Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 398 (2022) 115272
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

2-D local hp adaptive isogeometric analysis based on hierarchical Fup basis functions

G. Kamber*, H. Gotovac, V. Kozulić, B. Gotovac

Faculty of Civil Engineering, Architecture and Geodesy, University of Split, Matice hrvatske 15, 21000 Split, Croatia

Received 29 October 2021; received in revised form 4 May 2022; accepted 18 June 2022
Available online xxxx

Abstract

In this paper, 2-D hp local adaptive procedure is developed based on Control Volume Isogeometric Analysis (CV-IGA) and Hierarchical Fup (HF) basis functions. Contrary to the most common truncated hierarchical splines, HF enables local hp adaptation because higher resolution levels do not include only basis with smaller compact support or higher frequencies, but also with higher order. Consequence of this property is spectral convergence of the proposed adaptive algorithm which is presented on classical benchmarks such as L-shape benchmark and advection dominated problems. Even in non-smooth problems, spectral convergence is achieved contrary to the application of uniform grid. CV-IGA ensures local and global mass conservation which is potentially very important for fluid mechanics problems. 2-D proposed algorithm chooses regular control volumes in parametric space at all resolution levels closely related to the Greville points (vertices) of basis functions. Therefore, methodology is very simple requiring only overlapping of control volumes in the areas where different levels are connected, while its computational cost lies between Galerkin and collocation formulations.

Keywords: Hierarchical Fup basis functions; hp-refinement; Local refinement; Control volume; Isogeometric analysis; Adaptive methods

1. Introduction

Many industrial and real applicative problems in computational mechanics have been solved by numerical simulations that require large computational resources including parallel computing and the use of CPU/GPU clusters and/or supercomputers. Therefore, it is of great importance that computer resources are used as efficiently as possible.

Numerical modeling of different physical and engineering problems characterized with large range of spatial and temporal scales are typically faced by many difficulties. 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 classical methods are finite element method (FEM), finite difference method (FDM) and finite volume method (FVM) [1–5]. There are various other methods such as the spectral element method (SEM), boundary element method (BEM) [6], discrete element method (DEM) [7] which, together with various collocation, meshfree and other hybrid approaches, are usually practical for limited classes of problems.

* Corresponding author.
E-mail address: grgo.kamber@gradst.hr (G. Kamber).

https://doi.org/10.1016/j.cma.2022.115272
0045-7825/

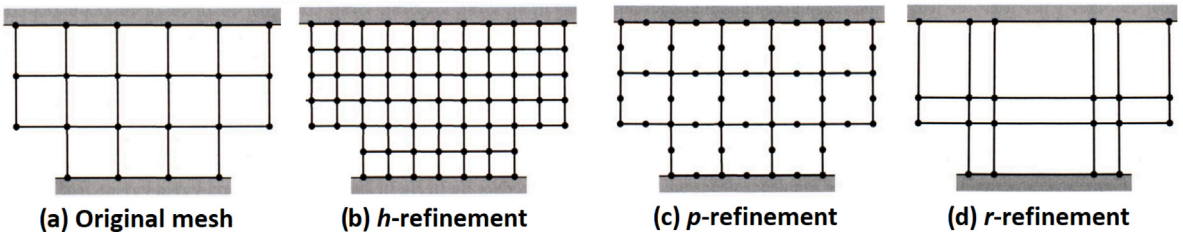


Fig. 1. Refinement procedures.

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, and mostly present due to differences in the used basis functions. Whereas classical polynomials have dominated in the field of numerical analysis, spline-based basis functions (e.g., B-splines, non-uniform rational B-splines (NURBS) [8], T-splines [9], hierarchical B-splines (HB) [10] 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. [8] and Cottrell et al. [11]). The main idea of isogeometric analysis (IGA) is to bridge the gap between FEA and a CAD by using the same type of spline basis functions for both systems. Therefore, 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, no cumbersome meshing, usage of higher-order basis functions, higher continuity of solution and geometry, more efficient refinement adaptive procedures and multiresolution approach (see e.g. [12]). 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 which is immersed to the background mesh defined on regular rectangle in 2-D or cube in 3-D (see for instance immersogeometric methods in Hsu et al. [13], Rvachev structure method by Rvachev et al. [14] or WEB-splines by Höllig et al. [15]).

The development of adaptive methods [16–20] for local refinement and coarsening became one of the most important 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 [9,21,22], hierarchical B-splines (HB) [10], truncated hierarchical B-splines (THB) [10,23,24] and locally refined B-splines (LR) [25]. Furthermore, linear independence, stability and partition of unity as well as local refinement and adaptation became center topics for these adaptive solutions.

Adaptive isogeometric methods attract a lot of attention and are a very active field of research which can generally be divided to h-refinement (Fig. 1b; spline functions of the same order but smaller knot intervals, i.e. higher frequencies), p-refinement (Fig. 1c; higher degree of basis functions), r-refinement (Fig. 1d; redesigning the mesh without changing the number of nodes and only adjusting their positions) and their combinations. 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 grid which preserve the non-negativity of standard B-splines and enables the possibility to properly deal with local problems [10]. However, since the hierarchical B-spline basis functions in non-rational form do not satisfy partition of unity, it may produce ill-conditioned control meshes at the refined level [26]. To overcome this deficiency, the truncated mechanism was first developed by Giannelli et al. [10] for the hierarchical B-spline basis functions (THB) 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 basis functions have been used in recent times (see Rvachev and Rvachev [27] and Gotovac [28]. 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 [29]). Gotovac [28] systematizes the existing knowledge about atomic basis functions and transforms them into a numerically appropriate form, especially Fup basis functions as a typical member of atomic class of basis functions. Kozulić [30] and Gotovac and Kozulić [31] showed the basic possibilities of using Fup basis functions in

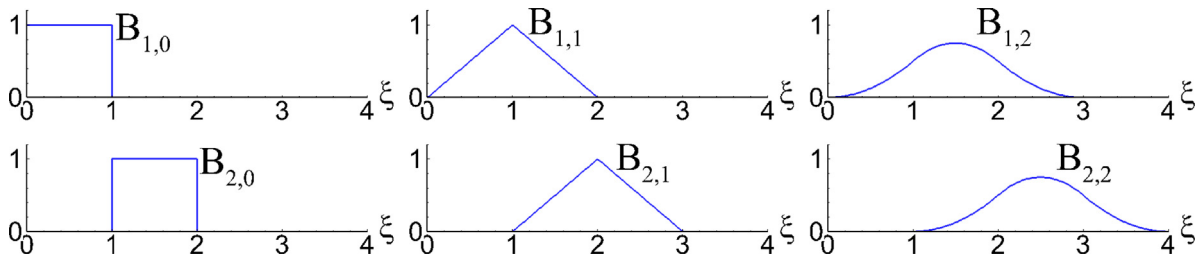


Fig. 2. Basis functions of order 0, 1, and 2 for uniform knot vector $\Xi = \{0, 1, 2, \ldots\}$.

structural mechanics and numerical analysis. The use of Fup basis functions has been shown to solve the problem of signal processing (see Kravchenko et al. [32]), the initial problem (see Gotovac and Kozulić [33]) and the boundary problems using the non-adaptive Fup collocation method (see Kozulić and Gotovac [34] and Gotovac et al. [35]).

Gotovac et al. [36] 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 and inability to describe the general irregular geometry. Malenica et al. [37] firstly developed Control Volume Isogometric Analysis (CV-IGA) applied to the karst groundwater flow model, while Gotovac et al. [12] presented CV-IGA in the context of other Galerkin and collocation formulations. Kamber et al. [38] set foundation for efficient adaptive spatial procedure developing 1-D hierarchical Fup (HF) basis functions inside CV-IGA. HF have the option of local hp-refinement such that they can replace certain Fup 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). Nested sequence of domains with hierarchical Fup basis functions ensures linear independence, stability, partition of unity and the possibility to properly deal with local problems.

In this work, we present a novel adaptive algorithm that is based on hierarchical 2-D Fup basis functions and CV-IGA, which are closely related to the HB and THB. HF provides spectral convergence and presents a substantial improvement in comparison to THB that enable only polynomial convergence.

2. Spline basis functions

2.1. Hierarchical B-spline basis functions

The B-spline basis functions are piecewise polynomial functions defined in parametric space. B-spline basis functions are defined recursively (see Cottrell et al. [11]) starting with piecewise constants ($n = 0$):

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

Fig. 2 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.

Although there is a general definition, such as Cox–de Boor recursion formula (see [11,23]), we will in this paper due to the presentation of hierarchical Fup splines and linking them with B-splines go only through the convolution theorem. $B_n(\xi)$ can be presented by using convolution theorem in the following form:

$$B_n(\xi) = \int_{-\infty}^{\infty} B_{n-1}(\xi - t)\, B_0(t)\ dt \tag{2}$$

or:

$$B_n(\xi) = B_{n-1}(\xi) * B_0(\xi) = \underbrace{B_0(\xi) * \ldots * B_0(\xi)}_{(n+1)\ \ times} \tag{3}$$

where n is the order of the B-spline. 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):

$$f_n(t) = \left(\frac{\sin(t/2)}{t/2}\right)^{n+1} \tag{4}$$

so the inverse FT of $B_n(\xi)$ is defined by:

$$B_n(\xi) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\left(\frac{\sin(t/2)}{t/2}\right)^{n+1}\cdot e^{-it\xi}dt. \tag{5}$$

Eq. (3) implies that the support of $B_n(\xi)$ is the union of the $(n+1)$ characteristic intervals $\Delta\xi$. 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.

In one-dimensional problems, a knot vector is a set of non-decreasing real numbers representing coordinates in the parametric space of the curve

$$\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\} \tag{6}$$

where ξ_i is the ith knot, i is the knot index, $i = 1, 2, \ldots, n+p+1$, n is the polynomial order of the B-spline, and p 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, otherwise they are non-uniform. 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. $B_n(\xi)$ is presented by the local polynomial of the nth order on each interval $[\xi_k, \xi_{k+1}]$.

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 nth derivative.
3. A linear combination of shifted B_n-splines by a characteristic interval describes algebraic polynomials up to the nth order.
4. A linear combination of m shifted B-splines by a characteristic interval describes a unit constant function ("partition of unity"), that is

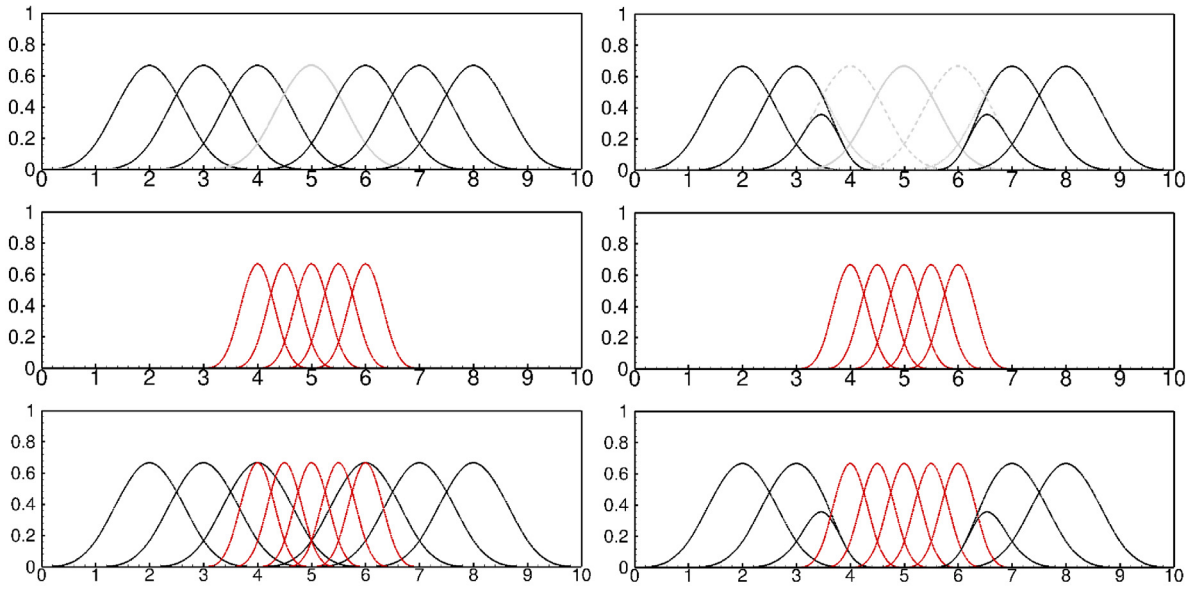
$$\sum_{i=1}^{m} B_{i,n}(\xi) = 1 \tag{7}$$

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., [16–19,21,23,24,26]).

B-spline basis functions are refinable, which enables the construction of HB and its truncated variant THB. Truncated hierarchical B-splines (THB) were introduced and analyzed in [10,39]. THB-splines can be considered as an upgrade for hierarchical B-splines (HB) i.e., an alternative base for the space of hierarchical splines, that retains 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 the domain Ω^{l+1} are truncated (see Fig. 3).

THB refinability (see [10,26]) 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,

$$B_{i,n}^l(\xi) = \sum_{k=0}^{n+1} c_{i,k}^n B_{2i+k,n}^{l+1}(\xi) \quad \text{with} \quad c_{i,k}^n = \frac{1}{2^n}\binom{n+1}{k}, i = 0, \ldots, m^l - 1 \tag{8}$$



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

Fig. 3. Comparison of univariate cubic HB- and THB-splines. (a) Three steps to construct univariate cubic HB-spline basis function without truncation and (b) three steps to construct univariate cubic HB-spline basis function with truncation (THB).

where $c^n_{i,k}$ are the refinement coefficients 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 $B^{l+1}_{2i+k,n}$ on the next level are called the children of $B^l_{i,n}(\xi)$ i.e., denoted as,

$$chd\, B^l_{i,n}(\xi) = \left\{ B^{l+1}_{2i+k,n}(\xi) | k = 0, 1, \ldots, n+1 \right\}. \tag{9}$$

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 Fig. 3). 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 [40], 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. Fig. 3 shows a construction process for univariate cubic THB 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 $active$ ($\mathcal{B}^l_a = \mathcal{B}^l \setminus \mathcal{B}^l_p$).
- Obtain the children at level $l+1$ (red solid curves) only for the $passive$ $\mathcal{B}^l_p$ and define them as $active$; $\mathcal{B}^{l+1}_a = chd\mathcal{B}^l_p$.
- 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,

$$\mathcal{B}^{l+1}_{hbf} = \mathcal{B}^{l+1} = \mathcal{B}^l_a \cup \mathcal{B}^{l+1}_a. \tag{10}$$

Eq. (10) 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 [10,26]. Fig. 3 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 Eq. (8) for its refinability gives,

$$B_i^l(\xi) = \sum_{supp B_j^{l+1} \subseteq supp B_i^l} c_{i,j} B_j^{l+1}(\xi), \tag{11}$$

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

$$trun B_i^l(\xi) = \sum_{supp B_j^{l+1} \not\subseteq \Omega^{l+1}} c_{i,j} B_j^{l+1}(\xi) \tag{12}$$

with respect to $\mathcal{B}_p^l$ [26].

Eq. (12) 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 $trun B_i^l$. In Fig. 3, 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 side; 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 [39]. It gives a sparser connectivity among basis functions at different levels, and it can preserve geometry when local refinement is performed [26].

2.2. Hierarchical Fup basis functions

Fup basis functions belong to the class of atomic functions (see [27],[31]) and span vector space of algebraic polynomials, while their properties are closely related to the B-splines, as will be explained in the sequel.

Function $up\,(\xi)$ can be obtained by an infinite number of convolutions of the contracted $B_0\,(\xi)$ with compact support 2^{-k} and vertex value 2^k, $k \in \mathbb{N}$, according to following convolution procedure:

$$up\,(\xi) = B_0\,(\xi) * B_0\,(2\xi) * \cdots * B_0\left(2^k\xi\right) * \cdots * B_0\left(2^\infty\xi\right) \tag{13}$$

From (13), the compact support of $up\,(\xi)$ is the union of an infinite number of finite intervals. However, its compact support is finite:

$$h_{up} = \sum_{k=0}^{\infty} \frac{1}{2^k} = 2 \quad \rightarrow \quad \text{supp } up\,(\xi) = [-1, 1] \tag{14}$$

The convolution procedure (13) causes $up(\xi)$ to contain all polynomial orders by parts of its compact support. Due to its infinite number of continuous and non-zero derivatives, function $up(\xi)$ can be regarded as a perfect spline.

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:

$$\xi_{br} = -1 + k \cdot 2^{-m}, \quad m \in N, \quad k = 1, \ \ldots, 2^{m+1}. \tag{15}$$

At all other points of the compact support calculation of $up\,(\xi)$ can be done only approximately, but up to the computer accuracy.

For the calculation of $up(\xi)$ values at arbitrary points, Gotovac and Kozulić [31] 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 nth order). Values of the even function $up(\xi)$ in arbitrary point $\xi \in [0, 1]$ can be presented as follows:

$$up\,(\xi) = 1 - up(\xi - 1) = 1 - \sum_{k=1}^{\infty}(-1)^{1+p_1+\cdots+p_k}\, p_k \sum_{j=0}^{k} C_{jk} \cdot \Delta_k{}^j \tag{16}$$

where the coefficients C_{jk} are rational numbers containing values of $up(\xi)$ at the binary-rational points $\xi_k = -1 + 1/2^m$ [31]:

$$C_{jk} = \frac{1}{j!} 2^{j(j+1)/2} up(-1 + 2^{-(k-j)})\ ; \qquad j = 0, 1, \ldots,\ k; \quad k = 1, 2, \ldots, \infty \tag{17}$$

Factor Δ_k in (16) 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:

$$\Delta_k = \xi - \sum_{i=1}^{k} p_i \cdot \frac{1}{2^i} \tag{18}$$

For an exact description of polynomials up to the nth 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.

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

$$supp \quad Fup_n(\xi) = \left[-(n+2)\cdot 2^{-n-1},\ (n+2)\cdot 2^{-n-1}\right] \tag{19}$$

For $n = 0$, the following holds:

$$Fup_0(\xi) = up(\xi) \tag{20}$$

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

$$Fup_n(\xi) = B_n(2^n\xi) * up(2^{n+1}\xi) \tag{21}$$

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 nth order of basis functions. Eq. (21) is not numerically favorable for calculating the value of the function $Fup_n(\xi)$.

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

$$Fup_n'\,(\xi) = 2\sum_{k=0}^{n+2}\left(C_n^k - C_n^{k-2}\right)\cdot Fup_n\left(2\xi - \frac{k}{2^n} + \frac{n+2}{2^{n+1}}\right) \tag{22}$$

where C_n^k are binomial coefficients defined as

$$C_n^k = \binom{n}{k} = \frac{(n)!}{(n-k)!\cdot k!} \tag{23}$$

Eq. (22) presents the atomic structure of these basis functions because its derivatives (but also a function values as will be shown in the sequel) are decomposed by a linear combination of these same functions (Rvachev and

Rvachev [27]). $Fup_n(\xi)$ can be calculated by a linear combination of $up(\xi)$ mutually shifted by the characteristic interval 2^{-n}:

$$Fup_n(\xi) = \sum_{k=0}^{\infty} C_k(n) \cdot up\left(\xi - 1 - \frac{k}{2^n} + \frac{n+2}{2^{n+1}}\right) \tag{24}$$

The zero coefficient in (24) is:

$$C_0(\mathrm{n}) = 2^{C_{n+1}^2} = 2^{n(n+1)/2} \tag{25}$$

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:

$$\begin{aligned} C'_0(n) &= 1 \\ C'_k(n) &= (-1)^k C_{n+1}^k - \sum_{j=1}^{min\left\{k\ ;\ 2^{n+1}-1\right\}} C'_{k-j}(n) \cdot \delta_{j+1} \end{aligned} \tag{26}$$

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 $\varphi_{n,j}$, $j = -[(n+1)/2], \ldots, [n/2]$ on the left domain boundary ξ_A, and $j = N - [n/2], \ldots, N + [(n+1)/2]$ on the right domain boundary ξ_B (N is the number of characteristic intervals $\Delta\xi_n$ inside the domain).

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

$$\begin{aligned} &\varphi_{n,j}^{(i)}(\xi_A) \neq 0 \quad for \quad j + [(n+1)/2] \leqslant i \leqslant n \\ &\varphi_{n,j}^{(i)}(\xi_A) = 0 \quad otherwise; \qquad\qquad i \in \mathbb{N} \end{aligned} \tag{27}$$

Modification of the right boundary basis functions is 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 each boundary.

We can summarize the properties of the Fup basis functions as follows:

1. Fup_n is positive on $n+2$ characteristic intervals and vanishes outside these intervals i.e., Fup basis functions have compact support where they have strictly positive non-zero values; elsewhere, they are zero, implying localized approximation properties.
2. Fup_n is infinitely differentiable.
3. A linear combination of m shifted Fup basis functions by a characteristic interval describes a unit constant function ("partition of unity"), that is

$$\frac{1}{2^n} \sum_{i=1}^{m} Fup_{i,n}(\xi) = 1 \tag{28}$$

4. Fup_n can be presented by a linear combination of the shifted Fup basis functions with the higher order, but using two-times-smaller supports. This implies that Fup basis functions enable multiresolution analysis and efficient adaptive numerical procedures (e.g., [38]).

Basis function Fup_n^l defined on Ξ^l can be represented as a linear combination of $n+2$ Fup_{n+1}^{l+1} basis functions defined on Ξ^{l+1},

$$Fup_n^l(\xi) = \sum_{k=0}^{n+1} C_{n+1}^k \cdot Fup_{n+1}^{l+1}\left(\xi - \frac{k}{2^{n+1}} + \frac{n+1}{2^{n+2}}\right), \tag{29}$$

where C_{n+1}^k are the refinement coefficients

$$C_{n+1}^k = \frac{1}{2^{n+1}} \binom{n+1}{k} \tag{30}$$

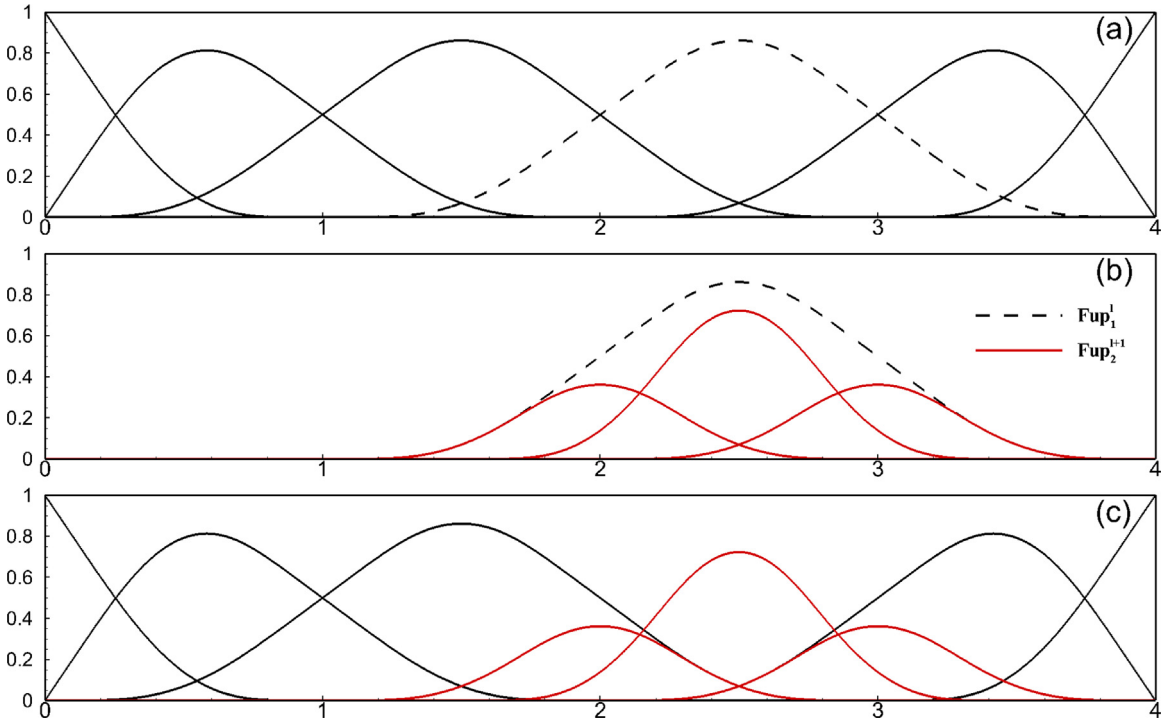


Fig. 4. The three steps to construct hierarchical Fup basis functions. (a) In level l, basis functions $\mathcal{F}_p^l$ that need to be refined are determined (black dashed curve $|Fup_1^l$) and they are defined as $passive$, while remaining basis functions are defined as $active$; (b) In level $l+1$, 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 $\mathcal{F}_{hbf}^{l+1}$.

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

$$chdFup_n^l(\xi) = \left\{ Fup_{n+1}^{l+1}\left(\xi - \frac{k}{2^{n+1}} + \frac{n+1}{2^{n+2}}\right) \middle| k = 0, 1, \ldots, n+1 \right\} \tag{31}$$

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).

Let $\Omega^0 \subseteq \Omega^1 \subseteq \cdots \subseteq \Omega^l$ be a sequence of nested domains. Each Ω^l represents the region selected to be refined at the level l. It should be noted that in our examples we are only considering dyadic refinement. 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. (29)) [31]. In the following, we will show only two consecutive levels and construct level $l+1$ from the level l.

Fig. 4 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 $active$; $\mathcal{F}_a^{l+1} = chd\mathcal{F}_p^l$.

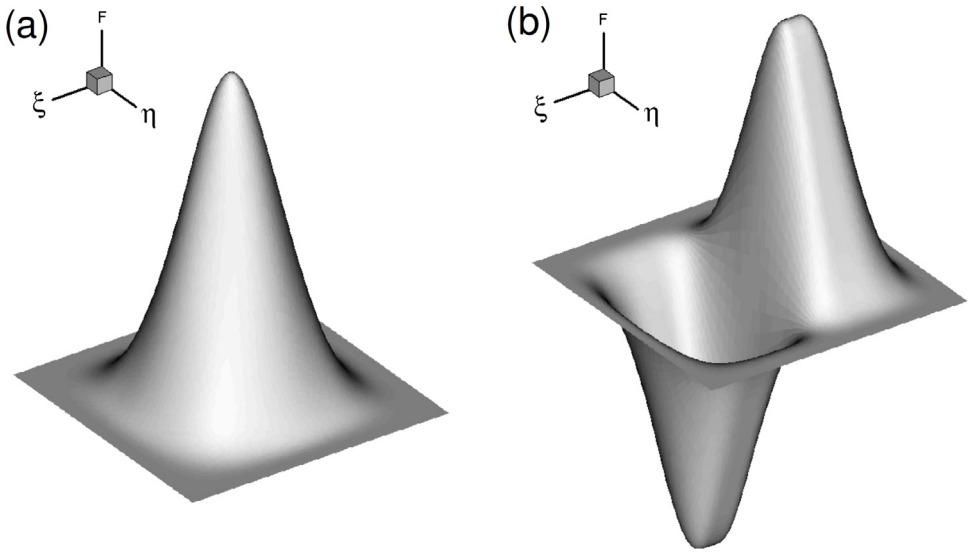


Fig. 5. 2D Fup basis functions. F=Fup$_1(\xi,\eta)$; (a)F and (b)$\frac{\partial F}{\partial \xi}$.

- Merge all of the basis functions that are $active$ from levels l and $l+1$ to obtain the hierarchical Fup basis functions,

$$\mathcal{F}_{hbf}^{l+1} = \mathcal{F}^{l+1} = \mathcal{F}_a^l \cup \mathcal{F}_a^{l+1}. \tag{32}$$

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} (see Eq. (28)). 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. (29)), it entails that all of the Fup basis functions that are created at higher resolution levels also satisfy partition of unity.

3. Adaptive methodology

The 2-D adaptive spatial strategy used in this work is a novel approach based on the Control Volume IsoGeometric Analysis, shortly CV-IGA (Malenica et al. [37,41], Gotovac et al. [12]) and hierarchical Fup basis functions (see Kamber et al. [38]). Firstly, 2-D Fup basis functions are presented, then local hp-refinement within CV-IGA concept is explained. In Section 3.3.1, hp-adaptive scheme for approximating known function is presented. It is used for easier understanding of whole adaptive process and serves as introduction for boundary value problems (BVPs) explained in Section 3.3.2.

3.1. 2-D Basis functions

Multi-dimensional 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,

$$Fup_n(\xi,\eta) = Fup_n(\xi) \cdot Fup_n(\eta) \tag{33}$$

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. Fig. 5 shows two-dimensional Fup$_1(\xi,\eta)$ basis function and its first partial derivative.

For 1-D 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. (29)). Fup$_n^l(\xi,\eta)$ defined on the level l can be represented as a linear combination of

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

$$Fup_n^l(\xi,\eta) = \sum_{i=0}^{n+1}\sum_{j=0}^{n+1} C_{n+1}^i C_{n+1}^j Fup_{n+1}^{l+1}\left(\xi - \frac{i}{2^{n+1}} + \frac{n+1}{2^{n+2}}\right) Fup_{n+1}^{l+1}\left(\eta - \frac{j}{2^{n+1}} + \frac{n+1}{2^{n+2}}\right) \tag{34}$$

where C_{n+1}^i and C_{n+1}^j are refinement coefficients (see Eq. (30)).

For example Fup_1^l is 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 its nine children Fup_2^{l+1} (see Eq. (31)) are 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 $H^{l+1} = \{0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}, 1\}$.

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 $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{m^\xi}\}$ and $H = \{\eta_1, \eta_2, \ldots, \eta_{m^\eta}\}$, 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

$$\Delta\xi^l = \frac{\xi_{m^\xi+n+2} - \xi_1}{(m^{l,\xi} - n - 1)2^l}; \quad \Delta\eta^l = \frac{\eta_{m^\eta+n+2} - \eta_1}{(m^{l,\eta} - n - 1)2^l}, \tag{35}$$

where ξ_1 and $\xi_{m^\xi+n+2}$ are the first and last members of the knot vector in ξ- direction and η_1 and $\eta_{m^\eta+n+2}$ are the first and last members 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 derivations (see Eq. (27)). In the vector of mutually displaced Fup_n basis functions in 2-D, it is necessary to modify the $(n+2)$ basis functions in ξ and/or η direction if they are near boundary of the domain.

3.2. Control volume isogeometric analysis

In IGA one 2-D subdomain or patch is transformed from the parameter (virtual) space to the physical (real) space using following spline representation

$$x(\xi,\eta) = \sum_{j=1} x_j \phi_j(\xi,\eta); \quad y(\xi,\eta) = \sum_{j=1} y_j \phi_j(\xi,\eta) \tag{36}$$

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 (36) are spline basis functions ϕ_j which in classic IGA are B-splines and NURBS. It is clear from (36) that IGA operates only with basis functions in the parametric regular domain since transformations from the parametric to real physical space, and vice versa are defined by the Jacobian and its inverse 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 [11].

The numerical solution in the parametric space is also described by independent set of spline basis functions

$$u(\xi,\eta) = \sum_{j=1} \alpha_j \varphi_j(\xi,\eta) \tag{37}$$

It should be noted that number and order of the basis functions in the (36) and (37) may not be the same.

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

$$\nabla\cdot(D\nabla u(\mathbf{x})) - \nabla\cdot(v u(\mathbf{x})) = 0 \quad in \quad \Omega \tag{38}$$

with appropriate boundary conditions:

$$u(\mathbf{x}) = u_D \quad on \quad \Gamma_D \tag{39}$$

$$(D\nabla u(\mathbf{x}) - v u(\mathbf{x}))\cdot\mathbf{n} = q_N(\mathbf{x}) \quad on \quad \Gamma_N \tag{40}$$

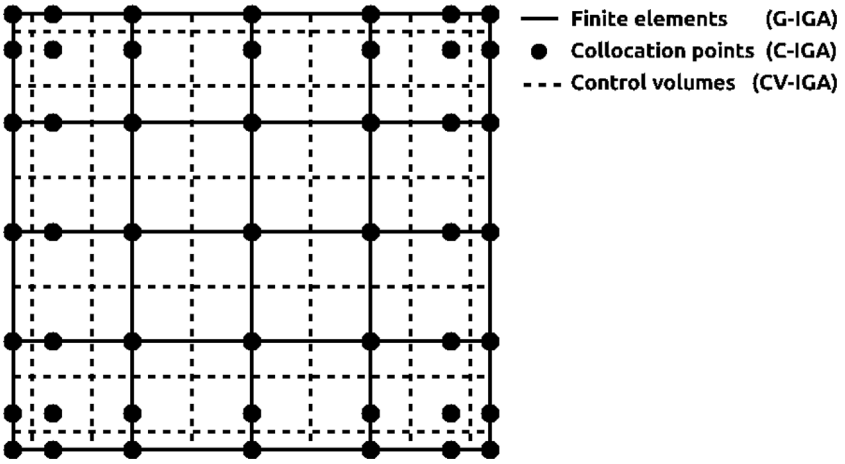


Fig. 6. Discretization of 2-D domain with three different IGA formulations. Note that Greville points (black circles) represents collocation points but also locations of the vertices of basis functions which are crucial for creating CV boundaries.

where $u(\mathbf{x})$ represents the solution, while the first and second term in Eq. (38) 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 (38) over the domain of interest and multiply it by a finite number of weighting (test) functions $w_i(\mathbf{x})$:

$$\int_\Omega \nabla \cdot (D\nabla u(\mathbf{x}))w_i(\mathbf{x})d\Omega - \int_\Omega \nabla \cdot (vu(\mathbf{x}))w_i(\mathbf{x})d\Omega = 0 \tag{41}$$

where the number of test functions (w_i) is generally the same as the number of basis functions. Two most used formulations in IGA are Galerkin (G-IGA; test functions are the same as basis functions, Hughes et al. [8]) and collocation formulation (C-IGA; test functions are Dirac functions located at Greville points, Schillinger et al. [42]). However in this work, formulation of control volume within IGA (CV-IGA) will be introduced [12,37,38].

The control volume formulation is performed by firstly dividing the parametric space by m control volumes (see Fig. 6) ($\Omega_i; i = 1, \ldots, m$). CV formulation [43] uses test functions defined in the following form:

$$w_i(x) = \begin{cases} 1 & \mathrm{x} \in \Omega_i \\ 0 & \mathrm{x} \notin \Omega_i, \end{cases} \Omega_i \in \Omega. \tag{42}$$

Substituting (42) in (41) and integrating only over the ith control volume (CV) due to the properties of the test functions (42), the volume integrals at left side over the control volume are transformed into a surface integrals across Ω_i boundaries Γ_i using Gauss's theorem:

$$\int_{\Gamma_i} \big(D\nabla u(\mathbf{x})\big)\boldsymbol{n}d\Gamma - \int_{\Gamma_i} \big(vu(\mathbf{x})\big)\boldsymbol{n}d\Gamma = 0 \tag{43}$$

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

Finally, weak formulation (43) is defined on each control volume using spline basis functions and unit compactly supported test functions (42) in order to get fully discretized control volume formulation:

$$\alpha_j \left[\int_{\Gamma_i} \big(D\nabla\varphi_j(\mathbf{x})\big)\boldsymbol{n}d\Gamma_i - \int_{\Gamma_i} \big(v\varphi_j(\mathbf{x})\big)\boldsymbol{n}d\Gamma_i \right] = \int_{\Gamma_{N_i}} q_N d\Gamma_N \tag{44}$$

where i denotes index of control volume and row of stiffness/conductance matrix, while j denotes index of spline basis function and column of the stiffness/conductance matrix. It is valid for all internal CV faces and boundary CV faces with Neumann boundary conditions. This implies that Neumann boundary conditions are weakly imposed by

incorporating the known value of q_N to the weak formulation. However, as in G-IGA, Dirichlet essential boundary conditions requires special treatment. In this paper, Dirichlet boundary conditions are satisfied in the strong sense by directly satisfying the boundary conditions values in the following form:

$$\int_{\Gamma_{D_i}} u(\mathbf{x}) d\Gamma_{D_i} = \int_{\Gamma_{D_i}} u_D(\mathbf{x}) d\Gamma_{D_i} \tag{45}$$

After using set of spline basis functions for representation of the numerical solution $u(\mathbf{x})$ (see Eq. (37)) yields

$$\alpha_j \int_{\Gamma_{D_i}} \varphi_j(\mathbf{x}) d\Gamma_D = \int_{\Gamma_{D_i}} u_D(\mathbf{x}) d\Gamma_D. \tag{46}$$

It should be noted that Dirichlet boundary conditions are satisfied in similar sense as in classical FEM. Equations linked with CVs which contain Dirichlet boundary conditions are replaced with (45), and later when we solve the system of equations and get unknown spline coefficients, from these memorized equations the Dirichlet boundary fluxes are calculated.

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 [43].

CV-IGA requires cheaper numerical integrations then G-IGA because control volume formulation (43) requires only integration over CV boundaries Γ_i, while Galerkin formulation requires (full) integration over the part of 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. On the other side, CV-IGA is more expensive than C-IGA which contains only one integration (collocation) point per degree of freedom and smaller number of nonzero elements in the stiffness/conductance matrix. Generally, CV-IGA lies between two classical IGA formulations enabling local and global mass conservation (see details in Gotovac et al. [12]).

3.2.1. Hierarchical mesh

CV-IGA directly depends on selection of dimensions and positions of CVs. Since control volume approach can be considered as subdomain collocation, selection of control volumes is directly related to the Greville collocation points (see IGA collocation for example in Schillinger et al. [42]). 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 and η axis by the length of the characteristic interval ($\Delta\xi$, $\Delta\eta$). However, not all vertices are uniformly spaced according to the length of the characteristic interval. Vertices of the modified boundary basis functions (see Section 2.2) are shifted and located inside the domain area. Their exact location can be calculated. 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 [44]. 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 (Fig. 6 - black circles) for the Fup_n basis functions can easily be computed from a knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{m+n+2}\}$

$$\hat{\xi}_i = \frac{1}{n+1}(\xi_{i+1} + \cdots + \xi_{i+n+1}), \quad i = 1, \ldots, m \tag{47}$$

where n is the order of the basis functions, and m is the number of basis functions. 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 vertex coordinates are represented by the Greville points.

Fig. 6 shows distribution of finite elements, collocation points and control volumes for all three IGA formulations (see also Gotovac et al. [12]). For each control volume (CV), there are four CV boundaries or faces. Each CV boundary represents side faces of CV in a manner that it lies in the middle between two adjacent Greville points (see also Figs. 7 and 8). Fig. 7 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 Greville point (vertex) i.e., the number of basis functions corresponds to the number of CVs.

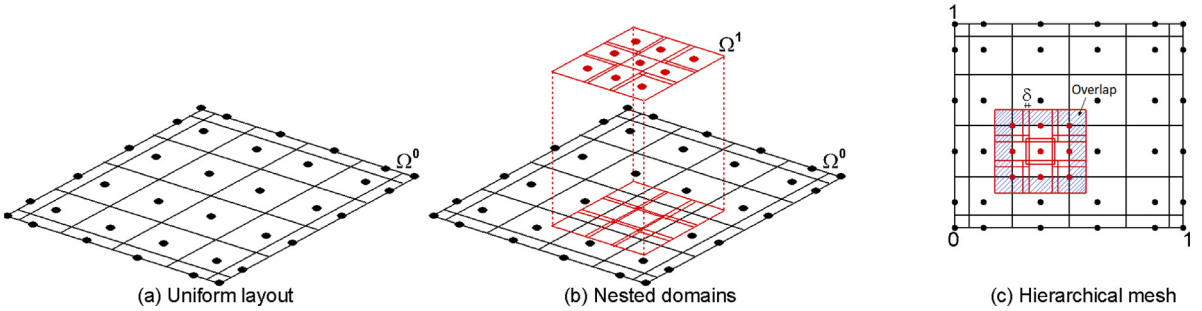


Fig. 7. A nested sequence of CV domains for the construction of the Fup hierarchy according to relation $\Omega^l \supseteq \Omega^{l+1}$ for two-dimensional case. (a) uniform nonoverlapping CV distribution at the first level; (b) CV distribution on the first and second resolution level with active Fup basis functions from $\mathcal{F}^0$ and $\mathcal{F}^1$; (c) Hierarchical mesh with overlapping CVs (dots represents basis functions vertices).

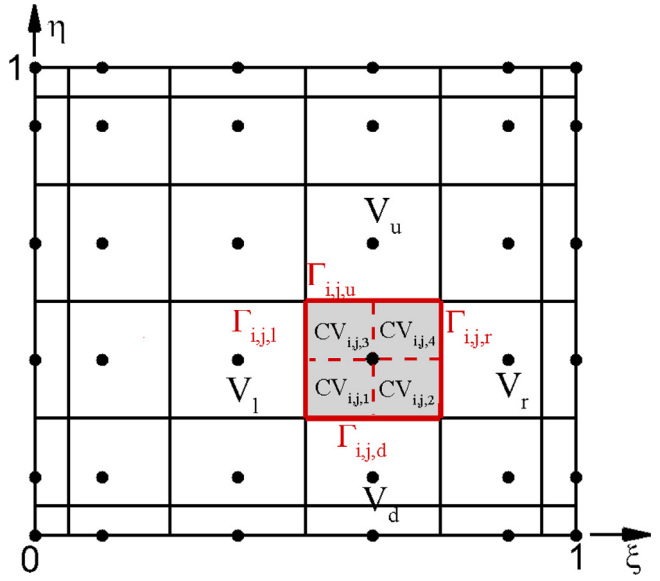


Fig. 8. Dividing ith CV into four equal parts ($\mathrm{CV}_{i,j,1}$, $\mathrm{CV}_{i,j,2}$, $\mathrm{CV}_{i,j,3}$ and $\mathrm{CV}_{i,j,4}$) while testing adaptive criteria on ith CV.

Fig. 7a shows uniformly distributed Fup_1 basis functions on the knot vectors $\Xi^0 = \{0,0,0,\frac{1}{8},\frac{2}{8},\frac{3}{8},\frac{4}{8},\frac{5}{8},\frac{6}{8},\frac{7}{8},1,1,1\}$ and $H^0 = \{0,0,0,\frac{1}{8},\frac{2}{8},\frac{3}{8},\frac{4}{8},\frac{5}{8},\frac{6}{8},\frac{7}{8},1,1,1\}$ with the position of vertices (black dots) and corresponding CVs represented with solid line. Furthermore, after replacing one Fup_1 (assigned as passive) basis function from the first resolution level, nine Fup_2 basis functions (red dots, assigned as active) are introduced to the second resolution level (Ω^1). Each CVs boundary on the higher second level (see Fig. 7b–c) are positioned exactly half the length of the characteristic intervals $\Delta\xi$, $\Delta\eta$ (see Eq. (35)) from the corresponding Fup basis function vertex, thus higher levels (CVs) are overlapping with lower levels (CVs). After assembling all active basis functions (assembles active Fup_1 and Fup_2 basis functions; $\mathcal{F}^1 = \mathcal{F}^0_a \cup \mathcal{F}^1_a$), CV overlapping distribution is defined for these two levels. However, it is possible that few CVs from higher level cover the same area as one larger CV from the lower level which creates problem of linearly dependent equations. Therefore, higher level that is in contact with lower level should have increased CV area to avoid problem of singular stiffness matrix. Enlargement of CV dimensions to $\Delta\xi(1+\delta)$, $\Delta\eta(1+\delta)$ can be chosen using parameter $\delta \in \langle 0, \frac{1}{2}\rangle$. Here, we choose $\delta = \frac{1}{4}$ (see Fig. 7c). All CVs from different resolution levels are rectangles in the parametric domain. Overlapping of some CVs makes this algorithm even more robust, but main advantage is easier process of constructing test (weight) functions in two-dimensional domains. Also, it should be emphasized that the CV overlapping is the simplest possible algorithm when compiling hierarchical mesh and to simplify the numerical integration across each control volume. Some different algorithm

could avoid overlapping of CVs, but would make integration process more complex over irregular CVs including cumbersome meshing procedure with Voronoi cells. We choose as simple as possible algorithm with all regular CVs in the parametric space, while avoiding any meshing procedure which can compromise meshless nature of CV-IGA.

3.3. Application of adaptive strategy

3.3.1. Function approximation

Hp-adaptive CV-IGA with hierarchical Fup basis functions is easy and effective to present firstly in the simple functions approximation. 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, while fine control volumes and a higher order of Fup basis functions are used in those areas where the solution varies strongly.

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

$$\varepsilon(x, y) = f(x, y) - \tilde{f}(x, y) = f(x, y) - \sum_{j=1}^{m} \alpha_j \varphi_j(x, y) \tag{48}$$

The meaning of the approximation is to minimize the error $\varepsilon(x, y)$. If the control volume formulation is applied, the unknown coefficients α_j are obtained from the following system of equations according to the mentioned weighted residuals:

$$\sum_{j=1}^{m} \alpha_j \int_{\Omega_i} \varphi_j(x, y) d\Omega = \int_{\Omega_i} f(x, y) d\Omega; \quad i, j = 1, 2, \ldots, m \tag{49}$$

which can be presented in a reduced matrix form:

$$a_{ij}\alpha_j = b_i; \quad i, j = 1, 2, \ldots, m \tag{50}$$

where

$$a_{ij} = \int_{\Omega_i} \varphi_j(x, y) d\Omega; \quad b_i = \int_{\Omega_i} f(x, y) d\Omega. \tag{51}$$

The adaptive criteria for the function approximation is defined as:

$$\int_{\Omega_A} \frac{1}{\Omega_A} \left(|f(x, y) - \tilde{f}(x, y)| \right) d\Omega < \varepsilon_A \tag{52}$$

where ε_A represents the defined threshold and Ω_A is the integration area. Adaptive criteria (ε_A) defines whether Fup basis functions are kept (assigned as $active$) or replaced (assigned as $passive$) while refining resolution level l following expression (34). For the ith control volume (CV), boundaries are defined via $\Gamma_{i,j,l}$, $\Gamma_{i,j,r}$, $\Gamma_{i,j,u}$ and $\Gamma_{i,j,d}$ (see Fig. 8), where subscript letter l (left), r (right), u (up) and d (down) represents side faces of the CV. Since the numerical approximation ($\tilde{f}$) satisfies the average function value of the known function (f) over every CV_i ($i = 1, 2, \ldots, m$) on the current resolution level, the main problem for enabling an adaptation is to test how close the numerical approximation ($\tilde{f}$) is with respect to the known function (f). Therefore, we perform testing on the each quarter of the CV (see Fig. 8). If all CVs satisfy adaptive criteria (52), the adaptive procedure stops. However, if one or more CVs did not satisfy Eq. (52), than those CVs are marked as refinable. Furthermore, all corresponding Fup basis functions that are at least partially located inside refinable CVs are marked as $passive$. Other Fup basis functions are marked as $active$, and they are kept in the next level. For the $passive$ Fup basis functions, the algorithm introduces their children, as it is earlier explained (see Eq. (29)). In this way, using mentioned adaptive criteria on CV quarters, adaptive hierarchical grid is created using the hierarchical Fup basis functions with different resolutions and orders over the adaptive grid.

The adaptive approximation procedure can be summarized as follows:

- The initial domain Ω^0 is defined, together with the trial function space of uniformly distributed Fup_n basis functions ($\mathcal{F}$) on the first resolution level of given order n and error threshold ε_A is set. Usually, Fup_1 basis of low order are chosen in the first starting resolution level.
- Unknown coefficients α are obtained by solving the system of equations (49).
- Adaptive criteria (52) needs to be satisfied on all CVs.
- If (52) is not satisfied for some CVs and related basis functions on the first resolution level, then adaptive procedure applies local hierarchical refinement (see Section 2.2) to the marked regions (CVs) and basis functions on the second resolution level. Next, i.e., higher resolution level contains higher order basis functions enabling hp-refinement and hp-adaptive procedure. Unknown coefficients in (49) are now solved for all CVs related to the basis functions for the first two resolution levels.

Adaptive procedure is stopped at certain resolution level where adaptive criteria (52) is satisfied over all CVs and resolution levels. The final result is nested sequence of domains with hierarchical Fup basis functions in regions where local refinement was necessary describing all needed spatial resolution scales.

3.3.2. Boundary value problems

The hp-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. The question is how to solve (approximate) the BVP. One of the possible approaches is shown considering the ADE (see Eq. (38)). In that case, solving ADE is reduced to the flux conservation over all CVs (see Eq. (44)). Since the CV formulation exactly satisfies Eq. (44) (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 quarter of the particular ith CV (see Fig. 8) i.e., the adaptive criteria for the ADE problem is defined as

$$\alpha_j \left[\int_{\Gamma_i} \big(D \nabla \varphi_j(\mathbf{x})\big) \boldsymbol{n} d\Gamma_i - \int_{\Gamma_i} \big(v \varphi_j(\mathbf{x})\big) \boldsymbol{n} d\Gamma_i \right] - \int_{\Gamma_{N_i}} q_N d\Gamma_N < \varepsilon_A \tag{53}$$

where ε_A represents the defined threshold. All CVs where at least one of the quarters has the conservation error greater than the prescribed threshold are marked as refinable, and the adaptive procedure refines selected (assigned as $passive$) basis functions in the next level in the same way as for the function approximation. Moreover, in BVP there are boundary conditions that needs to be satisfied. CVs with Neumann boundary conditions are satisfied in same sense as all internal CVs by checking conservation error for each quarter of the particular ith CV, i.e. Neumann boundary conditions are weakly imposed by incorporating the known value (40) to the weak formulation (44). However, Dirichlet boundary conditions are satisfied in the strong sense by directly satisfying the boundary values (39). This implies that CVs with Dirichlet boundary conditions use calculated boundary fluxes from memorized equations and check mass conservation with other internal fluxes. If mass conservation is not satisfied, those Dirichlet CVs are marked as refinable, thus adaptive procedure selects all boundary Fup basis functions that are in contact with that CV and marks them as $passive$. Boundary hierarchical basis functions are modified so that ith derivation is satisfied following expression (27). Adaptive procedure repeats local hierarchical refinement until mass conservation is satisfied over all boundary and interior CVs.

In the function approximation, adaptive criteria is set to be related to the function accuracy equation (52), while in BVP the criteria is set to be the mass conservation error. However, the adaptive criteria can be defined in many ways. 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, for BVP, the solution error between two resolution levels can be defined as criteria [45]. Moreover, satisfaction of the Peclet number can be very valuable for ADE problems (see [46]). Finally, any combination of these criteria can also be new obtained criteria.

4. Numerical examples

4.1. Aim of the numerical examples

The aim of the numerical tests herein is to investigate whether adaptive refinement using hierarchical Fup basis functions achieves spectral convergence rates, even while solving problems that may involve singularities, contrary to the application of uniform grid.

Numerical examples are started with function approximation for easier understanding of whole adaptive process. This example demonstrates HF's ability to capture sharp fronts by introducing new levels into a portion of the domain where it is needed. To demonstrate the potential of HF within CV-IGA we address the following classical benchmark 2-D problems:

- Poisson equation
- Heat conduction problem
- Advection–dispersion problem

Analytical solutions are available for all problems except advection–dispersion problem. All of the examples illustrate the ability of HF's to efficiently and accurately describe different resolution scales.

4.2. Verification tests

4.2.1. Function approximation

The selected test 2-D function is:

$$f(x, y) = \arctan\left(50\left(-0.25 + \sqrt{x^2 + y^2}\right)\right) \tag{54}$$

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. (48)) between the Fup approximation and the given function (54) over all CVs at all resolution levels must be less than this prescribed threshold. Fig. 9 shows the evolution of the adaptive procedure using HF at five consecutive resolution levels starting with uniform $\text{Fup}_1(x, y)$ basis functions.

Function approximations of the given function (54) over all levels are shown in Fig. 9(1a–5a). The error measure between the numerical approximation and given function is residual which can be calculated as the integral difference between those two functions on all quarters of the CVs. Fig. 9(1b–5b) 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 (in Fig. 9 red circle represents Fup_1 basis functions, blue circle Fup_2 basis functions, etc.). CVs are not directly shown but can be visualized with the help of the basis functions vertices, since every CVs edge is placed between the vertices of the adjacent functions (see Fig. 7).

The adaptive procedure is repeated until all residuals are less than the prescribed threshold. For given function (54) and adaptive threshold set as $\varepsilon_s = 10^{-7}$, adaptive procedure needs five levels to approximate given problem, as shown in Fig. 9. Note that fine CVs with a higher order of Fup basis functions are obtained only around the "well" edges describing high solution frequencies. Moreover, in other regions the adaptive grid uses lower order of Fup basis functions and coarse CVs which helps in reducing the computational cost and increases efficiency.

Fig. 10a shows efficiency in terms of the L_2 error norm as a function of the total degrees of freedom (DOFs) with slope representing the convergence rate (p) confirming that $p = n+1$ is valid for uniform grid if n is the Fup order. Adaptive procedure just like in one-dimensional case [38] yields spectral convergence (solid line with filled circles), contrary to the THB splines which ensures polynomial convergence ($p = n+1$). Spectral convergence enables that convergence rate is higher than $p = n + 1$ if n is the Fup order used at the highest resolution level. Spectral convergence is achieved due to using hp-refinement when higher resolution levels not only use basis functions with smaller scales or higher frequencies, but also increased order of basis functions. This hp property causes that new levels and DOFs more drastically increase accuracy than it is case with THB splines where all levels use the same order of the basis functions. Fig. 10a shows that convergence plot have larger slopes when new resolution levels and increased order of basis functions are introduced. Furthermore, adaptive procedure achieves a higher accuracy then the prescribed threshold (dashed line with empty circles, ε_A), thus proving the control of the numerical error.

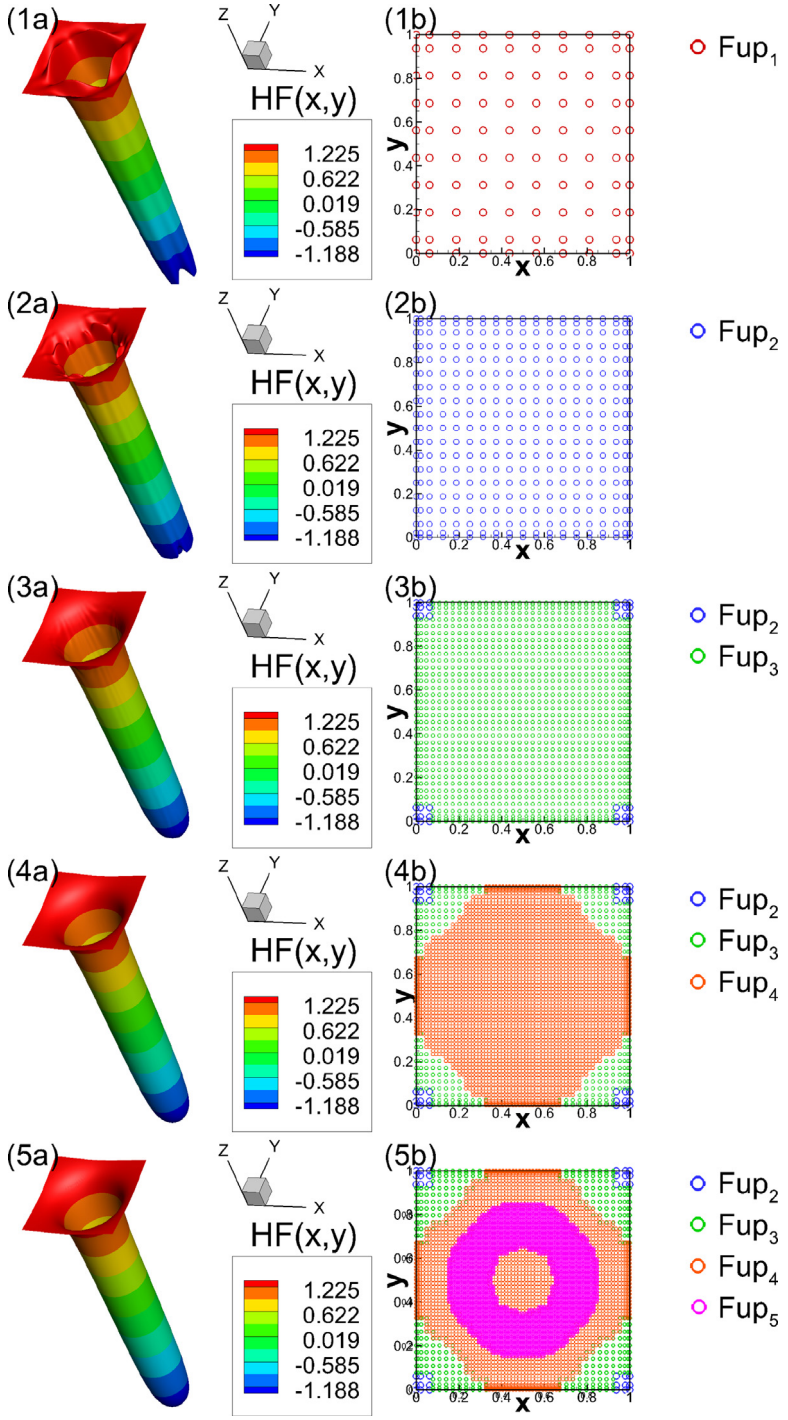


Fig. 9. CV-IGA approximation of the function (54). (a) HF approximations of the given function, (b) the adaptive grid on different resolution levels where each color represents Fup basis function vertices on different level.

This means that the real numerical error of the function approximation is strictly less than the prescribed threshold. Moreover, Fig. 10b shows CPU time in seconds vs L_2 error norm. It can be seen that adaptive method achieves more efficient solution in terms of computation cost in comparison with uniform grid. It should be noted that by

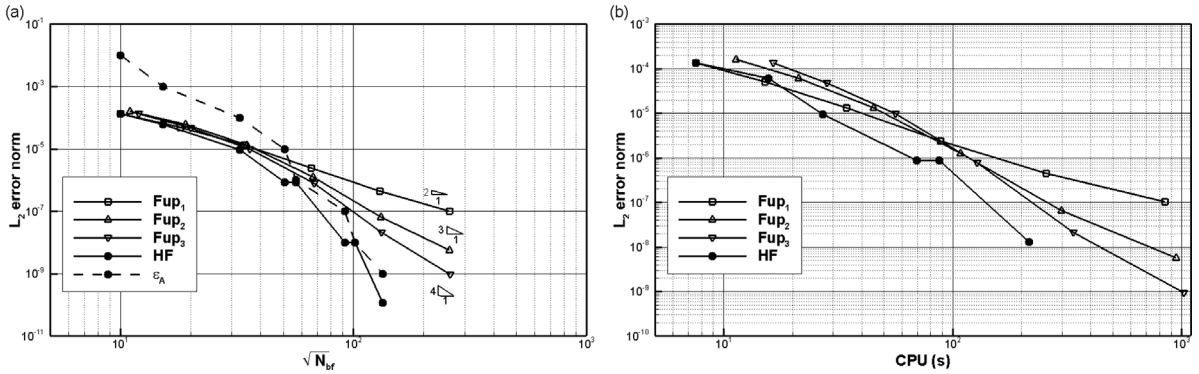


Fig. 10. (a) Convergence analysis obtained with uniform and adaptive Fup_n basis functions for the function approximation; (b) CPU time (in seconds) analysis for uniform and adaptive method.

changing adaptive criteria or setting new preconditioners we can maybe achieve even better results. However, we wanted to show the highlight of HF within adaptive procedure, i.e. that it is possible to do local hp-refinement without any constrains achieving spectral convergence.

4.2.2. Poisson Equation

For 2-D Poisson benchmark 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 [47–49]. 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 (54), 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

$$\nabla \cdot (-\kappa \nabla u(x, y)) = f(x, y), \quad (x, y) \in \Omega \tag{55}$$

with boundary conditions

$$u(x, y) = u_D(x, y), \quad (x, y) \in \partial\Omega \tag{56}$$

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 Fig. 11a). The exact analytical solution for the pressure field is given by:

$$u(x, y) = \arctan(\alpha\,(r - r_0)) \quad \text{where} \quad r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{57}$$

where x_c and y_c represent 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.

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 Eq. (57). The exact solution depicted in Fig. 11b 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

$$\kappa = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{58}$$

The adaptive simulation shown in Figs. 12 and 13 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 quarters CVs on every level must be less than this prescribed threshold.

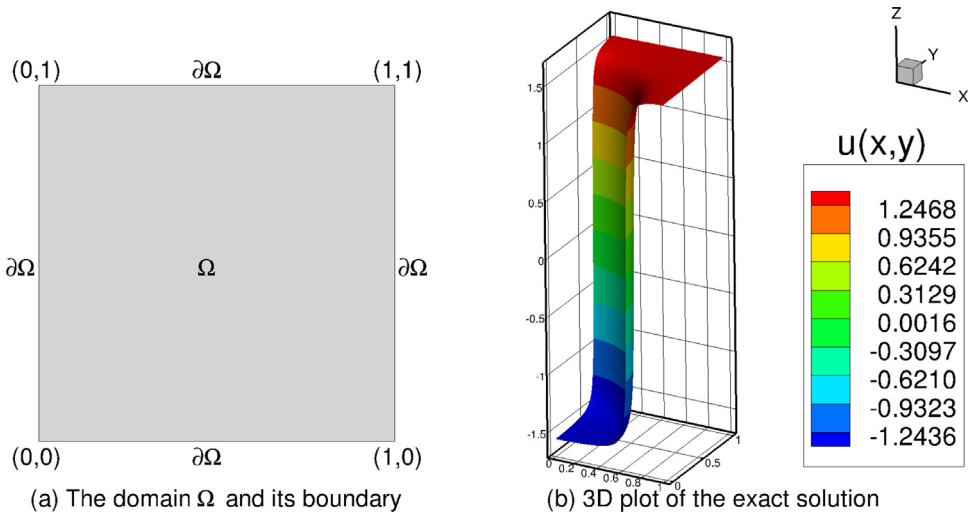


Fig. 11. Numerical solution domain and exact solution plot of the wave well problem defined by Eq. (55).

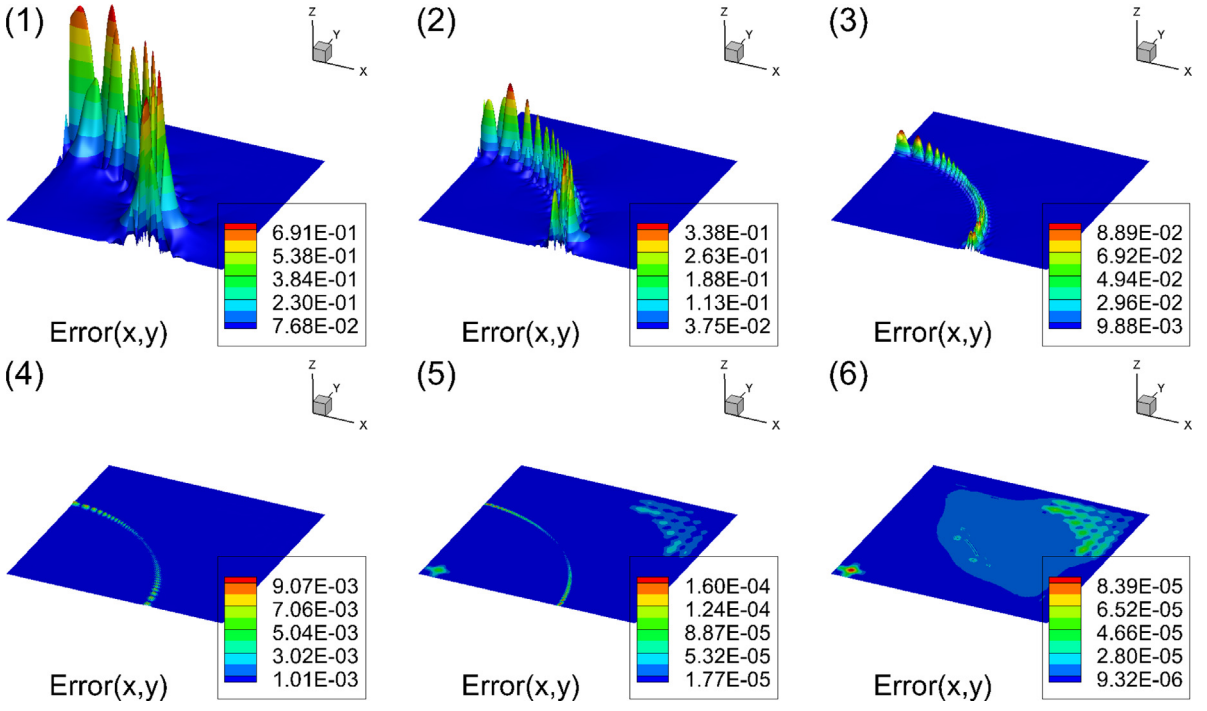


Fig. 12. The absolute difference between the numerical and exact (57) solution on different resolution levels (1–6).

Figs. 12 presents the absolute difference between the numerical and exact (see (57)) solution while Fig. 13 presents the adaptive grid on different resolution levels. With every new level, numerical solution becomes closer to the real solution (Fig. 12 1a–6a). Even though, difference between numerical solution and exact solution is presented in Fig. 12 because exact solution is known, it was not the adaptive criteria used for testing like in approximating function (54). Here, adaptive criteria is used to check conservation error for each quarters of the particular ith CV on the current resolution level. Quarters of the CVs are used because CV formulation exactly satisfies governing

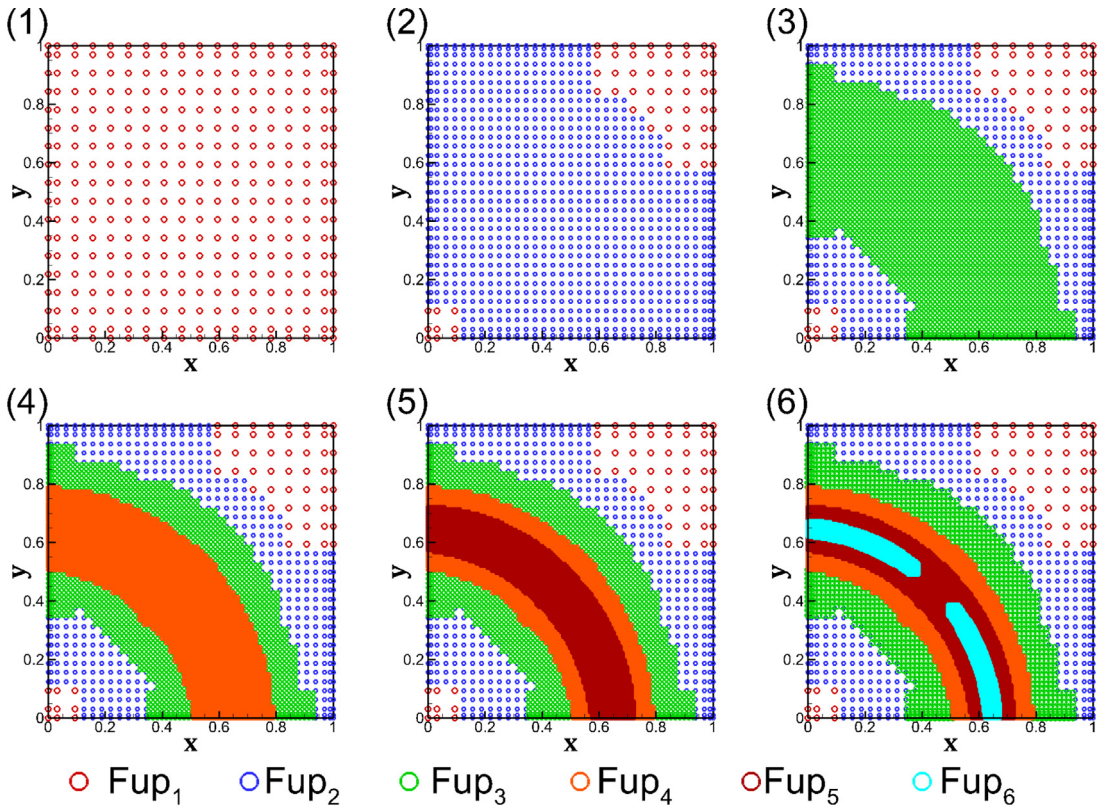


Fig. 13. The adaptive grid on different resolution levels, (1) first, (2) second, ..., (6) sixth level. Each color represents Fup basis function vertices on different level.

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 Fig. 13) and repeats adaptive procedure until conservation error is less than 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 than prescribed error threshold on all quarters of the CVs.

The convergence analysis for the uniform and adaptive procedure is shown in Fig. 14a. It 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 using the uniform grid is the optimal ($p = n + 1$) for odd and the suboptimal ($p = n$) for even order (n) of basis functions. 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 (collocation) yields suboptimal convergence rates of $p = n - 1$ for odd basis functions and $p = n$ for even basis functions [12,41]. The adaptive procedure for this diffusive-like boundary value problem exhibits spectral convergence (black solid line with filled triangles), just like in one-dimensional case [38]. CPU time analysis with hp-refinement method for given problem is shown in Fig. 14b. It shows that adaptive procedure achieves better CPU results then uniform tests. Taking convergence rates and CPU results for this test in consideration, we can say that presented local hp-refinement with HF shows great performance improvement when applied to numerical problems where adaptive strategies are needed.

4.2.3. Heat equation (Laplace)

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

$$\Delta u = 0 \tag{59}$$

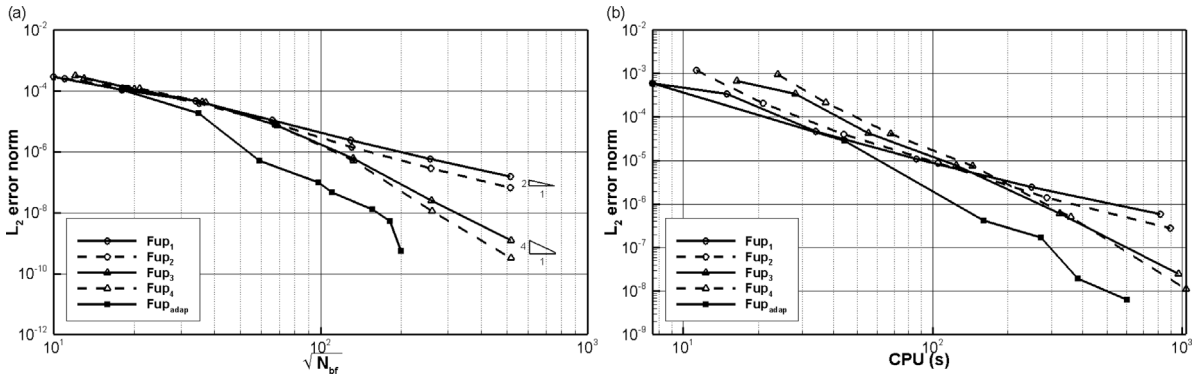


Fig. 14. (a) Convergence analysis of the wave front problem given in the form (55) for the uniform and adaptive procedure; (b) CPU time (in seconds) versus L_2 error norm.

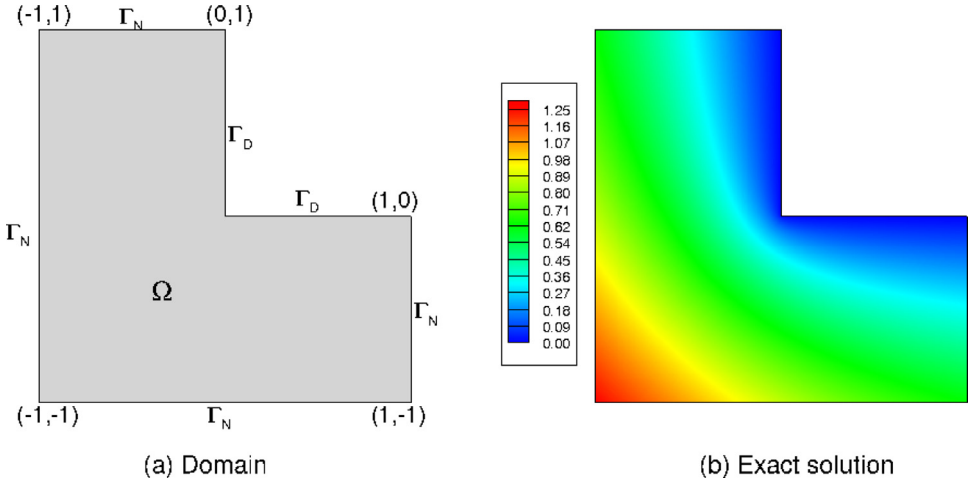


Fig. 15. The L-shape problem: (a) Numerical solution domain with boundary conditions and (b) exact solution plot.

on an L-shaped domain $\Omega = [-1, 1]^2 \setminus [0, 1]^2$, see Fig. 15a), with boundary conditions

$$u = 0 \quad on \quad \Gamma_D \tag{60}$$

$$\frac{\partial u}{\partial n} = q_N \quad on \quad \Gamma_N \tag{61}$$

such that the exact solution is given by

$$u = r^{2/3} \sin\left(\frac{2\theta - \pi}{3}\right) \tag{62}$$

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 (62). 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 [17].

Presented HF procedure starts with $m_x^0 = 18$, $m_y^0 = 18$ Fup basis functions on the first (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 than this prescribed threshold. The exact solution of the presented problem is shown in Fig. 15b.

L-shaped domain is discretized by two patches, as shown in Fig. 16b, while Fig. 16a shows control points for the coarse mesh.

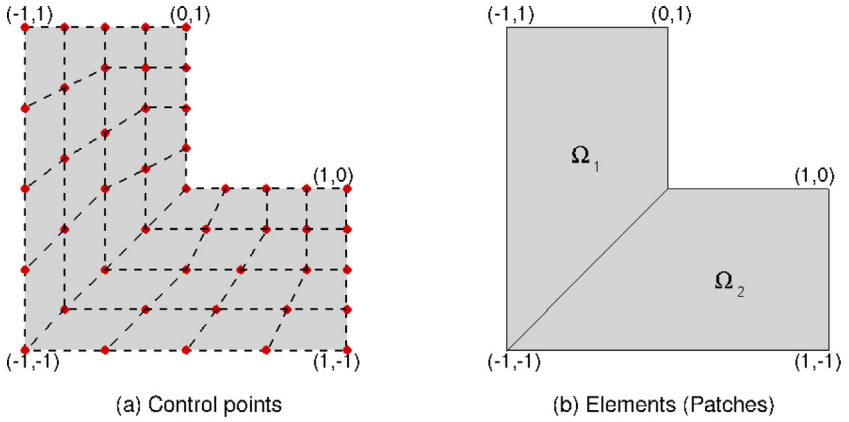


Fig. 16. The L-shape problem: (a) Fup discretized geometry with a $n_{cp} = 25$ number of control points per each element, and (b) for $n_{el} = 2$ number of patches. In a) red circles represent the control points, whereas the shaded region is the modeled geometry.

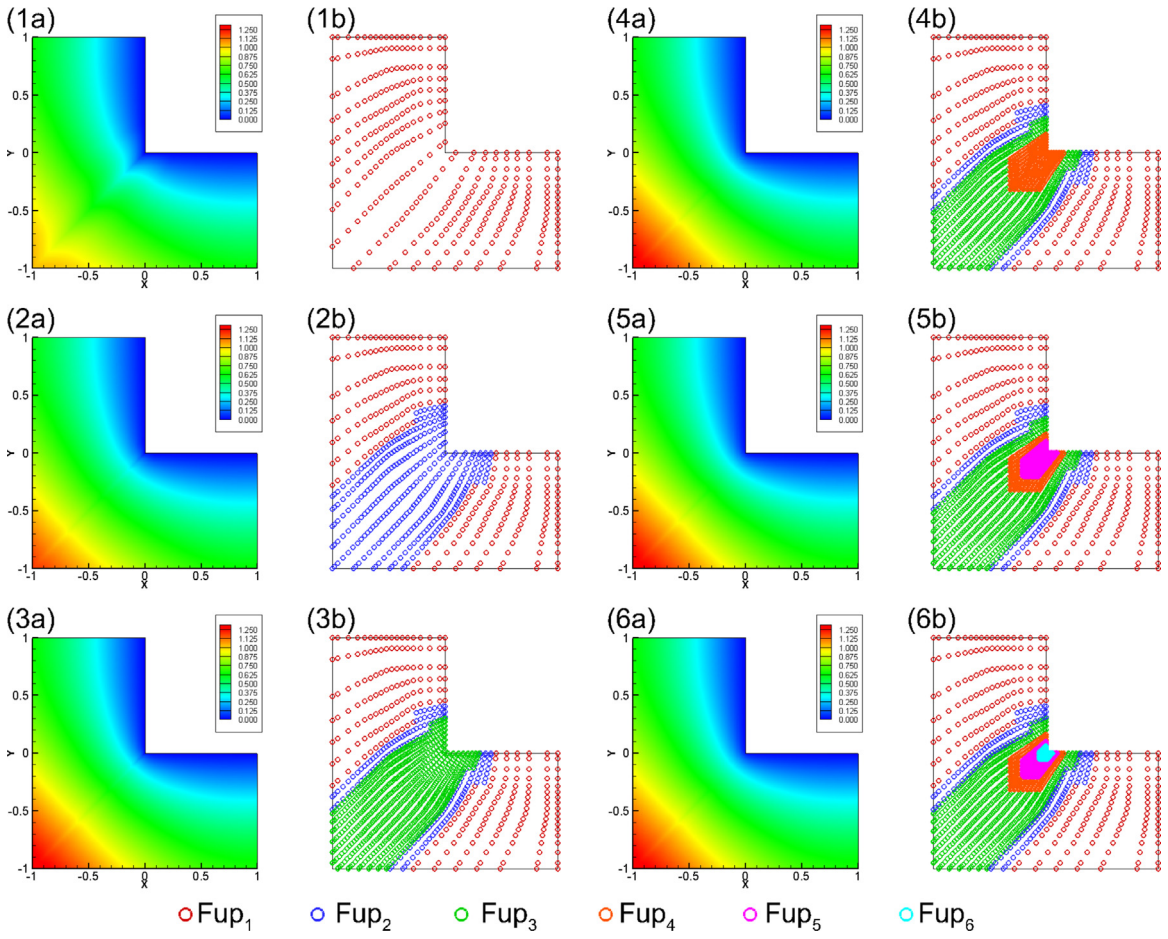


Fig. 17. Numerical solution of the stationary heat conduction problem defined over an L-shaped domain (governed by Laplace equation (59)) at different resolution levels; (1a–6a) HF approximations; (1b–6b) corresponding adaptive spatial grids.

Fig. 17 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 Fig. 17). Refinement captures

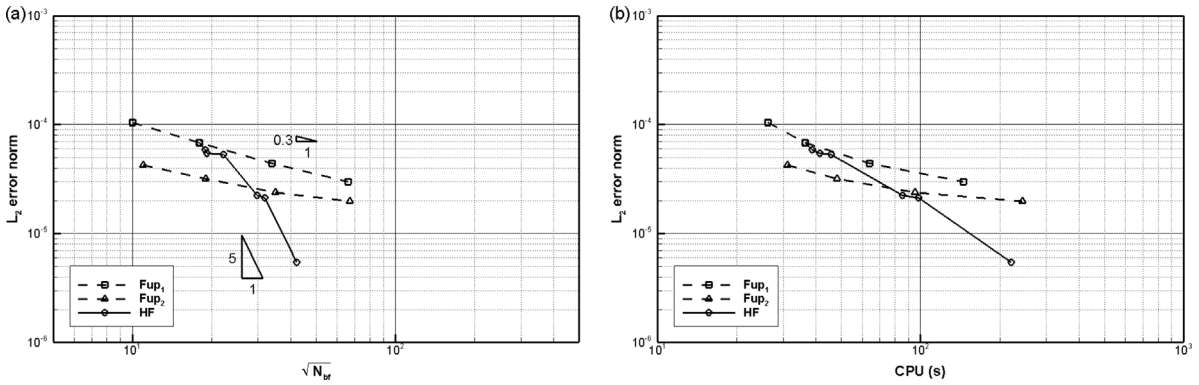


Fig. 18. (a) Convergence analysis for uniform and adaptive method for the L-shape problem; (b) CPU time (in seconds) analysis for uniform and adaptive method.

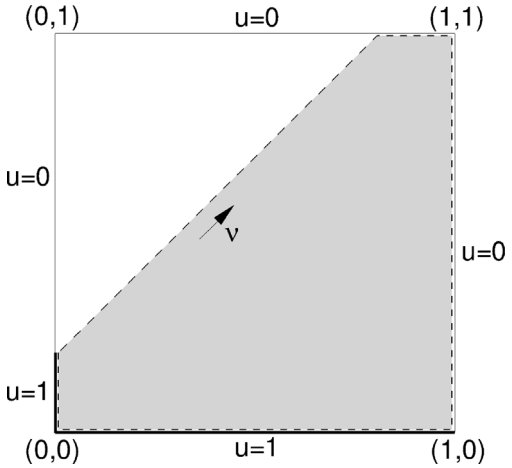


Fig. 19. Domain with discontinuous Dirichlet boundary conditions for the Advection–dispersion problem.

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 than prescribed error threshold on all quarters of the CVs.

The convergence analysis is performed using L_2 norm and is plotted in Fig. 18a for uniform Fup_1, Fup_2 and HF basis functions. It can be observed that adaptive HF basis functions again improves the convergence rate in comparison to the uniform layout. Moreover, uniform grids shows a significantly reduced convergence rate ($p = 0.3$) due to the re-entrant corner at $(0, 0)$ in the domain (singularity). The present numerical example thus confirms that adaptive algorithm significantly improves solution for rough problems still enabling spectral convergence. Convergence rate by parts is equal to uniform grid if new levels are not introduced. When new levels are introduced around singular corner, convergence rate exhibits spectral character. Overall, convergence is still spectral due to hp-refinement properties of the proposed method. Moreover, even in CPU tests we can see improvement as it is shown in Fig. 18b. Just like in previous two tests, HF adaptive procedure achieves great results by drastically reducing the DOF and CPU time while solving some classical benchmark problems.

4.2.4. Advection–dispersion equation

Steady-state discontinuous example

Two-dimensional benchmark example is taken from [8,11,19] which consists of solving the advection–dispersion equation

$$D\Delta u - v \cdot \nabla u = 0 \tag{63}$$

on the unit-square with discontinuous Dirichlet boundary conditions (see Fig. 19). 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 are considered. Sharp interior and boundary layers require stable numerical techniques as well as adaptive solutions in order to capture all resolution scales. Adaptation with hierarchical 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 [50] is employed as additional mechanism inside the adaptive procedure.

Adaptive resolution of the internal and boundary layers are investigated with the presented HF procedure starting from $m_x^0 = 18$, $m_y^0 = 18$ Fup basis functions on the first (uniform) resolution level. The error threshold is set as $\varepsilon_s = 1 \cdot 10^{-4}$. The exact solution of the presented problem is not known.

Firstly, Fig. 20 presents the evolution of the sharp boundary layer and corresponding adaptive spatial grids at five consecutive resolution levels in two-dimensional domain without stabilization. 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 advective term when this 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 levels are not placed only around the boundary layer and the internal layer, solely because adaptive algorithm just like in one-dimensional case [38] requires stabilization process to efficiently solve this advection dominated problem (see Fig. 21). It is more relevant to apply stabilization only to the first few levels (in this case, for the first three levels, $l = 1, 2, 3$) because the Peclet number is higher at the initial resolution levels. Fig. 21 presents the evolution of the numerical solution and corresponding adaptive spatial grids at four consecutive resolution levels with the stabilization method applied to the adaptive algorithm. Comparing grids with (Fig. 21) and without (Fig. 20) stabilization, methodology which uses stabilization yields significant improvement. 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.

Fig. 22 presents the convergence analysis of the adaptive algorithm using HF basis functions, with respect to the degrees of freedom used to achieve a certain accuracy. Since previous problem had reduced convergence rate for uniform test due to the singularity, in this test we skipped uniform analysis since problem has discontinuity within Dirichlet boundary conditions. As expected, HF adaptive algorithm achieves spectral convergence rate which is quite impressive for these type of problems. Note that previous authors did not present convergence plot for this ADE benchmark using THB.

Space–time advection–dispersion problem

This section describes the mixing of transport processes in the space–time domain, for instance in porous media [45]. Advection–dispersion process can be described by the following equation, in the form:

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} - v\frac{\partial C(x,t)}{\partial x} \tag{64}$$

with appropriate initial and boundary conditions:

$$C(x,0) = 0 \tag{65}$$

$$C(0,t) = C_0; \quad \frac{\partial C(2,t)}{\partial x} = 0 \tag{66}$$

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.

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. (65)) shows that initially the domain was occupied by fresh water ($C = 0$). However, the left boundary consists of 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.

Fig. 23a 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 and time. This change occurs in a narrow transition zone (see Fig. 23). Fig. 23b shows an adaptive grids 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

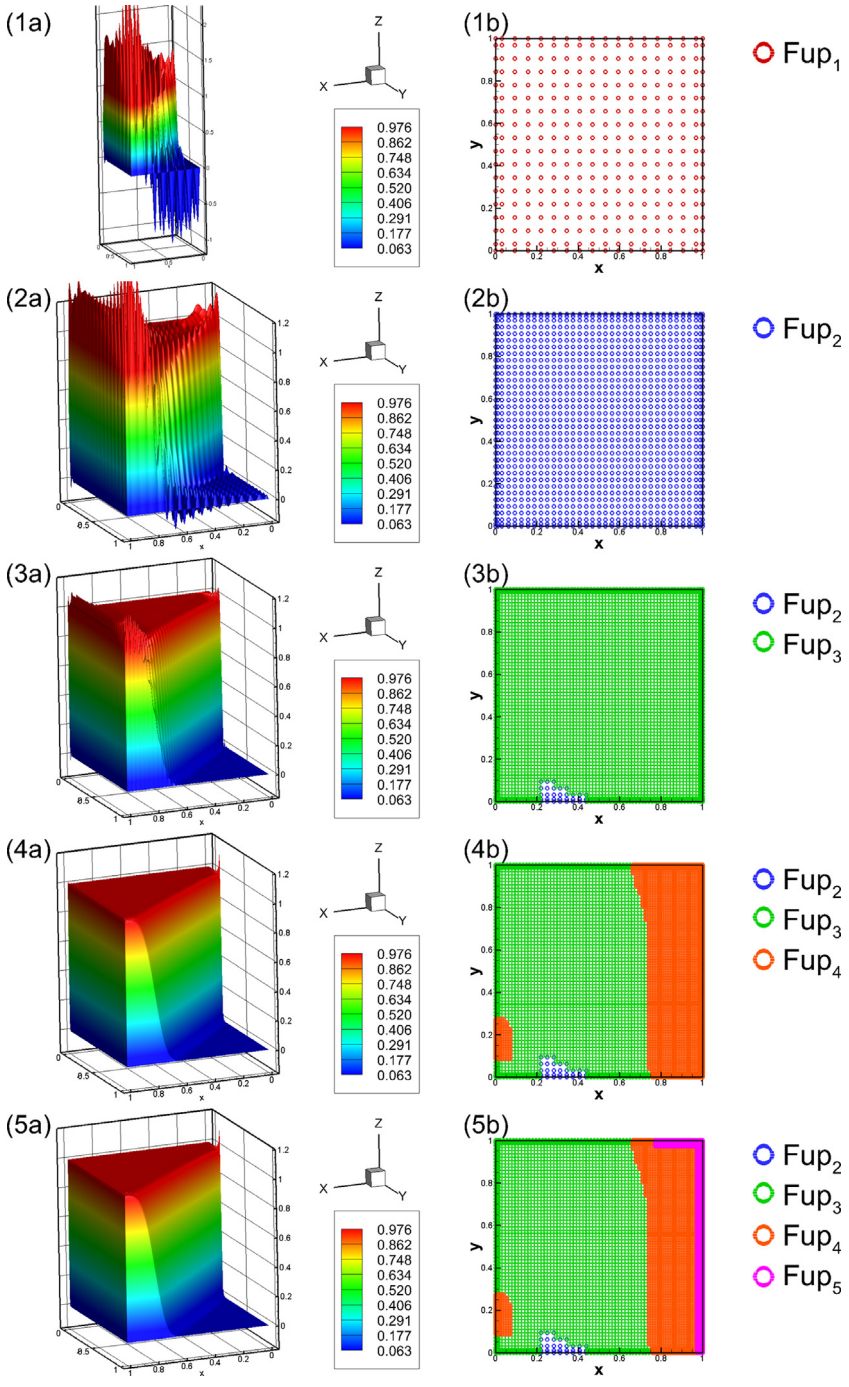


Fig. 20. Numerical solution of the ADE (63) at different resolution levels (without stabilization); (1a–5a) HF approximations; (1b–5b) corresponding adaptive spatial grids.

creation of a very sharp discontinuous concentration front. It should be noted that time domain is considered as one global time step. Furthermore, the initial error does not propagate further over time because proposed adaptive method converts the boundary–initial problem to a quasi-boundary problem controlling the global temporal/spatial error.

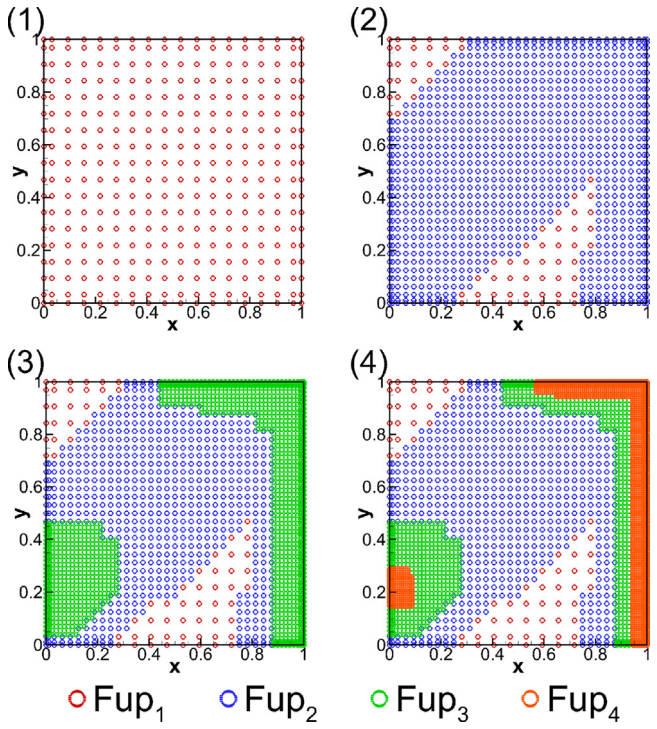


Fig. 21. Numerical solution of the ADE (63) with stabilization procedure at different resolution levels: corresponding adaptive spatial grids at the (1) first, (2) second, (3) third and (4) fourth level.

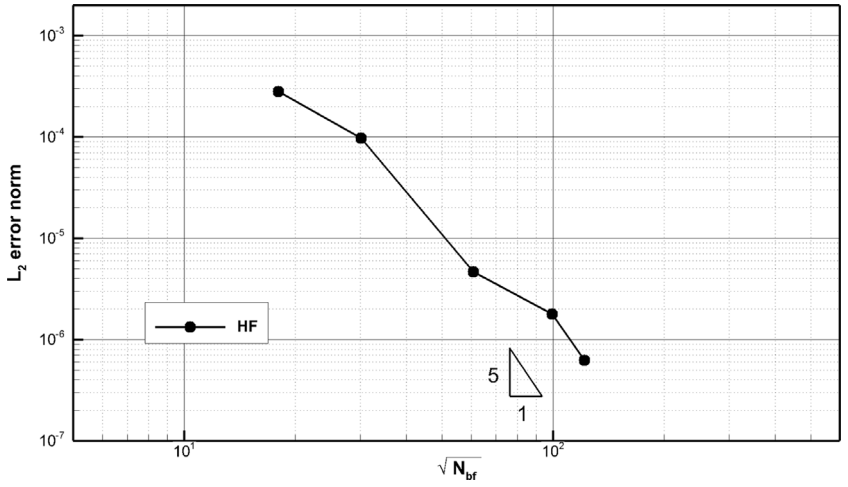


Fig. 22. Convergence analysis for adaptive method of steady state advection–dispersion problem (63).

Fig. 24 presents corresponding adaptive spatial grids at four consecutive resolution levels with the stabilization method. For two-dimensional cases, the idea of implementing stabilization method of upwinding cannot be easily applied. Various methods have been proposed to implement the basic idea of upwinding to 2-D analyses. Here, upwinding method adjusting for 2-D analyses with control volume procedure is used [51]. Moreover, the computational cost is reduced since fewer basis functions and levels (Fig. 23 vs Fig. 24) are needed to achieve the same mass conservation error on all quarters of the CVs.

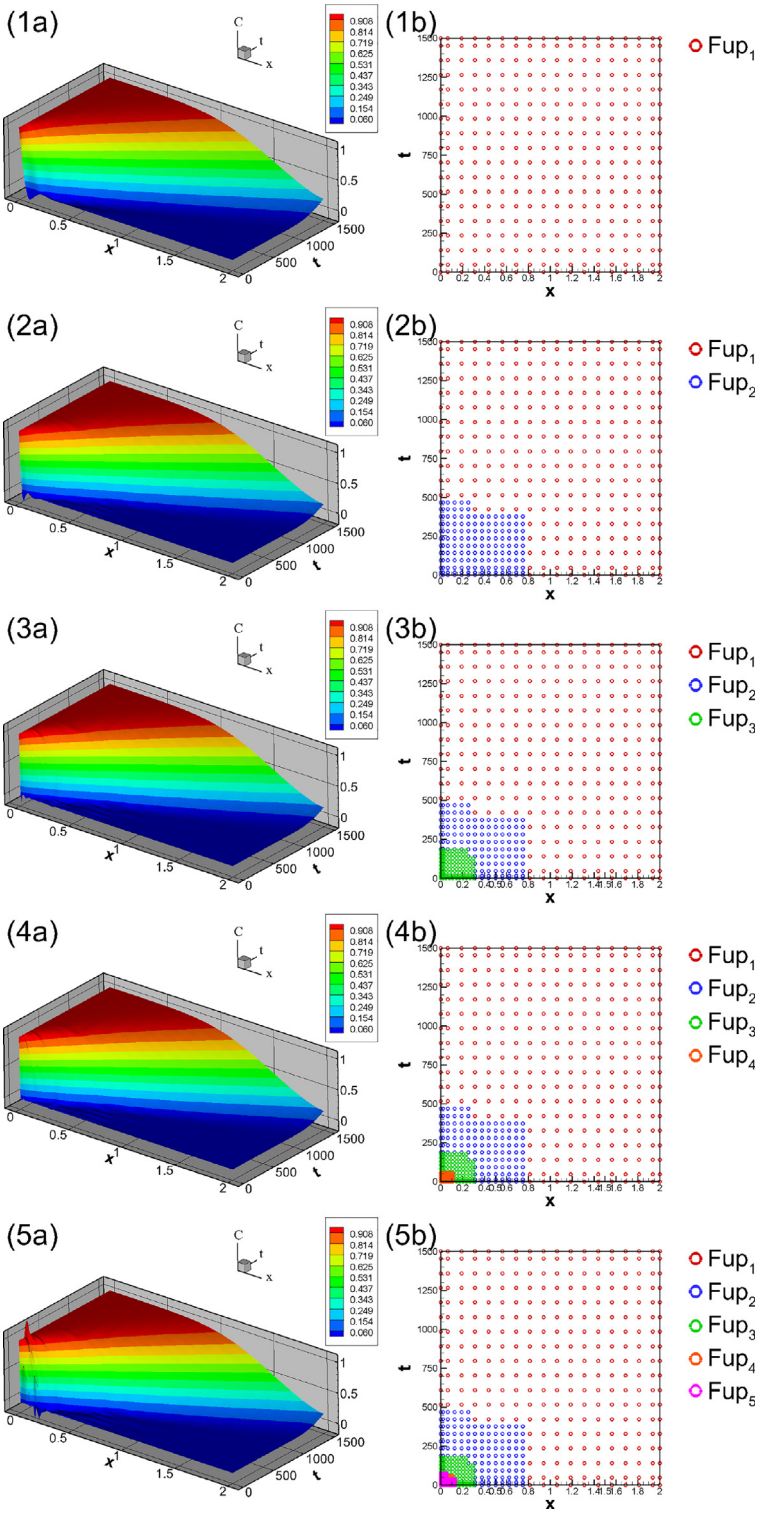


Fig. 23. Numerical solution of the ADE (64) at different resolution levels; (1a–5a) HF approximation (without stabilization), (1b–5b) corresponding adaptive time-spatial grids.

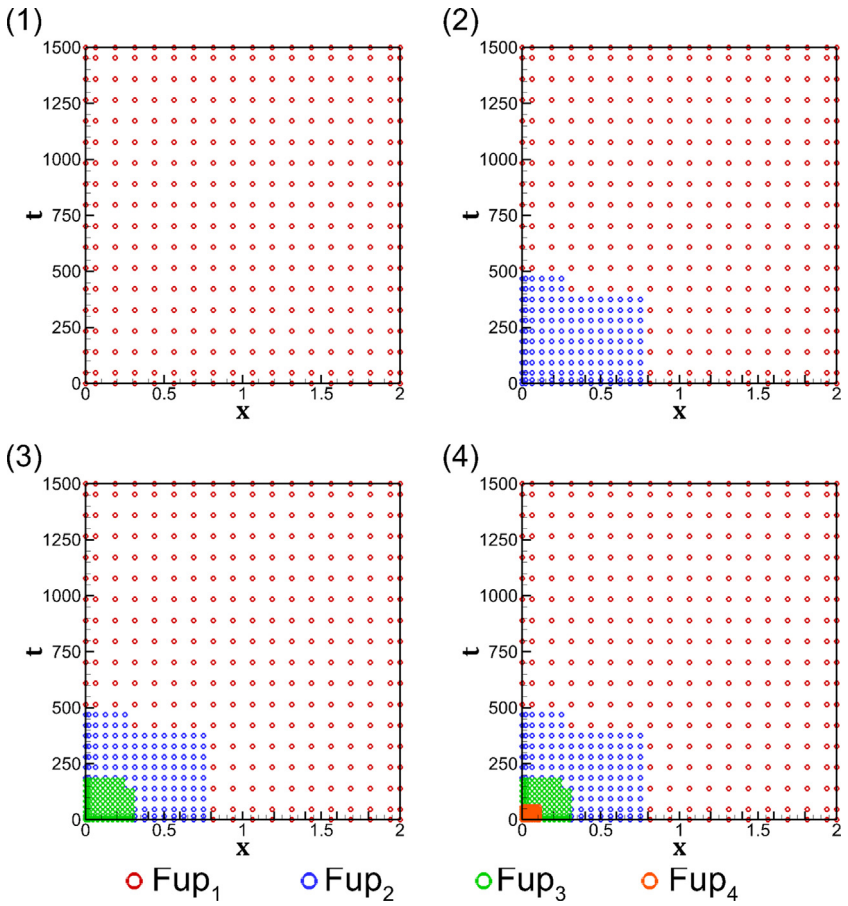


Fig. 24. Corresponding adaptive time-spatial grids at different resolution levels of the ADE (64) (with stabilization).

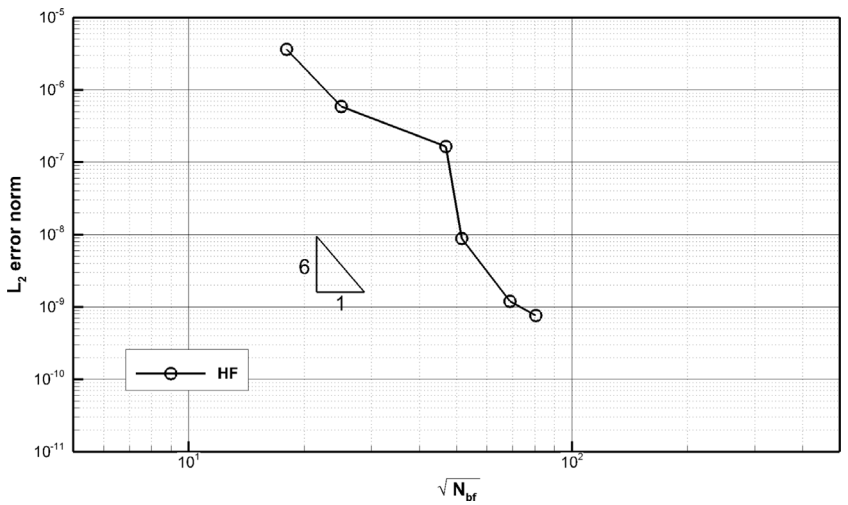


Fig. 25. Convergence analysis for adaptive method (without stabilization).

Fig. 25 shows convergence analysis using L_2 norm. Uniform analysis is skipped since presented problem has singularity due to discontinuity of boundary conditions, 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 grid handles moving fronts and have the ability to change the grid dynamically, following a front during the simulation while keeping the spectral convergence rate.

5. Conclusions

This paper presents the development of new 2-D hierarchical Fup (HF) basis functions that enable local hp-improvement inside adaptive control volume isogeometric analysis (CV-IGA). HF provides spectral convergence and presents a substantial improvement in comparison to THB that enable polynomial convergence. Hierarchical Fup basis functions do not require additional modifications to preserve the essential property of partition of unity that allow easy implementation of local hp-enhancements. Control volume formulation is simple, all control volumes are regular in the parametric space (also related to the Greville collocation points), only overlapping is needed in the zones of contact between different resolution levels.

The developed adaptive algorithm is presented first on a simple example of function approximation for the sake of simplicity of the presented adaptive algorithm, then to the application of the Poisson equation, which has wide implementation in structural and fluid mechanics. On the example of ADE, we show that even in cases when the advective member dominates and creates oscillations in solving using adaptive techniques, we achieve stability and accurate solutions. Even in non-smooth problems, spectral convergence is achieved contrary to the application of uniform grid. CV-IGA ensures local and global mass conservation which is potentially very important for fluid mechanics problems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by the Croatian Science Foundation (in Croatian: Hrvatska zaklada za znanost - HRZZ) through the scientific project “Multiphysics modelling of surface-subsurface water systems”, grant number: IP-2020-02-2298.

This research is partially supported through project KK.01.1.1.02.0027, a project co-financed by the Croatian Government and the European Union through the European Regional Development Fund - the Competitiveness and Cohesion Operational Programme.

References

[1] B.E. Rapp, Chapter 30 - finite difference method, in: Microfluidics: Modelling, Mechanics and Mathematics, in: Micro and Nano Technologies, Elsevier, Oxford, 2017, pp. 623–631.
[2] B.E. Rapp, Chapter 31 - finite volume method, in: Microfluidics: Modelling, Mechanics and Mathematics, in: Micro and Nano Technologies, Elsevier, Oxford, 2017, pp. 633–654.
[3] B.E. Rapp, Chapter 32 - finite element method, in: Microfluidics: Modelling, Mechanics and Mathematics, in: Micro and Nano Technologies, Elsevier, Oxford, 2017, pp. 655–678.
[4] A. Brandt, Multi-level adaptive solutions to boundary-value problems, Math. Comp. 31 (138) (1977) 333.
[5] I. Babushka, W.C. Rheinboldt, Error estimates for adaptive finite element computation, SIAM J. Num. Anal. 15 (4) (1978) 736–754.
[6] C.J. Leo, J.R. Booker, A boundary element method for analysis of contaminant transport in porous media I: homogeneous porous media, Int. J. Numer. Anal. Methods Geomech. 23 (14) (1999) 1681–1699.
[7] F.A. Tavarez, M.E. Plesha, Discrete element method for modelling solid and particulate materials, Internat. J. Numer. Methods Engrg. 70 (4) (2007) 379–404.
[8] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (39) (2005) 4135–4195.
[9] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, T. Lyche, T-spline simplification and local refinement, ACM Trans. Graph. 23 (3) (2004) 276–283.
[10] C. Giannelli, B. Jüttler, H. Speleers, THB-splines: The truncated basis for hierarchical splines, Comput. Aided Geom. Design 29 (7) (2012) 485–498, Geometric Modeling and Processing 2012.
[11] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, Isogeometric analysis toward intergration of CAD and FEA, ISBN: 978-0-470-74873-2, 2009, p. 335.

[12] H. Gotovac, L. Malenica, B. Gotovac, Control volume isogeometric analysis for groundwater flow modeling in heterogeneous porous media, Adv. Water Resour. 148 (2021) 103838.
[13] M.-C. Hsu, C. Wang, F. Xu, A.J. Herrema, A. Krishnamurthy, Direct immersogeometric fluid flow analysis using B-rep CAD models, Comput. Aided Geom. Design 43 (2016) 143–158, Geometric Modeling and Processing 2016.
[14] V.L. Rvachev, T.I. Sheiko, R-functions in boundary value problems in mechanics, Appl. Mech. Rev. 48 (4) (1995) 151–188.
[15] K. Höllig, C. Apprich, A. Streit, Introduction to the web-method and its applications, Adv. Comput. Math. 23 (2005) 215–237.
[16] J.A. Cottrell, T.J.R. Hughes, A. Reali, Studies of refinement and continuity in isogeometric structural analysis, Comput. Methods Appl. Mech. Engrg. 196 (41–44) (2007) 4160–4183.
[17] Y.W. Bekele, T. Kvamsdal, A.M. Kvarving, S. Nordal, Adaptive isogeometric finite element analysis of steady-state groundwater flow, Int. J. Numer. Anal. Methods Geomech. 40 (5) (2016) 738–765.
[18] G. Lorenzo, M. Scott, K. Tew, T. Hughes, H. Gomez, Hierarchically refined and coarsened splines for moving interface problems, with particular application to phase-field models of prostate tumor growth, Comput. Methods Appl. Mech. Engrg. 319 (2017) 515–548.
[19] A.-V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 200 (49–52) (2011) 3554–3567.
[20] L. Coradello, D. D'Angella, M. Carraturo, J. Kiendl, S. Kollmannsberger, E. Rank, A. Reali, Hierarchically refined isogeometric analysis of trimmed shells, Comput. Mech. 66 (2) (2020) 431–447.
[21] D.C. Thomas, M.A. Scott, J.A. Evans, K. Tew, 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, Comput. Methods Appl. Mech. Engrg. 284 (2015) 55–105.
[22] X. Wei, Y. Zhang, L. Liu, T.J. Hughes, Truncated T-splines: Fundamentals and methods, Comput. Methods Appl. Mech. Engrg. 316 (2017) 349–372, Special Issue on Isogeometric Analysis: Progress and Challenges.
[23] D. D'Angella, S. Kollmannsberger, E. Rank, A. Reali, Multi-level bézier extraction for hierarchical local refinement of isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 328 (2018) 147–174.
[24] M. Carraturo, C. Giannelli, A. Reali, R. Vázquez, Suitably graded THB-spline refinement and coarsening: Towards an adaptive isogeometric analysis of additive manufacturing processes, Comput. Methods Appl. Mech. Engrg. 348 (2019) 660–679.
[25] K.A. Johannessen, F. Remonato, T. Kvamsdal, On the similarities and differences between classical hierarchical, truncated hierarchical and LR B-splines, Comput. Methods Appl. Mech. Engrg. 291 (2015) 64–101.
[26] X. Wei, Y. Zhang, T.J. Hughes, M.A. Scott, Truncated hierarchical Catmull–Clark subdivision with local refinement, Comput. Methods Appl. Mech. Engrg. 291 (2015) 1–20.
[27] V.L. Rvachev, V.A. Rvachev, On a finite function, Dokl. Akad. Nauk Ukrainian SSR, Ser. A (6) (1971) 705–707.
[28] B. Gotovac, Numerical Modelling of Engineering Problems by Smooth Finite Functions (In Croatian), (Ph.D. thesis), Faculty of Civil Engineering, University of Zagreb, 1986.
[29] G. Beylkin, J.M. Keiser, On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases, J. Comput. Phys. 132 (2) (1997) 233–259.
[30] V. Kozulić, Numerical Modelling by the Fragment Method with Rbf Functions (In Croatian), (Ph.D. thesis), Faculty of Civil Engineering, University of Split, 1999.
[31] B. Gotovac, V. Kozulić, On a selection of basis functions in numerical analyses of engineering problems, Int. J. Eng. Modell. 12 (1–4) (1999) 25–41.
[32] V.F. Kravchenko, M.A. Basarab, V.I. Pustovoit, H. Pérez-Meana, New constructions of weight windows based on atomic functions in problems of speech-signal processing, Doklady Phys. 46 (2001) 166–172.
[33] B. Gotovac, V. Kozulić, Numerical solving of initial-value problems by Rbf basis functions, Struct. Eng. Mech. 14 (3) (2002) 263–285.
[34] V. Kozulić, B. Gotovac, Numerical analyses of 2D problems using fupn (x, y) basis functions, Int. J. Eng. Modell. 13 (1–2) (2000) 7–18.
[35] H. Gotovac, R. Andricevic, B. Gotovac, V. Kozulić, M. Vranjes, An improved collocation method for solving the Henry problem, J. Contam. Hydrol. 64 (1) (2003) 129–149.
[36] H. Gotovac, V. Cvetkovic, R. Andricevic, Adaptive fup multi-resolution approach to flow and advective transport in highly heterogeneous porous media: Methodology, accuracy and convergence, Adv. Water Resour. 32 (6) (2009) 885–905.
[37] L. Malenica, H. Gotovac, G. Kamber, S. Simunovic, S. Allu, 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 10 (12) (2018).
[38] G. Kamber, H. Gotovac, V. Kozulić, L. Malenica, B. Gotovac, Adaptive numerical modeling using the hierarchical Fup basis functions and control volume isogeometric analysis, Internat. J. Numer. Methods Fluids 92 (10) (2020) 1437–1461.
[39] C. Giannelli, B. Jüttler, H. Speleers, Strongly stable bases for adaptively refined multilevel spline spaces, Adv. Comput. Math. 40 (2) (2014) 459–490.
[40] R. Kraft, Adaptive and Linearly Independent Multilevel B-Splines, SFB 404, Geschäftsstelle, 1997.
[41] L. Malenica, Numerical Modeling Based on Spline Basis Functions: Application to Groundwater Flow Modeling in Karst Aquifers and Advection Dominated Problems (Ph.D. thesis), University of Split,Faculty of Civil Engineering, Architecture and Geodesy, Split, 2019.
[42] D. Schillinger, J.A. Evans, A. Reali, M.A. Scott, T.J. Hughes, Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations, Comput. Methods Appl. Mech. Engrg. 267 (February) (2013) 170–232.
[43] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Pub. Corp., 1980, p. 197.
[44] R.W. Johnson, Higher order B-spline collocation at the Greville abscissae, Appl. Numer. Math. 52 (1) (2005) 63–75.

[45] H. Gotovac, R. Andricevic, B. Gotovac, Multi-resolution adaptive modeling of groundwater flow and transport problems, Adv. Water Resour. 30 (5) (2007) 1105–1126.
[46] 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.
[47] W. Rachowicz, J. Oden, L. Demkowicz, Toward a universal h-p adaptive finite element strategy part 3. design of h-p meshes, Comput. Methods Appl. Mech. Engrg. 77 (1–2) (1989) 181–212.
[48] J. Oden, A. Patra, A parallel adaptive strategy for hp finite element computations, Comput. Methods Appl. Mech. Engrg. 121 (1) (1995) 449–470.
[49] W.F. Mitchell, A collection of 2D elliptic problems for testing adaptive grid refinement algorithms, Appl. Math. Comput. 220 (2013) 350–364.
[50] C. Swaminathan, V. Voller, Streamline upwind scheme for control-volume finite elements, part i. formulations, Numer. Heat Transfer B 22 (1) (1992) 95–107.
[51] C. Swaminathan, V. Voller, S. Patankar, A streamline upwind control volume finite element method for modeling fluid flow and heat transfer problems, Finite Elem. Anal. Des. 13 (2) (1993) 169–184.