



HAL
open science

Compatible Discrete Operator schemes for the steady incompressible Stokes and Navier-Stokes equations

Riccardo Milani, Jérôme Bonelle, Alexandre Ern

► **To cite this version:**

Riccardo Milani, Jérôme Bonelle, Alexandre Ern. Compatible Discrete Operator schemes for the steady incompressible Stokes and Navier-Stokes equations. 2019. hal-02417514v1

HAL Id: hal-02417514

<https://hal.science/hal-02417514v1>

Preprint submitted on 18 Dec 2019 (v1), last revised 2 Feb 2020 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Compatible Discrete Operator schemes for the steady incompressible Stokes and Navier–Stokes equations

Riccardo Milani

EDF R&D and ENPC (CERMICS) and INRIA Paris,
6 Quai Watier 78400 Chatou, France, riccardo.milani@edf.fr

Jérôme Bonelle

EDF R&D,
6 Quai Watier 78400 Chatou, France, jerome.bonelle@edf.fr

Alexandre Ern

ENPC (CERMICS) and INRIA Paris,
6-8 Av. Blaise Pascal, 77455 Marne-la-Vallée, France, alexandre.ern@enpc.fr

December 18, 2019

Abstract

We extend the Compatible Discrete Operator (CDO) schemes to the steady incompressible Stokes and Navier–Stokes equations. The main features of the CDO face-based schemes are recalled: a hybrid velocity discretization with degrees of freedom at faces and cells, a stabilized velocity gradient reconstruction defined on the face-based sub-cell pyramids, and a discrete pressure attached to the mesh cells. We introduce a discrete divergence operator that will account for the velocity-pressure coupling, and a hybrid discretization of the convection term. The results of several benchmark test cases validate the framework.

Keywords: CDO, Stokes, Navier–Stokes, structure-preserving schemes

MSC (2010): 65N08, 76D05, 76M12

1 Introduction

The Compatible Discrete Operator (CDO) schemes provide a low-order framework which is part of the so-called mimetic or structure-preserving methods. One of the main advantages of the CDO schemes is that they can handle polytopal, nonmatching (cf. left part of Fig. 1) and deformed meshes. Taking advantage of a dual mesh, invisible to the end user, discrete differential operators are carefully designed to satisfy conservation laws and properties typical of their continuous counterparts. This special treatment results in optimal order of convergence in space [3] (first order for the reconstructed gradient and second for the original variable) and a cell-wise and fully parallelizable building stage ensures good performances of the overall method. Thanks to its flexibility, the CDO framework allows us to define the main problem variables on different mesh entities, according to their physical nature. Hence, one can choose to use a cell-, vertex- [3], edge- [8] or face-based [2] scheme.

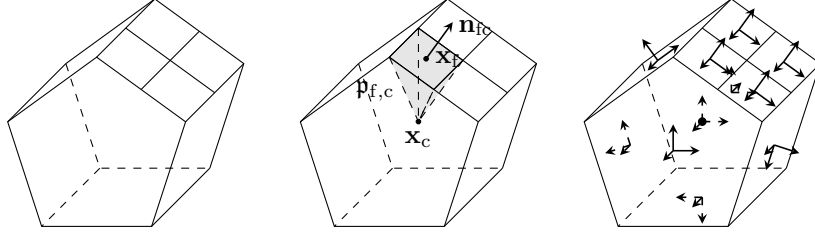


Figure 1: Example of cell compatible with CDO. **Left** cell with hanging nodes. **Center** cell with one of its subpyramids $\mathbf{p}_{f,c}$. **Right** cell with velocity (arrows) and pressure (circle) DoFs.

Here, the Stokes and Navier–Stokes equations (NSE) are discretized by means of face-based CDO (CDO-Fb) schemes. In this case, the velocity is defined at faces and cells, and the pressure is defined at cells only. CDO-Fb was introduced initially for the Poisson problem [2] and its key ingredient is a stabilized subcell gradient reconstruction, which can be bridged to the one used in the Hybrid Mixed Mimetic (HMM) framework [12] and to a generalization of the Crouzeix-Raviart framework [10] (GCR). A divergence operator is derived from this gradient and it is the tool on which the velocity-pressure coupling hinges. Finally, the discretization of the convection term is inspired by the lowest-order case of the Hybrid High-Order (HHO($k = 0$)) method [9]. The Stokes problem in its curl formulation has been already treated by means of CDO with vertex- and cell-based schemes [4] but here we retain the face-based one.

Let $\mathcal{D} \in \mathbb{R}^d$, $d = 2, 3$ be a bounded connected polyhedral domain and denote with $\partial\mathcal{D}$ its boundary. We consider the following model problem:

$$-\nu\Delta \mathbf{u} + \chi(\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f}, \quad \text{in } \mathcal{D} \quad (1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad \text{in } \mathcal{D} \quad (1b)$$

where $\nu > 0$ is the viscosity, and $\chi = 0$ for the Stokes equations or $\chi = 1$ for the NSE. For the sake of simplicity, homogeneous Dirichlet boundary conditions (BC) are considered. The pressure is uniquely defined by requiring that $\int_{\mathcal{D}} p = 0$.

2 Space discretization

A mesh discretizing \mathcal{D} is a finite collection $\mathcal{C} := \{c\}$ of nonempty, disjoint, open, polytopal elements of \mathbb{R}^d , $d = 2, 3$, usually referred to as cells c . The faces f are assumed to be planar and are gathered in the set F which may be subdivided in two disjoint sets: $F^b := \{f \mid f \subset \partial\mathcal{D}\}$ collects the boundary faces and $F^i := F \setminus F^b$ the interfaces. One associates with each face f a normal vector \mathbf{n}_f : if $f \in F^b$, \mathbf{n}_f points outward \mathcal{D} and, if $f \in F^i$, the direction is chosen arbitrarily. For a mesh entity $z = c, f$, \mathbf{x}_z denotes its barycenter and $|z|$ its measure. Consider now a generic cell c . Define the set of faces of the cell c as $F_c := \{f \in F \mid f \subset \partial c\}$. For every $f \in F_c$, $\mathbf{n}_{fc} := \pm \mathbf{n}_f$ is the normal vector to the face f pointing outward c , the sign depends thus on the direction chosen for \mathbf{n}_f . The sub-pyramid obtained by joining the vertices of f to the barycenter \mathbf{x}_c of the cell (cf. the central part of Fig. 1) is denoted by $\mathbf{p}_{f,c}$.

2.1 Discrete functional spaces and differential operators

Given a generic mesh entity $z = c, f$, $\mathbb{P}^0(z) \equiv \mathbb{R}$ denotes the scalar-valued, zero-th order polynomials defined on z . We denote with $\pi_z: L^1(z) \rightarrow \mathbb{P}^0(z) \equiv \mathbb{R}$ the L^2 -projection (average): for all $s \in L^1(z)$, $\pi_z(s) = \int_z s / |z|$. For vector-valued functions $\mathbf{s} \in [L^1(z)]^d$ the projection is applied component-wise: $\boldsymbol{\pi}_z(\mathbf{s}) := (\pi_z(s_i))_{i=1, \dots, d}$.

In the CDO-Fb framework the velocity is hybrid, meaning that it has cell- and face-based degrees of freedom (DoFs). Hence, the global velocity space is

$$\widehat{\mathbf{U}}_h := \prod_{c \in \mathcal{C}} [\mathbb{P}^0(c)]^d \times \prod_{f \in \mathcal{F}} [\mathbb{P}^0(f)]^d. \quad (2)$$

An element of $\widehat{\mathbf{U}}_h$ is denoted by $\widehat{\mathbf{u}}_h := ((\mathbf{u}_c)_{c \in \mathcal{C}}, (\mathbf{u}_f)_{f \in \mathcal{F}})$, where, for a generic $z = c, f$, \mathbf{u}_z is the z -based DoF. Notice that the value at the interfaces is uniquely defined. The velocity DoFs associated with a cell c are denoted $\widehat{\mathbf{u}}_c := (\mathbf{u}_c, (\mathbf{u}_f)_{f \in \mathcal{F}_c}) \in \widehat{\mathbf{U}}_c := [\mathbb{P}^0(c)]^d \times \prod_{f \in \mathcal{F}_c} [\mathbb{P}^0(f)]^d$. The pressure is defined at cells only: $P_h := \prod_{c \in \mathcal{C}} P_c \ni p_h := (p_c)_{c \in \mathcal{C}}$, where $P_c := \mathbb{P}^0(c)$. In order to take into consideration the velocity BC and the constraint on the pressure average, one also needs $\widehat{\mathbf{U}}_{h,0} := \{\widehat{\mathbf{v}}_h \in \widehat{\mathbf{U}}_h \mid \mathbf{u}_f = 0 \ \forall f \in \mathcal{F}^b\}$, $P_{h,*} := \{p_h \in P_h \mid \sum_{c \in \mathcal{C}} |c| p_c = 0\}$. The right part of Fig. 1 gives an example of local velocity and pressure DoFs for a cell.

2.1.1 Discrete velocity gradient and divergence

For each cell $c \in \mathcal{C}$, the discrete local gradient G_c is piecewise constant on the pyramid partition $\{\mathbf{p}_{f,c}\}_{f \in \mathcal{F}_c}$ (cf. central part of Fig. 1) and is defined as follows: $G_c: \widehat{\mathbf{U}}_c \rightarrow [\mathbb{P}^0(\{\mathbf{p}_{f,c}\}_{f \in \mathcal{F}_c})]^{d \times d}$ such that for all $f \in \mathcal{F}_c$

$$G_c(\widehat{\mathbf{u}}_c)|_{\mathbf{p}_{f,c}} := G_c^0(\widehat{\mathbf{u}}_c) + \beta \frac{|f|}{|\mathbf{p}_{f,c}|} ((\mathbf{u}_f - \mathbf{u}_c) - G_c^0(\widehat{\mathbf{u}}_c)(\mathbf{x}_f - \mathbf{x}_c)) \otimes \mathbf{n}_{fc}, \quad (3)$$

where $\beta > 0$ is an arbitrary stability parameter and $G_c^0(\widehat{\mathbf{u}})$ is a \mathbb{P}_0 -consistent gradient, constant inside each cell and defined as $G_c^0(\widehat{\mathbf{u}}_c) := 1/|c| \sum_{f \in \mathcal{F}_c} |f| (\mathbf{u}_f - \mathbf{u}_c) \otimes \mathbf{n}_{fc}$. The definition (3) is the vector-valued version of the gradient introduced in [2]. In the numerical tests, we will use $\beta = 1$, which recovers the GCR framework [10]; the choice $\beta = 1/\sqrt{d}$ gives the HMM one [12].

For each cell $c \in \mathcal{C}$, the discrete velocity divergence $D_c: \widehat{\mathbf{U}}_c \rightarrow \mathbb{P}^0(c)$ is defined as follows

$$D_c(\widehat{\mathbf{u}}_c) := \text{trace}(G_c^0(\widehat{\mathbf{u}}_c)) = \frac{1}{|c|} \sum_{f \in \mathcal{F}_c} |f| \mathbf{u}_f \cdot \mathbf{n}_{fc}. \quad (4)$$

Notice that only the face-based DoFs are used (since faces are planar). The discrete velocity divergence is the tool on which the velocity-pressure coupling hinges. This divergence operator can be found also in the HMM framework [11].

2.1.2 Discrete advection scheme

The design of the advection scheme is inspired by HHO($k=0$) [9]. We aim at discretizing the classical advective trilinear form such that $\int_{\mathcal{D}} ((\mathbf{w} \cdot \nabla) \mathbf{u}) \cdot \mathbf{v}$. Given $\widehat{\mathbf{u}}_h, \widehat{\mathbf{v}}_h, \widehat{\mathbf{w}}_h \in \widehat{\mathbf{U}}_{h,0}$, we use

$$\begin{aligned} a_{\text{adv}}(\widehat{\mathbf{w}}_h; \widehat{\mathbf{u}}_h, \widehat{\mathbf{v}}_h) := & \frac{1}{2} \sum_{c \in \mathcal{C}} \sum_{f \in \mathcal{F}_c} |f| (\mathbf{w}_f \cdot \mathbf{n}_{fc}) (\mathbf{u}_f - \mathbf{u}_c) (\mathbf{v}_f + \mathbf{v}_c) \\ & + \theta^{\text{upw}} \sum_{f \in \mathcal{F}^i} \sum_{c \in \mathcal{C}_f} |f| |\mathbf{w}_f \cdot \mathbf{n}_f| (\mathbf{u}_f - \mathbf{u}_c) (\mathbf{v}_f - \mathbf{v}_c). \end{aligned} \quad (5)$$

where $\theta^{\text{upw}} := 1$ in one wants a stabilization by upwinding, or $\theta^{\text{upw}} := 0$ for a centered scheme. Suppose, for now, that $\theta^{\text{upw}} = 0$. One has:

$$a_{\text{adv}}(\widehat{\mathbf{w}}_h; \widehat{\mathbf{u}}_h, \widehat{\mathbf{v}}_h) + a_{\text{adv}}(\widehat{\mathbf{w}}_h; \widehat{\mathbf{v}}_h, \widehat{\mathbf{u}}_h) = - \sum_{c \in \mathcal{C}} |c| D_c(\widehat{\mathbf{w}}_c) \mathbf{u}_c \cdot \mathbf{v}_c + \sum_{f \in \mathcal{F}^b} |f| (\mathbf{w}_f \cdot \mathbf{n}_f) \mathbf{u}_f \cdot \mathbf{v}_f \quad (6)$$

obtained by using (4) and by discarding the internal face-defined DoFs since they sum to zero. The boundary DoFs are kept in order to better show that (6) is the discrete counterpart of a known integral-by-parts result. Plugging $\widehat{\mathbf{v}}_h = \widehat{\mathbf{u}}_h$ into (6) shows that

$$a_{\text{adv}}(\widehat{\mathbf{w}}_h; \widehat{\mathbf{u}}_h, \widehat{\mathbf{u}}_h) = -\frac{1}{2} \sum_{c \in \mathcal{C}} |c| D_c(\widehat{\mathbf{w}}_c) \mathbf{u}_c^2. \quad (7)$$

Supposing there exists $\mu > 0$ such that $-1/2 D_c(\widehat{\mathbf{w}}_c) \geq \mu$ for all $c \in \mathcal{C}$ (this is a discrete counterpart of the well-known stability hypothesis of the continuous advection problem), then (7) proves the coercivity of $a_{\text{adv}}(\widehat{\mathbf{w}}_h; \cdot, \cdot)$.

2.2 Discrete bilinear form

The discrete counterpart of problem (1)-(1) stemming from the CDO-Fb scheme writes: Find $(\widehat{\mathbf{u}}_h, p_h) \in \widehat{\mathbf{U}}_{h,0} \times P_{h,*}$ such that, $\forall \widehat{\mathbf{v}}_h \in \widehat{\mathbf{U}}_{h,0}$ and $\forall q_h \in P_{h,*}$

$$\sum_{c \in \mathcal{C}} \int_c \{ \nu G_c(\widehat{\mathbf{u}}_c) : G_c(\widehat{\mathbf{v}}_c) - p_c D_c(\widehat{\mathbf{v}}_c) \} + \chi a_{\text{adv}}(\widehat{\mathbf{u}}_c; \widehat{\mathbf{u}}_c, \widehat{\mathbf{v}}_c) = \sum_{c \in \mathcal{C}} \int_c \mathbf{f} \cdot \mathbf{v}_c, \quad (8a)$$

$$\sum_{c \in \mathcal{C}} -D_c(\widehat{\mathbf{u}}_c) q_c = 0. \quad (8b)$$

The Stokes problem ($\chi = 0$ in (1)) has been analyzed in [11].

A static condensation procedure eliminating the cell-based velocity DoFs can be performed in order to reduce the size of the global system, which thus becomes $d \text{Card}(\mathcal{F}) + \text{Card}(\mathcal{C})$. The discarded DoFs are recovered after the solving stage, as a post-processing.

3 Numerical results

Following the benchmark proposed in [7], four test cases are considered, two for the Stokes equations (in 2D and 3D), and two for the NSE (both in 2D). When considering the latter, the nonlinear equations are solved by Picard iterations, and the stopping criterion is $\|\mathbf{u}^k - \mathbf{u}^{k-1}\|_C / \|\mathbf{u}^{k-1}\|_C < \varepsilon$. We consider the discrete L^2 -norm of the velocity $\|\widehat{\mathbf{u}}_h\|_C^2 := \sum_{c \in \mathcal{C}} |c| \|\mathbf{u}_c\|_2^2$, and of its gradient $\|\widehat{\mathbf{u}}_h\|_{G,C}^2 := \sum_{c \in \mathcal{C}} |c| \|G(\widehat{\mathbf{u}}_c)\|_2^2$, the discrete L^2 -norm of the pressure $\|p_h\|_C^2 := \sum_{c \in \mathcal{C}} |c| p_c^2$, and to the following relative error norms:

$$\text{erru} := \frac{\|\widehat{\mathbf{u}}_h - \widehat{\boldsymbol{\pi}}_h(\mathbf{u})\|_C}{\|\widehat{\boldsymbol{\pi}}_h(\mathbf{u})\|_C}, \quad \text{errgu} := \frac{\|\widehat{\mathbf{u}}_h - \widehat{\boldsymbol{\pi}}_h(\mathbf{u})\|_{G,C}}{\|\widehat{\boldsymbol{\pi}}_h(\mathbf{u})\|_{G,C}}, \quad \text{errp} := \frac{\|p_h - \pi_h(p)\|_C}{\|\pi_h(p)\|_C}, \quad (9)$$

where $\widehat{\boldsymbol{\pi}}_h(\mathbf{u}) := ((\boldsymbol{\pi}_c(\mathbf{u}))_{c \in \mathcal{C}}, (\boldsymbol{\pi}_f(\mathbf{u}))_{f \in \mathcal{F}})$ and $\pi_h(p) := (\pi_c(p))_{c \in \mathcal{C}}$. Let n_{u} (resp. n_{p}) stand for the number of velocity (pressure) unknowns. They will be used to evaluate the orders of convergence in space.

We will use the CDO implementation available via *Code_Saturne* [1], an open-source multi-purpose CFD solver developed at EDF. The computations have been performed on a octa-core, Intel i7 laptop with 32GB RAM using PETSc and MUMPS libraries to solve the linear systems.

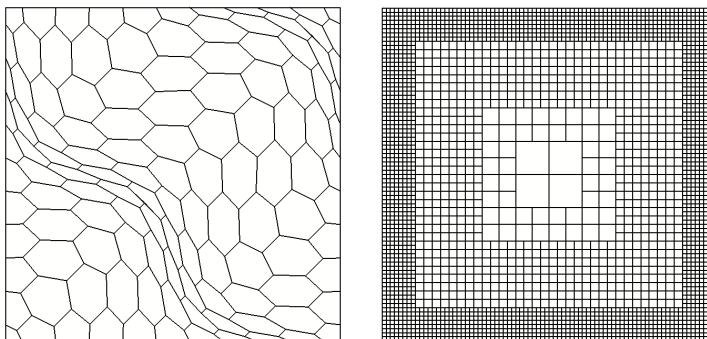


Figure 2: Examples of 2D meshes. **Left** polygonal. **Right** progressively refined Cartesian.

3.1 Stokes equations

Two test cases are considered for the Stokes equations ($\chi = 0$ in (1)).

2D Bercovier–Engelman test case It is proposed in the test case 2.1 of the benchmark [7]. The sequence of Cartesian meshes (denoted by Hn where n is the number of segments an edge of the domain is divided into) from [7] and a 2D polygonal family (similarly denoted by Pn , cf. left part of Fig. 2) have been considered. The results are collected in Table 1.

Table 1: Errors for the 2D Bercovier–Engelman test case - Cartesian and polygonal meshes

mesh	nuu	npu	errgu	order	erru	order	errp	order
H32	4224	1024	$9.15 \cdot 10^{-4}$	–	$7.71 \cdot 10^{-4}$	–	$1.06 \cdot 10^{-1}$	–
H64	16640	4096	$3.16 \cdot 10^{-4}$	1.55	$1.93 \cdot 10^{-4}$	2.02	$2.87 \cdot 10^{-2}$	1.98
H128	66048	16384	$1.35 \cdot 10^{-4}$	1.24	$4.82 \cdot 10^{-5}$	2.01	$7.36 \cdot 10^{-3}$	1.99
H256	263168	65536	$6.41 \cdot 10^{-5}$	1.07	$1.21 \cdot 10^{-5}$	2.01	$1.85 \cdot 10^{-3}$	1.99
P10	720	121	$9.99 \cdot 10^{-2}$	–	$3.33 \cdot 10^{-2}$	–	$2.63 \cdot 10^0$	–
P20	2640	441	$5.81 \cdot 10^{-2}$	0.83	$9.07 \cdot 10^{-3}$	2.00	$7.42 \cdot 10^{-1}$	1.96
P30	5760	961	$4.07 \cdot 10^{-2}$	0.91	$4.07 \cdot 10^{-3}$	2.05	$3.36 \cdot 10^{-1}$	2.03
P40	10080	1681	$3.13 \cdot 10^{-2}$	0.94	$2.29 \cdot 10^{-3}$	2.05	$1.91 \cdot 10^{-1}$	2.02

3D Taylor–Green vortex This test case is proposed in the test case 2.2 of the benchmark [7]. The meshes used were the Cartesian (Hn) and prismatic with triangular bases ($PrTn$) sequences proposed in [7], and one composed of tetrahedra (Tn , the refinement is achieved by dividing each tetrahedra into 8 subtetrahedra). The results are collected in Table 2.

3.2 Navier–Stokes equations

Two test cases are considered for the Navier–Stokes equations ($\chi = 1$ in (1)).

Burggraf flow It consists in a manufactured polynomial solution of the 2D NSE presented in [6]. The centered scheme was considered ($\theta^{upw} = 0$ in (5)). The viscosity is $\nu = 1/100$. About 15 Picard iterations were needed to reach the prescribed tolerance $\varepsilon = 10^{-7}$. Two sequences of meshes have been considered: the Cartesian one (Hn) from [7], and one composed

Table 2: Errors for the 3D Taylor–Green Vortex- Cartesian, tetrahedral and prismatic meshes

mesh	nuu	npu	errgu	order	erru	order	errp	order
H4	720	64	$4.36 \cdot 10^{-1}$	–	$3.18 \cdot 10^{-1}$	–	$4.83 \cdot 10^{-1}$	–
H8	5184	512	$2.60 \cdot 10^{-1}$	0.79	$1.05 \cdot 10^{-1}$	1.69	$1.49 \cdot 10^{-1}$	1.70
H16	39168	4096	$1.36 \cdot 10^{-1}$	0.96	$2.82 \cdot 10^{-2}$	1.95	$3.95 \cdot 10^{-2}$	1.92
H32	304128	32768	$6.91 \cdot 10^{-2}$	1.00	$7.18 \cdot 10^{-3}$	2.00	$1.00 \cdot 10^{-2}$	1.98
T6	15090	2383	$3.39 \cdot 10^{-1}$	–	$8.99 \cdot 10^{-2}$	–	$1.45 \cdot 10^{-1}$	–
T12	117552	19064	$1.76 \cdot 10^{-1}$	0.96	$2.39 \cdot 10^{-2}$	1.94	$5.65 \cdot 10^{-2}$	1.36
T24	927744	152512	$8.86 \cdot 10^{-2}$	1.00	$6.07 \cdot 10^{-3}$	1.99	$2.49 \cdot 10^{-2}$	1.18
PrT10	16200	2000	$3.12 \cdot 10^{-1}$	–	$9.18 \cdot 10^{-2}$	–	$1.64 \cdot 10^{-1}$	–
PrT20	124800	16000	$1.67 \cdot 10^{-1}$	0.92	$2.72 \cdot 10^{-2}$	1.79	$6.60 \cdot 10^{-2}$	1.32
PrT30	415800	54000	$1.13 \cdot 10^{-1}$	0.97	$1.28 \cdot 10^{-2}$	1.88	$3.96 \cdot 10^{-2}$	1.26
PrT40	979200	128000	$8.54 \cdot 10^{-2}$	0.99	$7.40 \cdot 10^{-3}$	1.92	$2.81 \cdot 10^{-2}$	1.20

of nonmatching squares (HR n , cf. right part of Fig. 2), obtained by progressively refining the Cartesian meshes. The results are collected in Table 3.

Table 3: Errors for the 2D Burggraf flow - Cartesian and refined Cartesian meshes

mesh	nuu	npu	errgu	order	erru	order	errp	order
H32	4224	1024	$2.40 \cdot 10^{-1}$	–	$1.73 \cdot 10^{-2}$	–	$1.33 \cdot 10^{-2}$	–
H64	16640	4096	$1.20 \cdot 10^{-1}$	1.01	$4.35 \cdot 10^{-3}$	2.01	$3.39 \cdot 10^{-3}$	1.98
H128	66048	16384	$6.01 \cdot 10^{-2}$	1.00	$1.09 \cdot 10^{-3}$	2.00	$8.53 \cdot 10^{-4}$	1.99
H256	263168	65536	$3.01 \cdot 10^{-2}$	1.00	$2.73 \cdot 10^{-4}$	2.02	$2.14 \cdot 10^{-4}$	1.99
HR80	12984	3124	$1.74 \cdot 10^{-1}$	–	$2.19 \cdot 10^{-2}$	–	$2.38 \cdot 10^{-2}$	–
HR160	50960	12496	$8.85 \cdot 10^{-2}$	0.99	$5.95 \cdot 10^{-3}$	1.90	$6.59 \cdot 10^{-3}$	1.85
HR320	201888	49984	$4.44 \cdot 10^{-2}$	1.00	$1.51 \cdot 10^{-3}$	1.99	$1.69 \cdot 10^{-3}$	1.96

2D lid-driven cavity It is proposed in the test case 6 of the benchmark [7]. Two values of the viscosity has been considered: $\nu = 1/400, 1/1000$. Computations have been run on Cartesian meshes with edges divided into 127, 255, and 511 segments. The upwind stabilization has been considered ($\theta^{\text{upw}} = 1$ in (5)). The prescribed tolerance for the Picard iterations is $\varepsilon = 10^{-7}$, less than 15 iterations were needed for $\nu = 1/400$ and less than 30 for $1/1000$. Fig. 3 gathers the computed vertical and horizontal profiles on the symmetry axes for three Cartesian meshes as well as those from references [13] and [5].

References

- [1] Archambeau, F., Méchitoua, N., Sakiz, M.: Code saturne: A finite volume code for turbulent flows - industrial applications. *Int. J. Finite Vol.* **1**(1) (2004)
- [2] Bonelle, J.: Compatible discrete operator schemes on polyhedral meshes for elliptic and Stokes equations. Ph.D. thesis, Université Paris-Est (2014)

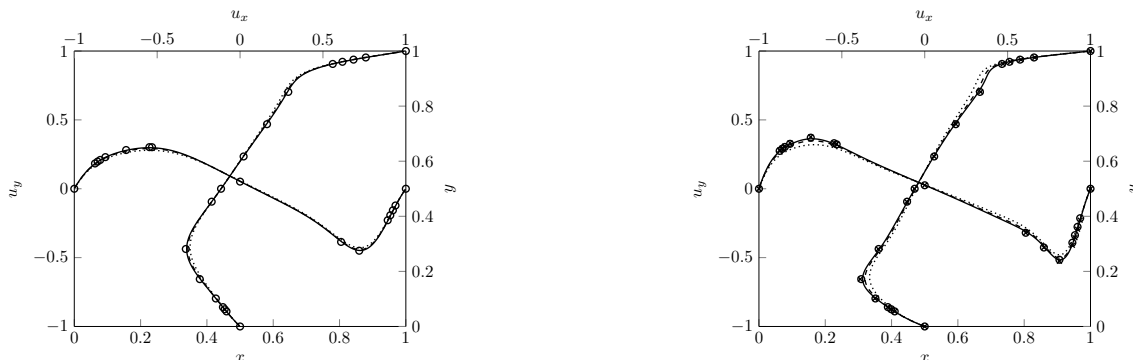


Figure 3: Lid-driven cavity, vertical and horizontal velocity profiles at the axis of symmetry. Data: CDO H127 (*solid line*), CDO H255 (*dashed line*), CDO H511 (*dotted line*), [13] (*circle*), [5] (*cross*). **Left** $\nu = 1/400$. **Right** $\nu = 1/1000$.

- [3] Bonelle, J., Ern, A.: Analysis of compatible discrete operator schemes for elliptic problems on polyhedral meshes. *ESAIM Math. Model. Numer. Anal.* **48**(2), 553–581 (2014)
- [4] Bonelle, J., Ern, A.: Analysis of Compatible Discrete Operator schemes for the Stokes equations on polyhedral meshes. *IMA J. Numer. Anal.* **35**(4), 1672–1697 (2015)
- [5] Botella, O., Peyret, R.: Benchmark spectral results on the lid-driven cavity flow. *Comput. Fluids* **27**(4), 421–433 (1998)
- [6] Burggraf, O.R.: Analytical and numerical studies of the structure of steady separated flows. *J. Fluid Mech.* **24**(1), 113–151 (1966)
- [7] Cancès, C., Omnes, P. (eds.): *Finite Volumes for Complex Applications VIII - Methods and Theoretical Aspects, Springer Proceedings in Mathematics & Statistics*, vol. 199. Springer International Publishing, Lille, France (2017)
- [8] Cantin, P., Ern, A.: Vertex-based compatible discrete operator schemes on polyhedral meshes for advection-diffusion equations. *Comput. Methods Appl. Math.* **16**(2), 187–212 (2016)
- [9] Di Pietro, D.A., Droniou, J., Ern, A.: A discontinuous-skeletal method for advection-diffusion-reaction on general meshes. *SIAM J. Numer. Anal.* **53**(5), 2135–2157 (2015)
- [10] Di Pietro, D.A., Lemaire, S.: An extension of the Crouzeix-Raviart space to general meshes with application to quasi-incompressible linear elasticity and stokes flow. *Math. Comput.* **84**(291), 1–31 (2015)
- [11] Droniou, J., Eymard, R., Feron, P.: Gradient Schemes for Stokes problem. *IMA J. Numer. Anal.* **36**(4), 1636–1669 (2015)
- [12] Droniou, J., Eymard, R., Gallouët, T., Herbin, R.: A unified approach to mimetic finite difference, hybrid finite volume and mixed finite volume methods. *Math. Model. Methods Appl. Sci.* **20**(2), 265–295 (2010)
- [13] Ghia, U., Ghia, K.N., Shin, C.T.: High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.* **48**(3), 387–411 (1982)