Conservative Mimetic Cut-Cell Method for Incompressible Navier-Stokes Equations

René Beltman, Martijn Anthonissen, and Barry Koren

Abstract We introduce a mimetic Cartesian cut-cell method for incompressible viscous flow that conserves mass, momentum, and kinetic energy in the inviscid limit, and determines the vorticity such that the global vorticity is consistent with the boundary conditions. In particular we discuss how the no-slip boundary conditions should be applied in a conservative way on objects immersed in the Cartesian mesh. We use the method to compute the flow around a cylinder. We find a good comparison between our results and benchmark results for both a steady and an unsteady test case.

To compute fluid flow in complicated geometries often curvilinear or unstructured meshes are used. The generation of these meshes is difficult and time-consuming. When the geometry depends on time, the mesh has to be updated after every time step and the cost of mesh generation will take a significant part of the total computing time.

Immersed boundary methods form an increasingly popular alternative. Immersed boundary methods are methods in which one Cartesian mesh is used for the complete flow domain, with the boundaries of objects immersed in this Cartesian mesh. Near the immersed boundary the Cartesian method is adapted for the no-slip boundary condition.

Within the class of immersed boundary methods essentially two approaches for modeling the boundary conditions exist. In the first approach the influence of the boundary on the fluid is modeled by an extra force term in the Navier-Stokes equations. In the second approach the sharp interface of the boundary is maintained and the boundary condition is taken into account by adjusting the discretized Navier-Stokes equations.

R. Beltman (✉) · M. Anthonissen · B. Koren
Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, Netherlands
e-mail: r.beltman@tue.nl; m.j.h.anthonissen@tue.nl; b.koren@tue.nl

© Springer Nature Switzerland AG 2019

F. A. Radu et al. (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2017*, Lecture Notes in Computational Science and Engineering 126,
https://doi.org/10.1007/978-3-319-96415-7_98

Cut-cell methods follow the second approach. Close to the object non-Cartesian cells occur, the so-called cut cells. These cells demand a special treatment because of their more difficult shape and the no-slip boundary condition on one of their faces.

One of the most popular discretization methods for the incompressible Navier-Stokes equations on Cartesian meshes is the MAC-method [1]. It uses a staggered mesh, which means that the velocity variables are located on the faces of the Cartesian cells and the pressure variables are located in the centers of the cells. From this variable positioning, through central difference approximations a method follows that has a compact stencil, no spurious pressure oscillations, conservation of both mass and momentum as well as conservation of secondary quantities as vorticity and energy (in the inviscid limit).

Most cut-cell methods use a colocated mesh, where the velocity and pressure variables are all located in the centers, which makes the treatment of cut cells simpler. However, methods on colocated meshes do not have all the favorable properties that the MAC method has. For example, spurious pressure oscillations have to be suppressed by introducing artificial diffusion, making these methods less suitable for turbulent flow computations.

A few extensions of the MAC method to cut-cell meshes have been presented in the literature [2, 3]. In these methods different cut-cell configurations are treated case by case and a mass, momentum and energy conserving extension is derived using, in 2D, a 5-point stencil and a finite-volume rationale. However, achieving a fully conservative method using the 5-point stencil is impossible. Moreover, an extension of the method to 3D is problematic due to the many possible cut-cell configurations. To our knowledge only a quasi-3D extension, where the immersed boundary is parallel to one of the Cartesian coordinate axes, has been published so far [4].

We will use the recent developments in mimetic discretization methods [5], that allow for a conservative treatment of non-Cartesian cells. Using these methods on a primal-dual mesh structure, upon completing the dual mesh near the boundary, we derive a mass, momentum and energy conserving cut-cell method that calculates vorticity in such a way that the global vorticity is consistent with no-slip boundary conditions. We first introduce the method and then present results for the flow around a cylinder.

1 The Mimetic Cut-Cell Method

1.1 The Cut-Cell Primal-Dual Cell-Complex

We discuss the cut-cell mesh for the specific case of a cylinder in a rectangular domain. From this discussion it will be clear how it can be generalized to other domains. We give the cylinder its own discrete representation, independent of the

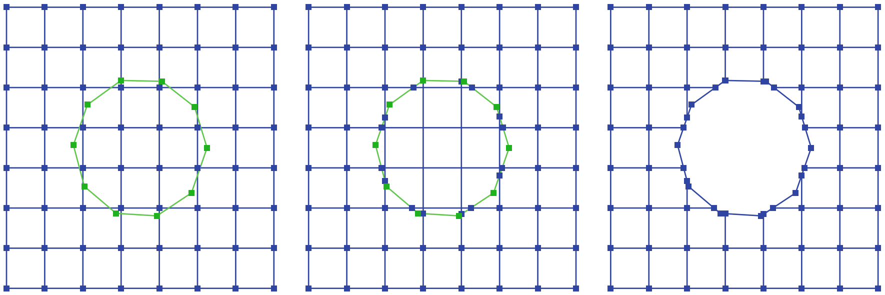

Fig. 1 On the left the Cartesian mesh (blue) and the discretized cylinder (green) are shown. In the middle one sees the cylinder with extra intersection vertices added. Finally, on the right the resulting primal mesh is shown

Cartesian mesh, because in a future extension to general time-dependent geometries this will be needed for mass conservation.

We cover the rectangular domain by a Cartesian mesh with $N_x \times N_y$ cells. For simplicity we take the mesh to be uniform. We discretize the boundary of the cylinder by uniformly taking N_θ points on the circle and connecting these by straight line segments. Subsequently, we immerse the discretized cylinder in the Cartesian mesh, add the intersections between edges of the Cartesian mesh and the cylinder as extra vertices, and remove all the edges and vertices in the interior of the cylinder. This process has been depicted in Fig. 1.

The resulting computational mesh consists of two-dimensional cells, one-dimensional edges and zero-dimensional vertices. We denote the mesh by $\mathcal{G} := \{\mathcal{C}_{(0)}, \mathcal{C}_{(1)}, \mathcal{C}_{(2)}\}$, where $\mathcal{C}_{(k)}$, for $k = 0, 1, 2$, is the set of k-dimensional cells. This mesh is a so-called cell-complex. This means that for each k-dimensional cell $\sigma_{(k)} \in \mathcal{C}_{(k)}$ in the mesh its boundary $\partial\sigma_{(k)}$ is made up of lower dimensional cells that are also part of the mesh. The cell-complex $\mathcal{G}$ covers the flow domain Ω.

Our cut-cell method will also need a dual mesh. The dual mesh is a second mesh for the flow domain that is geometrically dual to the primal mesh in the sense that for each k-dimensional cell $\sigma_{(k)} \in \mathcal{C}_{(k)}$ of the primal mesh, there exists a $(2-k)$-dimensional dual cell $\tilde{\sigma}_{(k)} \in \tilde{\mathcal{C}}_{(k)}$ in the dual mesh, where we denote by $\tilde{\mathcal{C}}_{(k)}$ the set of $(2-k)$-dimensional dual cells. We use a barycentric dual mesh. In contrast to the primal mesh, the dual mesh $\tilde{\mathcal{G}} = \{\tilde{\mathcal{C}}_{(2)}, \tilde{\mathcal{C}}_{(1)}, \tilde{\mathcal{C}}_{(0)}\}$ is not a cell-complex. However, to formulate a fully conservative cut-cell method it is important to extend the dual mesh to a cell-complex. This can be done as follows. We consider the restriction of the primal mesh to the boundary $\partial\Omega$ of Ω, denoted by $\mathcal{G}^b = \{\mathcal{C}^b_{(0)}, \mathcal{C}^b_{(1)}\}$, and then take the dual mesh to the primal mesh within $\partial\Omega$, which we denote by $\tilde{\mathcal{G}}^b = \{\tilde{\mathcal{C}}^b_{(1)}, \tilde{\mathcal{C}}^b_{(0)}\}$. The union of the original dual mesh and the boundary dual mesh, $\bar{\mathcal{G}} := \tilde{\mathcal{G}} \cup \tilde{\mathcal{G}}^b$, constitutes a dual cell-complex. This construction works in arbitrary dimension. The construction of the dual cell-complex has been depicted in Fig. 2. More details concerning the primal and dual cell-complex can be found in [6].

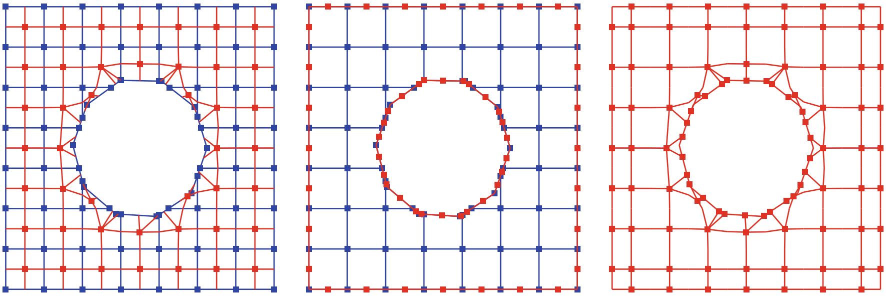

Fig. 2 On the left the primal (blue) and dual (red) mesh are depicted and in the middle the primal mesh and boundary dual mesh are shown. On the right one sees the dual cell-complex which is the union of the interior and boundary dual mesh

1.2 The Incidence Matrices

We discretize the velocity field $\underline{u}$ as the fluxes through the edges of the primal mesh. For each edge $\sigma_{(1)} \in \mathcal{C}_{(1)}$ we set

$$u_{\sigma_{(1)}} := \int_{\sigma_{(1)}} \underline{u} \cdot \underline{n} \, \mathrm{d}L, \tag{1}$$

where $\underline{n}$ is the normal on $\sigma_{(1)}$. Next, let $\boldsymbol{u}^{(1)} = (u^{(1)}_{\sigma_{(1)}})$ be the vector of all these fluxes. We denote the finite linear space corresponding to unknowns on the edges by $\mathcal{C}^{(1)}$, so $\boldsymbol{u}^{(1)} \in \mathcal{C}^{(1)}$, and define $\mathcal{C}^{(0)}$ (vertices) and $\mathcal{C}^{(2)}$ (cells) analogously.

Using the integral values as variables we can discretize the continuity equation $\nabla \cdot \underline{u} = 0$ without introducing a discretization error. To see this, we consider a two-dimensional cell $\sigma_{(2)} \in \mathcal{C}_{(2)}$, integrate the continuity equation over $\sigma_{(2)}$ and apply the divergence theorem to obtain

$$0 = \sum_{\sigma_{(1)} \in \partial\sigma_{(2)}} o_{\sigma_{(2)}\sigma_{(1)}} u^{(1)}_{\sigma_{(1)}}, \tag{2}$$

where $o_{\sigma_{(2)}\sigma_{(1)}} = 1$ if the orientations of $\sigma_{(2)}$ and $\sigma_{(1)}$ agree and -1 otherwise. For example, if we take $\sigma_{(2)}$ to be oriented such that the normal on $\partial\sigma_{(2)}$ points outward and the normal on $\sigma_{(1)}$ given by the orientation of $\sigma_{(1)}$ points into $\sigma_{(2)}$, then $o_{\sigma_{(2)}\sigma_{(1)}} = -1$.

We can write Eq. (2) for all cells in $\mathcal{C}_{(2)}$ together as $\mathbf{0}^{(2)} = \mathbb{D}^{(2,1)}\boldsymbol{u}^{(1)}$, where $\mathbf{0}^{(2)}$ is the zero element of $\mathcal{C}^{(2)}$ and $\mathbb{D}^{(2,1)} : \mathcal{C}^{(1)} \rightarrow \mathcal{C}^{(2)}$ is defined by $\mathbb{D}^{(2,1)}_{\sigma_{(2)}\sigma_{(1)}} = o_{\sigma_{(2)}\sigma_{(1)}}$, where we extend the definition of $o_{\sigma_{(2)}\sigma_{(1)}}$ by 0 to the instances when $\sigma_{(1)} \notin \partial\sigma_{(2)}$.

Similarly, we define an incidence matrix $\mathbb{D}^{(1,0)} : \mathcal{C}^{(0)} \rightarrow \mathcal{C}^{(1)}$. Let $\boldsymbol{\omega}^{(0)} \in \mathcal{C}^{(0)}$ be a discretization of the 2D scalar vorticity field ω by evaluation on the vertices of the mesh, then the value of $\mathbb{D}^{(1,0)}\boldsymbol{\omega}^{(0)}$ for some edge $\sigma_{(1)}$ is, as a result of the fundamental theorem of calculus, given by

$$[\mathbb{D}^{(1,0)}\boldsymbol{\omega}^{(0)}]_{\sigma_{(1)}} = \int_{\sigma_{(1)}} \underline{\mathrm{rot}}\, \omega \cdot \underline{n}\, \mathrm{d}L = \sum_{\sigma_{(0)} \in \partial\sigma_{(1)}} o_{\sigma_{(1)}\sigma_{(0)}} \omega^{(0)}_{\sigma_{(0)}}, \tag{3}$$

with $\underline{\mathrm{rot}}\, \omega := (\partial_y \omega, -\partial_x \omega)$.

We choose the orientation of the dual cells in relation to the orientation of their corresponding primal cells. From this it follows [6] that the incidence matrices on the dual mesh $\tilde{\mathcal{G}}$, i.e., $\tilde{\mathbb{D}}^{(1,2)} : \tilde{\mathcal{C}}^{(2)} \to \tilde{\mathcal{C}}^{(1)}$ and $\tilde{\mathbb{D}}^{(0,1)} : \tilde{\mathcal{C}}^{(1)} \to \tilde{\mathcal{C}}^{(0)}$ are given by $\tilde{\mathbb{D}}^{(1,2)} = \left(\mathbb{D}^{(2,1)}\right)^T$ and $\tilde{\mathbb{D}}^{(0,1)} = \left(\mathbb{D}^{(1,0)}\right)^T$.

For Eq. (3) to hold for all $\sigma_{(1)} \in \mathcal{C}_{(1)}$ it is crucial that the two boundary points that make up $\partial\sigma_{(1)}$ are in $\mathcal{C}_{(0)}$. The dual mesh $\tilde{\mathcal{G}}$ is not a cell-complex and therefore if $\tilde{\boldsymbol{p}}^{(2)} \in \tilde{\mathcal{C}}^{(2)}$ is, for example, a discretization of the continuous pressure field p on the vertices of the dual mesh, then

$$[\tilde{\mathbb{D}}^{(1,2)}\tilde{\boldsymbol{p}}^{(2)}]_{\tilde{\sigma}_{(1)}} = \int_{\tilde{\sigma}_{(1)}} \nabla p \cdot \mathrm{d}\underline{L}$$

only holds for an edge $\tilde{\sigma}_{(1)}$ for which its end points are again part of the mesh, i.e., for an edge that is dual to a primal edge $\sigma_{(1)}$ that is not part of the boundary mesh $\mathcal{C}^b_{(1)}$.

To get a consistent discretization of the pressure gradient on all the dual edges we extend $\tilde{\mathbb{D}}^{(1,2)}$ to $\bar{\mathcal{G}}$. This extension maps from $\bar{\mathcal{C}}^{(2)}$ to $\bar{\mathcal{C}}^{(1)}$ and we denote it by $\bar{\mathbb{D}}^{(1,2)}$. The part of $\bar{\mathbb{D}}^{(1,2)}$ that maps from $\bar{\mathcal{C}}^{(2)}$ to $\tilde{\mathcal{C}}^{(1)}$ will be used separately in the discretization and denoted by $\bar{\mathbb{D}}^{(1,2)}_i$. It consists of an interior and boundary part, i.e., $\bar{\mathbb{D}}^{(1,2)}_i = [\tilde{\mathbb{D}}^{(1,2)}\ \tilde{\mathbb{I}}^{(1,2)}_b]$. For more details on the construction of $\bar{\mathbb{D}}^{(1,2)}$ and, analogously, $\bar{\mathbb{D}}^{(0,1)}$, see [6].

1.3 The Discrete Hodge Operators

So far, we only discussed the exact discretization of differential operators on both the primal and dual cell-complex in terms of the incidence matrices. These primal and dual incidence matrices map from $\mathcal{C}^{(k)}$ to $\mathcal{C}^{(k+1)}$ and from $\tilde{\mathcal{C}}^{(k+1)}$ to $\tilde{\mathcal{C}}^{(k)}$, respectively. To be able to discretize higher order differential operators, like for example the vector Laplacian, and to end up with a square solvable system of equations we need an interpolation map between the primal and dual meshes.

We introduce interpolation operators $\mathbb{H}^{(k)} : \mathcal{C}^{(k)} \to \tilde{\mathcal{C}}^{(k)}$ that interpolate a field discretized on the k-cells in $\mathcal{C}_{(k)}$ to a consistent discretization of the same field on $\tilde{\mathcal{C}}_{(k)}$. For example, if $\boldsymbol{u}^{(1)} \in \mathcal{C}^{(1)}$ is a discretization of the velocity field on primal edges as in (1), then $\tilde{\boldsymbol{u}}^{(1)} := \mathbb{H}^{(1)}\boldsymbol{u}^{(1)}$ is an approximation of the velocity field discretized on the dual edges in $\tilde{\mathcal{C}}_{(1)}$.

We use the mimetic scalar product matrices [5], which can also be interpreted as interpolation operators [6]. These operators are constructed locally for each 2-cell and the global operator is constructed by an assembly process as in finite element methods. The operators are symmetric positive definite and diagonal in the Cartesian part of the mesh.

1.4 The Numerical Scheme

We use the incidence matrices and discrete Hodge matrices to discretize the incompressible Navier-Stokes equations in space. The discrete variables are velocity, vorticity and pressure. As velocity variables we use the fluxes on the edges, i.e., $\boldsymbol{u}^{(1)} \in \mathcal{C}^{(1)}$. The pressure variables are located on the vertices of the dual cell-complex, i.e., $\bar{\boldsymbol{p}}^{(2)} \in \bar{\mathcal{C}}^{(2)}$. The vorticity variables will be located in the vertices of the primal mesh, i.e., $\boldsymbol{\omega}^{(0)} \in \mathcal{C}^{(0)}$. However, as we will see shortly, we only need vorticity variables in the vertices of the non-Cartesian cut cells.

We discretize the incompressible Navier-Stokes equations in the form

$$\partial_t \underline{u} + \nabla \cdot (\underline{u} \otimes \underline{u}) + \nu \nabla \times \underline{\omega} + \nabla p = 0,$$

with additionally $\underline{\omega} = \nabla \times \underline{u}$ and the continuity equation $\nabla \cdot \underline{u} = 0$. We discretize the momentum equation on all edges of the dual mesh, i.e., we approximate the integral of the momentum equation over every dual edge. In summary, we discretize the spatial part as

$$\begin{bmatrix} \mathbb{H}^{(1)}\partial_t + \mathbb{C}[\boldsymbol{u}^{(1)}] & \mathbb{H}^{(1)}\mathbb{D}^{(1,0)} & \bar{\mathbb{D}}_i^{(1,2)} \\ \tilde{\mathbb{D}}^{(0,1)}\mathbb{H}^{(1)} & -\nu^{-1}\mathbb{H}^{(0)} & 0 \\ (\bar{\mathbb{D}}_i^{(1,2)})^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{(1)} \\ \boldsymbol{\omega}^{(0)} \\ \bar{\boldsymbol{p}}^{(2)} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{0}}^{(1)} \\ -\tilde{\mathbb{I}}_b^{(0,1)}\tilde{\boldsymbol{v}}_b^{(0)} \\ \mathbb{I}^{(1,1)}\boldsymbol{v}_b^{(1)} \end{bmatrix}. \tag{4}$$

The Dirichlet boundary conditions are incorporated in the last line. The matrix $\tilde{\mathbb{I}}^{(1,1)}$ contains only entries equal to 1, 0 or, -1, and consists, just like $(\bar{\mathbb{D}}_i^{(1,2)})^T = [(\mathbb{D}^{(2,1)})^T \ \ \tilde{\mathbb{I}}_b^{(1,2)}]^T$, of two parts. The matrix $\tilde{\mathbb{I}}^{(1,1)}$ is defined such that the equation $(\bar{\mathbb{D}}_i^{(1,2)})^T\boldsymbol{u}^{(1)} = \tilde{\mathbb{I}}^{(1,1)}\boldsymbol{v}_b^{(1)}$ brings forth $\mathbb{D}^{(2,1)}\boldsymbol{u}^{(1)} = \boldsymbol{0}^{(2)}$ and $(\tilde{\mathbb{I}}_b^{(1,2)})^T\boldsymbol{u}^{(1)} = (\tilde{\mathbb{I}}_b^{(1,2)})^T\boldsymbol{v}_b^{(1)}$, i.e., the discrete incompressibility constraint and the definition of the Dirichlet boundary conditions, imposing that the fluxes for the boundary edges are equal to the prescribed values $\boldsymbol{v}_b^{(1)} \in \mathcal{C}_b^{(1)}$.

The no-slip boundary condition for the diffusive term is incorporated in the equation that defines the vorticity. The vector $\tilde{\boldsymbol{v}}_b^{(0)} \in \tilde{\mathcal{C}}_b^{(0)}$ contains the integrals of the tangential velocity over the boundary dual edges. The boundary term $-\tilde{\mathbb{I}}_b^{(0,1)}\tilde{\boldsymbol{v}}_b^{(0)}$ gives the contribution of boundary edges in the dual cell-complex to the vorticity integral over the dual cells. The entries of $\boldsymbol{\omega}^{(0)}$ are actually the vorticity variables multiplied with the viscosity ν. This definition of the variables increases the symmetry of the discrete system.

The equation for the vorticity variables, i.e., $\mathbb{H}^{(0)}\boldsymbol{\omega}^{(0)} = \nu(\tilde{\mathbb{D}}^{(0,1)}\mathbb{H}^{(1)}\boldsymbol{u}^{(1)} + \tilde{\mathbb{I}}_b^{(0,1)}\tilde{\boldsymbol{v}}_b^{(0)})$, is explicit for variables in the Cartesian parts of the mesh, because the restriction of $\mathbb{H}^{(0)}$ to these parts of the mesh is diagonal. Consequently, these variables become superfluous and are not used. As a result, the vorticity variables are only necessary in the cut cells. For convenience we left the corresponding adjustment of the discrete equations out of (4).

For the convection term we use the central approximation introduced in [7], which conserves both momentum and kinetic energy. This discretization of the convection term fits well in the framework presented above [6].

2 Numerical Results: The Flow Around a Cylinder

We consider the benchmark case [8] of the flow around an asymmetrically placed cylinder in a channel. On the inflow boundary a parabolic profile is prescribed, on the top and bottom of the channel no-slip boundary conditions apply and on the outflow boundary we impose zero stress. We test for two meshes that are uniform in the y-direction and non-uniform in the x-direction such that mesh lines are concentrated near the cylinder. We use the standard four-stage explicit Runge-Kutta method for the time integration.

Results are shown for two test cases. The first case is a steady case with average inflow velocity corresponding to Re $= 20$. We compute the drag coefficient c_D, the lift coefficient c_L, the length of the recirculation zone L_a and the pressure difference between the front and back of the cylinder. The results are shown in Table 1.

The second case is an unsteady periodic case with inflow corresponding to Re $= 100$. In this case we compute for one period the maximum drag coefficient $c_{D\max}$, the maximum lift coefficient $c_{L\max}$, the Strouhal number St, and the pressure difference between front and back of the cylinder halfway the period, where the start of the period coincides with the moment that $c_L(t) = c_{L\max}$. The results can be found in Table 2. In Fig. 3 the vorticity field is shown.

Despite the fact that the meshes used are not body-conforming and have relatively few cells near the cylinder, characteristic values close to the benchmark values are found. Especially in the unsteady case the drag and lift coefficients are somewhat

Table 1 Results for the steady test case

$N_x \times N_y \times N_\theta$	$\#\boldsymbol{u}^{(1)}$	$\#\boldsymbol{\omega}^{(1)}$	$\#\tilde{\boldsymbol{p}}^{(2)}$	c_D	c_L	L_a	Δp
160 × 130 × 30	40,656	226	20,732	5.5876	0.0109	0.0858	0.1171
300 × 255 × 60	149,584	444	75,539	5.5835	0.0122	0.0849	0.1178
[8] (lower bound)	–	–	–	5.57	0.0104	0.0842	0.1172
[8] (upper bound)	–	–	–	5.59	0.0110	0.0852	0.1176

Table 2 Results for the unsteady test case

$N_x \times N_y \times N_\theta$	$\#\boldsymbol{u}^{(1)}$	$\#\boldsymbol{\omega}^{(1)}$	$\#\tilde{\boldsymbol{p}}^{(2)}$	$c_{D\max}$	$c_{L\max}$	St	Δp
160 × 130 × 30	40,656	226	20,732	3.3884	1.1601	0.2989	2.5300
300 × 255 × 60	149,584	444	75,539	3.2839	1.0423	0.3003	2.4997
[8] (lower bound)	–	–	–	3.22	0.99	0.295	2.46
[8] (upper bound)	–	–	–	3.24	1.01	0.305	2.50

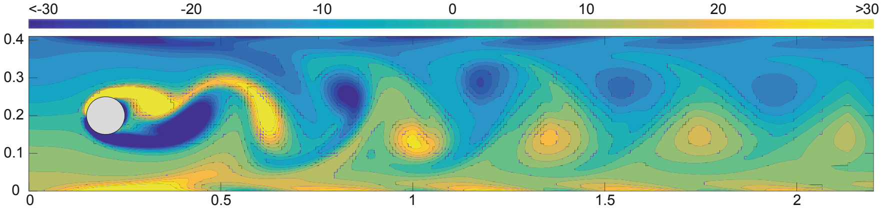

Fig. 3 The vorticity on the finer mesh in the unsteady case for $c_L(t) = c_{L\max}$

overestimated, however, the values corresponding to the finer mesh are already close to the benchmark interval. The fact that we find relatively accurate results despite the use of a non-optimal mesh and the low order of accuracy of the method in the cut cells, is to be attributed to the physical accuracy of the method. The method conserves mass, momentum and energy, even in cut cells, and determines a vorticity corresponding to a physically correct global vorticity. For proofs and numerical verifications of these properties see [6].

Acknowledgements This research is part of the EUROS program, which is supported by NWO domain Applied and Engineering Sciences and partly funded by the Ministry of Economic Affairs.

References

1. F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Phys. Fluids **8**, 2182–2189 (1965)
2. M. Dröge, R. Verstappen, A new symmetry-preserving Cartesian-grid method for computing flow past arbitrarily shaped objects. Int. J. Numer. Methods Fluids **47**, 979–985 (2005)
3. Y. Cheny, O. Botella, The LS-STAG method: a new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties. J. Comput. Phys. **229**, 1043–1076 (2010)
4. Y. Cheny, F. Nikfarjam, O. Botella, Towards a fully 3D version of the LS-STAG immersed boundary/cut-cell method, in *Eighth International Conference on Computational Fluid Dynamics*, paper ICCFD8-2014-0371 (2014)

5. K. Lipnikov, G. Manzini, M. Shaskov, Mimetic finite difference method. J. Comput. Phys. **257**, 1163–1227 (2014)
6. R. Beltman, M.J.H. Anthonissen, B. Koren, Conservative polytopal mimetic discretization of the incompressible Navier-Stokes equations. J. Comput. Appl. Math. **340**, 443–473 (2018). https://doi.org/10.1016/j.cam.2018.02.007
7. B. Perot, Conservation properties of unstructured staggered mesh schemes. J. Comput. Phys. **159**, 58–89 (2000)
8. M. Schäfer, S. Turek, F. Durst, E. Krause, R. Rannacher, Benchmark computations of laminar flow around a cylinder, in *Flow Simulation with High-Performance Computers II* (Vieweg+ Teubner Verlag, Braunschweig, 1996), pp. 547–566