Chapter 15
Isogeometric Analysis of Navier-Stokes Flow Using Locally Refinable B-Splines

Peter Nørtoft and Tor Dokken

15.1 Introduction

In recent years, isogeometric analysis (IGA) has gained increasing interest as a numerical method for solving engineering problems within fluid mechanics [7, 15]. This popularity may be attributed to its ability to model complex geometries exactly, to approximate the flow fields with arbitrarily high degree of smoothness, and to couple the geometric modeling and the flow analysis into one single framework. At the very heart of the isogeometric paradigm is the unification of finite element analysis (FEA) for solving the governing flow equations, and computer-aided design (CAD) for modeling the geometry of the flow domain.

One of the early challenges of the isogeometric paradigm was the concept of *local refinement*. To resolve the flow around some obstacle, say, a *fine* approximation of the field is often required in the boundary layer close to the obstacle, whereas a *coarse* approximation suffices in the far-field away from the obstacle. Here, efficient local refinement is of paramount importance: a coarse representation of the entire flow domain leaves the boundary layer unresolved, and the results are useless; a fine representation of the entire flow domain yields a fatal blow-up in the number of degrees of freedom, rendering the approach useless. This is sketched in Fig. 15.1. Although well-established within FEA, efficient local refinement was initially prohibited in IGA, primarily owing to the tensor-product structures inherited from CAD, that only allowed for global refinements, or a "poor man's" local refinement

P. Nørtoft (✉)
DTU Compute, Technical University of Denmark, Matematiktorvet 303B, DK-2800 Kgs. Lyngby, Denmark
e-mail: penn@dtu.dk

T. Dokken
SINTEF ICT, P.O. Box 124 Blindern, N-0314 Oslo, Norway
e-mail: Tor.Dokken@sintef.no

T. Dokken, G. Muntingh (eds.), *SAGA – Advances in ShApes, Geometry, and Algebra*, Geometry and Computing 10, DOI 10.1007/978-3-319-08635-4_15

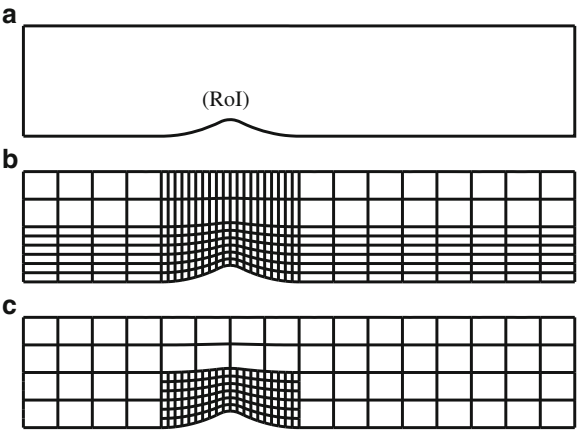

Fig. 15.1 Difference between global (**b**) and local (**c**) refinement of a region of interest (*RoI*) inside a flow domain (**a**)

through patching. Several ways to achieve local refinement in an isogeometric setting have been proposed since the birth of IGA, including in particular T-splines [2, 9, 18], and hierarchical splines [14, 19].

In this work, we study a novel approach to local refinement in the context of fluid mechanics, namely through the recently proposed locally refinable (LR) B-splines [4, 8, 16]. We investigate two families of locally refined B-spline discretizations of the flow variables for solving the mixed formulation of the stationary, incompressible Navier-Stokes equations in 2 dimensions using IGA. These two LR B-spline discretizations are motivated by recent results for ordinary tensor-product B-spline discretizations of the flow variables [3, 5, 17]. Our focus here is primarily on *how* to refine the flow discretizations, as dictated by a refinement strategy, and not on *whether* or *where* to refine, as dictated by some error estimator. In the context of a full-blown adaptive mesh refinement setup, of course, one needs to address all of these issues.

The outline of the rest of the work is as follows. We start by introducing the governing Navier-Stokes equation in Sect. 15.2, after which we introduce the LR B-splines as refinable building blocks for solving it in Sect. 15.3. Then, in Sect. 15.4, we briefly outline the general isogeometric framework, and in Sect. 15.5 we present the two flow discretizations. Numerical examples are presented in Sect. 15.6 with focus on the numerical stability, error convergence during refinements, and benchmarking. Finally, we summarize our findings and outline future work.

15.2 Navier-Stokes Equation

We start by introducing the steady-state, incompressible Navier-Stokes equation. This is the equation that governs the motion of fluids under sufficiently simple conditions.

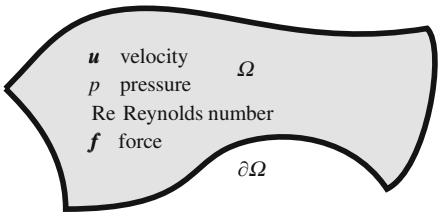

Fig. 15.2 A fluid contained in a flow domain

We consider a fluid in a 2-dimensional domain Ω as depicted in Fig. 15.2. We assume the fluid is isothermal, i.e., at constant temperature, incompressible, i.e., the density is constant, and Newtonian, i.e., the stress and the strain rate are linearly related through the viscosity, which is also assumed to be constant. Finally, we assume that the flow is stationary, i.e., time-independent. The state of the fluid is then given by the velocity and the pressure, and these are governed by the Navier-Stokes and mass-continuity equations:

$$(\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} + \nabla p - \frac{1}{\mathrm{Re}} \Delta \boldsymbol{u} + \boldsymbol{f} = \boldsymbol{0}, \tag{15.1a}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{15.1b}$$

Here $\boldsymbol{u} = (u, v)$ is the velocity, p is the pressure, and $\boldsymbol{f}$ are additional body forces acting on the fluid, all in dimensionless form, while $\mathrm{Re} := \rho U L / \mu$ is the Reynolds number, where ρ is the density, μ is the viscosity, and U and L are characteristic velocity and length scales of the problem, respectively. In somewhat loose terms, Re is a measure of the degree of nonlinearity, and hence complexity, of the flow problem. We will consider primarily *laminar* flows with $\mathrm{Re} \lesssim 2{,}000$, as opposed to *turbulent* flows with $\mathrm{Re} \gtrsim 2{,}000$.

These equations govern the flow in the interior of the domain Ω, and they must be augmented by suitable boundary conditions. Here, we consider so-called full Dirichlet boundary conditions, along with a condition on the average pressure:

$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{on } \partial\Omega, \tag{15.2a}$$

$$\int_\Omega p \, \mathrm{d}\boldsymbol{x} = p_0. \tag{15.2b}$$

15.3 Locally Refinable B-Splines

We now proceed to give a brief introduction to Locally Refinable (LR) B-splines. First, the underlying B-splines are introduced, after which their LR extensions are described in 2 dimensions. The intention is to give the reader an intuitive

understanding of LR B-splines, and thereby pave the road for using them to solve the Navier-Stokes equation in the next sections. For a more rigorous introduction to LR B-splines, we refer to [8].

15.3.1 B-Splines

We start by recalling the definition of univariate B-splines. Given a polynomial degree $d \geq 0$ and a non-decreasing sequence of $d+2$ knots $\boldsymbol{\eta} = \{\eta_1, \ldots, \eta_{d+2}\}$, a univariate B-spline $B[\boldsymbol{\eta}] : \mathbb{R} \to \mathbb{R}$ is defined recursively through:

$$B[\boldsymbol{\eta}](\xi) = \frac{\xi - \eta_1}{\eta_{d+1} - \eta_1} B[\eta_1, \ldots, \eta_{d+1}](\xi) + \frac{\eta_{d+2} - \xi}{\eta_{d+2} - \eta_2} B[\eta_2, \ldots, \eta_{d+2}](\xi), \tag{15.3a}$$

starting with

$$B[\eta_i, \eta_{i+1}](\xi) = \begin{cases} 1 & \text{if } \eta_i \leq \xi < \eta_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{15.3b}$$

for $i = 1, \ldots, d+1$, and where terms with zero denominator are defined to be zero. A univariate B-spline is thus a piecewise polynomial function of degree d. Its support is the interval $[\eta_1, \eta_{d+2}]$, and the continuity across a knot η_i is $d - m$, where m denotes the multiplicity of the knot η_i.

Multivariate B-splines can be formed quite naturally through tensor-product structures based on multiple univariate B-splines. In two parametric dimensions we have the following: Given two polynomial degrees d_i and two non-decreasing sequences of $d_i + 2$ knots $\boldsymbol{\eta}_i = \{\eta_{i,1}, \ldots, \eta_{i,d_i+2}\}$ for $i = 1, 2$, a bivariate tensor-product B-spline $B[\boldsymbol{\eta}_1, \boldsymbol{\eta}_2] : \mathbb{R}^2 \to \mathbb{R}$ is given by:

$$B[\boldsymbol{\eta}_1, \boldsymbol{\eta}_2](\xi_1, \xi_2) = B[\boldsymbol{\eta}_1](\xi_1)\, B[\boldsymbol{\eta}_2](\xi_2). \tag{15.4}$$

The support of a tensor-product B-spline is the rectangle $[\eta_{1,1}, \eta_{1,d_1+2}] \times [\eta_{2,1}, \eta_{2,d_2+2}]$. Figure 15.3 illustrates the construction of a bivariate tensor-product B-spline from two univariate B-splines. In the example shown, the two univariate B-splines that form the bivariate bi-quadratic tensor-product B-spline are constructed from the polynomial degrees $d_1 = d_2 = 2$, and the knot vectors $\boldsymbol{\eta}_1 = \{0, 1/3, 2/3, 1\}$ and $\boldsymbol{\eta}_2 = \{0, 0, 1/3, 2/3\}$. The knot vectors $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ are extracted from two identical global knot vectors $\overline{\boldsymbol{\eta}}_1 = \overline{\boldsymbol{\eta}}_2 = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$. Each of these gives rise to 5 univariate B-splines, resulting in a total of 25 bivariate B-splines.

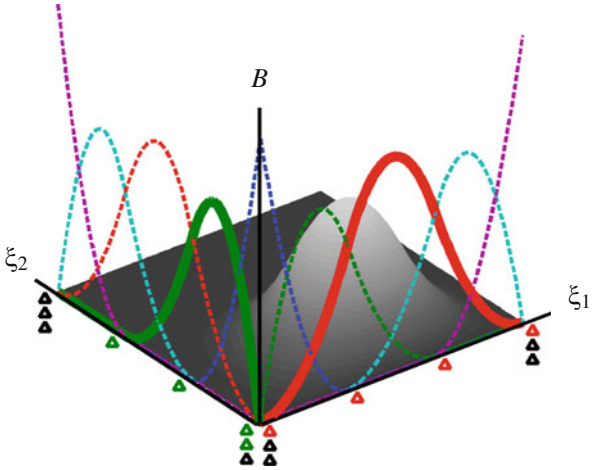

Fig. 15.3 Construction of a bi-quadratic tensor-product B-spline (*grayscale surface*) from two univariate quadratic B-splines (*lines in bold*) with given knot vectors (*colored triangles*)

15.3.2 LR Mesh

Locally refinable B-splines rest naturally on B-splines. They include the tensor-product B-splines introduced above as a special case, but in addition provide a much more "local" framework for multivariate B-splines. Before understanding the notion of an LR B-spline, we must, however, understand the notion of an LR *mesh*.

Just like any tensor-product B-spline is formed on a tensor-product mesh, as sketched in Fig. 15.3, any LR B-spline is formed on an LR mesh. A mesh holds information about essentially two things: the *location* and the *multiplicity* of all knots. For tensor-product B-splines, as defined in (15.4), the mesh is specified simply through the two global knot vectors $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$. As the LR mesh cannot be defined by global knot vectors, it has to be defined by its N_m knotline segments and their multiplicities. Each knotline segment is defined by a start point and an end point. When the multiplicities of all knotline segments in a given mesh are all set to 1, we will refer to it as the *grid*.

An LR mesh is a special kind of a mesh. The life of an LR mesh has two different stages:

1. The initial tensor-product construction
2. The subsequent local refinements

From the beginning, the LR mesh is constructed simply as a standard tensor-product mesh. An example is shown in Fig. 15.4a. The global knot vectors are $\boldsymbol{\eta}_1 = \boldsymbol{\eta}_2 = \{0, 0, 0, 1/5, 2/5, 3/5, 4/5, 1, 1, 1\}$. Using this tensor-product mesh as a starting point, the mesh is then refined by subsequently inserting knotline segments into it, such that the mesh remains a *box-partition*, i.e., consists of a collection of quadrilaterals throughout each refinement. Let us assume that we want to refine the highlighted box in Fig. 15.4a. First, Fig. 15.4b shows the result of a usual tensor-product refinement, i.e., when inserting one vertical and one horizontal knotline

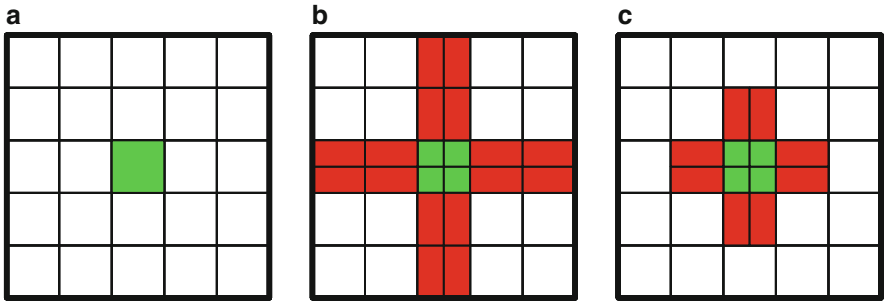

Fig. 15.4 Three different meshes: the initial tensor-product mesh (**a**), a globally refined tensor-product mesh (**b**), and a locally refined mesh (**c**). Multiplicities are three on the boundary and one elsewhere

segments through the box of interest and letting these extend all the way to the boundaries. This clearly identifies the problem with the tensor-product approach; along with the actual box of interest, *all* boxes towards the boundaries are also refined. The LR mesh, however, allows for much more local refinements. An example is shown in Fig. 15.4c. Here, we have inserted two short knotline segments, one vertical and one horizontal. As we shall see below, the knotline segments must be specified in such a way that each of them splits an LR B-spline. This is part of the reason why the knotline segments extend outside the highlighted box, and why the neighboring boxes are still refined. For consistency with the notation in different dimensions and settings, we usually refer to the knotline segments as *mesh-rectangles* and to the boxes as *elements*. These are central ingredients of an LR mesh.

15.3.3 LR B-Splines

With the LR mesh introduced, we now turn to the LR B-splines. An LR mesh gives rise to a number of LR B-splines, just like a tensor-product mesh gives rise to a number of tensor-product B-splines, cf. Fig. 15.3. By inserting local mesh-rectangles into an LR mesh, we enrich the space of B-splines living on it, and this in a much more local sense than by inserting global knots into a tensor-product mesh.

To illustrate this, let us return to the example from above, and assume that the polynomial degrees are $d_1 = d_2 = 2$. Figure 15.5a sketches the B-splines on the initial tensor-product mesh in Fig. 15.4a. To be more precise, the plot shows the Greville abscissae of the B-splines, which for any B-spline is just the average of the d_i central knots $\{\eta_{i,2}, \ldots, \eta_{i,d_i+1}\}$ in each parameter direction $i = 1, 2$, and thus a condensed way of visualizing the functions. By making the tensor-product refinements as in Fig. 15.4b, we end up with the B-splines sketched in Fig. 15.5b. The global nature of the refinement is again evident, as B-splines appear also

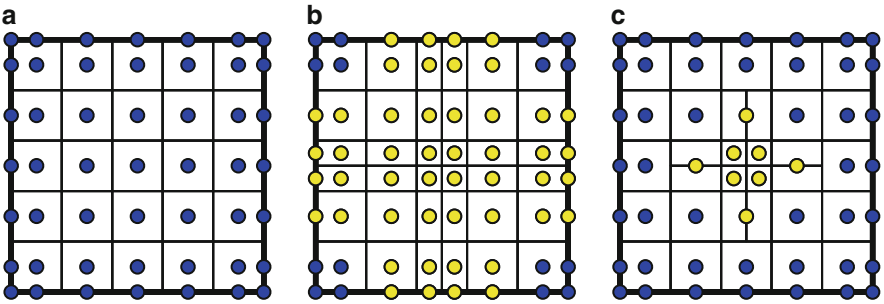

Fig. 15.5 Greville abscissae of all bi-quadratic B-splines on three different meshes: the initial tensor-product mesh (**a**), a globally refined tensor-product mesh (**b**), and a locally refined mesh (**c**). Multiplicities are three on the boundary and one elsewhere

away from the element of interest. On the other hand, by inserting the local mesh rectangles as in Fig. 15.4c, we obtain the LR B-splines shown in Fig. 15.5c. All new LR B-splines appear in close proximity of the element of interest. We mention in passing that, as the tensor-product meshes in Fig. 15.4a, b are indeed also LR meshes, the tensor-product B-splines in Fig. 15.5a, b are also LR B-splines.

But how do the new LR B-splines in Fig. 15.5c actually come about from the LR mesh in Fig. 15.4c? To answer this question, we consider Fig. 15.6. We first consider the insertion of the vertical mesh-rectangle. Remembering that all the tensor-product B-splines have support over 3×3 knot spans, we easily see that there are exactly three B-splines for which the mesh-rectangle traverses their entire support in the vertical direction, as indicated in Fig. 15.6a (left). These are the coarse functions that are to be refined. The resulting functions after this first refinement are shown in Fig. 15.6a (right). Note that each of the new LR B-splines has exactly the same underlying knot structure as a standard tensor-product B-spline. Next, we consider the insertion of the horizontal mesh-rectangle. Now, there are four LR B-splines for which the mesh-rectangle traverses its entire support in the vertical direction, as indicated in Fig. 15.6b (left). These are now the coarse functions that are to be refined. The resulting functions after this second refinement are shown in Fig. 15.6b (right). The order of insertion turns out to play no role. Thus, we may as well insert the horizontal mesh-rectangle first, and then vertical afterward; the final outcome will be the same.

LR B-splines possess many of the properties that standard tensor-product B-splines do. They are piecewise polynomial functions, they have compact support, and they form a partition of unity, i.e., they sum to one in all points, a property ensured through a simple scaling of each of the functions. Linear independence of a set of LR B-splines is not guaranteed per se. This is crucial when using them to solve equations like the Navier-Stokes equation. To ensure linear independence, the functions either have to be established through refinement schemes known a priori to result in linear independent LR B-splines, or they must be tested a posteriori through, e.g., a so-called peeling algorithm [8]. In Sect. 15.5 below, we shall return

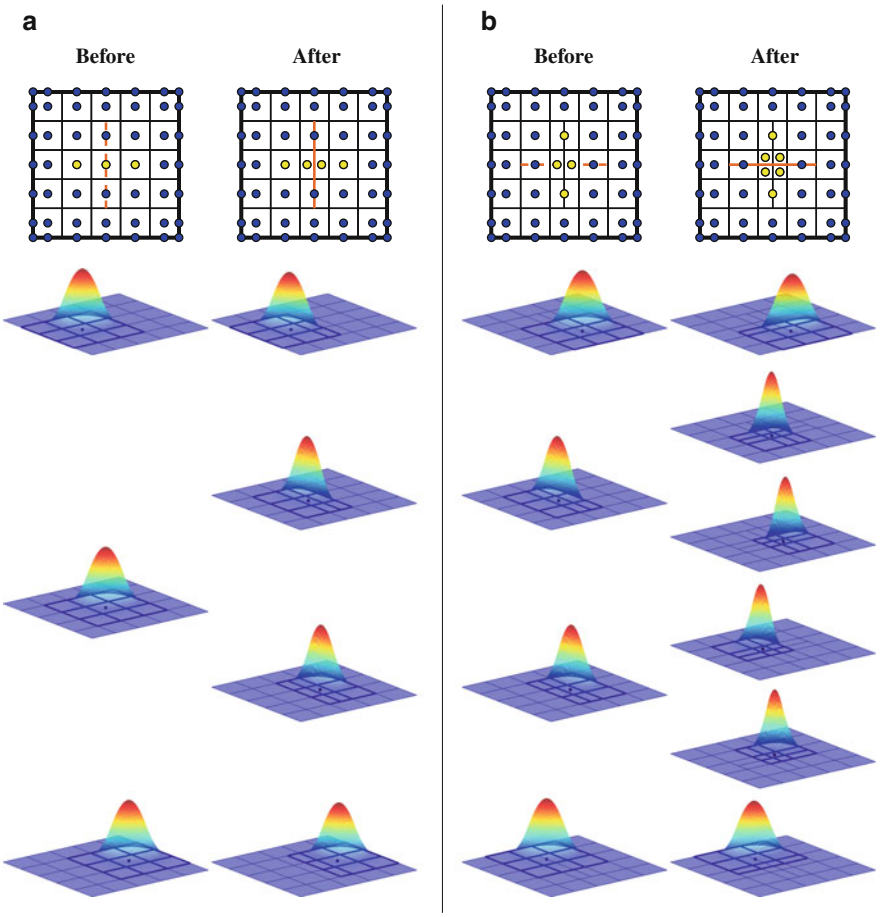

Fig. 15.6 B-splines before (*left*) and after (*right*) insertion of the vertical mesh rectangle (**a**) and the horizontal mesh rectangle (**b**). LR B-splines with Greville abscissae shown in *blue* are unaffected by the insertion, whereas LR B-splines with Greville abscissae shown in *yellow* are removed or inserted as a result of the insertion

to the construction of LR B-splines in the context of approximation of flow pressure and velocities for solving the Navier-Stokes equations.

15.4 Isogeometric Analysis

In this section, we outline the fundamentals of how to solve the Navier-Stokes and continuity equations (15.1) in an isogeometric framework based on LR B-splines. This essentially involves three ingredients: a parametrization of the geometry, a

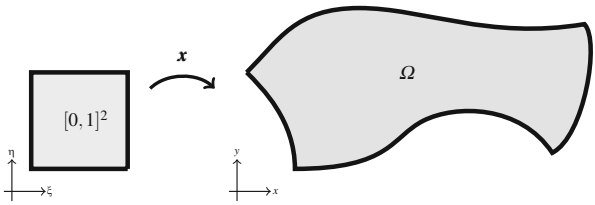

Fig. 15.7 A parametrization of the flow domain Ω

discretization of the flow variables for the Galerkin projection, and the weak form of governing equations.

First, we construct a parametrization of the flow domain Ω, as sketched in Fig. 15.7. We take the parameter domain $\hat{\Omega}$ as the unit square and use bivariate LR B-splines as basis functions. The parametrization $\boldsymbol{x} : [0,1]^2 \to \mathbb{R}^2$ reads:

$$\boldsymbol{x}(\xi_1, \xi_2) = \sum_{i=1}^{N_g} \underline{\boldsymbol{x}}_i \mathcal{P}_i^g(\xi_1, \xi_2), \tag{15.5}$$

where $\underline{\boldsymbol{x}}_i$ are the control points, $\mathcal{P}_i^g$ are the LR B-splines, N_g is the number of LR B-splines and control points, and the superscript g indicates that the functions refer to the geometry parametrization.

Next, we seek approximations of the velocity $\boldsymbol{u}_h : [0,1]^2 \to \mathbb{R}^2$ and pressure $p_h : [0,1]^2 \to \mathbb{R}$ as linear combinations of LR B-splines, just like the geometry representation in Eq. (15.5) above:

$$\boldsymbol{u}_h(\xi_1, \xi_2) = \sum_{i=1}^{N_u} \underline{\boldsymbol{u}}_i \mathcal{P}_i^u(\xi_1, \xi_2), \qquad p_h(\xi_1, \xi_2) = \sum_{i=1}^{N_p} \underline{p}_i \mathcal{P}_i^p(\xi_1, \xi_2). \tag{15.6}$$

Here, $\mathcal{P}_i^u$ and $\mathcal{P}_i^p$ denote the LR B-spline basis functions for the velocity and pressure, N_u and N_p are the number of velocity and pressure B-splines, while $\underline{\boldsymbol{u}}$ and $\underline{p}$ are the unknown control variables for the velocity and pressure, respectively. For simplicity, we discretize the two components of the velocity identically. The approximations in Eqs. (15.6) are defined in parameter space, whereas the governing equations (15.1) are posed in physical space. To evaluate the pressure in physical space $p : \Omega \to \mathbb{R}$, we use the inverse of the geometry parametrization as $p \circ \boldsymbol{x}^{-1}$. To evaluate the velocity in physical space $\boldsymbol{u} : \Omega \to \mathbb{R}^2$, we simply map each component as a scalar $\boldsymbol{I}\boldsymbol{u} \circ \boldsymbol{x}^{-1}$, where $\boldsymbol{I}$ is the identity map. Note here that, with abuse of notation, we use p and $\boldsymbol{u}$ to denote the pressure and the velocity, respectively, both over the physical space and over the parameter space. In Sect. 15.5 below, we describe in greater detail how to construct the LR B-spline discretizations of the pressure and velocity fields.

Finally, we cast the governing equations (15.1) and (15.2) into their weak form, which reads: find $(\boldsymbol{u}, p)$ with $\boldsymbol{u} = \boldsymbol{g}$ on $\partial\Omega$ and $\int_\Omega p \, \mathrm{d}\boldsymbol{x} = 0$ such that

$$0 = \int_\Omega \left(\left(\frac{1}{\mathrm{Re}} \nabla v_k + v_k \, \boldsymbol{u} \right) \cdot \nabla u_k - \left(p \nabla v_k + v_k \, \boldsymbol{f} \right) \cdot \boldsymbol{e}_k \right) \mathrm{d}\boldsymbol{x}, \quad k = 1, 2, \tag{15.7a}$$

$$0 = \int_\Omega q \, (\nabla \cdot \boldsymbol{u}) \, \mathrm{d}\boldsymbol{x} \tag{15.7b}$$

for all $(\boldsymbol{v}, q)$ with $\boldsymbol{v} = \boldsymbol{0}$ on $\partial\Omega$, where we have used integration by parts in the derivation. Here, $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ are the standard Cartesian basis vectors, and the functions p and q must be square-integrable, while $\boldsymbol{u}$ and $\boldsymbol{v}$ as well as all their first-order derivatives must be square-integrable.

By using the LR B-splines approximations (15.6) as test and weight functions in the weak equations (15.7), and pulling the integrals back to the parameter domain based on the parametrization (15.5), a non-linear system of equations of the form $\boldsymbol{M}(\boldsymbol{U})\,\boldsymbol{U} = \boldsymbol{F}$ may be obtained:

$$\begin{bmatrix} \frac{1}{\mathrm{Re}}\boldsymbol{K} + \boldsymbol{C}(\underline{\boldsymbol{u}}) & \boldsymbol{0} & -\boldsymbol{G}_1^T \\ \boldsymbol{0} & \frac{1}{\mathrm{Re}}\boldsymbol{K} + \boldsymbol{C}(\underline{\boldsymbol{u}}) & -\boldsymbol{G}_2^T \\ \boldsymbol{G}_1 & \boldsymbol{G}_2 & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{u}}_1 \\ \underline{\boldsymbol{u}}_2 \\ \underline{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \boldsymbol{0} \end{bmatrix}, \tag{15.8}$$

where

$$\boldsymbol{K}_{i,j} = \int_{[0,1]^2} \nabla^T \mathcal{P}_i^u \, \boldsymbol{J}^{-1} \, \boldsymbol{J}^{-T} \, \nabla \mathcal{P}_j^u \, \det\left(\boldsymbol{J}\right) \mathrm{d}\boldsymbol{\xi}, \tag{15.9a}$$

$$\boldsymbol{C}_{i,j}(\boldsymbol{u}) = \int_{[0,1]^2} \mathcal{P}_i^u \, \boldsymbol{u}^T \, \boldsymbol{J}^{-T} \, \nabla \mathcal{P}_j^u \, \det\left(\boldsymbol{J}\right) \mathrm{d}\boldsymbol{\xi}, \tag{15.9b}$$

$$\boldsymbol{G}_{k_{i,j}} = \int_{[0,1]^2} \mathcal{P}_i^p \, \boldsymbol{e}_k^T \, \boldsymbol{J}^{-T} \, \nabla \mathcal{P}_j^u \, \det\left(\boldsymbol{J}\right) \mathrm{d}\boldsymbol{\xi}, \quad k = 1, 2 \tag{15.9c}$$

$$\boldsymbol{F}_{k_i} = \int_{[0,1]^2} \mathcal{P}_i^u \, \boldsymbol{e}_k^T \, \boldsymbol{f} \, \det\left(\boldsymbol{J}\right) \mathrm{d}\boldsymbol{\xi}, \quad k = 1, 2 \tag{15.9d}$$

where $\boldsymbol{J}_{i,j} := \partial x_i / \partial \xi_j$ is the Jacobian matrix of the parametrization (15.5).

To solve the governing partial differential equations (15.1) using LR B-spline based isogeometric analysis, we thus need to solve the system of algebraic equations (15.8). To do this, we evaluate the integrals in (15.9) using Gaussian quadrature, and use an iterative Newton-Raphson solver. The Dirichlet boundary conditions (15.2a) on the velocity are enforced strongly, perhaps only in an approximative sense, while the additional condition (15.2b) on the mean pressure is imposed weakly through a least-square approach.

15.5 Flow Discretizations

We now proceed to introduce two families of discretizations of the pressure and velocity fields based on LR B-splines, thus substantiating the flow approximations (15.6) introduced above. We refer to these as the *Taylor-Hood* and *multigrid* families.

As indicated in Sect. 15.3, LR B-splines are characterized by a high degree of flexibility. In loose terms, we can play around with the two polynomial degrees, the two vectors of unique, global knots, and the multiplicities of each of the small mesh-rectangles along each of the global knots. When discretizing the flow variables, however, we narrow the scope slightly. As explained below, we choose to characterize the LR B-spline flow discretizations simply through one polynomial degree d, from which we then specify the polynomial degrees, regularities (smoothnesses) α, and levels of refinement r of both the pressure and the velocity fields.

The construction of both the Taylor-Hood and the multigrid flow discretizations comprises the same two stages as outlined in Sect. 15.3:

1. The tensor-product initialization
2. The subsequent local refinements

For now, we assume that a tensor-product spline representation $\boldsymbol{x}$ of the geometry is provided to us as input. We shall relax this assumption later.

In the initialization of both the Taylor-Hood and the multigrid discretization, we construct tensor-product spline representations of the velocity $\boldsymbol{u}_h$ and the pressure p_h in the usual fashion [5,17]. For both $\boldsymbol{u}_h$ and p_h, we take the global knot vectors $\boldsymbol{\eta}_i$ to be open, i.e., the multiplicity of the first and last knots are $d_i + 1$, and we take all interior knots to have the same multiplicity $m_i < d_i$ for the parametric dimensions $i = 1, 2$. Furthermore, in order to limit the number of parameters, and thus simplify the notation in the following, we assume for both $\boldsymbol{u}_h$ and p_h that the degree and the regularity are the same in both parametric dimensions, i.e., $d_1 = d_2 = d$ and $\alpha_1 = \alpha_2 = \alpha$, although these assumptions are not strictly required.

From the given tensor-product spline representation $\boldsymbol{x}$ of the geometry, we now choose a degree d and construct the tensor-product spline discretization p_h of the pressure using the same grid as for the geometry $\boldsymbol{x}$, with the degree $d^p = d$, full regularity $\alpha^p = d-1$, and no refinements $r^p = 0$. The assumption of full regularity is not strictly required, but again it limits the number of parameters.

Next, we construct the discretization $\boldsymbol{u}_h$ of the velocity from p_h through one of two approaches: In the Taylor-Hood approach, we increase the polynomial degree $d^u = d^p + 1$, fix the regularity $\alpha^u = \alpha^p = d-1$ by increasing the knot multiplicity, and keep the refinement level $r^u = 0$. In the multigrid approach, we increase both the polynomial degree $d^u = d^p + 1$, the regularity $\alpha^u = \alpha^p + 1 = d$, and the refinement level through insertion of, say, one additional knot in each regular knot span, such that $r^u = 1$. These tensor-product initializations are illustrated in the top of Fig. 15.8 for both the Taylor-Hood and multigrid discretization.

With the tensor-product initialization in place, we turn to the subsequent local refinements. For both the Taylor-Hood and the multigrid discretizations, we base

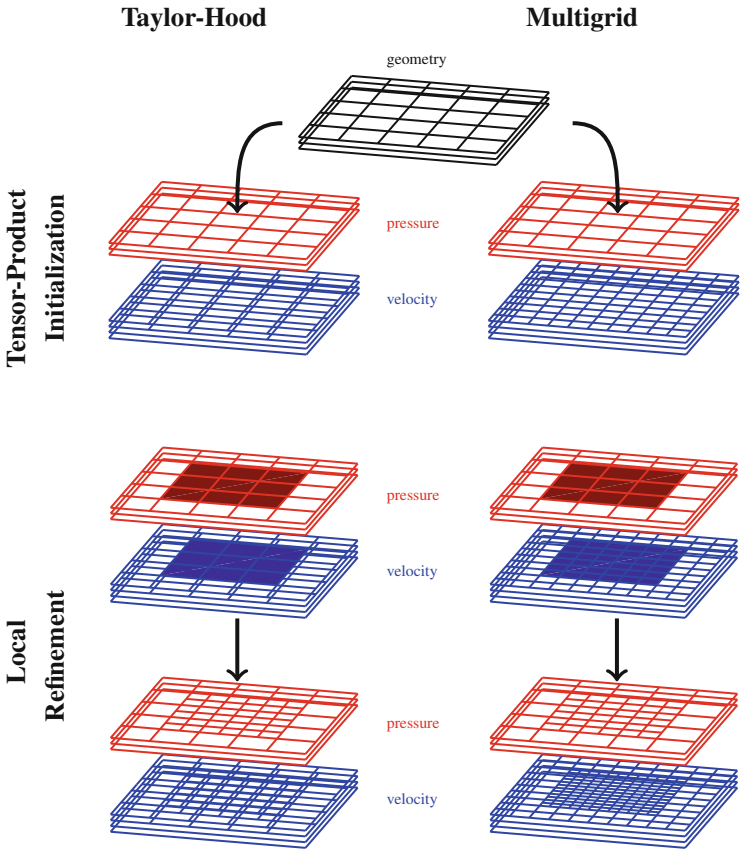

Fig. 15.8 Illustration of the tensor-product initialization (*top*) and subsequent local refinements (*bottom*) following the Taylor-Hood (*left*) and multigrid (*right*) approaches for degree $d = 2$

these on the structured mesh approach [16] with constant multiplicities $m^f = d^f - \alpha^f$ for both fields $f \in \{\boldsymbol{u}, p\}$. The idea, when refining according to the structured mesh approach, is to reason in terms of basis functions. First we choose a set of LR B-splines to refine, and then we insert new mesh-rectangles with the same multiplicity as the existing ones, in such a way that each regular span of mesh-rectangles whose two mesh-rectangles are both contained in the same LR B-spline within the specified set is split uniformly into n new spans of mesh-rectangles. Here, we will for simplicity use $n = 2$.

Now, when refining the flow discretization according to the structured mesh approach, we have both the LR B-splines of the *velocity* discretization and the LR B-splines of the *pressure* discretization to take into consideration. We choose to specify refinements for *both* the pressure and the velocity through a set of *pressure* LR B-splines. For both the pressure and the velocity discretization, we follow the structured mesh approach and base the insertions of mesh-rectangles on the support

Table 15.1 Characteristics of the two families of LR B-spline discretizations of the flow fields for given degree d

	Pressure			Velocity		
Discretization	Degree	Regularity	Refinement	Degree	Regularity	Refinement
Taylor-Hood	d	$d-1$	0	$d+1$	$d-1$	0
Multigrid	d	$d-1$	0	$d+1$	d	1

of a specified collection of pressure LR B-splines. This gives us *two* collections of mesh-rectangles; one collection is inserted in the *pressure* mesh, and one in the *velocity* mesh. These two collections of mesh-rectangles either differ in the number of mesh-rectangles or in the multiplicity of the mesh-rectangles, depending on the discretization in question. The number of pressure and velocity mesh-rectangles is dictated by the refinement level (e.g., N for both fields for the Taylor-Hood, and N and $n(N+1)-1$ for the pressure and velocity field, respectively, for the multigrid). The multiplicity of the pressure and velocity mesh-rectangles is given by the regularity (1 and 2, respectively, for the pressure and velocity fields for the Taylor-Hood, and 1 for both fields for the multigrid). These local refinements procedures are illustrated in the bottom of Fig. 15.8 for both the Taylor-Hood and the multigrid discretization, and their characteristics are summarized in Table 15.1.

It should be emphasized that the number of velocity elements for the multigrid discretization is larger than the number of velocity elements for the Taylor-Hood discretization by a factor of $(r+1)^2$. Since integrals in the matrices in Eq. (15.9) are evaluated based on elements, this makes the multigrid discretization computationally more expensive than the Taylor-Hood discretization.

We conclude by noting that the assumption of the geometry being discretized by a tensor-product spline can easily be relaxed. One obvious way to achieve this, while still ensuring that the geometry grid is contained within the velocity and pressure grids, is by assuming instead that the geometry is represented by an LR spline, that was initialized as a tensor-product spline of some degree based on open knot vectors with single interior knots, and whose subsequent refinements were all obtained using the structured mesh approach. In this case, we may construct the Taylor-Hood and multigrid flow discretization as before, except we must first set the initial tensor-product pressure discretization equal to the initial tensor-product geometry discretization, construct the tensor-product velocity discretization as before, and then go through the exact same steps of local refinements of the flow discretizations as for the final geometry discretization.

15.6 Numerical Examples

In this section, we test the LR B-spline flow discretizations introduced above in different numerical examples. Through these, we investigate the stability of the discretizations, we study their ability to reproduce an analytical solution, and we examine their performance on a standard benchmark problem.

15.6.1 Wall-Driven Annular Cavity: Stability

In the first example, we investigate the stability of the discretizations. We consider the problem outlined in Fig. 15.9a, in which a fluid is contained in an annular cavity, approximated by cubic B-splines. We are interested in the flow problem in the limit of small Reynolds numbers. Hence, we neglect the nonlinear term in the Navier-Stokes equation (15.1a), which then reduces to the Stokes equation. The sliding movement of the lower circular part of the boundary induces a rotating flow in the interior, while singularities in the pressure field form in the two lower corners, as indicated in Fig. 15.9b. Unstable discretizations manifest themselves qualitatively through spurious oscillations in the pressure field. Quantitatively, they violate the so-called inf–sup condition:

$$\inf_{p_h} \sup_{\boldsymbol{u}_h} \frac{\int_\Omega p_h \nabla \cdot \boldsymbol{u}_h \, \mathrm{d}\boldsymbol{x}}{\|p_h\|_{L^2} \|\boldsymbol{u}_h\|_{H^1}} \geq \beta > 0, \tag{15.10}$$

where the constant β is independent of the mesh resolution h.

In the following, we perform a series of numerical tests of whether the discretizations fulfill the inf–sup condition (15.10), i.e., whether they are stable or not, based on the rather degenerate problem sketched in Fig. 15.9. For each discretization, we refine a coarse mesh repeatedly, and estimate the value of β in each step [1, 6]. We follow two different schemes for choosing which LR B-splines to refine: by the first scheme, we refine the LR B-splines with support in one of the two lower corners, where the pressure singularities occur. By the second scheme, we refine a number of randomly chosen LR B-splines. Examples of the pressure grids produced by these two schemes are shown in Fig. 15.9c, d, respectively.

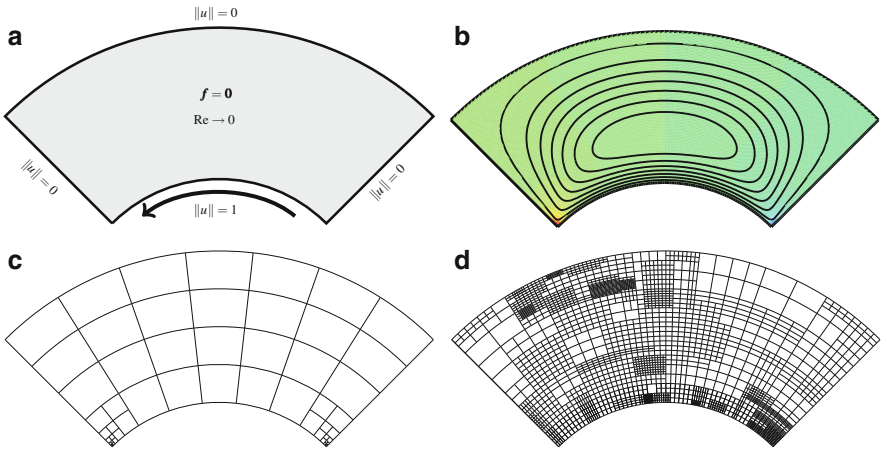

Fig. 15.9 Wall-driven annular cavity: problem setup (**a**), streamlines and pressure field (**b**), and pressure grid examples (**c** and **d**)

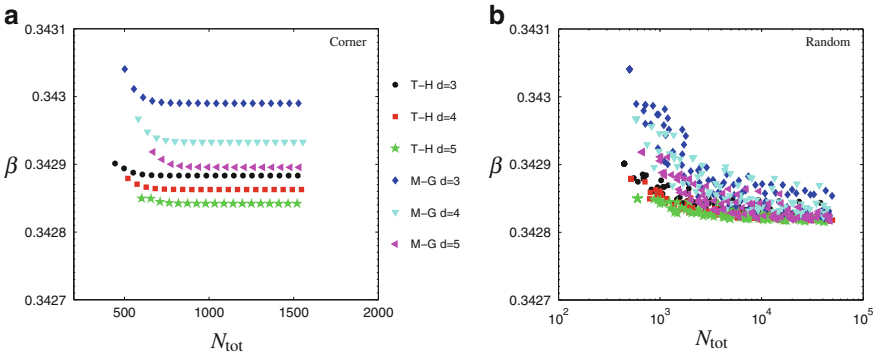

Fig. 15.10 Wall-driven annular cavity: Numerical estimates of the inf-sup "constant" β as a function of total number of analysis degrees of freedom N_{tot} for different discretizations (*T-H*: Taylor-Hood and *M-G* multigrid) based on corner function refinements (**a**) and random function refinements (**b**)

The estimated values of β for the Taylor-Hood and multigrid discretizations based on polynomial degrees 3–5, corresponding to degrees 3–5 for the pressure and 4–6 for the velocity, respectively, are shown in Fig. 15.10. A given discretization is said to pass the inf–sup test, if the estimated value of β does *not* tend to zero as the number of degrees of freedom is increased; if the value *does* tend to zero, the discretization fails the test. From these results, we are led to conclude that all the investigated discretizations among both the Taylor-Hood and the multigrid families pass the test. We emphasize that these conclusions are drawn on a (large but) finite number of numerical tests, and not on mathematical proofs. Furthermore, in addition to the Taylor-Hood and multigrid discretizations shown here, a discretization known to be unstable, based on a bi-quadratic pressure approximation and a bi-quartic velocity approximation, both fields having full regularity, was also tested and failed the test as expected.

15.6.2 *Forced Wedge-Shaped Cavity: Manufactured Solution and Error Convergence*

In this example, we study the ability of the discretizations to reproduce an analytical solution. Motivated by examples in [11, 20], we consider the problem outlined in Fig. 15.11a. A fluid is contained in the wedge-shaped region $\Omega = \{(x, y) \in \mathbb{R}^2 | \, 0 \leq x \leq 2, \, 0 \leq y \leq 1 + x^2/4\}$. We represent the domain exactly using quadratic B-splines. As target solution, we use the following velocity and pressure fields:

$$u^* = \frac{5}{532} \mathrm{e}^x \, x^2 \, (x-2)^2 \, y \, (4 - 4\,y + x^2) \times (16 - 56\,y + 8\,x^2 + 40\,y^2 - 14\,yx^2 + x^4), \tag{15.11a}$$

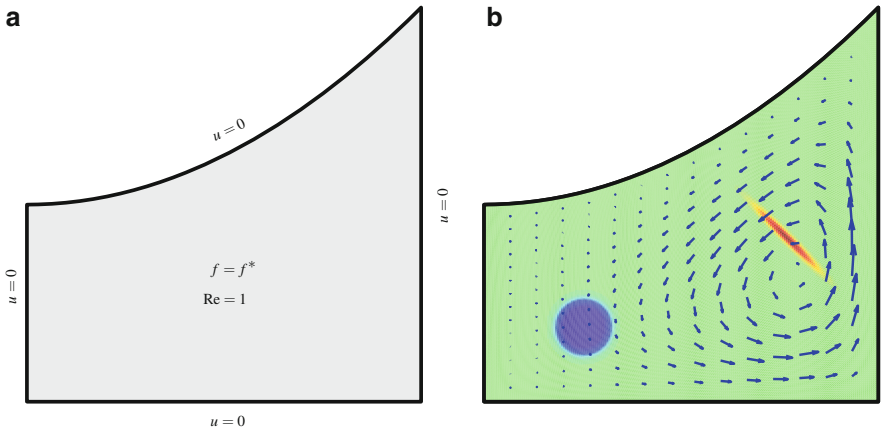

Fig. 15.11 Forced wedge-shaped cavity: problem setup (**a**) and velocity and pressure field (**b**)

$$v^* = \frac{5}{266}\,\mathrm{e}^x\,x\,(x-2)\,y^2\,(4-4\,y+x^2)^2$$
$$\times(16-8\,x-16\,y+12\,x^2+8\,xy-8\,x^3+4\,yx^2-x^4), \tag{15.11b}$$
$$p^* = \frac{1}{4}+\frac{1}{4}\tanh\left(200\,(x-1/2)^2+200\,(y-3/8)^2-4\right)$$
$$+\frac{1}{2}\,e^{-\left(5/2\sqrt{2}x-5/2\sqrt{2}y-5/2\right)^2-\left(25\sqrt{2}x+25\sqrt{2}y-250/3\right)^2}, \tag{15.11c}$$

as sketched in Fig. 15.11b. The velocity field is incompressible and fulfills the boundary conditions. By deriving the body force $\boldsymbol{f}^*$ through direct insertion into the Navier-Stokes equation, Eq. (15.11) is a manufactured analytical solution to the governing equations (15.1) and (15.2). As is evident from Fig. 15.11b, both the velocity and pressure fields exhibit phenomena that clearly call for local refinement.

To select which B-splines to refine, we follow two different approaches: a global approach and a local approach. By the global approach, we refine *all* functions, i.e., we are back in the tensor-product setting. By the local approach, we base the selection on the strong residual of the governing equations (15.1):

$$\boldsymbol{R}_h = \begin{pmatrix} (\boldsymbol{u}_h\cdot\nabla)u_h + \frac{\partial p_h}{\partial x} - \frac{1}{\mathrm{Re}}\Delta u_h + f_x \\ (\boldsymbol{u}_h\cdot\nabla)v_h + \frac{\partial p_h}{\partial y} - \frac{1}{\mathrm{Re}}\Delta v_h + f_y \\ \nabla\cdot\boldsymbol{u}_h \end{pmatrix}. \tag{15.12}$$

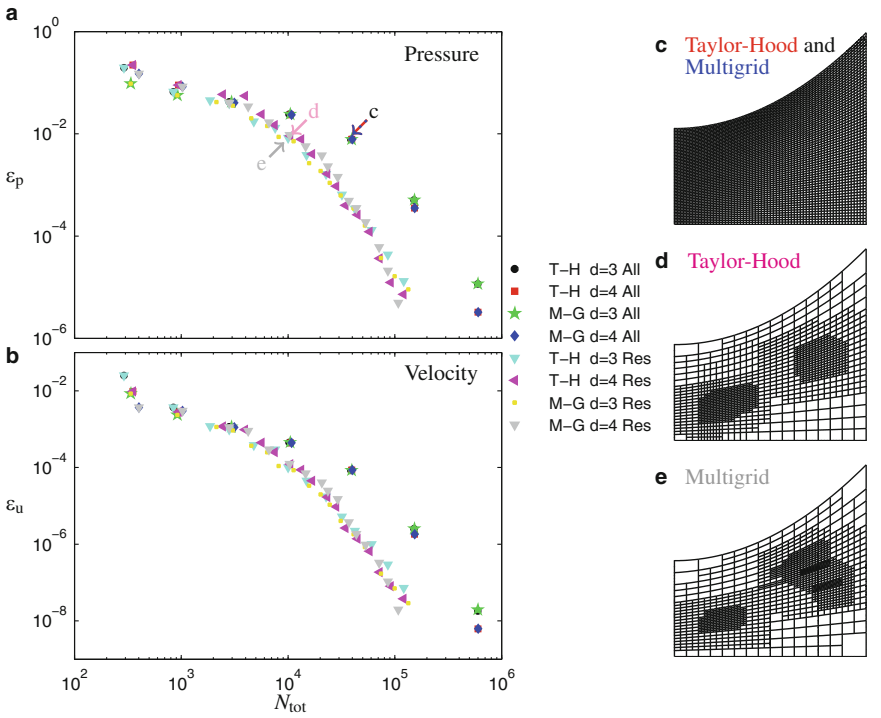

Fig. 15.12 Forced wedge-shaped cavity: Error convergence for the pressure (**a**) and velocity (**b**), and examples of the pressure grids produced by the global (**c**) and local refinement schemes for the Taylor-Hood (**d**) and the multigrid discretization (**e**) for degree $d = 4$

As error indicator, we use the L^2-norm of the residual vector $\boldsymbol{R}_h$. We integrate this on each element in the mesh, and for each (pressure) LR B-spline, we sum the errors from each of the elements in their support. Ordering the LR B-splines in a decreasing order according to their sum of errors, we refine the smallest number of LR B-splines that account for at least, say, 25 % of the total error. This may be seen as a *Dörfler* marking of LR B-splines.

In the following, we study how the global integrals of the L^2-norms of the errors on the pressure and the velocity fields behave as we refine the two discretization families based on each of the two refinement schemes. The results are shown in Fig. 15.12a, b for the pressure and the velocity fields, respectively, using polynomial degrees of 3 and 4. For any given number of degrees of freedom, the local refinement scheme is seen to yield significantly lower errors than the global refinement scheme, when comparing corresponding discretizations and degrees. For any given tolerance on the errors, hence, local refinement reduces the required number of degrees of freedom by up to an order of magnitude compared to global refinement. Comparing the Taylor-Hood and the multigrid discretizations for any given refinement strategy and polynomial degree, the two discretizations are seen to perform remarkably

alike. As expected, we observe that higher polynomial degrees are associated with smaller errors. Also shown in Fig. 15.12c–e are examples of the pressure grids produced through global and local refinement of the Taylor-Hood and the multigrid discretizations, respectively, for degree $d = 4$. Although the globally refined grids in Fig. 15.12c have around four times as many degrees of freedom as the locally refined grids in Fig. 15.12d, e, all four produce results with similar errors. The local refinement scheme is seen to yield refinements in regions of strong gradients in the target pressure field, cf. Fig. 15.11b.

We mention at last that the quantitative aspects of these results of course depend on the specific problem, the error estimator, the refinement scheme, etc. We believe, however, that their qualitative aspects often will be the same.

15.6.3 Lid-Driven Square Cavity: Benchmark

In the last example, we examine the performance of the discretizations on a standard benchmark flow problem: the lid-driven square cavity. As outlined in Fig. 15.13a, the fluid in the square container is set in motion by the sliding movement of the lid of the container. The problem resembles the one in Sect. 15.6.1 above. However, although the geometry is simpler, the flow is now (weakly) turbulent with $\mathrm{Re} = 5{,}000$. This introduces new challenges that we can test the LR B-spline flow discretizations against.

We solve the problem based on the LR Taylor-Hood and multigrid flow discretizations of degree $d = 3$ using the pressure grids shown in Fig. 15.13b, yielding a total of 22,515 and 22,785 degrees of freedom, respectively. The results of the computations are shown in Fig. 15.14. In Fig. 15.14a, the computed streamlines using the Taylor-Hood discretization clearly capture the counter-rotating eddies that are known to form in the NW, SE, and SW corners [10, 13]. The streamline pattern

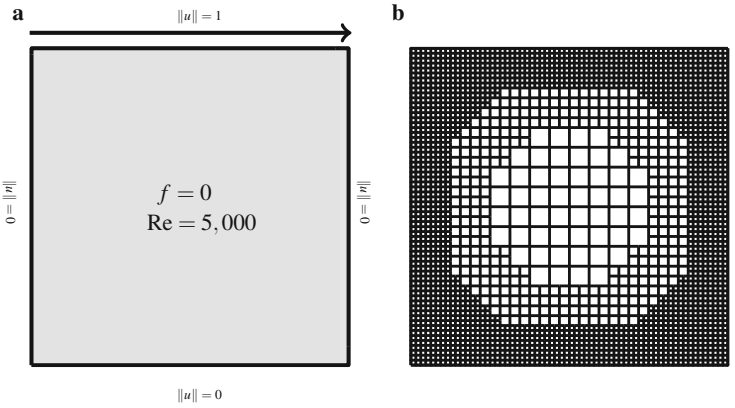

Fig. 15.13 The lid-driven square cavity: problem formulation (**a**) and pressure grids (**b**)

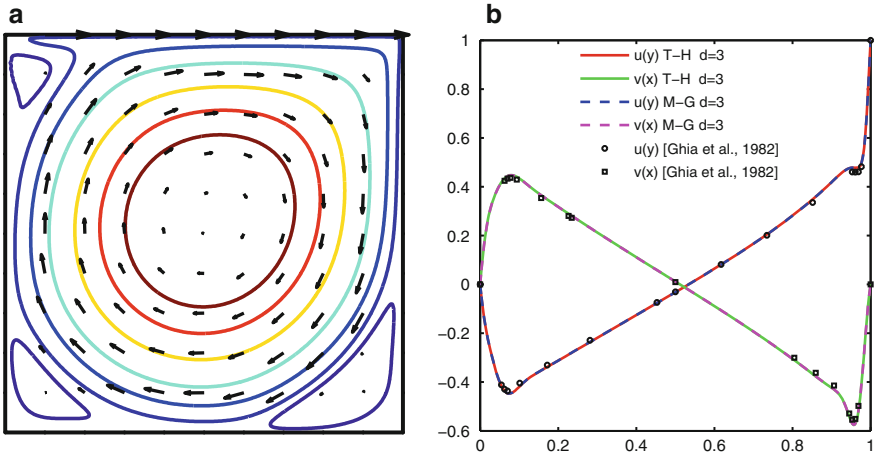

Fig. 15.14 The lid-driven square cavity: computed streamlines using the Taylor-Hood discretization (**a**) and normal velocity profiles through the two cavity center lines using both the Taylor-Hood (T-H) and multigrid (M-G) discretization (**b**)

looks the same for the multigrid discretization. In Fig. 15.14b, the normal velocity profiles through the cavity center lines computed using both the Taylor-Hood and the multigrid discretizations are seen to match very well with each other and with literature data [13].

Conclusions

The ability to achieve local refinement is crucial in all computer methods for flow problems. Locally Refinable B-splines represent a novel approach to local refinement within the context of isogeometric analysis. In this study, we have proposed two families of LR B-spline discretizations of the pressure and velocity fields for solving the mixed formulation of the steady-state, incompressible Navier-Stokes equations in two dimensions using isogeometric analysis. These LR flow discretizations represent direct extensions of well-known tensor-product flow discretizations, namely the Taylor-Hood and the multigrid discretizations. Through representative examples, we have performed a series of numerical investigations of the use of LR B-splines in isogeometric analysis of flow problems, including the stability of the discretizations, error convergence during refinement based on a manufactured solution, and benchmarking based on the lid-driven cavity problem.

Future investigations will hopefully reveal more insight into the properties of the flow discretizations. Their straightforward extensions to three dimensions should be studied. Extending the very promising but slightly more

(continued)

complicated Raviart-Thomas discretization to support local refinement is also of extreme interest, since this element satisfies the incompressibility condition exactly [11, 12]. Furthermore, efficient error estimators and refinement schemes should be studied to allow for efficient adaptive mesh refinement.

References

1. K. Bathe, The inf-sup condition and its evaluation for mixed finite element methods. Comput. Struct. **79**, 243–252 (2001)
2. Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, T.W. Sederberg, Isogeometric analysis using T-splines. Comput. Methods Appl. Mech. Eng. **199**, 229–263 (2010)
3. A. Bressan, Isogeometric regular discretization for the Stokes problem. IMA J. Numer. Anal. **31**(4), 1334–1356 (2011)
4. A. Bressan, Some properties of LR B-splines, Computer Aided Geometric Design, **30**(8), 778–794, (2013) doi: http://dx.doi.org/10.1016/j.cagd.2013.06.004
5. A. Buffa, C. de Falco, G. Sangalli, Isogeometric analysis: stable elements for the 2D Stokes equation. Int. J. Numer. Methods Fluids **65**(11–12), 1407–1422 (2011)
6. D. Chapelle, K. Bathe, The inf-sup test. Comput. Struct. **47**(4/5), 537–545 (1993)
7. J. Cottrell, T. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA* (Wiley, Chichester/Hoboken, 2009)
8. T. Dokken, K.F. Pettersen, T. Lyche, Locally refinable B-splines over box partions. Comput. Aided Geom. Des. **30**(3), 331–356 (2013)
9. M. Dörfel, B. Jüttler, B. Simeon, Adaptive isogeometric analysis by local h-refinement with T-splines. Comput. Methods Appl. Mech. Eng. **199**, 264–275 (2010)
10. E. Erturk, T. Corke, C. Gokcol, Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers. Int. J. Numer. Methods Fluids **48**, 747–774 (2005)
11. J.A. Evans, T.J.R. Hughes, Isogeometric divergence-conforming B-splines for the steady Navier-Stokes equations. Math. Models Methods Appl. Sci. **23**(8), 1421–1478 (2013)
12. J.A. Evans, T.J.R. Hughes, Isogeometric divergence-conforming B-splines for the unsteady Navier-Stokes equations. J. Comput. Fluids **241**, 141–167 (2013)
13. U. Ghia, K. Ghia, C. Shin, High-Re solution for incompressible flow using the Navier-Stokes equations and a multigrid method. J. Comput. Phys. **48**, 387–411 (1982)
14. C. Giannelli, B. Jüttler, H. Speleers, THB-splines: the truncated basis for hierarchical splines. Comput. Aided Geom. Des. **29**(7), 485–498 (2012)
15. T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**, 4135–4195 (2005)
16. K.A. Johannessen, T. Kvamsdal, T. Dokken, Isogeometric analysis using LR B-splines. Comput. Methods Appl. Mech. Eng. **269**, 471–514 (2014)
17. P.N. Nielsen, A.R. Gersborg, J. Gravesen, N.L. Pedersen, Discretizations in isogeometric analysis of Navier-Stokes flow. Comput. Methods Appl. Mech. Eng. **200**, 3242–3253 (2011)
18. M.A. Scott, X. Li, T.W. Sederberg, T.J.R. Hughes, Local refinement of analysis-suitable T-splines. Comput. Methods Appl. Mech. Eng. **213–216**, 206–222 (2012)
19. A.V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis. Comput. Methods Appl. Mech. Eng. **200**, 3554–3567 (2011)
20. G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Optimal analysis-aware parametrization of computational domain in isogeometric analysis, in *Advances in Geometric Modeling and Processing*, ed. by B. Mourrain, S. Schaeffer, G. Xu (Springer, Berlin, 2010), pp. 236–254