first (uniform) level. Moreover, the computational cost is reduced since fewer basis functions are needed on higher levels to achieve the same mass conservation error on all quarters of the CVs.

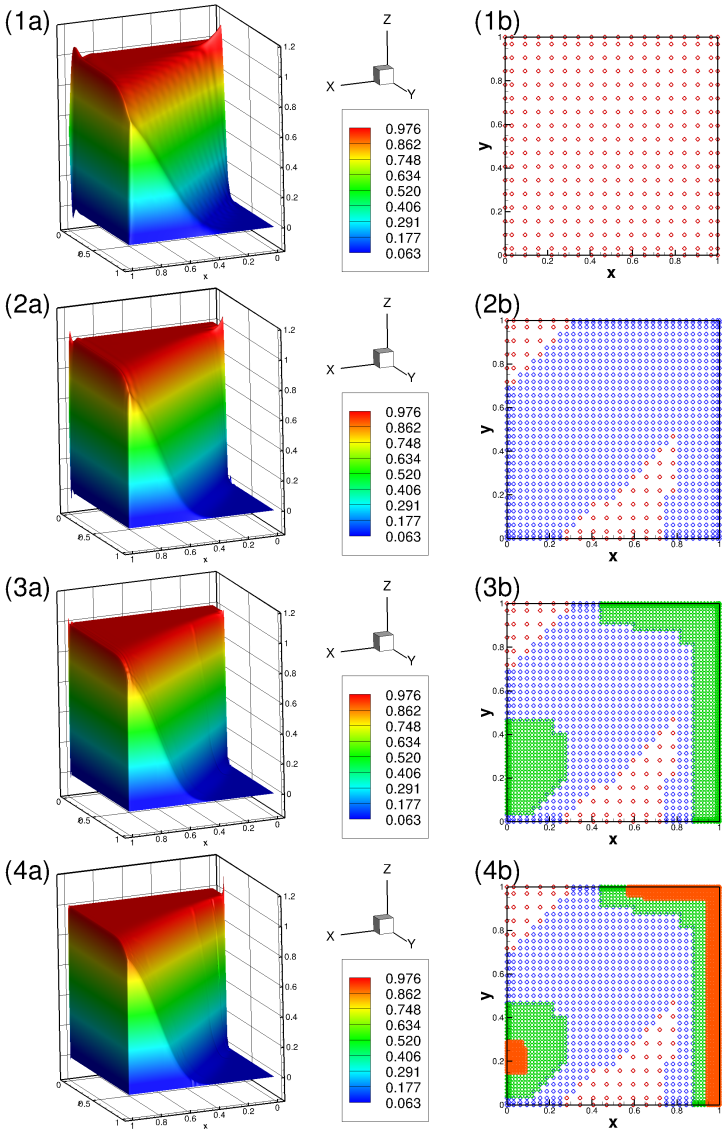


FIGURE 7.20: Numerical solution of the ADE (7.16) with stabilization procedure at different resolution levels; (1a-4a) HF approximations; (1b-4b) corresponding adaptive spatial grids

Figure 7.21 presents the convergence analysis of the uniform and adaptive algorithms using Fup_n basis functions, with respect to the degrees of freedom used to achieve a certain accuracy. The results for the uniform ($n = 1, 2, 3, 4$) grids shows a significantly reduced convergence rate due to the discontinuity in the boundary conditions (see Figure 7.18). However, as expected, HF adaptive algorithm is again superior to a uniform strategy, and achieves spectral convergence rate which is quite impressive for this type of problems.

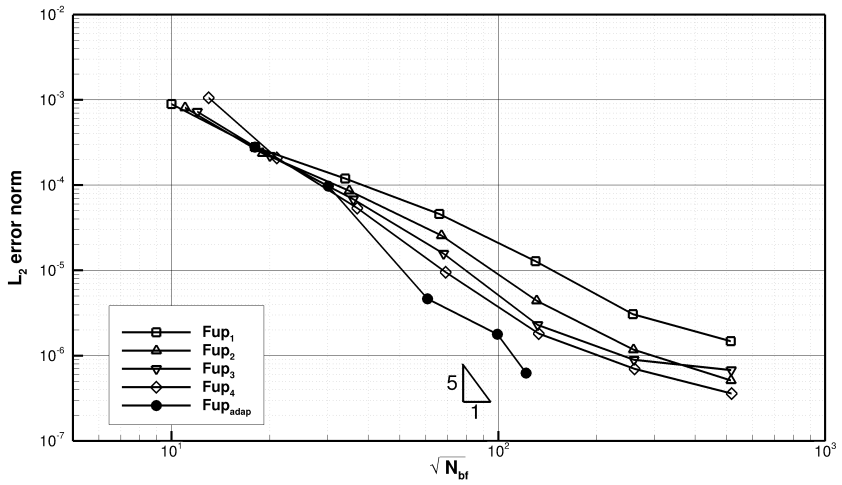


FIGURE 7.21: Convergence analysis for uniform and adaptive method

7.4 Engineering problems with irregular geometry

In this part, results obtained with adaptive algorithm on problems with irregular geometry are presented. Considering stationary heat conduction problem

$$\Delta u = 0 \tag{7.17}$$

on an L-shaped domain $\Omega = [-1,1]^2 \setminus [0,1]^2$, see Figure 7.22a), with boundary conditions

$$u = 0 \quad on \quad \Gamma_D \tag{7.18}$$

$$\frac{\partial u}{\partial n} = q_N \quad on \quad \Gamma_N \tag{7.19}$$

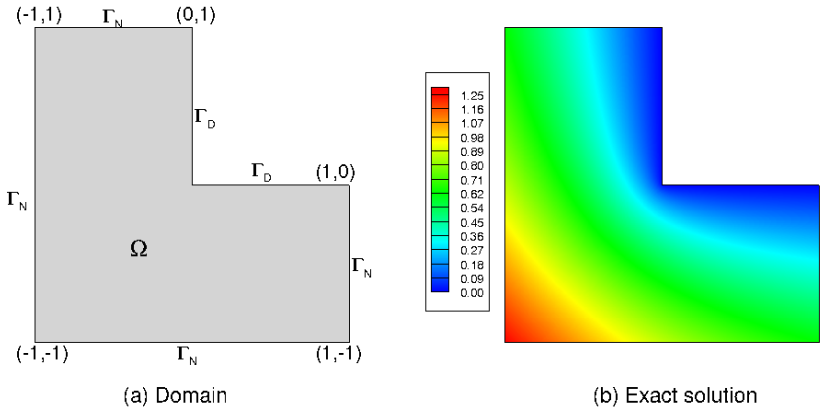


FIGURE 7.22: The L-shape problem: a) Numerical solution domain with boundary conditions and b) exact solution plot.

such that the exact solution is given by

$$u = r^{2/3} \sin\left(\frac{2\theta - \pi}{3}\right) \tag{7.20}$$

in polar coordinates (r, θ), where $r^2 = x^2 + y^2$ and $\theta = \arctan(y/x)$. The expression for the Neumann boundary condition q_N is derived based on the exact solution (7.20). For the given elliptic problem, the re-entrant corner at $(0, 0)$ in the domain causes a singularity in the solution. An optimal convergence rate is not obtained when uniform mesh refinement is performed for the problems where the solution is not sufficiently smooth [22].

Presented HF procedure starts with $m_x^0 = 18$, $m_y^0 = 18$ Fup basis functions on the starting (uniform) resolution level. The error threshold is set as $\varepsilon_s = 9 \cdot 10^{-3}$, which implies that the mass conservation error over all quarters CVs on every level must be less then this prescribed threshold. The exact solution of the presented problem is shown in Figure Figure 7.22b).

L-shaped domain is discretized by two elements (patches), as shown in Figure 7.23b.

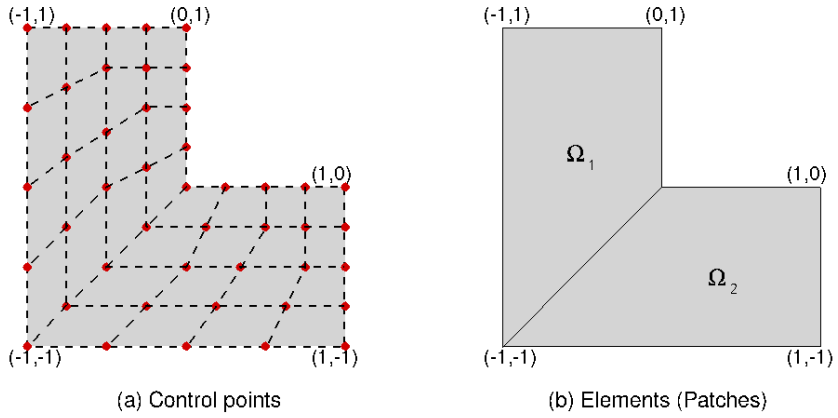


FIGURE 7.23: The L-shape problem: a) Fup discretized geometry with a $n_{cp} = 25$ number of control points per each element, and b) for $n_{el} = 2$ number of elements. In a) red circles represent the control points, whereas the shaded region is the modeled geometry.

Figure 7.23a) shows control points for the coarse mesh. The coordinates of the control points serve as coefficients (see (4.1)) in spline representation for transforming one 2D subdomain from the parameter space to the physical space.

Figure 7.24 presents the numerical solution for the stationary heat conduction problem in two-dimensional domain. The area of interest is detected and resolved locally using HF basis functions (see Figure 7.24). Refinement captures the re-entrant corner in the domain at $(0, 0)$ where a singularity in the solution occurs. For given parameters and using HF, six levels are needed in order to find numerical solution that has conservation error less then prescribed error threshold on all quarters of the CVs.

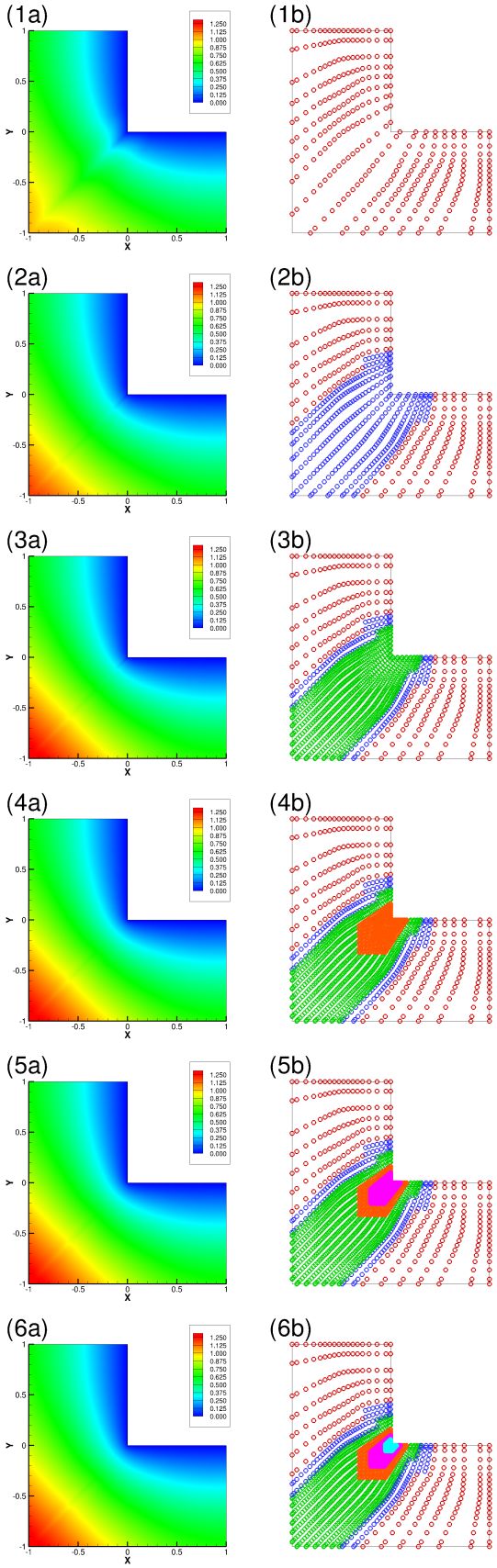


FIGURE 7.24: Numerical solution of the stationary heat conduction problem defined over an L-shaped domain (governed by Laplace equation (7.17)) at different resolution levels; (1a-6a) HF approximations; (1b-6b) corresponding adaptive spatial grids.

It should be noted that CVs that are linked with odd order of Fup basis functions on level l and have one side of CVs boundary on top of the Jacobian discontinuity (see Figure 7.23 where two elements share a common border) are enlarged to resolve numerical problems that can occur while solving governing equations on those CVs. For more details about enlargement process see 6.3.3.

The convergence analysis is performed using L_2 norm and is plotted in Figure 7.25 for uniform Fup_1, Fup_2 and HF basis functions. It can be observed that adaptive HF basis functions again improves the convergence rate by achieving spectral convergence rate. However, uniform grids shows a significantly reduced convergence rate due to the re-entrant corner at $(0, 0)$ in the domain (singularity). The present numerical example thus confirms that adaptive algorithm works well for rough problems.

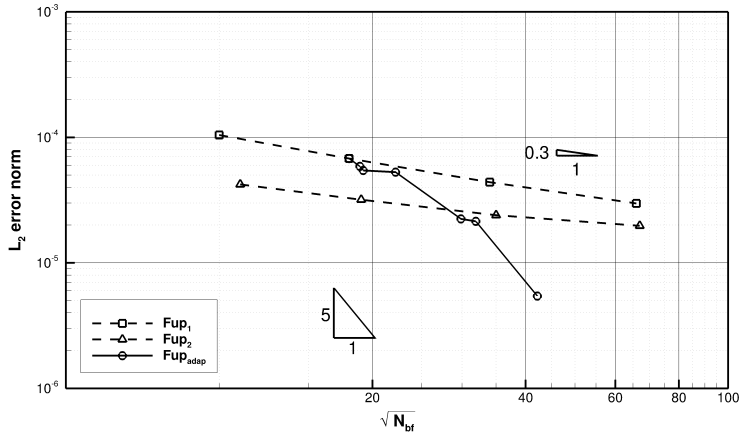


FIGURE 7.25: Convergence analysis for uniform and adaptive method

7.5 Space-time advection-dispersion problem

This section describes the mixing of transport processes, for instance in porous media. The one-dimensional advection-dispersion process can be described by the following equation, in the form:

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} - v\frac{\partial C(x,t)}{\partial x} \tag{7.21}$$

with appropriate boundary conditions:

$$C(x,0) = 0 \tag{7.22}$$

$$C(0,t) = C_0; \quad \frac{\partial C(2,t)}{\partial x} = 0 \tag{7.23}$$

where C represents the dependent variable (concentration $[M/L^3]$), while D is the dispersion coefficient and v is the transport velocity in the x direction.

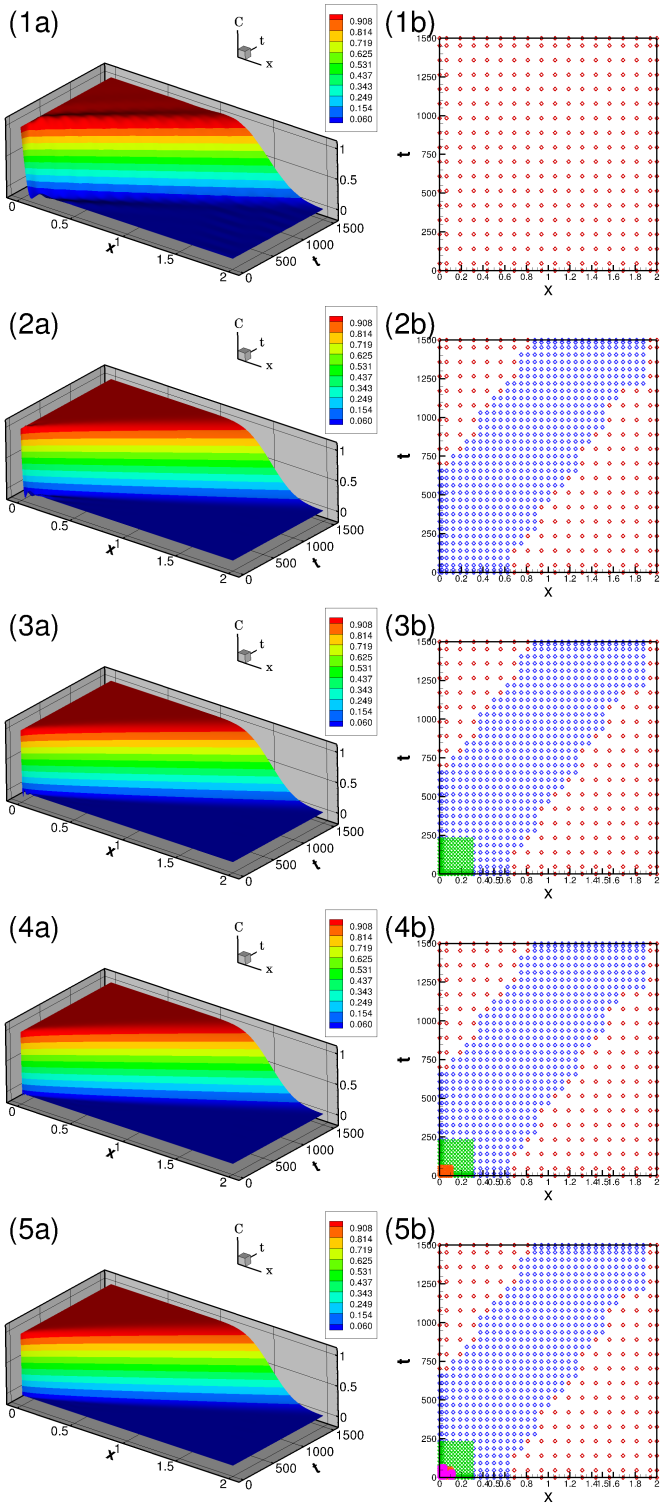


FIGURE 7.26: Numerical solution of the ADE (7.22) at different resolution levels; (1a-5a) HF approximation (without stabilization), (1b-5b) corresponding adaptive time-spatial grids.

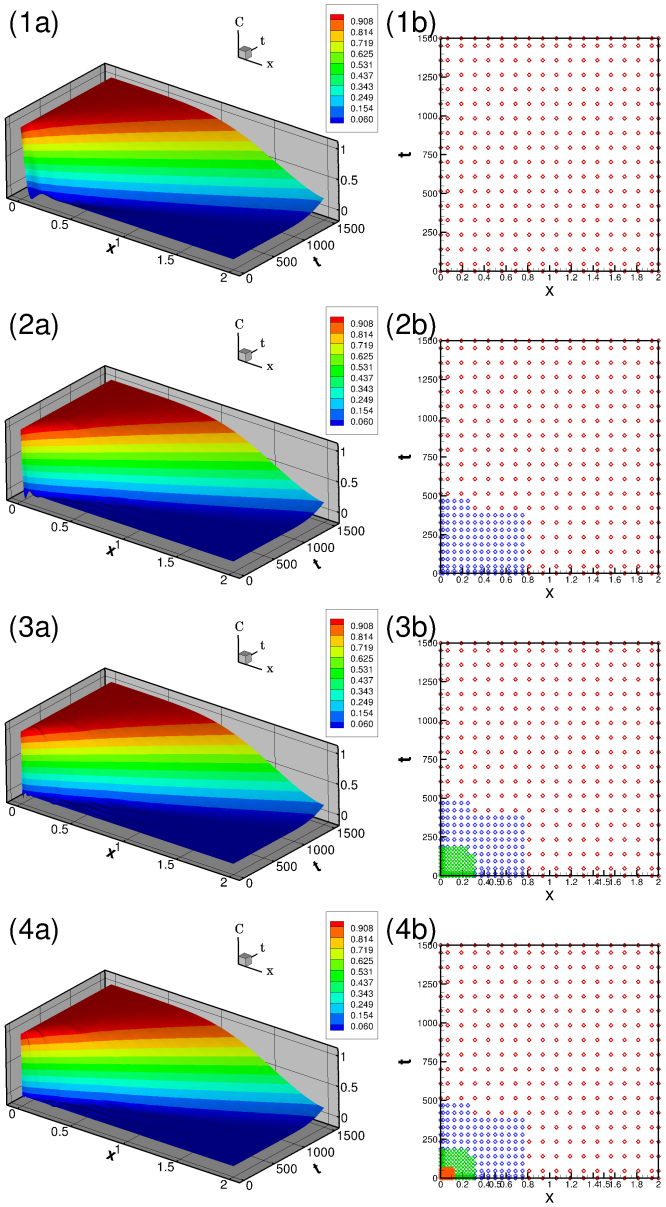


FIGURE 7.27: Numerical solution of the ADE (7.22) at different resolution levels; (1a-4a) HF approximation (with stabilization), (1b-4b) corresponding adaptive time-spatial grids.

The domain, dispersion, velocity and threshold are defined by: $L = 2m$; $D = 10^{-5}$, $v = 10^{-3}$, $\varepsilon = 5 \cdot 10^{-4}$. The initial condition (see Eq. (7.22)) shows that initially the domain was occupied by fresh water. However, the left boundary consists of some denser fluid (for example the salt source) that continuously flows into the domain, and the right boundary states that there is no dispersion flux through that boundary.

Figure 7.26a) shows the numerical solution in the $x - t$ domain obtained with space-time HF basis functions. It represents the change in the solute concentration over the space through time. This change occurs in a narrow transition zone (see Figure 7.26). Figure 7.26b) shows an adaptive grid in the space-time domain. In initial stages of the process, a fine CVs with higher

order of Fup basis functions are needed due to very challenging initial conditions and the creation of a very sharp concentration front. It should be noted that only one time step is used, and the difference between presented adaptive numerical and analytical solutions (see [73]) is strictly less then the prescribed threshold for $t > 200$. However, for lower t, the numerical error is slightly higher then the prescribed threshold, due to discontinuity of initial conditions (see Eq. (7.22) and (7.23)). It should be noted that this initial error does not propagate further over time because HF adaptive method converts the boundary-initial problem to a quasi-boundary problem controlling the global temporal/spatial error.

Figure 7.27 presents the evolution of the numerical solution and corresponding adaptive spatial grid at four consecutive resolution levels with the stabilization method just like in section 7.3. Again the computational cost is reduced since fewer basis functions and levels (Figure 7.26-5b vs 7.27-4b) are needed to achieve the same mass conservation error on all quarters of the CVs.

Figure 7.28 shows convergence analysis using L_2 norm. Uniform analysis is skipped since presented problem has singularity due to discontinuity of initial conditions similar to the L-shape problem (see previous section 7.4), thus only adaptive algorithm without stabilization is tested. It can be observed that adaptive HF basis functions achieves spectral convergence rate. This example is used to show how adaptive HF method handles moving fronts and have the ability to change the grid dynamically, following a front during the simulation while keeping spectral convergence rate.

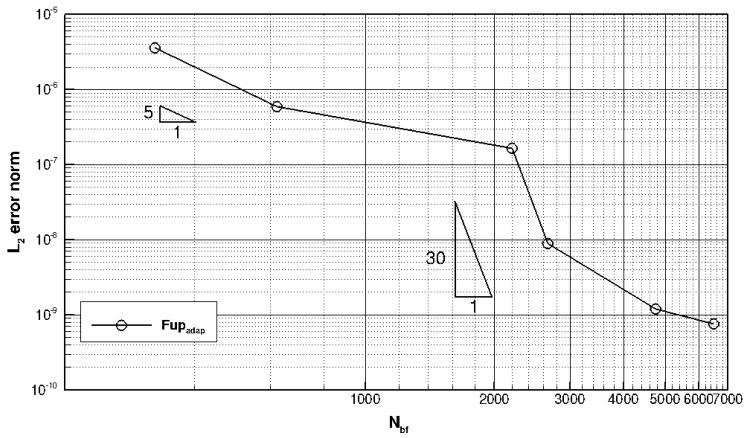


FIGURE 7.28: Convergence analysis for adaptive method (without stabilization)

Chapter 8

Conclusions

8.1 Summary

In this thesis, for the first time hierarchical Fup (HF) basis functions are created and implemented into control volume IGA formulation. Control volume formulation was used in numerical modeling because it contains several interesting properties such as:

- Local and global conservation properties,
- Very close accuracy to the Gallerkin's solution for much lower computational cost,
- Direct physical meaning of the discretized equations,
- Increased accuracy and stability in relation to the collocation method.

Furthermore, conservation is an interesting feature of the control volume formulation. The conservation is exactly satisfied over any control volume (local conservation), as well as over the whole computational domain (global conservation).

Fup functions are used to develop new type of hierarchical basis functions, which can be summarized as follows:

- The usage of Fup basis functions for numerical solutions of engineering problems provides the opportunity for exact CAD geometry description.
- Enables higher continuity throughout the whole domain.
- Refinement can be performed without affecting the computational geometry.
- Fup functions enables local hp-refinement which is superior to refinement procedures done with classical B-spline or NURBS.
- HF achieves spectral convergence rate unlike uniform Fup_n and B_n basis functions which achieves the maximum theoretical convergence order of $p = n + 1$.

Isogeometric analysis (IGA) approach is used as a unified framework for representation of the geometry and solution in multiscale sense. By using a linear combination of smooth hierarchical Fup basis functions all fields are described as continuous and smooth functions. However, since classical IGA uses the Galerkin (G-IGA) or collocation (C-IGA) approach with B-spline or NURBS basis functions, here, the developed model is based on the hierarchical Fup basis functions and the control volume formulation (CV-IGA). Moreover, CV-IGA enables local and global mass conservation (excellent for error testing), as well as approximate solutions of higher smoothness.

These advantages are the main reasons for adopting the control volume formulation as the foundation, and using Fup basis functions as representative member of spline basis functions. Developed adaptive procedure is specially efficient in solving demanding engineering problems with highly localized gradients. First, an analysis of the one-dimensional and two-dimensional problems is performed starting with the approximation of the known function to present basic parameters of the adaptive algorithm. It also serves as introduction problem for easier understanding of the adaptive procedure. Second type of problem is commonly used Poisson example for testing adaptive refinement algorithms because of a steep fronts that occurs in the interior of the domain. For the one dimensional case, diffusion type of problem in order to describe flow in a heterogeneous porous media is used. The adaptive grid has been shown to accurately captures the front and that the adaptive algorithm achieves spectral convergence rate. For two-dimensional case, so called wavefront well problem is considered. Just like for one-dimensional problem, here, the adaptive grid captures the front while reducing conservation error on each resolution level. Demonstration in the terms of the L_2 error norm as a function of DOF is presented. Just like in one-dimensional case, the convergence rate for uniform CV-IGA is the optimal ($p = n + 1$) for odd and suboptimal ($p = n$) for even order of basis functions, unlike for the adaptive procedure which again for the diffusive-like boundary value problems exhibits spectral convergence.

Since many problems in fluid dynamics can be regarded as advection-dominated, third example addresses these type of problems. In many cases, advection-dominated problems feature sharp interfaces and large gradients, which can be strongly localized in spatial direction. For this type of problems, fine spatial resolutions are needed only in specific locations to resolve the demanding small-scale solution features. Moreover, for the remaining part of the domain, coarse resolution is usually sufficient. This adaptive procedure represents one of the most efficient way to address these types of problems. Furthermore, the efficiency of the adaptive algorithm is tested and again shows a significant improvement compared to the uniform case, achieving spectral convergence rate. However, additional stabilization procedures are needed to obtain even better numerical results since some nonphsysical oscillations can occur when convective term dominates the other terms in the governing equation. It is important to point out that the adaptive procedure with stabilization method achieves even better efficiency.

Thus in this work, a novel spatial adaptive algorithm is developed using

new type of hierarchical Fup basis functions. Hierarchical Fup basis functions enables local hp-refinement which means that higher resolution levels have basis functions not only of smaller length of the compact support (higher frequencies; h-refinement) but also contain basis functions of higher order (p-refinement). Thus, the method obtains spectral convergence properties, while the existing hierarchical B-splines (HB) and truncated hierarchical B-splines (THB) achieve convergence determined by the degree of the polynomial of the basis functions $p = n$.

8.2 Scientific contributions

New type of hierarchical Fup basis functions that enable local hp-refinement are developed in this thesis. Local hp-refinement is done by replacing certain basis functions at one resolution level with new basis functions at the next resolution level that have a smaller length of the compact support (h-refinement) but also higher order (p-refinement). This adaptive technique (local hp-refinement) has not been performed in this way so far, which opens the way to new knowledge and techniques in solving real engineering problems characterized by a wide range of spatial scales (due to concentrated force, boundary conditions, pumping or recharging, advective dominant problems in transfer processes, etc.). Hierarchical Fup basis functions do not require additional modifications to preserve the essential property of the partition of unity and allow easy implementation of (local) hp-refinement. Adaptive algorithm is verified on classic 1D and 2D benchmark test cases and compared with classical B-spline hierarchical functions. It should be noted that the presented adaptive procedure achieves a higher accuracy then the prescribed threshold, thus providing the control of the numerical error. This means that the real numerical error of the approximations is strictly less then the prescribed threshold. Furthermore, control volume formulation with adaptive hierarchical Fup$_n$ basis functions yields spectral convergence, in comparison with adaptive THB procedure that achieves higher accuracy then the uniform procedure but keeps the maximum theoretical convergence order of $p = n + 1$.

The first applications of the developed adaptive algorithm are presented on 1-D and 2-D examples of approximation of the known function, Poisson equation which has wide application in structural mechanics and fluid mechanics, elasticity problems and on equations of mass and energy conduction generally shown with diffusive and advection-dispersive equation. In particular, a comparison with analogous h-adaptive procedures based on IGA and hierarchically modified B-splines is shown. A special contribution is the numerical proof that the proposed method (using Fup basis functions) converges in a spectral way.

It is important to emphasize that CV-IGA with adaptive procedure for the diffusive-like boundary value problem exhibits spectral convergence, whereas for uniform layout CV-IGA yields the optimal convergence $p = n + 1$ for odd basis functions, but the suboptimal rate $p = n$ for even basis functions.

In the case of the advection-dominated transport problem, on the first level significant onset of instabilities can occur due to the uniform grid and high corresponding grid Peclet number. However, the adaptive grid accurately captures the front and controls the numerical error and numerical oscillations with higher resolutions levels, thus instabilities are reduced at higher (adaptive) resolution levels. Furthermore, stabilization is also applied to the adaptive algorithm for the advection-dominated problem yielding significant improvement on the first few resolution levels. Moreover, the computational cost is reduced since fewer basis functions are used to achieve the same mass conservation error on all CVs.

The developed new adaptive CV-IGA with the hierarchical Fup basis functions used here for the first time is original and as such represents certain alternative for solving problems compared to classical numerical methods.

8.3 Future perspectives

Finally, some possibilities and directions for future progress can be summarized as follows:

- In this work, hierarchical Fup basis functions that consist of the whole family of algebraic atomic functions are presented. Future work could investigate advantages of using exponential atomic Fup basis functions [48], especially for solving advection-dominated problems where spourious numerical oscillations occur. Namely, these functions are defined using frequency parameter which is closely related to the Peclet number. Introducing numerical and physical consequences in the structure of the basis functions will cause that developed adaptive algorithm could lead to more stable algorithms without oscillations enabling even greater convergence rate.

- The developed 2D adaptive algorithm adds new Fup basis functions at a higher level in both directions (2D Fup basis functions are made as tensor product of the 1D Fup basis functions; see (6.12)). By introducing an improvement so that the algorithm adds basis functions only in the direction where the change of solution is most demanding could significantly reduce a number of degrees of freedom, thus obtain even cheaper numerical solutions and accordingly even better convergence rate.

- Apply adaptive strategy for geometry and material description (i.e., heterogenuos porous material) where proposed algorithm becomes unified framework describing material, geometry and solution in adaptive way resolving all spatial and temporal scales and expand it to three-dimensional problems.

- Finally, an ambitious attempt to extend the proposed adaptive algorithm with Fup basis functions to the resolution of multiple-variable solution space-time scales arising in complex multiphysics problems

would be very interesting. It should include additional adaptive process tied to the time variable. Presented adaptive spatial strategy could serve for initial testing process with the basic hypothesis that the solution would not "move" outside the border of the adaptive grid.

Bibliography

[1] A. Brandt, "Multi-Level Adaptive Solutions to Boundary-Value Problems," *Mathematics of Computation*, vol. 31, no. 138, p. 333, 1977, ISSN: 00255718. DOI: 10.2307/2006422. [Online]. Available: http://www.jstor.org/stable/2006422?origin=crossref.

[2] I Babushka and W. C. Rheinboldt, "Error Estimates for Adaptive Finite Element Computation," *SIAM J. Num. Anal.*, vol. 15, no. 4, pp. 736–754, 1978.

[3] L. Demkowicz, P. Devloo, and J. Oden, "On an h-type mesh-refinement strategy based on minimization of interpolation errors," *Computer Methods in Applied Mechanics and Engineering*, vol. 53, no. 1, pp. 67–89, Oct. 1985, ISSN: 0045-7825. DOI: 10.1016/0045-7825(85)90076-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0045782585900763.

[4] M. J. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics," *Journal of Computational Physics*, vol. 82, no. 1, pp. 64–84, 1989, ISSN: 10902716. DOI: 10.1016/0021-9991(89)90035-1. arXiv: 1607.02212.

[5] L. Demkowicz, J. Oden, W. Rachowicz, and O. Hardy, "Toward a universal h-p adaptive finite element strategy, part 1. Constrained approximation and data structure," *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1-2, pp. 79–112, Dec. 1989, ISSN: 0045-7825. DOI: 10.1016/0045-7825(89)90129-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0045782589901291.

[6] J. Oden, L. Demkowicz, W. Rachowicz, and T. Westermann, "Toward a universal h-p adaptive finite element strategy, part 2. A posteriori error estimation," *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1-2, pp. 113–180, Dec. 1989, ISSN: 0045-7825. DOI: 10.1016/0045-7825(89)90130-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0045782589901308.

[7] W. Rachowicz, J. Oden, and L. Demkowicz, "Toward a universal h-p adaptive finite element strategy part 3. design of h-p meshes," *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1-2, pp. 181–212, Dec. 1989, ISSN: 0045-7825. DOI: 10.1016/0045-7825(89)90131-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/004578258990131X.

[8] A. Harten, *Adaptive Multiresolution Schemes for Shock Computations*, 1994. DOI: 10.1006/jcph.1994.1199. [Online]. Available: http://dx.doi.org/10.1006/jcph.1994.1199.

[9] A. Harten, "Multiresolution algorithms for the numerical solution of hyperbolic conservation laws," *Communications on Pure and Applied Mathematics*, vol. 48, no. 12, pp. 1305–1342, 1995, ISSN: 10970312. DOI: 10.1002/cpa.3160481201.

[10] S. V. Patankar, *Numerical heat transfer and fluid flow*. Hemisphere Pub. Corp., 1980, p. 197, ISBN: 9780891165224.

[11] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *Finite element method : its basis and fundamentals*. Butterworth-Heinemann, 2013, ISBN: 9780080951355.

[12] K.-J. Bathe, *Finite Element Procedures*. Jan. 2006, vol. 2, ISBN: 9780979004902.

[13] T. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, ser. Dover Civil and Mechanical Engineering. Dover Publications, 2000, ISBN: 9780486411811. [Online]. Available: https://books.google.hr/books?id=yarmSc7ULRsC.

[14] T. Hughes, J. Cottrell, and Y. Bazilevs, "Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39, pp. 4135 –4195, 2005, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2004.10.008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782504005171.

[15] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche, "T-spline simplification and local refinement," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 276–283, Aug. 2004, ISSN: 0730-0301. DOI: 10.1145/1015706.1015715. [Online]. Available: http://doi.acm.org/10.1145/1015706.1015715.

[16] C. Giannelli, B. Jüttler, and H. Speleers, "Thb-splines: The truncated basis for hierarchical splines," *Computer Aided Geometric Design*, vol. 29, no. 7, pp. 485 –498, 2012, Geometric Modeling and Processing 2012, ISSN: 0167-8396. DOI: https://doi.org/10.1016/j.cagd.2012.03.025. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167839612000519.

[17] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, "Isogeometric Analysis Toward Intergration of CAD and FEA," p. 335, 2009.

[18] V. L. Rvachev, T. I. Sheiko, V. Shapiro, and I. Tsukanov, "On completeness of rfm solution structures," *Computational Mechanics*, vol. 25, pp. 305 –317, 2000. DOI: 10.1007/s004660050479. [Online]. Available: https://doi.org/10.1007/s004660050479.

[19] K. Höllig, C. Apprich, and A. Streit, "Introduction to the web-method and its applications," *Advances in Computational Mathematics*, vol. 23, pp. 215 –237, 2005. DOI: 10.1007/s10444-004-1811-y. [Online]. Available: https://doi.org/10.1007/s10444-004-1811-y.

[20] V. Kozulić and B. Gotovac, "Application of the solution structure method in numerically solving poisson's equation on the basis of atomic functions," *International Journal of Computational Methods*, vol. 15, no. 05, p. 1 850 033, 2018. DOI: 10.1142/S0219876218500330. eprint: https://doi.org/10.1142/S0219876218500330. [Online]. Available: https://doi.org/10.1142/S0219876218500330.

[21] J. A. Cottrell, T. J. R. Hughes, and A. Reali, "Studies of refinement and continuity in isogeometric structural analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 41-44, pp. 4160–4183, Sep. 2007, ISSN: 00457825. DOI: 10.1016/j.cma.2007.04.007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782507001703.

[22] Y. W. Bekele, T. Kvamsdal, A. M. Kvarving, and S. Nordal, "Adaptive isogeometric finite element analysis of steady-state groundwater flow," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 40, no. 5, pp. 738–765, Apr. 2016, ISSN: 10969853. DOI: 10.1002/nag.2425. arXiv: nag.2347 [10.1002]. [Online]. Available: http://doi.wiley.com/10.1002/nag.2425.

[23] G. Lorenzo, M. Scott, K. Tew, T. Hughes, and H. Gomez, "Hierarchically refined and coarsened splines for moving interface problems, with particular application to phase-field models of prostate tumor growth," *Computer Methods in Applied Mechanics and Engineering*, vol. 319, pp. 515–548, Jun. 2017, ISSN: 00457825. DOI: 10.1016/j.cma.2017.03.009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0045782516318254.

[24] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon, "A hierarchical approach to adaptive local refinement in isogeometric analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 49-52, pp. 3554–3567, Dec. 2011, ISSN: 0045-7825. DOI: 10.1016/J.CMA.2011.09.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782511002933.

[25] D. C. Thomas, M. A. Scott, J. A. Evans, K. Tew, and E. J. Evans, "Bézier projection: A unified approach for local projection and quadrature-free refinement and coarsening of NURBS and T-splines with particular application to isogeometric design and analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 55–105, Feb. 2015, ISSN: 00457825. DOI: 10.1016/j.cma.2014.07.014. arXiv: 1404.7155. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782514002448.

[26] Y. Zhang, W. Wang, and T. J. Hughes, "Solid t-spline construction from boundary representations for genus-zero geometry," *Computer Methods in Applied Mechanics and Engineering*, vol. 249-252, pp. 185 –197, 2012, Higher Order Finite Element and Isogeometric Methods, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2012.01.014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782512000254.

[27] Y. Zhang, W. Wang, and T. J. R. Hughes, "Conformal solid t-spline construction from boundary t-spline representations," *Computational Mechanics*, vol. 51, pp. 1051 –1059, 2013. DOI: https://doi.org/10.1007/s00466-012-0787-6. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010448512002230.

[28] X. Wei, Y. Zhang, L. Liu, and T. J. Hughes, "Truncated t-splines: Fundamentals and methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 316, pp. 349 –372, 2017, Special Issue on Isogeometric Analysis: Progress and Challenges, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.07.020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S004578251630771X.

[29] H. Casquero, L. Liu, Y. Zhang, A. Reali, and H. Gomez, "Isogeometric collocation using analysis-suitable t-splines of arbitrary degree," *Computer Methods in Applied Mechanics and Engineering*, vol. 301, pp. 164 –186, 2016, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2015.12.014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S004578251500420X.

[30] H. Casquero, L. Liu, Y. Zhang, A. Reali, J. Kiendl, and H. Gomez, "Arbitrary-degree t-splines for isogeometric analysis of fully nonlinear kirchhoff–love shells," *Computer-Aided Design*, vol. 82, pp. 140 –153, 2017, Isogeometric Design and Analysis, ISSN: 0010-4485. DOI: https://doi.org/10.1016/j.cad.2016.08.009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010448516300987.

[31] X. Wei, Y. Zhang, T. J. Hughes, and M. A. Scott, "Truncated hierarchical catmull–clark subdivision with local refinement," *Computer Methods in Applied Mechanics and Engineering*, vol. 291, pp. 1 –20, 2015, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2015.03.019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782515001292.

[32] X. Wei, Y. J. Zhang, T. J. Hughes, and M. A. Scott, "Extended truncated hierarchical catmull–clark subdivision," *Computer Methods in Applied Mechanics and Engineering*, vol. 299, pp. 316 –336, 2016, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2015.10.024. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782515003618.

[33] M. Carraturo, C. Giannelli, A. Reali, and R. Vázquez, "Suitably graded thb-spline refinement and coarsening: Towards an adaptive isogeometric analysis of additive manufacturing processes," *Computer Methods in Applied Mechanics and Engineering*, vol. 348, pp. 660 –679, 2019, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2019.01.044. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782519300696.

[34] D. D'Angella, S. Kollmannsberger, E. Rank, and A. Reali, "Multi-level bézier extraction for hierarchical local refinement of isogeometric analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 328, pp. 147 –174, 2018, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2017.08.017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782517305972.

[35] X. Li, X. Wei, and Y. J. Zhang, "Hybrid non-uniform recursive subdivision with improved convergence rates," *Computer Methods in Applied Mechanics and Engineering*, vol. 352, pp. 606 –624, 2019, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2019.04.036. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782519302464.

[36] K. A. Johannessen, F. Remonato, and T. Kvamsdal, "On the similarities and differences between classical hierarchical, truncated hierarchical and lr b-splines," *Computer Methods in Applied Mechanics and Engineering*, vol. 291, pp. 64 –101, 2015, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2015.02.031. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782515001073.

[37] V. L. Rvachev and V. A. Rvachev, "On a finite function," *Dokl. Akad. Nauk Ukrainian SSR, ser. A*, no. 6, pp. 705–707. 1971.

[38] B. Gotovac, "Numerical modelling of engineering problems by smooth finite functions (in croatian)," PhD thesis, 1986.

[39] G. Beylkin and J. M. Keiser, "On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases," *Journal of Computational Physics*, vol. 132, no. 2, pp. 233 –259, 1997, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1996.5562. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199919695562X.

[40] V. Kozulić, "Numerical modelling by the fragment method with rbf functions (in croatian)," PhD thesis, 1999.

[41] B. Gotovac and V. Kozulić, "On a selection of basis functions in numerical analyses of engineering problems," *International Journal for Engineering Modelling*, vol. 12, no. 1-4, pp. 25–41, 1999, ISSN: 13301365.

[42] V. F. Kravchenko, M. A. Basarab, V. I. Pustovoit, and H. Pérez-Meana, "New constructions of weight windows based on atomic functions in problems of speech-signal processing," *Doklady Physics*, vol. 46, pp. 166 –172, 2001. DOI: 10.1134/1.1364722. [Online]. Available: https://doi.org/10.1134/1.1364722.

[43] B. Gotovac and V. Kozulić, "Numerical solving of initial-value problems by rbf basis functions," *Structural Engineering and Mechanics*, vol. 14, no. 3, pp. 263–285, 2002.

[44] V. Kozulić and B. Gotovac, "Numerical analyses of 2d problems using fupn (x, y) basis functions," *International Journal for Engineering Modelling*, vol. 13, no. 1-2, pp. 7–18, 2000.

[45] H. Gotovac, R. Andricevic, B. Gotovac, V. Kozulić, and M. Vranjes, "An improved collocation method for solving the henry problem," *Journal of Contaminant Hydrology*, vol. 64, no. 1, pp. 129 –149, 2003, ISSN: 0169-7722. DOI: https://doi.org/10.1016/S0169-7722(02)00055-4. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169772202000554.

[46] H. Gotovac, V. Cvetkovic, and R. Andricevic, "Adaptive fup multi-resolution approach to flow and advective transport in highly heterogeneous porous media: Methodology, accuracy and convergence," *Advances in Water Resources*, vol. 32, no. 6, pp. 885 –905, 2009, ISSN: 0309-1708. DOI: https://doi.org/10.1016/j.advwatres.2009.02.013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0309170809000414.

[47] H. Gotovac, V. Cvetkovic, and R. Andricevic, "Flow and travel time statistics in highly heterogeneous porous media," *Water Resources Research*, vol. 45, no. 7, 2009. DOI: https://doi.org/10.1029/2008WR007168. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2008WR007168. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008WR007168.

[48] N. Brajcic Kurbasa, "Eksponencijalne atomske bazne funkcije: Razvoj i primjena: Disertacija," PhD thesis, 2016.

[49] L. Malenica, "Numerical Modeling Based on Spline Basis Functions: Application to Groundwater Flow Modeling in Karst Aquifers and Advection Dominated Problems," PhD thesis, 2019.

[50] G. Kamber, H. Gotovac, V. Kozulić, L. Malenica, and B. Gotovac, "Adaptive numerical modeling using the hierarchical fup basis functions and control volume isogeometric analysis," *International Journal for Numerical Methods in Fluids*, vol. 92, no. 10, pp. 1437–1461, 2020. DOI: https://doi.org/10.1002/fld.4830. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.4830. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4830.

[51] H. Gotovac, R. Andricevic, and B. Gotovac, "Multi-resolution adaptive modeling of groundwater flow and transport problems," *Advances in Water Resources*, vol. 30, no. 5, pp. 1105–1126, 2007, ISSN: 03091708. DOI: 10.1016/j.advwatres.2006.10.007.

[52] L. Malenica, H. Gotovac, G. Kamber, S. Simunovic, S. Allu, and V. Divic, "Groundwater flow modeling in karst aquifers: Coupling 3d matrix and 1d conduit flow via control volume isogeometric analysis experimental verification with a 3d physical model," *Water*, vol. 10, no. 12, 2018, ISSN: 2073-4441. DOI: 10.3390/w10121787. [Online]. Available: http://www.mdpi.com/2073-4441/10/12/1787.

[53] B. E. Rapp, "Chapter 31 - finite volume method," in *Microfluidics: Modelling, Mechanics and Mathematics*, ser. Micro and Nano Technologies, Oxford: Elsevier, 2017, pp. 633 –654, ISBN: 978-1-4557-3141-1. DOI: https://doi.org/10.1016/B978-1-4557-3141-1.50031-9. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781455731411500319.

[54] F. A. Tavarez and M. E. Plesha, "Discrete element method for modelling solid and particulate materials," *International Journal for Numerical Methods in Engineering*, vol. 70, no. 4, pp. 379–404, 2007. DOI: 10.1002/nme.1881. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1881. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1881.

[55] C. J. Leo and J. R. Booker, "A boundary element method for analysis of contaminant transport in porous media i: Homogeneous porous media," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 23, no. 14, pp. 1681–1699, 1999. DOI: 10.1002/(SICI)1096-9853(19991210)23:14<1681::AID-NAG971>3.0.CO;2-U. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291096-9853%2819991210%2923%3A14%3C1681%3A%3AAID-NAG971%3E3.0.CO%3B2-U. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291096-9853%2819991210%2923%3A14%3C1681%3A%3AAID-NAG971%3E3.0.CO%3B2-U.

[56] W. Rudin, *Functional Analysis*, ser. International series in pure and applied mathematics. McGraw-Hill, 1991, ISBN: 9780070542365. [Online]. Available: https://books.google.hr/books?id=Sh_vAAAAMAAJ.

[57] T. W. Sederberg, G. T. Finnigan, X. Li, H. Lin, and H. Ipson, "Watertight trimmed nurbs," in *ACM SIGGRAPH 2008 Papers*, ser. SIGGRAPH '08, Los Angeles, California: Association for Computing Machinery, 2008, ISBN: 9781450301121. DOI: 10.1145/1399504.1360678. [Online]. Available: https://doi.org/10.1145/1399504.1360678.

[58] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche, "T-spline simplification and local refinement," *ACM Trans. Graph.*, vol. 23, no. 3, 276–283, Aug. 2004, ISSN: 0730-0301. DOI: 10.1145/1015706.1015715. [Online]. Available: https://doi.org/10.1145/1015706.1015715.

[59] B. R. Baliga and S. V. Patankar, "A new finite-element formulation for convection-diffusion problems," *Numerical Heat Transfer*, vol. 3, no. 4, pp. 393–409, 1980. DOI: 10.1080/01495728008961767. eprint: https://doi.org/10.1080/01495728008961767. [Online]. Available: https://doi.org/10.1080/01495728008961767.

[60] B. R. Baliga and S. V. Patankar, "A control volume finite-element method for two-dimensional fluid flow and heat transfer," *Numerical Heat Transfer*, vol. 6, no. 3, pp. 245–261, 1983. DOI: 10.1080/01495728308963086. eprint: https://doi.org/10.1080/

01495728308963086. [Online]. Available: https://doi.org/10.1080/01495728308963086.

[61] V. R. Voller, *Basic Control Volume Finite Element Methods for Fluids and Solids*. World Scientific Publishing Company, 2009, pp. i–xiv. DOI: 10.1142/9789812834997_fmatter. eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789812834997_fmatter. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/9789812834997_fmatter.

[62] D. Schillinger, J. A. Evans, A. Reali, M. A. Scott, and T. J. Hughes, "Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations," *Computer Methods in Applied Mechanics and Engineering*, vol. 267, no. February, pp. 170–232, 2013, ISSN: 00457825. DOI: 10.1016/j.cma.2013.07.017.

[63] R. W. Johnson, "Higher order b-spline collocation at the greville abscissae," *Applied Numerical Mathematics*, vol. 52, no. 1, pp. 63 –75, 2005, ISSN: 0168-9274. DOI: https://doi.org/10.1016/j.apnum.2004.04.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016892740400073X.

[64] C. Giannelli, B. Jüttler, and H. Speleers, "Strongly stable bases for adaptively refined multilevel spline spaces," *Adv. Comput. Math.*, vol. 40, no. 2, 459–490, Apr. 2014, ISSN: 1019-7168. DOI: 10.1007/s10444-013-9315-2. [Online]. Available: https://doi.org/10.1007/s10444-013-9315-2.

[65] R. Kraft, *Adaptive and linearly independent multilevel B-splines*. SFB 404, Geschäftsstelle, 1997.

[66] C. Swaminathan and V. Voller, "Streamline upwind scheme for control-volume finite elements, part i. formulations," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 22, no. 1, pp. 95–107, 1992. DOI: 10.1080/10407799208944972. eprint: https://www.tandfonline.com/doi/pdf/10.1080/10407799208944972. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/10407799208944972.

[67] D. Hendriana, "On finite element and control volume upwinding methods for high peclet number flows," Master's thesis, Massachusetts Institute of Technology. Dept. of Mechanical Engineering, 1994.

[68] J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics*. Springer, 2002, p. 423, ISBN: 978-3-540-78091-5. DOI: 10.1007/978-3-642-98037-4. arXiv: arXiv:1011.1669v3. [Online]. Available: http://link.springer.com/10.1007/978-3-642-98037-4.

[69] C.-C. Yu and J. C. Heinrich, "Petrov-galerkin method for multidimensional, time-dependent, convective-diffusion equations," *International Journal for Numerical Methods in Engineering*, vol. 24, no. 11, pp. 2201–2215, 1987. DOI: https://doi.org/10.1002/nme.1620241112. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1620241112. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620241112.

[70] A. N. Brooks and T. J. Hughes, "Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 32, no. 1, pp. 199 –259, 1982, ISSN: 0045-7825. DOI: https://doi.org/10.1016/0045-7825(82)90071-8. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0045782582900718.

[71] J. Oden and A. Patra, "A parallel adaptive strategy for hp finite element computations," *Computer Methods in Applied Mechanics and Engineering*, vol. 121, no. 1, pp. 449 –470, 1995, ISSN: 0045-7825. DOI: https://doi.org/10.1016/0045-7825(94)00705-R. [Online]. Available: http://www.sciencedirect.com/science/article/pii/004578259400705R.

[72] W. F. Mitchell, "A collection of 2d elliptic problems for testing adaptive grid refinement algorithms," *Applied Mathematics and Computation*, vol. 220, pp. 350 –364, 2013, ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2013.05.068. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0096300313006449.

[73] A. Ogata and R. Banks, "A solution of the differential equation of longitudinal dispersion in porous media," *U.S. Geological Survey, Washington, D.C.*, no. 411-A, 1961.