An octree-based solver for the incompressible Navier-Stokes equations with enhanced stability and low dissipation

3 authors:

Maxim Olshanskii
University of Houston
120 PUBLICATIONS   **2,108** CITATIONS

SEE PROFILE

Kirill M Terekhov
Russian Academy of Sciences
26 PUBLICATIONS   **155** CITATIONS

SEE PROFILE

Yuri Vassilevski
Russian Academy of Sciences
164 PUBLICATIONS   **1,447** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

GeRa - hydrogeological modeling code. View project

Automatic Differentiation General Purpose Research Simulator (AD-GPRS) View project

An octree-based solver for the incompressible Navier-Stokes equations with enhanced stability and low dissipation.

Maxim A. Olshanskii

Department of Mathematics, University of Houston, Houston, Texas 77204 and Department of Mechanics and Mathematics, Moscow State University, Moscow 119899

Kirill M. Terekhov

Institute of Numerical Mathematics, Russian Academy of Sciences, Moscow 119333

Yuri V. Vassilevski

Institute of Numerical Mathematics, Russian Academy of Sciences, Moscow 119333 and Moscow Institute of Physics and Technology, Moscow Region, Dolgoprudny 141700

Abstract

The paper introduces a finite difference solver for the unsteady incompressible Navier-Stokes equations based on adaptive cartesian octree grids. The method extends a stable staggered grid finite difference scheme to the graded octree meshes. It is found that a straightforward extension is prone to produce spurious oscillatory velocity modes on the fine-to-coarse grids interfaces. A local linear low-pass filter is shown to reduce much of the bad influence of the interface modes on the accuracy of numerical solution. We introduce an implicit upwind finite difference approximation of advective terms as a low dissipative and stable alternative to semi-Lagrangian methods to treat the transport part of the equations. The performance of method is verified for a set of benchmark tests: a Beltrami type flow, the 3D lid-driven cavity and channel flows over a 3D square cylinder.

Keywords: Octree grid, staggered grid, MAC scheme, incompressible viscous fluid, benchmarking

1. Introduction

Octree grids are gaining popularity in computational mechanics and physics due to the their simple cartesian structure and embedded hierarchy, which makes mesh adaptation, reconstruction and data access fast and easy. As an example, such grids were used for adaptive discontinuous Galerkin and finite volume methods with application to hyperbolic conservation laws, see, e.g., [1,

2, 3, 4]. Fast dynamic remeshing with octree grids makes them a natural choice for the simulation of moving interfaces and free surface flows, see, e.g., [5, 6, 7, 8, 9, 10, 11], octree grids became a standard tool in image processing [12].

In this paper, we study the application of octree grids to numerical solution of the incompressible Navier-Stokes equations, which in non-dimension form read

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{0} \quad \text{in } \Omega \times (0, T),$$

$$\text{div } \mathbf{u} = 0 \quad \text{in } \Omega \times [0, T),$$

$$\mathbf{u}|_{t=0} = \mathbf{u}_0 \quad \text{in } \Omega, \tag{1}$$

$$\mathbf{u}|_{\Gamma_1} = \mathbf{g}, \quad \left(\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p\mathbf{n} \right)\Bigg|_{\Gamma_2} = \mathbf{0},$$

where $\mathbf{u}$, p are unknown fluid velocity and kinematic pressure, ν is the viscosity parameter, Γ_2 is the outflow part of the boundary, $\mathbf{n}$ is the normal vector to Γ_2, and Γ_1 is the rest of $\partial\Omega$. Discretizations on octree (quadtree) cartesian grids already enjoyed an employment in incompressible viscous and inviscid fluid flow computations. Thus, Popinet in [13] developed a finite volume Godunov type scheme, which uses a collocated arrangement of velocity unknowns in cells vertices. Min and Gibou [14, 15] introduced a finite difference method on non-graded octree grids, where all unknowns were collocated to cell vertices and semi-Lagrangian techniques was applied to treat advection. A special stabilization was applied in those papers to avoid spurious pressure modes typical for the collocated arrangement of unknowns. In [6, 7, 11] the finite difference MAC scheme [16, 17, 18], with staggered location of unknowns, was extended to octree meshes.

The advantages of using staggered location of unknowns are the cell-wise enforcement of the incompressibility condition and the well-known pressure stability of such schemes: odd-even oscillatory pressure modes do not emerge. However, such arrangement of unknowns makes building higher order accurate methods on octree grids technically more difficult or computationally expensive. In particular, in papers [6, 7, 11] a first order semi-Lagrangian method was applied to treat advection terms. In this paper, we develop a second order accurate finite difference scheme with the staggered location of unknowns on graded octree cartesian meshes. To reduce numerical dissipation, we build higher order upwind finite difference approximations of advective terms. The discretization invokes simple linear interpolation or quadratic interpolation built upon second degree polynomials of only two variables. This leads to compact nodal stencils and makes implicit treatment of diffusion and advection terms feasible by solving algebraic systems of equations, with sparse matrices, by preconditioned Krylov subspace iterative methods. The implicit advection step removes the Courant stability condition for the time step, which can be otherwise rather restrictive for locally refined meshes. Applying low dissipative approximations reveals, however, a (seemingly) previously unknown issue: For octree staggered grids, the discrete Helmholtz decomposition is unstable due to oscillatory spurious *velocity* modes tailored to course-to-fine grid interfaces. If a fluid viscosity or numerical diffusion is sufficiently large, then such modes are suppressed, otherwise they propagate and destroy the accuracy of numerical solution. In the paper, we introduce a linear low-pass filter which eliminates the spurious modes and improves the accuracy of numerical solution significantly.
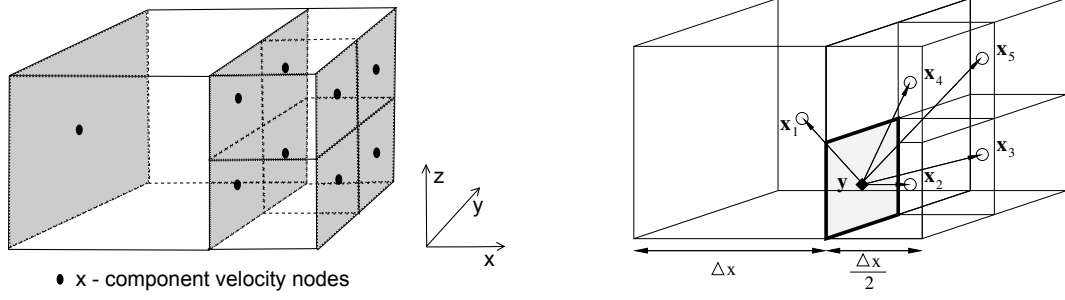
2

● x - component velocity nodes

Figure 1: Left: Each shared face holds a node for velocity x-component. The nodes are located at faces barycenters. Right: Discretization stencil for $\partial p / \partial x$.

The remainder of the paper is organized as follows. In section 2, we consider the discrete Helmholtz decomposition on octree meshes with the staggered location of unknowns. It turns out that the decomposition is specifically unstable. However, it can be stabilized by a local enrichment of discrete pressure space or by introducing a local low-pass filter. The discretizations of advective and diffusion operators are described in section 3. We note right away that only the case of cubic cells is treated in the paper. For curvilinear boundaries, this means a first order staircase approximation. An extension of the discretization for cut cells will be reported elsewhere. Further, in section 4 we collect few time-stepping splitting schemes that are used further in numerical experiments. Finally, in section 5 we present the results of numerical experiments for several benchmark problems: a 3D Beltrami type flow, the 3D lid-driven cavity problem and channel flows around a 3D square cylinder. Conclusions are given in section 6.

2. Staggered grid discretization and the Helmholtz decomposition

For the staggered location of velocity and pressure unknowns on cubic meshes, the pressure degrees of freedom are assigned to cells centers and velocity variables are located at cells faces in such a way that every face stores normal velocity flux. If a face is shared by cells from different grid levels, then velocity degrees of freedom are assigned to the faces centers of fine grid cells (in the case of graded octree mesh, the corresponding face of the coarse grid cell holds 4 velocity unknowns), cf. Figure 1 (left). The staggered FD discretization is well-known, cf. [19], to be stable on a uniform mesh. Although we do not have a rigorous proof, which is a non-trivial exercise even for the uniform grid, results of numerical experiments strongly suggest that the scheme remains *pressure*-stable for octree meshes as well.

The approximation of div $\mathbf{u}$ in the center $\mathbf{x}_V$ of a grid cell V makes use of the Gauss formula

$$\int_V \operatorname{div} \mathbf{u} \, d\mathbf{x} = \int_{\partial V} \mathbf{u} \cdot \mathbf{n} \, ds, \tag{2}$$

where $\mathbf{n}$ is the outward unit normal to the cells boundary. Let $\mathcal{F}(V)$ be the set of all faces F of V, i.e. $\partial V = \cup_{F \in \mathcal{F}(V)} F$, and $\mathbf{x}_F$ denotes the center of $F \in \mathcal{F}(V)$. We define the grid divergence operator by

$$(\operatorname{div}_h \mathbf{u}_h)(\mathbf{x}_V) = |V|^{-1} \sum_{F \in \mathcal{F}(V)} |F|(\mathbf{u}_h \cdot \mathbf{n})(\mathbf{x}_F). \tag{3}$$

3

Thanks to the staggered location of velocity nodes, the fluxes $(\mathbf{u}_h \cdot \mathbf{n})(\mathbf{x}_F)$ are well-defined.

One way to introduce the discrete gradient is to define it as the adjoint of the discrete divergence. We define ∇_h differently based on the formal Taylor expansions. For every internal face we assign the corresponding component of $\nabla_h p$ as follows. Since the octree mesh is graded, there can be only two geometric cases. If a face is shared by two equal-size cells, then the central difference approximation is used. Otherwise, the approximation of p_x at the face center node $\mathbf{y}$ is illustrated in Figure 1 (right): Consider the centers of five surrounding cells $\mathbf{x}_1, \ldots, \mathbf{x}_5$ and expand the pressure value $p(\mathbf{x}_i)$ with respect to $p(\mathbf{y})$:

$$p(\mathbf{x}_i) = p(\mathbf{y}) + \nabla p(\mathbf{y}) \cdot (\mathbf{x}_i - \mathbf{y}) + O(|\mathbf{x}_i - \mathbf{y}|^2).$$

Neglecting the second-order terms, we obtain the following over-determined system:

$$\begin{pmatrix} 1 & -\Delta/2 & \Delta/4 & \Delta/4 \\ 1 & \Delta/4 & 0 & 0 \\ 1 & \Delta/4 & \Delta/2 & 0 \\ 1 & \Delta/4 & 0 & \Delta/2 \\ 1 & \Delta/4 & \Delta/2 & \Delta/2 \end{pmatrix} \begin{pmatrix} p(\mathbf{y}) \\ p_x(\mathbf{y}) \\ p_y(\mathbf{y}) \\ p_z(\mathbf{y}) \end{pmatrix} = \begin{pmatrix} p(\mathbf{x}_1) \\ p(\mathbf{x}_2) \\ p(\mathbf{x}_3) \\ p(\mathbf{x}_4) \\ p(\mathbf{x}_5) \end{pmatrix}, \tag{4}$$

where $\Delta \equiv \Delta x$. The least squares solution of (4) gives the stencil for the x-component of the gradient:

$$p_x(\mathbf{y}) \approx \frac{1}{3\Delta}(p_2 + p_3 + p_4 + p_5 - 4p_1). \tag{5}$$

The finite difference gradient and divergence defined above are similar to what can be found in [7]. Yet the gradient stencil is slightly different: For the cells arrangement given in Figure 1 (right) the reference [7] uses

$$p_x(\mathbf{y}) \approx \frac{1}{2\Delta}(p_2 + p_3 + p_4 + p_5 - 4p_1). \tag{6}$$

We found that using (5) shows slightly better results for smooth solutions compared to (6), although the convergence order and the accuracy were comparable. The super-position of the discrete gradient and divergence operators generally leads to the non-symmetric matrix for the pressure problem. However, the corresponding linear algebraic systems are solved efficiently by a Krylov subspace method with a multigrid preconditioner, see details in section 5.

The key ingredient of many splitting algorithms for the time-integration of the incompressible Navier-Stokes equations is the (discrete) Helmholtz decomposition of a given (grid) vector function $\mathbf{f}$ such that $\int_{\partial\Omega} \mathbf{f} \cdot \mathbf{n} = 0$:

$$\begin{cases} \mathbf{f} = \mathbf{u} + \nabla p, \\ \text{div } \mathbf{u} = 0, \\ \mathbf{u} \cdot \mathbf{n}|_{\partial\Omega} = \mathbf{f} \cdot \mathbf{n}|_{\partial\Omega}. \end{cases} \iff \begin{cases} -\text{div } \nabla p = \text{div } \mathbf{f}, \\ \left.\dfrac{\partial p}{\partial \mathbf{n}}\right|_{\partial\Omega} = 0, \\ \mathbf{u} = \mathbf{f} - \nabla p. \end{cases} \tag{7}$$

It occurs that for the finite difference discretization as described above, the decomposition is not stable in the following sense. For a given smooth function $\mathbf{f}$, the error for $\mathbf{u}$ in the discrete decomposition may significantly increase with every level of *local* refinement. This is illustrated below

Table 1: Errors for the discrete Helmholtz decomposition on uniform and one-level refined grids.

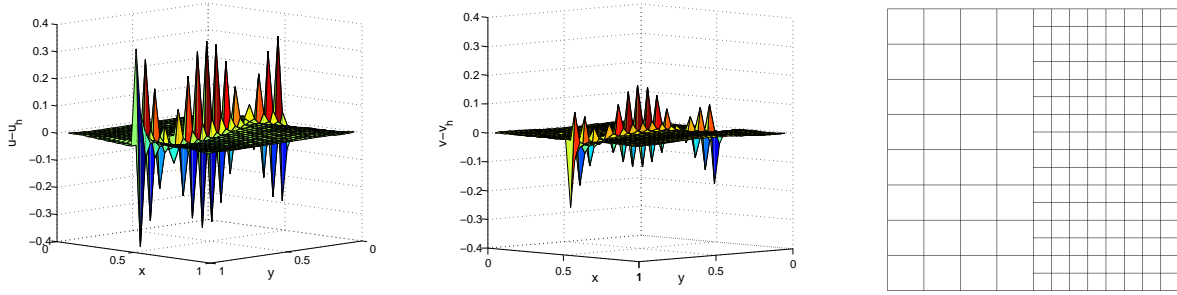| quantity	mesh size h							
	1/8	1/16	1/32	1/64	1/8	1/16	1/32	1/64
	uniform mesh				locally refined mesh			
$\|\mathbf{u} - \mathbf{u}_h\|_{L^\infty}$	1.1e-1	2.9e-2	1.1e-2	3.8e-3	1.4e-1	7.0e-1	3.5e-1	1.8e-1
$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	6.7e-2	1.7e-2	4.2e-3	1.1e-3	3.5e-1	1.2e-1	4.2e-2	1.5e-2
$\|p - p_h\|_{L^2}$	2.5-2	6.4e-3	1.6e-3	4.0e-4	1.4e-2	3.3e-3	8.1e-4	2.0e-4

Figure 2: The velocity error is dominated by the div-free modes occurring on the coarse-to-fine grids interface. The left figure shows the u-component of the error, the middle figure shows the v-component, and the right figure shows the grid for $h = 1/8$.

by a simple 2D example and one refinement level (another example of a 3D problem and more levels of refinement can be found in the next section). We consider two meshes in $\Omega = (0, 1)^2$: The first is uniform with the mesh size h, the second mesh results from the first one by applying one refinement step for cells in the right half of the square, $x > \frac{1}{2}$ (Figure 2, right). The function $\mathbf{f}$ is such that the exact solution to (7) reads

$$u = \sin\left(\frac{2\pi(e^x - 1)}{e - 1}\right)\left(1 - \cos\left(\frac{2\pi(e^{ay} - 1)}{e^a - 1}\right)\right)\frac{1}{2\pi}\frac{e^x}{(e - 1)},$$

$$v = \left(1 - \cos\left(\frac{2\pi(e^x - 1)}{e - 1}\right)\right)\sin\left(\frac{2\pi(e^{ay} - 1)}{e^a - 1}\right)\frac{a}{2\pi}\frac{e^{ay}}{(e^a - 1)},$$

$$p = a\cos\left(\frac{2\pi(e^x - 1)}{e - 1}\right)\cos\left(\frac{2\pi(e^{ay} - 1)}{e^a - 1}\right)\frac{e^{a+1}}{(e - 1)(e^a - 1)},$$

with $a = 0.1$, $\mathbf{u} = (u, v)^T$. The discrete decomposition (7) on two grids was computed and the L^∞-norm and the (discrete) L^2-norm of the errors for $\mathbf{u}$ and p are shown in Table 1 (for the locally refined grid, h denotes the size of the coarse grid cells). The results in Table 1 show that introducing more degrees of freedom in a local way may lead to the significant loss of accuracy in velocity. The first explanation of this error growth could be the formal decrease of the discretization consistency order from the second to the first one at the nodes on the interface between the coarse and fine

5

grids. However, a closer look at the velocity error reveals the appearance of specific interface modes, which dominate the entire error function, see Figure 2 (left). These are local discretely div-free modes, which occur on the coarse-to-fine grids interface and are subgrid modes for the coarse grid, see Figure 3 (left).
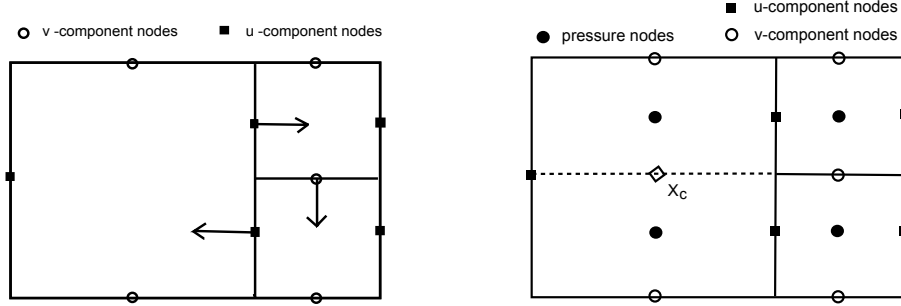

Figure 3: Left: A single local div-free mode, which is subgrid for the coarse grid cell. Right: Introducing additional pressure d.o.f. for the coarse cell next to fine cells. This yields a stronger discrete div-free condition and precludes the occurrence of subgrid div-free modes such as shown in the left figure.

A straightforward way to filter out the specific modes is to enforce a stronger divergence free condition such that these modes are no longer discretely divergence free. To demonstrate this, we introduce extra pressure degrees of freedom in the coarse grid cell near the coarse-to-fine grid interface, see Figure 3 (right). For this coarse cell, the div-free condition is now enforced separately for two sub-cells, by interpolating the v-component into the center point $\mathbf{x}_c$. The discrete gradient operator is altered in the obvious way. We call this method the "pressure enrichment" and show results for the discrete Helmholtz decomposition (7) in the left part of Table 2: The error in velocity is substantially reduced. However, for general 3D octree meshes the pressure enrichment may not be the best method for filtering the subgrid modes for the following reasons: When a coarse-grid cell is neighboring fine-grid cells from different sides, introducing up to 7 extra pressure d.o.f. may be required. This complicates the scheme and can lead to the pressure instability.

Table 2: Errors for the discrete Helmholtz decomposition, with one-level refined grids, using pressure enrichment and differential filter stabilizations.

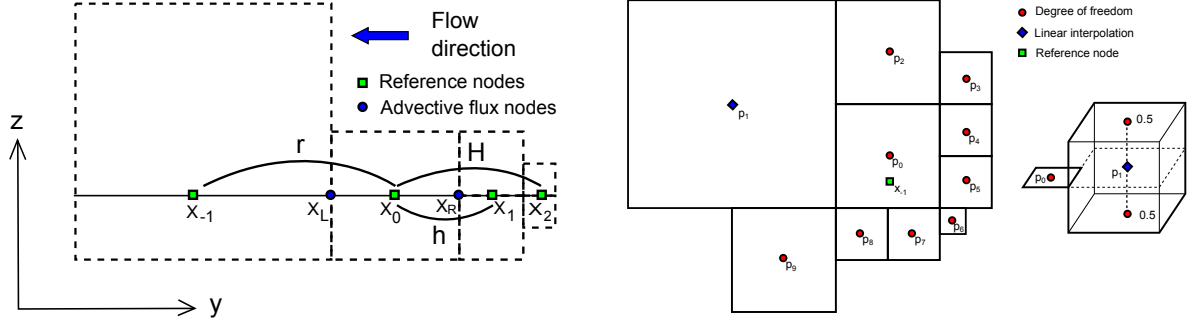| quantity	mesh size h							
	1/8	1/16	1/32	1/64	1/8	1/16	1/32	1/64
	pressure enrichment				differential filter			
$\|\mathbf{u} - \mathbf{u}_h\|_{L^\infty}$	1.4e-1	8.1e-2	4.3e-2	2.2e-2	1.2e-1	4.9e-1	2.2e-2	1.0e-2
$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	5.1e-2	1.5e-2	4.5e-3	1.5e-3	3.8e-2	1.0e-2	3.1e-3	1.0e-3
$\|p - p_h\|_{L^2}$	1.1e-2	2.8e-3	6.9e-4	1.7e-4	1.1e-2	2.6e-3	6.3e-4	1.6e-4

Figure 4: Left: Reference points for the third-order upwind approximation of advection. This illustration is for the derivative tangential to a face, where velocity degree of freedom is located. Right: An example of the set $\mathcal{P}$, if the velocity value (u_{-1}) is sought in the reference point $\mathbf{x}_{-1}$. All points $\mathbf{p}_0$, $\mathbf{p}_i$, $i \geq 2$, appear to be velocity nodes in this example. To assign a velocity value to p_1 one uses the linear interpolation.

The oscillatory behavior of the interface velocity error also suggests the application of a low-pass filter as an alternative way to enhance the stability of the discrete Helmholtz decomposition. To test the idea, we introduce the interface diffusion in (7) as

$$\begin{cases} \mathbf{f} = (I - \alpha h^2 \Delta_\Gamma)\mathbf{u} + \nabla p, \\ \operatorname{div} \mathbf{u} = 0, \\ \mathbf{u} \cdot \mathbf{n}|_{\partial\Omega} = \mathbf{f} \cdot \mathbf{n}|_{\partial\Omega}. \end{cases} \quad (8)$$

Here Δ_Γ is the vector Laplace-Beltrami operator for the coarse-to-fine grids interface Γ_{cf} ($\Delta_\Gamma v := v_{yy}|_{\Gamma_{cf}}$ for our test example), h is the size of coarse cells and $\alpha \geq 0$ is a parameter. Thus, (8) is closely related to the idea of low-pass differential filters, see, e.g., [20]. The results for the discrete decomposition (8) with $\alpha = 4$ are shown in the right part of Table 2. The error in velocity is reduced versus the non-stabilized case and the accuracy is comparable to the pressure enrichment stabilization (as usual for stabilized method, the parameter α has to be tuned). Note that introducing interface diffusion in (8) makes the corresponding pressure operator, $\operatorname{div}(I - \alpha h^2 \Delta_\Gamma)^{-1}\nabla$, non-local and the corresponding matrix is not sparse. Hence, the repeated solution of the pressure problem becomes expensive. To avoid this, in the splitting scheme for the Navier-Stokes equations we shall introduce *explicit* filter, rather than implicit as in (8).

We conclude that the error of the discrete Helmholtz decomposition on octree-refined meshes may be dominated by specific divergence-free coarse-to-fine grid interface velocity modes. However, using simple low-pass local filters may reduce the error significantly. In the next section, we define the remaining discrete operators and introduce a low-pass filter for unsteady flow computations.

3. Advection, diffusion and filtering

First, we describe how the advection terms are treated. Consider the advection term for the velocity x-component: $\mathbf{a} \cdot \nabla u$. We distinct between derivatives in the normal and the tangential directions to a face, where the velocity degree of freedom is located. Consider the discretization

of the tangential derivative, $a_y \frac{\partial u}{\partial y}$, in the face center $\mathbf{x}$. Depending on the sign of $a_y(\mathbf{x})$, (a_y is computed in $\mathbf{x}$ with the help of an interpolation procedure described in Remark 3.2), four 'reference' points ($\mathbf{x}_{-1}, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_0 := \mathbf{x}$) are taken as shown in Figure 4 (left). Note that $\mathbf{x}_{-1}, \mathbf{x}_1$, and $\mathbf{x}_2$ are not necessarily grid nodes. Values u_{-1}, u_1, and u_2 in these nodes are then defined based on the following interpolation procedure. If the reference point belongs to a cell *smaller* than the cell of $\mathbf{x}_0$ (points $\mathbf{x}_1$ and $\mathbf{x}_2$ in the figure), then the linear interpolation between the *two* nodes of adjunct faces is used. If the node belongs to a cell *larger* than the cell of $\mathbf{x}_0$ (point $\mathbf{x}_{-1}$ in the figure), then one considers the plane P such that $\mathbf{x}_{-1} \in P$ and $P \perp Oy$. Further, consider the cross sections of P with the cell possessing $\mathbf{x}_{-1}$ (denote this cell by V) and all cells sharing a face or an edge with V. Any such cross-section can be either a square or the empty set. All center points of square cross-sections form the set $\mathcal{P}$, cf. Figure 4 (right). Now we assign u-values to all points from $\mathcal{P}$. Due to the octree mesh structure, any $\mathbf{p} \in \mathcal{P}$ can be either a u-node (thus the value is assigned trivially) or it lies on the x-midline of a cell (denote this cell by V'). In the latter case, we first assign u-values to the centers of two x-faces of V': if the center is not a u-node, we take the average of velocity values from four u-nodes on this face. Further, we take the linear interpolation of these values from the face centers and assign a u-value to $\mathbf{p}$. When all $\mathbf{p} \in \mathcal{P}$ receive their u-values, the least-square second order interpolant Q_2 is computed for the set of $\mathbf{p} \in \mathcal{P}$ (Q_2 is the second order polynomial of y and z variables) and $u_{-1} := Q_2(\mathbf{x}_{-1})$. Once the values $\{u_i\}$, $i = -1, \ldots, 2$, are defined, we compute $a_y \frac{\partial u}{\partial y}$ at $\mathbf{x}$ using the third order upwind discretization stencil:

$$c_1 = -\frac{hH}{r(h+r)(r+H)}, \quad c_3 = \frac{rh}{H(r+H)(H-h)}, \quad c_2 = -\frac{r}{h(r+h)} - \frac{1}{H} + \frac{h}{(r+h)(r+H)}$$

$$u_L := c_1 u_1 + c_2 u_0 + c_3 u_{-1}, \quad u_R := c_1 u_2 + c_2 u_1 + c_3 u_0,$$

$$a_y \frac{\partial u}{\partial y}(\mathbf{x}) \approx a_y(\mathbf{x})(u_R - u_L),$$

where notations r, h, H are illustrated in Figure 4 (left). A special care is taken near boundaries, since the reference point $\mathbf{x}_2$ may be not available. In this case, we use the second order difference:

$$a_y \frac{\partial u}{\partial y}(\mathbf{x}) \approx a_y(\mathbf{x})\Big(\frac{r}{h(r+h)}u_1 - \frac{r^2 - h^2}{hr(h+r)}u_0 - \frac{h}{r(h+r)}u_{-1}\Big).$$

The finite difference approximation of the derivative in the normal direction, $a_x \frac{\partial u}{\partial x}$, is constructed in the similar manner. The alterations to the algorithm described above are the following: a_x is defined in $\mathbf{x}$ (no interpolation required), and the reference points $\mathbf{x}_{-1}, \mathbf{x}_1, \mathbf{x}_2$ are always lying on cells x-faces (although not necessarily in the centers and the interpolation is done as above).

Remark 3.1. Note that the interpolation procedure invokes computing the second order polynomial of only 2 variables. Moreover, only a limited number of different auxiliary matrices should be inverted to find the interpolation polynomials for a given grid. It may look inconsistent that some reference nodes from $\{\mathbf{x}_{-1}, \mathbf{x}_1, \mathbf{x}_2\}$ receive velocity values by the simplest linear interpolation, while others by more elaborated interpolation procedure. From numerical experiments we observed that using simple linear interpolation for fictitious nodes located in cells larger than current cell, where $\mathbf{x}_0$ is located, leads to perceptibly less accurate results, especially if a larger cell lies from the upwind side of the current cell. Applying the quadratic interpolation for larger cells, as described above, was found to produce stable and accurate disretization.
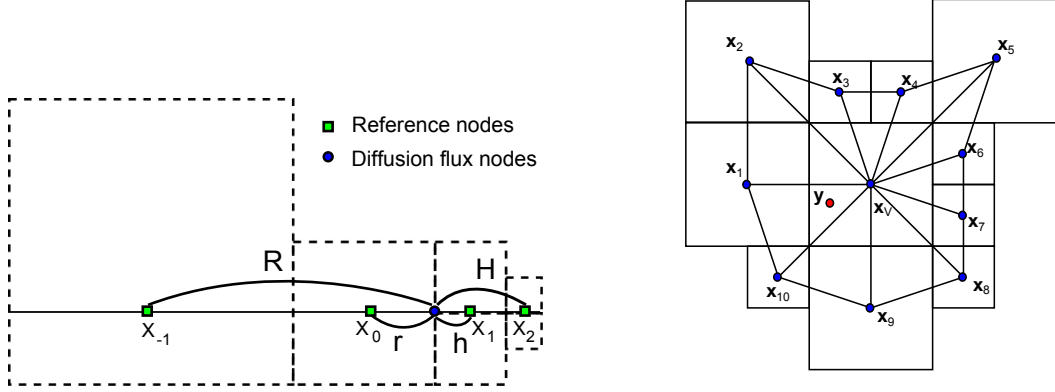
Figure 5: Left: Reference points for the diffusion flux approximation; Right: $u_h(\mathbf{y})$ is defined by a linear interpolation based on the fan triangulation with the center in $\mathbf{x}_V$, i.e. interpolation of $u_h(\mathbf{x}_V)$, $u_h(\mathbf{x}_1)$, $u_h(\mathbf{x}_{10})$ in this example.

Now we explain how the finite difference approximation of viscous terms is computed. Consider a velocity component u node $\mathbf{x}$ lying on a face F and define a cubic control volume V', such that $\mathbf{x}$ is the center of V' and F is a middle cross section of V'. We set

$$(\Delta_h u)(\mathbf{x}) = -|V'|^{-1} \sum_{F' \in \mathcal{F}(V')} |F'|(\nabla_h u \cdot \mathbf{n})(\mathbf{y}_{F'}). \tag{9}$$

In order to approximate the diffusion flux in the center $\mathbf{y}_{F'}$ of $F' \in \mathcal{F}(V')$, we take four reference points $(\mathbf{x}_{-1}, \mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2)$ as shown in Figure 5 (left). Velocity values u_{-1}, u_0, u_1, and u_2 are assigned to reference points in the same way as for the advective terms described above. Using the notation from Figure 5, the formal third order approximation of the diffusion flux $(\nabla u \cdot \mathbf{n})$ can be written out as

$$\begin{aligned}
(\nabla u \cdot \mathbf{n}) \approx D^{-1}[&(h^2 H^3 + h^3 R^2 - H^3 R^2 + h^2 R^3 - H^2 R^3 - h^3 H^2)u_0 \\
&+ (H^3 R^2 + r^3 R^2 + H^2 R^3 - r^2 R^3 - H^3 r^2 - H^2 r^3)u_1 \\
&+ (h^3 r^2 + h^2 r^3 - h^3 R^2 - r^3 R^2 - h^2 R^3 + r^2 R^3)u_{-1} \\
&+ (h^3 H^2 - h^2 H^3 - h^3 r^2 + H^3 r^2 - h^2 r^3 + H^2 r^3)u_2],
\end{aligned}$$

with $D = (H - h)(h + r)(H + r)(h + R)(H + R)(R - r)$. If the reference point in $\mathbf{x}_2$ is not available, we use the point $\mathbf{x}_{-2}$.

Finally, following the discussion of the previous section, we define the low-pass filter G acting on the coarse-to-fine grid interface Γ_{cf}:

$$G \circ u(\mathbf{x}) = \begin{cases} \dfrac{1}{4} \displaystyle\sum_{i=1}^{4} u(\mathbf{x}_i) & \text{if } \mathbf{x} \in \Gamma_{cf}, \\[2mm] u(\mathbf{x}) & \text{otherwise,} \end{cases} \quad \text{for every internal velocity component node } \mathbf{x}.$$

Here Γ_{cf} denotes the union of all octree cells faces, which are shared by cells of different sizes; $\mathbf{x}_i$ are four velocity nodes lying on the same large cells face as $\mathbf{x}$ (obviously $\mathbf{x} \in \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$).

9

Remark 3.2. We recall that for computing finite difference advection derivative $\mathbf{a} \cdot \nabla \mathbf{u}$, we need the approximation of every component of the grid vector function $\mathbf{a}$ in all velocity nodes. For a given point $\mathbf{y}$ in computational domain we evaluate $\mathbf{a}(\mathbf{y})$ as follows. Assume $\mathbf{y}$ belongs to a cell V and we are interested in interpolating the x-component of velocity to $\mathbf{y}$, i.e. $a_x(\mathbf{y})$. Consider a plane $\mathcal{P}$ such that $\mathbf{y} \in \mathcal{P}$ and $\mathcal{P}$ is orthogonal to the Ox axis. Let $\mathbf{x}_V \in \mathcal{P}$ be the orthogonal projection of the center of V on $\mathcal{P}$ and $\mathbf{x}_k$, $k = 1, \dots, m$, $m \leq 12$, are the projections of centers of all cells sharing a face with V. The values $a_x(\mathbf{x}_V)$ and $a_x(\mathbf{x}_k)$ can be defined by a linear interpolation of the velocity values at u-nodes. Once $a_x(\mathbf{x}_V)$ and $a_x(\mathbf{x}_k)$, $k = 1, \dots, m$, are computed, we consider the triangle fan based on $\mathbf{x}_V$ and $\mathbf{x}_k$, $k = 1, \dots, m$, as shown in Figure 5 (right). Now $a_x(\mathbf{y})$ is defined by a linear interpolation between the values of a_x in the vertices of the triangle, which contains $\mathbf{y}$.

Note that the discretization of momentum and continuity equations is based on control volumes. The method is, however, not consistent with a classical finite volume approach, since for the momentum equation the control volumes may fail to cover the entire computational domain.

4. Numerical time-integration

Our method of choice for numerical time-integration is the semi-implicit splitting scheme (also known as projection scheme [21]): Given $\mathbf{u}^n$, p^n approximating $\mathbf{u}(t)$, $p(t)$, find approximations $\mathbf{u}^{n+1}$, p^{n+1} to $\mathbf{u}(t + \Delta t^n)$, $p(t + \Delta t^n)$ in several steps. First, solve for auxiliary velocity $\widetilde{\mathbf{u}^{n+1}}$ the convection-diffusion problem with the filter acting on the advection terms,

$$\begin{cases} \dfrac{\alpha \widetilde{\mathbf{u}^{n+1}} + \beta \mathbf{u}^n + \gamma \mathbf{u}^{n-1}}{\Delta t^n} + G \circ (\mathbf{u}^n + \xi(\mathbf{u}^n - \mathbf{u}^{n-1})) \cdot \nabla \widetilde{\mathbf{u}^{n+1}}) - \nu \Delta \widetilde{\mathbf{u}^{n+1}} = -\nabla p^n, \\[2mm] \widetilde{\mathbf{u}^{n+1}}|_{\Gamma_1} = \mathbf{g}, \quad \dfrac{\partial \widetilde{\mathbf{u}^{n+1}}}{\partial \mathbf{n}}\bigg|_{\Gamma_2} = 0. \end{cases} \tag{10}$$

Here $\xi = \Delta t^n / \Delta t^{n-1}$, $\alpha = 1 + \xi/(\xi + 1)$, $\beta = -(\xi + 1)$, $\gamma = \xi^2/(\xi + 1)$. Next, project $\widetilde{\mathbf{u}^{n+1}}$ on the divergence-free space to recover $\mathbf{u}^{n+1}$:

$$\begin{cases} \alpha(\mathbf{u}^{n+1} - \widetilde{\mathbf{u}^{n+1}})/\Delta t^n - \nabla q = 0, \\[1mm] \operatorname{div} \mathbf{u}^{n+1} = 0, \\[1mm] \mathbf{n} \cdot \mathbf{u}^{n+1}|_{\Gamma_1} = \mathbf{n} \cdot \mathbf{g}, \quad q|_{\Gamma_2} = 0. \end{cases} \tag{11}$$

The problem (11) is reduced to the Poisson problem for q:

$$\begin{cases} -\Delta q = \alpha \operatorname{div} \widetilde{\mathbf{u}^{n+1}}/\Delta t^n, \\[2mm] q|_{\Gamma_2} = 0, \quad \dfrac{\partial q}{\partial \mathbf{n}}\bigg|_{\Gamma_1} = 0. \end{cases} \tag{12}$$

Finally, update the pressure:

$$p^{n+1} = p^n - q + \nu \operatorname{div} \widetilde{\mathbf{u}^{n+1}}. \tag{13}$$

10

The 'extra' divergence term in the pressure correction step (13) is known, see, e.g., [22, 23], to reduce numerical boundary layers in the pressure.

Further, we refer to the scheme (10)–(13) as *the linearized BDF2 projection scheme*, since it relies on BDF2 time discretization of the momentum equation at time t^{n+1} and the linearization of the advective terms. Note that the step (10) is implicit. The matrix of the corresponding linear algebraic system is non-symmetric, but it is well conditioned thanks to the scaled identity matrix resulting from the term $\frac{\alpha \widetilde{\mathbf{u}^{n+1}}}{\triangle t^n}$. A preconditioned Krylov subspace iterative method turns out to be efficient solver, see section 5. The implicit diffusion-advection step largely relaxes the Courant condition for the time step, which is now restricted by accuracy, rather than by stability requirements.

For a uniform time step, $\triangle t^n = \triangle t$, $n = 1, 2, \ldots$, the splitting scheme (10)–(13) is found, for example, in [22] § 3.3. In that paper, the method was shown to be second order accurate in time, if $\Gamma_2 = \emptyset$. We shall demonstrate in the series of numerical experiments that the scheme retains second order accuracy if $\triangle t$ varies smoothly. In the case of outflow boundary conditions, building a second order accurate stable pressure projection method is a well-known problem, see, e.g., [24, 25]. It is not our intention to address this problem in the present paper. If one sets $\nu \frac{\partial \widetilde{\mathbf{u}^{n+1}}}{\partial \mathbf{n}}\Big|_{\Gamma_2} = p^n \mathbf{n}$ in (10), then Guermond et al. [24] proved that the splitting method is up to $\frac{3}{2}$ order the accurate (the actual order depends on a certain regularity index). However, for such explicit treatment of pressure on outflow boundary, our experiments show instability if ν is not sufficiently large. Therefore, we modified the splitting of boundary conditions to ensure the numerical stability for higher Reynolds numbers. We are not aware of a convergence analysis for such modified splitting. In numerical experiments, no significant upstream influence was observed for outflow conditions we use: Developed vortex structures leave the computational domain through outflow boundary smoothly and retaining their shapes.

Remark 4.1. Instead of the implicit filter in the projection step, as in (8), we use the explicit filter G acting on advection terms. The rationality behind such choice is that computing projection, with implicit differential filter, on every time step is computationally 'expensive'. Let us briefly explain why the explicit filter still is efficient. Noting that $G \circ \nabla = \nabla$, we combine (10) and (11) to get

$$\frac{\alpha \mathbf{u}^{n+1} + \beta \mathbf{u}^n + \gamma \mathbf{u}^{n-1}}{\triangle t^n} + G \circ \left[(\mathbf{u}^n + \xi(\mathbf{u}^n - \mathbf{u}^{n-1})) \cdot \nabla \widetilde{\mathbf{u}^{n+1}}) + \nabla(p^n - q) \right] = \nu \Delta \widetilde{\mathbf{u}^{n+1}}. \qquad (14)$$

Hence, for inviscid fluid, if $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$ are free of spurious modes, then the same is true for $\mathbf{u}^{n+1}$. For $\nu > 0$, the viscous term can be the source of spurious modes. However for small ν, we observe that the production of such modes is not significant and the scheme remains stable, while for large ν, there is enough physical viscosity to damp the spurious modes.

For the purpose of comparison, we consider few alternative splitting schemes that have been shown to be efficient in other settings. Thus, consider the second order projection scheme, with the semi-Lagrangian method for treating the advection terms, as introduced in [26, 27]. The

differences from the scheme (10)–(13) are the following. The first predictor step (10) now reads

$$
\begin{cases}
\dfrac{\alpha\widetilde{\mathbf{u}^{n+1}} + \beta\mathbf{u}_d^n + \gamma\mathbf{u}_d^{n-1}}{\triangle t} - \nu\Delta\widetilde{\mathbf{u}^{n+1}} = -\nabla p^n, \\[2mm]
\widetilde{\mathbf{u}^{n+1}}|_{\Gamma_1} = \mathbf{g}, \qquad \nu\dfrac{\partial\widetilde{\mathbf{u}^{n+1}}}{\partial\mathbf{n}}\bigg|_{\Gamma_2} = 0,
\end{cases}
\tag{15}
$$

where $\mathbf{u}_d^n$ and $\mathbf{u}_d^{n-1}$ are given by the semi-Lagrangian method: $\mathbf{u}_d^n(\mathbf{x}^{n+1}) = \mathbf{u}(t_n, \mathbf{x}_d^n)$, where $\mathbf{x}^{n+1}$ is the grid node and $\mathbf{x}_d^n$ is found as a departing point at time t_n of a characteristic that passes through $\mathbf{x}^{n+1}$ at time t_{n+1}. The equation for characteristics is integrated using the second order method:

$$
\begin{aligned}
\mathbf{y} &= \mathbf{x}^{n+1} - \frac{\triangle t^n}{2}\,\mathbf{u}^n(\mathbf{x}^{n+1}), \\[2mm]
\mathbf{x}_d^n &= \mathbf{x}^{n+1} - \triangle t^n\left(\left(1 + \frac{\xi}{2}\right)\mathbf{u}^n(\mathbf{y}) - \frac{\xi}{2}\mathbf{u}^{n-1}(\mathbf{y})\right).
\end{aligned}
\tag{16}
$$

Similarly, one sets $\mathbf{u}_d^{n-1}(\mathbf{x}^{n+1}) = \mathbf{u}(t_{n-1}, \mathbf{x}_d^{n-1})$, where the departure point $\mathbf{x}_d^{n-1}$ is found from:

$$
\begin{aligned}
\mathbf{y} &= \mathbf{x}^{n+1} - \frac{\triangle t^n + \triangle t^{n-1}}{2}\mathbf{u}^n(\mathbf{x}^{n+1}), \\[2mm]
\mathbf{x}_d^{n-1} &= \mathbf{x}^{n+1} - \frac{\triangle t^n + \triangle t^{n-1}}{2}\left((1 + \xi)\mathbf{u}^n(\mathbf{y}) + (1 - \xi)\mathbf{u}^{n-1}(\mathbf{y})\right).
\end{aligned}
\tag{17}
$$

Of course, $\mathbf{x}_d^n$ and $\mathbf{x}_d^{n-1}$ are not necessarily grid nodes and an interpolation should be done to define $\mathbf{u}(t_n, \mathbf{x}_d^n)$ and $\mathbf{u}(t_{n-1}, \mathbf{x}_d^{n-1})$. In [27] the quadratic Hermite interpolation was used to interpolate between unknowns located in the vertices of cubic cells. To interpolate between face-centered unknowns on octree grids, in [6, 7] the piecewise linear interpolation was applied. To the best of our knowledge, an extension of higher order semi-Lagrangian methods to staggered octree grids is not available in the literature. For the purpose of comparison, we shall use both linear and quadratic interpolation. We stress that building more accurate and stable semi-Lagrangian methods, for example, based on non-linear oscillatory-free interpolation procedures [28], may be another way of developing staggered octree grid schemes. However, such developments are not within the scope of the present paper. In what follows, the scheme (15), (12), (13) is referred to as *the BDF2 with semi-Lagrangian step*.

A popular alternative, e.g., [29], to BDF2 scheme is the second order projection method often attributed to van Kan [30]. This method approximates the equations at time $t^{n+1/2}$. On the first step of the method, one finds $\widetilde{\mathbf{u}^{n+1}}$ from

$$
\begin{cases}
\dfrac{\widetilde{\mathbf{u}^{n+1}} - \mathbf{u}^n}{\triangle t^n} + G \circ \left(\mathbf{u}^n + \dfrac{\xi}{2}(\mathbf{u}^n - \mathbf{u}^{n-1})\right) \cdot \nabla\dfrac{\widetilde{\mathbf{u}^{n+1}} + \mathbf{u}^n}{2}) - \nu\Delta\dfrac{\widetilde{\mathbf{u}^{n+1}} + \mathbf{u}^n}{2} = -\nabla p^n, \\[3mm]
\widetilde{\mathbf{u}^{n+1}}|_{\Gamma_1} = \mathbf{g}, \qquad \nu\dfrac{\partial\widetilde{\mathbf{u}^{n+1}}}{\partial\mathbf{n}}\bigg|_{\Gamma_2} = 0.
\end{cases}
\tag{18}
$$

We shall refer to this method as *the linearized van Kan (VK) projection scheme*.

5. Numerical experiments

In this section, the performance of the method is verified for a set of benchmark tests. We compare several options for spacial and temporal discretizations. First, a smooth 3D Beltrami type flow with known analytical solution is considered. Next we compute the 3D lid-driven cavity flow for Re={100, 400, 1000} and channel flows around a 3D square cylinder for Re={20, 100} and a variable Reynolds number.

5.1. Example with an analytical solution

To assess the accuracy of the scheme on smooth solutions, we consider the well known Ethier-Steinman exact NSE solution from [31]. This problem was developed as a 3D analogue to the Taylor vortex problem, for the purpose of benchmarking. Although unlikely to be physically realized, it is a good test problem because it is an exact NSE solution and has non-trivial vortical structure. For chosen parameters a, d and viscosity v, the exact NSE solution is given on $[-1, 1]^3$ by

$$
\begin{aligned}
u &= -a \left(e^{ax} \sin(ay + dz) + e^{az} \cos(ax + dy) \right) e^{-vd^2 t} \\
v &= -a \left(e^{ay} \sin(az + dx) + e^{ax} \cos(ay + dz) \right) e^{-vd^2 t} \\
w &= -a \left(e^{az} \sin(ax + dy) + e^{ay} \cos(az + dx) \right) e^{-vd^2 t} \\
p &= -\frac{a^2}{2} (e^{2ax} + e^{2ay} + e^{2az} + 2 \sin(ax + dy) \cos(az + dx) e^{a(y+z)} \\
&\quad + 2 \sin(ay + dz) \cos(ax + dy) e^{a(z+x)} \\
&\quad + 2 \sin(az + dx) \cos(ay + dz) e^{a(x+y)}) e^{-2vd^2 t}.
\end{aligned}
$$

In our experiment we set $a = \pi/4$, $d = \pi/2$ and vary v.

First we compare the performance of different temporal discretizations if the spacial grid is uniformly refined. The errors in velocity and pressure are measured at time $t = 0.1$ and the results are shown in Table 3. The time steps for 'BDF2 with semi-Lagrangian (quadratic interpolation)' was set two times smaller than for other methods, otherwise we observed no convergence with this method. All schemes except 'BDF2 with semi-Lagrangian (linear interpolation)' demonstrated the expected second order of convergence. The semi-Lagrangian method with quadratic interpolation demonstrates the second order of convergence in L^2, but the convergence deteriorates in L^∞ norm. This may indicate the loose of the monotonicity of the scheme and non-physical oscillations in numerical solution. Further, we will see that this is indeed the case for the example of the flow around a square cylinder. 'BDF2 with FD advective fluxes' and VK show very similar accuracy, although the results with VK are slightly more accurate. Thus, our method of choice for further experiments is the 'BDF2 with FD advective fluxes' as potentially more robust than the van Kan scheme, while demonstrating similar accuracy (we note that the van Kan scheme is based on the trapezoidal rule, which is well known to require certain care, when applied to flow equations [32]).

In section 3 we built an upwind discretization of the advection terms. We shall refer to it as the third order upwind (TOU) discretization (the 'order' is related to uniformly refined grids). Since the entire scheme is of the second order, one may be interested in using the second order upwind stencil for advective derivatives, next referred to as SOU. Table 4 compares the accuracy of two

13

Table 3: Errors for different temporal discretizations on uniform meshes.

viscosity	$\nu = 10^{-5}$			$\nu = 10^{-2}$		
mesh size h	1/16	1/32	1/64	1/16	1/32	1/64
time step $\triangle t$	1/200	1/400	1/800	1/200	1/400	1/800
	BDF2 with FD advective terms					
$\|\mathbf{u} - \mathbf{u}_h\|_{L^\infty}$	1.7e-3	4.2e-4	1.1e-4	1.7e-3	3.5e-4	7.6e-5
$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	4.0e-4	9.7e-5	2.4e-5	4.0e-4	9.5e-5	2.3e-5
$\|p - p_h\|_{L^2}$	8.4e-3	2.2e-3	5.7e-4	8.0e-3	2.0e-3	4.7e-4
	BDF2 with semi-Lagrangian (linear interpolation)					
$\|\mathbf{u} - \mathbf{u}_h\|_{L^\infty}$	3.3e-2	1.8e-2	9.7e-3	3.2e-2	1.7e-2	9.0e-3
$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	8.6e-3	4.4e-3	2.2e-3	8.5e-3	4.3e-3	2.1e-3
$\|p - p_h\|_{L^2}$	2.9e-1	1.3e-1	6.4e-2	2.8e-1	1.3e-1	6.4e-2
	BDF2 with semi-Lagrangian (quadratic interpolation)[†]					
$\|\mathbf{u} - \mathbf{u}_h\|_{L^\infty}$	3.0e-4	8.3e-4	5.2e-4	2.5e-3	8.4e-4	1.0e-3
$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	5.6e-4	1.4e-4	3.8e-5	5.7e-4	1.4e-4	3.4e-5
$\|p - p_h\|_{L^2}$	1.4e-2	6.3e-3	5.0e-3	2.2e-2	5.4e-3	1.3e-3
	VK					
$\|\mathbf{u} - \mathbf{u}_h\|_{L^\infty}$	1.7e-3	4.2e-4	1.2e-4	1.6e-3	3.4e-4	7.5e-5
$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	4.0e-4	9.7e-5	2.4e-5	3.9e-4	9.4e-5	2.3e-5
$\|p - p_h\|_{L^2}$	8.0e-3	2.1e-3	5.6e-4	7.0e-3	1.6e-3	2.8e-4

† Here the time step was equal $\triangle t/2$

discretizations. From this experiment we conclude that the higher order approximation of the advection terms leads to more accurate solutions.Further we will see that for the benchmark problem with non-smooth solutions and adaptively refined grids TOU is still advantageous compared to SOU. Thus, the third order upwind scheme for advective terms is our preferred approach.

In the next two sets of experiments we demonstrate the role of the low-pass filter on the coarse-to-fine cells interface and the convergence of the method on a sequence of refined non-uniform grids. To compute the results in Tables 5 and 6, we refined the mesh inside the sphere of the radius 0.5 with the center in $(0, 0, 0)$. The sizes of the largest and the smallest cells are $h_{\max}$ and $h_{\min}$, respectively. Obviously, $h_{\max} = h_{\min}$ corresponds to the uniform grid. Table 5 shows the results of computations for two values of viscosity, $\nu = 0$ (the Euler limit) and $\nu = 1$ (diffusion dominated

Table 4: Errors for two FD discretizations of the advective operator on uniform meshes, $\nu = 10^{-2}$.

upwind order	SOU			TOU		
mesh size h	1/16	1/32	1/64	1/16	1/32	1/64
time step $\triangle t$	1/100	1/200	1/400	1/100	1/200	1/400
$\|\mathbf{u} - \mathbf{u}_h\|_{L^\infty}$	4.3e-3	8.4e-4	1.6e-4	1.6e-3	3.4e-4	7.5e-5
$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	8.3e-4	1.6e-4	3.2e-5	3.9e-4	9.4e-5	2.3e-5
$\|p - p_h\|_{L^2}$	1.5e-2	4.1e-3	1.1e-3	8.0e-3	2.1e-3	5.5e-4

Table 5: Dependence of errors on the number of tree levels for locally refined meshes. The effect of the filter.

viscosity	$\nu = 0$				$\nu = 1$			
mesh size $h_{\max}$	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/16
mesh size $h_{\min}$	1/16	1/32	1/64	1/128	1/16	1/32	1/64	1/128
	BDF2 with FD advective terms, no filter							
$\|\mathbf{u} - \mathbf{u}_h\|_{L^\infty}$	1.7e-3	1.6e-2	2.2e-2	2.9e-2	1.2e-3	1.3e-3	1.6e-3	1.8e-3
$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	4.0e-4	1.1e-3	1.4e-3	1.4e-3	3.1e-4	3.9e-4	5.3e-4	6.5e-4
$\|p - p_h\|_{L^2}$	8.0e-3	6.9e-3	6.2e-3	5.4e-3	4.0e-3	9.5e-3	1.4e-2	1.7e-2
	BDF2 with FD advective terms, with filter							
$\|\mathbf{u} - \mathbf{u}_h\|_{L^\infty}$	1.7e-3	2.6e-3	3.3e-3	3.2e-3	1.2e-3	1.4e-3	1.6e-3	1.8e-3
$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	4.0e-4	5.7e-4	7.8e-4	9.1e-4	3.1e-4	3.9e-4	5.3e-4	6.5e-4
$\|p - p_h\|_{L^2}$	8.0e-3	6.9e-3	5.8e-3	5.2e-3	4.0e-3	9.5e-3	1.4e-2	1.7e-2

Table 6: Convergence of the method on a sequence of refined non-uniform octree grids, $\nu = 0.01$; Results are shown for the BDF2 with FD advective terms, with filter.

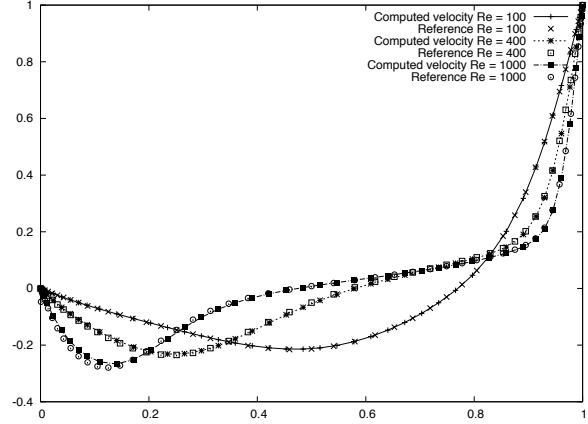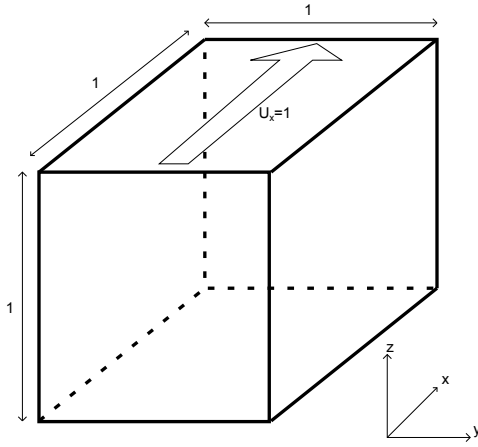| mesh size $h_{\max}$	1/8	1/16	1/32	1/64
mesh size $h_{\min}$	1/32	1/64	1/128	1/256
time step dt	1/50	1/100	1/200	1/400
$\|\mathbf{u} - \mathbf{u}_h\|_{L^\infty}$	1.1e-2	3.1e-3	7.9e-4	1.6e-4
$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$	3.6e-3	7.6e-4	1.8e-4	4.3e-5
$\|p - p_h\|_{L^\infty}$	6.3e-2	9.8e-3	2.5e-3	7.0e-4
$\|p - p_h\|_{L^2}$	1.5e-2	5.8e-3	1.7e-3	4.9e-4

15

Figure 6: Left: The 3D driven cavity problem setup; Right: The centerline $((0.5, 0.5, z), \ 0 \le z \le 1)$ u-velocities compared to reference data from [33]

case). Similar to what was observed for the discrete Helmholtz decomposition, for small values of the viscosity parameters the local mesh refinement leads to the growth of the error. Since the spurious modes are oscillatory and tailored to the coarse-to-fine grid interface, the L^∞-norm of the velocity error is more sensitive indicator of the instability than the L^2 norm. The error in pressure is not much influenced by the local refinement. For $\nu = 1$ the spurious oscillatory modes are damped by the dominating physical diffusion, so the error growth is very modest in this case. Otherwise, the course-to-fine grid interface filter provides the auxiliary damping of the modes and stabilizes the problem. In further experiments of this paper, we always use the filter with TOU discretization of advection terms (unless otherwise noted). Table 6 demonstrates the second order convergence for velocity and almost the second order convergence for pressure on the sequence of refined octree grids.

5.2. The 3D driven cavity

The next numerical example is the standard lid driven cavity benchmark problem. The problem setup is illustrated in Figure 6. One looks for the steady solution of the flow equations (1) in $\Omega = (0, 1)^3$, with $\mathbf{u} = (1, 0, 0)^T$ for $z = 1$ and no-slip/no-penetration conditions on other parts of the boundary. In spite of the simplest of geometrical settings, the cavity flows display many important fluid mechanical phenomena [34]. Note that due to the discontinues boundary conditions, the solution to the problem is singular in the neighborhood of upper edges.

We are interested in steady solutions for $Re = 100, 400, 1000$. The projection method (10)–(13) was used to integrate in time until the equilibrium steady state is recovered. For this benchmark problem, the coarsest mesh of $h_{max} = 1/32$ was used; further the grid was refined towards the boundary (five grid layers with $h = 1/64$ and two grid layers with $h_{min} = 1/256$). This resulted in the 983256 pressure degrees of freedom and 2936724 velocity degrees of freedom. Figure 6 (right) shows the computed centerline $((0.5, 0.5, z), \ 0 \le z \le 1)$ u-velocities of the steady state
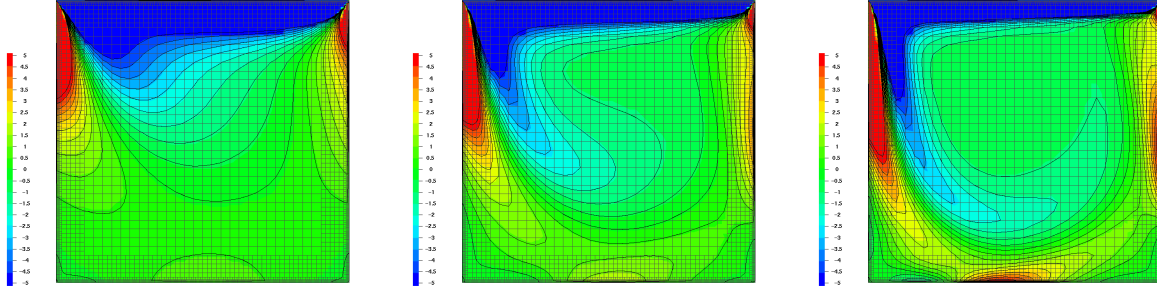
16

Figure 7: Spanwise vorticity for the midplane $y = 0.5$ and $Re = 100$ (left), $Re = 400$ (middle), $Re = 1000$ (right).

solution. They are in a very good agreement with the reference results of Wong and Baker [33], who used a solution-adapted tetrahedral mesh and a conforming finite element method to compute the solution in velocity-vorticity variables.

Figure 7 shows the contours of spanwise vorticity for the midplane $y = 0.5$. The plots of the midplane velocity fields are shown in Figure 8. Both vorticity contours and velocity fields agree well with those of [33, 35] and fairly well illustrate the recovered cavity flow dynamics. Besides the primary eddy (here we use the terminology of [34]), the method is able to capture upstream secondary eddy for $Re = 1000$ and $Re = 400$, bottom end-wall vortices for all values of Re and upper end-wall vortices for $Re = 1000$ and $Re = 400$; downstream swirls are visible for $Re = 1000$ and $Re = 400$. All this flow structures transit smoothly over coarse-to-fine meshes interfaces. The ability of the method to correctly predict secondary flow structures indicates the low numerical diffusion of the scheme.

5.3. Flow around cylinder

The final numerical example is the laminar 3D channel flow around a cylinder of square cross-section. The problem was defined within the DFG priority research program "Flow simulation on high performance computers" by Schäfer and Turek in [36] and further studied in, e.g., [37, 38].

The flow domain is shown in Figure 9. The no-slip and no-penetration boundary condition $\mathbf{u} = 0$ is prescribed on the channel walls and the cylinder surface. The parabolic velocity profile is set on the inflow boundary:

$$\mathbf{u} = (0, 0, 16\widetilde{U}xy(H - x)(H - y)/H^4)^T \quad \text{on } \Gamma_{\text{inflow}},$$

with $H = 0.41$ and a peak velocity $\widetilde{U}$. The Reynolds number, $Re = \nu^{-1}D\widetilde{U}$, is defined based on the cylinder width $D = 0.1$. The viscosity coefficient ν is set to 10^{-3}. In [36] three benchmark problems were suggested:

- Problem Q1: Steady flow with $Re = 20$ ($\widetilde{U} = 0.45$);

- Problem Q2: Unsteady periodic flow with $Re = 100$ ($\widetilde{U} = 2.25$);

- Problem Q3: Unsteady flow with varying Reynolds number for $\widetilde{U} = 2.25 \sin(\pi t/8)$.

17

Oxz   Oyz   Oxy
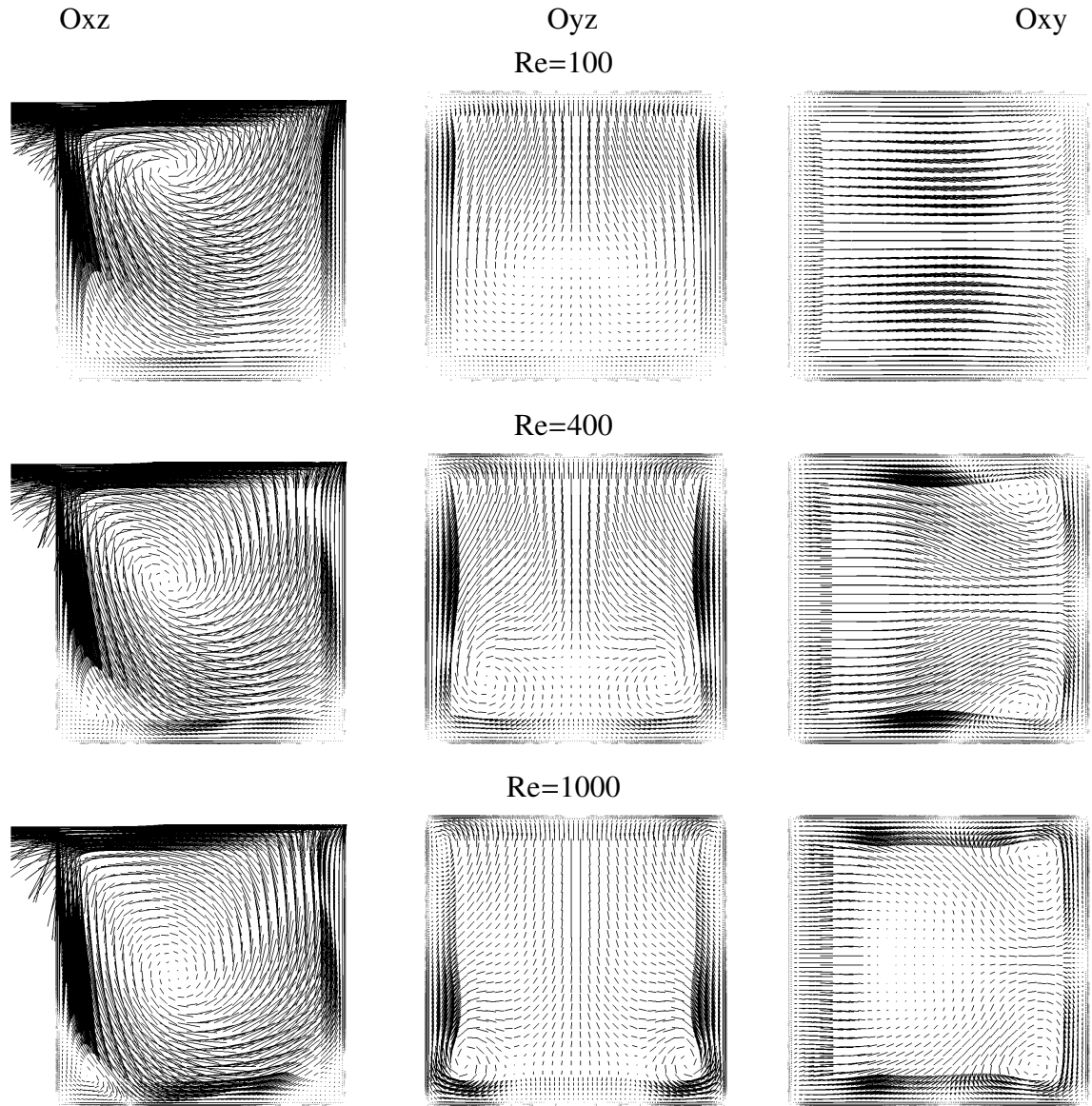
Re=100

Re=400

Re=1000

Figure 8: The 2D planar projections of steady state velocity fields at the midplanes for the 3D driven cavity problem with $Re = 100, 400, 1000$.
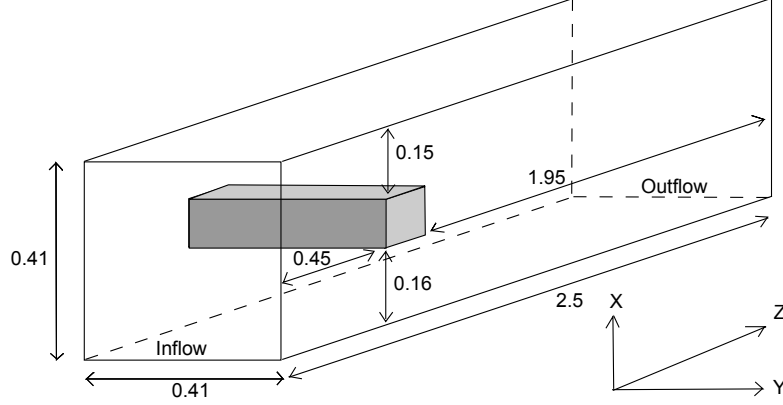
Figure 9: Computational domain for flow around a cylinder with square cross-section.

The initial condition is $\mathbf{u} = 0$ for $t = 0$.

To realize possible stability limitations of the proposed techniques, we also compute the flow around cylinder for $Re = 10^3$ and $Re = 10^4$.

The statistics of interest are the following:

- The difference $\Delta p = p(\mathbf{x}_2) - p(\mathbf{x}_1)$ between the pressure values in points $\mathbf{x}_1 = \{0.2, 0.205, 0.55\}$ and $\mathbf{x}_2 = \{0.2, 0.205, 0.45\}$.

- The drag coefficient given by an integral over the surface of the cylinder S:

$$C_{\text{drag}} = \frac{2}{DH\widetilde{U}^2} \int_S \left(\nu \frac{\partial(\mathbf{u} \cdot \mathbf{t})}{\partial \mathbf{n}} n_x - p n_z \right) ds \tag{19}$$

Here $\mathbf{n} = (n_x, n_y, n_z)^T$ is the normal vector to the cylinder surface pointing to Ω and $\mathbf{t} = (-n_z, 0, n_x)^T$ is a tangent vector.

- The lift coefficient given by an integral over the surface of the cylinder:

$$C_{\text{lift}} = -\frac{2}{DH\widetilde{U}^2} \int_S \left(\nu \frac{\partial(\mathbf{u} \cdot \mathbf{t})}{\partial \mathbf{n}} n_z + p n_x \right) ds \tag{20}$$

- If a periodic regime is attained by the solution to problem Q2, then one is interested in the Strouhal number $Df\widetilde{U}^{-1}$, where f is the frequency of vortices separation.

For problem Q3, the reference velocity in C_{drag} and C_{lift} is taken for $t = 4$.

The feature of the problem is the singularity of geometry: the corners on a square cylinder are likely to destroy the regularity of the solution to (1). This makes the accurate numerical prediction of the lift and drag coefficients difficult and a local grid refinement in the neighborhood of the cylinder is necessary.
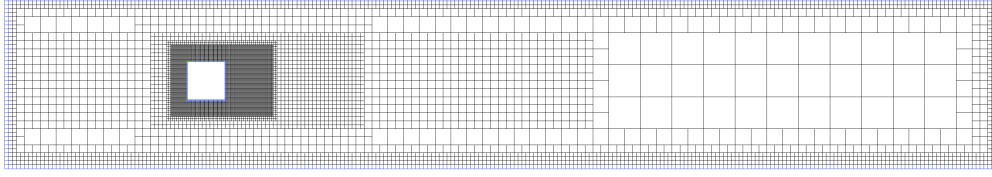
19

Figure 10: The cutaway of the grid at $y = 0.205$ for $h_{max} = 1/32$ and $h_{min} = 1/1024$.

Table 7: Number of velocity and pressure d.o.f. for different meshes. N_{CD} and N_{PP} are the average numbers of iterations in convection-diffusion and pressure solvers for problem Q1, respectively.

h_{min}	h_{max}	$\mathbf{u}$ d.o.f.	p d.o.f.	N_{CD}	N_{PP}
1/256	1/256	1246359	416150	23	30
1/512	1/256	1402593	467110	27	37
1/1024	1/256	2707497	897330	47	41
1/2048	1/256	12828221	4245010	112	66
1/1024	1/32	1969827	645393	45	44

To compute the drag and lift coefficient, one may replace the surface integrals in (19) and (20) by integration over the whole domain. This evaluation technique is known in the finite element community and has been used in [39, 37].

Assume $\mathbf{u} = (u, v, w)^T$ and p solve (1), then applying the integration by part one checks (cf. [37]) the following identities:

$$
\begin{aligned}
C_{\text{drag}} &= \widetilde{C} \int_{\Omega} \left[\left(\frac{\partial w}{\partial t} + (\mathbf{u} \cdot \nabla) w \right) \varphi + \nu \nabla w \cdot \nabla \varphi - p \partial_z \varphi \right] d\mathbf{x} \\
C_{\text{lift}} &= \widetilde{C} \int_{\Omega} \left[\left(\frac{\partial u}{\partial t} + (\mathbf{u} \cdot \nabla) u \right) \varphi + \nu \nabla u \cdot \nabla \varphi - p \partial_x \varphi \right] d\mathbf{x},
\end{aligned}
\tag{21}
$$

$\widetilde{C} = \frac{2}{DH\widetilde{U}^2}$, for any $\varphi \in H^1(\Omega)$ such that $\varphi|_S = 1$ and $\varphi|_{\partial\Omega/S} = 0$.

If the Navier-Stokes solution is sufficiently smooth and one computes drag and lift coefficients of a finite element numerical solution, then it is proved in [37] that using the volume based formulas (21) gives more accurate values of drag and lift coefficients compared to (19) and (20). Although for this test problem the solution is not regular, it turned out that using (21) still leads to more accurate results.

Now we discuss few technical details of evaluating (21). The identities (21) hold for any φ satisfying $\varphi|_S = 1$ and $\varphi|_{\partial\Omega/S} = 0$. However, if integrals in (21) are evaluated for a numerical solution, then the analysis of [37] suggests that accuracy may depend on the regularity of φ. For the finite difference method we define φ in pressure nodes and consider the discretely harmonic function ($\text{div}_h \nabla_h \varphi = 0$). All derivatives in (21) were approximated with the second order of accuracy.

The numerical solutions to problems Q1–Q3 were computed on a sequence of locally refined meshes, see Table 7 for the account of corresponding discrete space dimensions. The cutaway of

20

Table 8: Problem Q1: Convergence of drag, lift, and pressure drop to reference values for locally refined grid.

h_{min}	h_{max}	C_{drag}	C_{lift}	Δp	C_{drag}	C_{lift}	Δp
		SOU			TOU		
1/256	1/256	7.766	0.05951	0.1720	7.726	0.07122	0.1717
1/512	1/256	7.589	0.06671	0.1727	7.683	0.06814	0.1724
1/1024	1/256	7.609	0.06796	0.1732	7.706	0.06829	0.1745
1/2048	1/256	7.631	0.06868	0.1737	7.727	0.06864	0.1750
Braack & Richter		7.767	0.06893	0.1757	7.767	0.06893	0.1757
Schäfer & Turek		7.5-7.7	0.06-0.08	0.172-0.18	7.5-7.7	0.06-0.08	0.172-0.18
1/1024	1/32	7.631	0.06821	0.1727	7.716	0.0678	0.1749

a grid with $h_{min} = 1/32$ and $h_{max} = 1/1024$ is shown in Figure 10. The linear algebraic systems were solved iteratively. Thus, the discrete convection-diffusion-reaction problem arising on the predictor step (10) of the method was solved by the BiCGstab method with ILU(0) preconditioner and the pressure Poisson problem was solved with the GMRES method preconditioned by one V-cycle of the algebraic multigrid method from [40]. Average numbers of iterations required to ensure the Euclidian norm of residual is less than 1e-13 are shown in Table 7. These values correspond to the experiments with the problem Q1 and time step $\Delta t = 0.1$. For problems Q2 and Q3 we set $\Delta t^n = \max\{0.1, 10h_{min}(\max |\mathbf{u}^n|)^{-1}\}$. While iteration numbers in pressure solve were nearly the same for problems Q2 and Q3, the iteration numbers in advection-diffusion solve were smaller and almost independent of the refinement level due to the dominant zero order term $\alpha \frac{\widetilde{\mathbf{u}^{n+1}}}{\Delta t^n}$.

For all three problems, the reference [36] collects several DNS results based on various finite element, finite volume discretizations of the Navier-Stokes equations and the Lattice Boltzmann method. In [36], the authors provide reference intervals where the statistics should converge. Using a higher order finite element method and locally refined adaptive meshes, more accurate reference values of C_{drag}, C_{lift} and Δp were found in [37] for problem Q1. Thus, we first present in Table 8 the results for problem Q1 computed with second and third order approximations of advective terms. We note that the differences in CPU times for both cases were negligibly small. Similar to the case of uniform grids and analytical solution, the TOU discretization shows somewhat more accurate results. For a sequence of locally refined octree meshes, Table 8 demonstrates the convergence of computed drag, lift, and pressure drop to reference values. Here and in Table 9 we additionally include the results for the 5-refinement levels mesh ($h_{max} = 1/32$, $h_{min} = 1/1024$), which appear to be very close to those for 2-refinement levels ($h_{max} = 1/256$, $h_{min} = 1/1024$), indicating that the refinement around cylinder rather than in the bulk domain is crucial for accurate computation of the statistics.

Less accurate reference data is available for problems Q2 and Q3. Table 9 summarizes the results computed by the present method and those available in the literature. For problem Q2, C_{drag} and C_{lift} are the maximum lift and drag coefficients after the flow attains a periodic regime. For problem Q3, C_{drag} and C_{lift} are the maximum lift and drag coefficients over the whole time interval $t \in [0, 8]$, the pressure drop is computed at $t = 8$. Note that [36] does not give reference intervals for problem Q2, and we simply show the maximum and minimum values of lift, drag,

Table 9: Lift, drag, and the Strouhal number for problem Q2; Lift, drag, and pressure drop for problem Q3

		Problem Q2			Problem Q3		
h_{min}	h_{max}	C_{drag}	C_{lift}	St	C_{drag}	C_{lift}	Δp
1/256	1/256	6.204	0.07631	*	6.038	0.3497	-0.1461
1/512	1/256	5.222	0.04407	0.326	5.178	0.0381	-0.1284
1/1024	1/256	4.679	0.02697	0.297	4.655	0.0168	-0.1367
1/2048	1/256	4.484	0.03166	0.307	4.475	0.0300	-0.1407
Schafer & Turek		4.32–4.67[†]	0.015–0.05[†]	0.27–0.35[†]	4.3–4.5	0.01–0.05	-0.14 – -0.12
1/1024	1/32	4.671	0.02666	0.306	4.658	0.0172	-0.1374

∗ Solution has not attained a periodic regime for $t \in [0, 16]$.

† Reference intervals for problem Q2 may be not very accurate.

and the Strouhal numbers for the DNS results included in [36]. However, these intervals can be not very accurate. It is expected that $Re = 100$ is close to the critical Reynolds number (when transition from the steady state to unsteady periodic flow occurs). Therefore, the simplest reasonable criteria of the success of a numerical method for this problem: Is a stable periodic flow (including vortex separation and von Karman vortex street) captured by a method for $Re = 100$? If no, then the method is likely to be excessively diffusive or unstable.
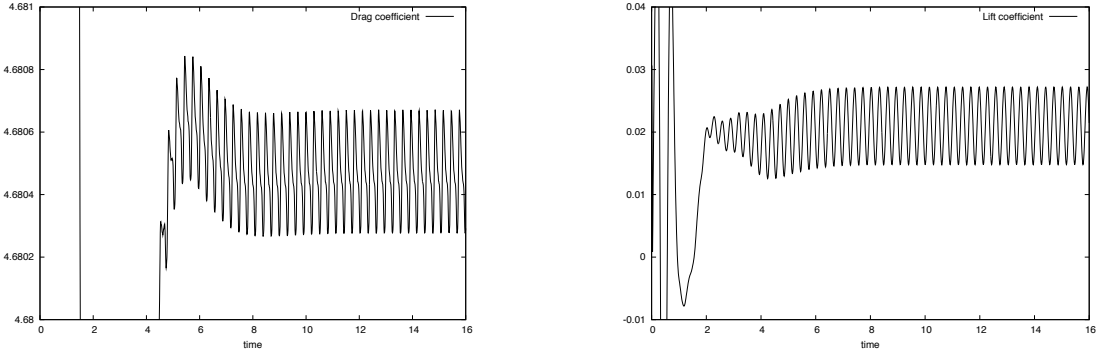

Figure 11: The evolution of drag (left) and lift (right) coefficients for the flow around a square cylinder with $Re = 100$ (problem Q2) computed with $h_{max} = 1/32$, $h_{min} = 1/1024$.

For the present methods a stable periodic flow is recovered starting with $h_{min} = 1/512$. Figure 11 plots the evolution of the drag and lift coefficients for time interval $t \in [0, 16]$. It is clear that the periodic regime is attained. In Figure 12 we show the snapshot of the spanwise vorticity contours at time t=16 for the midplane $y = 0.205$. The figure illustrates the developed von Karman vortex street behind the cylinder. We note that the periodic unsteady solution is recovered with BDF2 with FD advective terms scheme, while the semi-Lagrangian method with linear interpolation produces steady solutions. The semi-Lagrangian method with quadratic interpolation was found unstable for this problem.
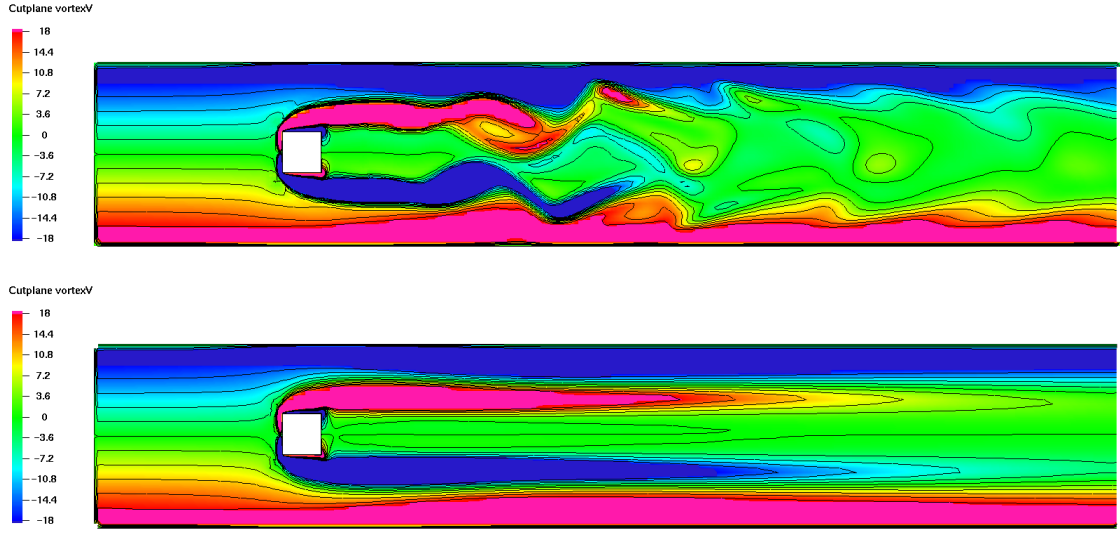
Figure 12: Problem Q2 (Re=100): Spanwise vorticity at time t=16 for the midplane $y = 0.205$. The top plot shows the development of vortex street for solution by BDF2 with FD advective terms; the bottom plot shows an over-diffusive solution computed by BDF2 with semi-Lagrangian method (linear interpolation). Both solutions were computed with $h_{\max} = 1/256$ and $h_{\min} = 1/1024$ at $t = 16$.

From the regularity theory of the linearized Navier-Stokes problem [41], we may expect that pressure and velocity become less regular in the neighborhoods of cylinder edges: The theory predicts $p \notin H^1(\Omega)$ and $\mathbf{u} \notin H^2(\Omega)^3$. This, in particular, implies that the pressure gradient and the second velocity derivatives are unbounded in the vicinity of the edges. Indeed, Figure 12 shows sharp internal layers in vorticity originating from upstream edges of the cylinder and Figure 13 presents the midplane pressure contours around the cylinder. The pressure has large gradients near the upstream edges. This lack of solution smoothness explains why local grid refinement is necessary and why accurate evaluation of drag and lift coefficients for the flow around a square cylinder is hard.

The 3D structure of the flow with $Re = 100$ is seen from Figures 14 and 15, where we show the streamlines of the developed flow, pressure isosurfaces and the isosurfaces $|\mathbf{w}| = 20$ of vorticity colored by the absolute values of velocity, $|\mathbf{u}|$.

Finally, we run the same test with higher Reynolds numbers, $Re = 10^3$ and $Re = 10^4$. We have not found other data in the literature for this problem with higher Re numbers to make a comparison. The goal of performing the tests is to verify if the method remains stable when Re is increasing and to quantify stability limitations (if any). Note that for smooth solutions we got stable results for arbitrary small values of the viscosity coefficient, see Table 5. For the flow around a square cylinder problem, the situation is more complicated: A solution has sharp boundary and internal layers and the flow becomes turbulent for sufficiently large Reynolds number. This was eventually the case for $Re = 10^4$.

For flow around a square cylinder at $Re = 10^3$, we observe a stable numerical solution, which demonstrates a quasi-periodic behaviour. Now large vortices periodically form in the internal
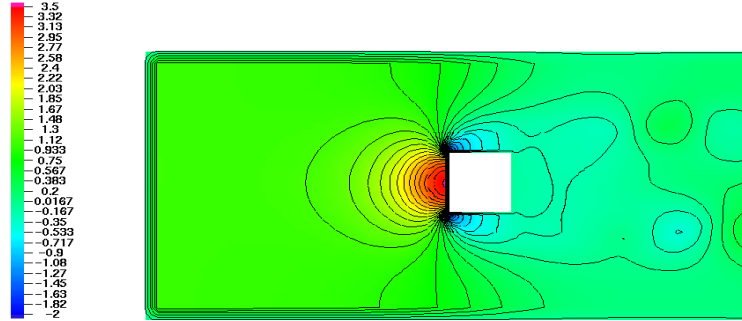
23

Figure 13: Problem Q2 (Re=100): The midplane pressure contours around a square cylinder. The solution shown was computed with $h_{\max} = 1/256$ and $h_{\min} = 1/1024$ at $t = 16$.
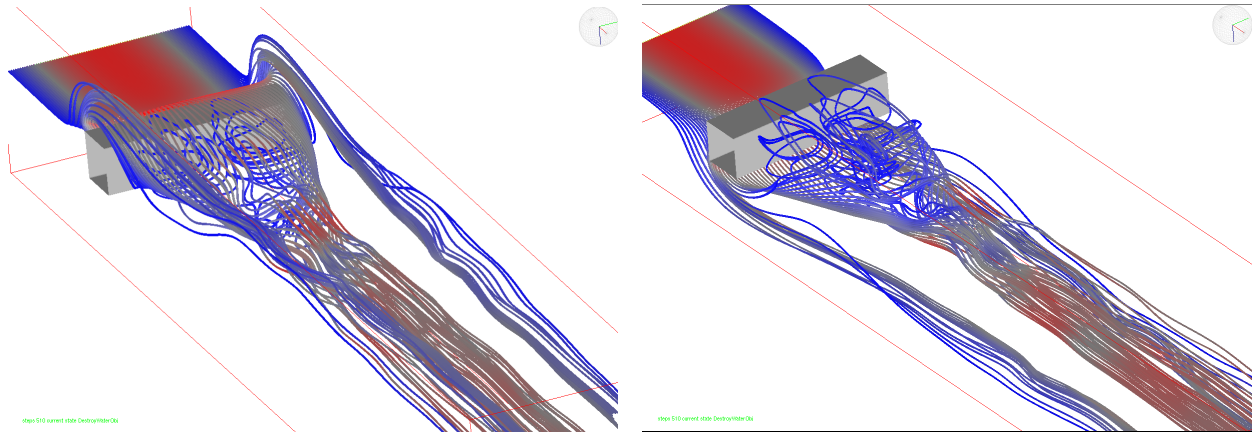

Figure 14: Problem Q2: Streamlines of the developed flow around a square cylinder for Re=100. The streamlines are selected for two fluid layers entering the domain slightly above and below $x = 0.21$.

layers originating from two upstream edges of the cylinder rather than shed behind the cylinder. These vortices are convected downstream and interact in a complicated way with each other and smaller eddies created near upper and bottom walls. A flow in the recirculation region behind the cylinder is close to chaotic. All these make an intricate picture of the (still laminar) flow over a square cylinder in a channel at $Re = 10^3$. The computed solution is illustrated in Figure 16, where we show spanwise vorticity isolines and isosurfaces for pressure and vorticity. For the vorticity, we choose to show the isosurface $|\mathbf{w}| = 100$ as a good illustration of vorticity generation around the cylinder.

For $Re = 10^4$ we observe no regular flow pattern: An unstable boundary layer develops near channel walls close to the inlet (approximately at $z = 0.2$), traveling vortices appear on many scales and interact with each other in a stochastic way. We may conclude that the flow is turbulent. Nevertheless, for this type of flow the method produces a numerically stable solution up to the time $t = 8$, which was the final time of computations. A multiscale structure of the flow is illustrated in Figure 17.
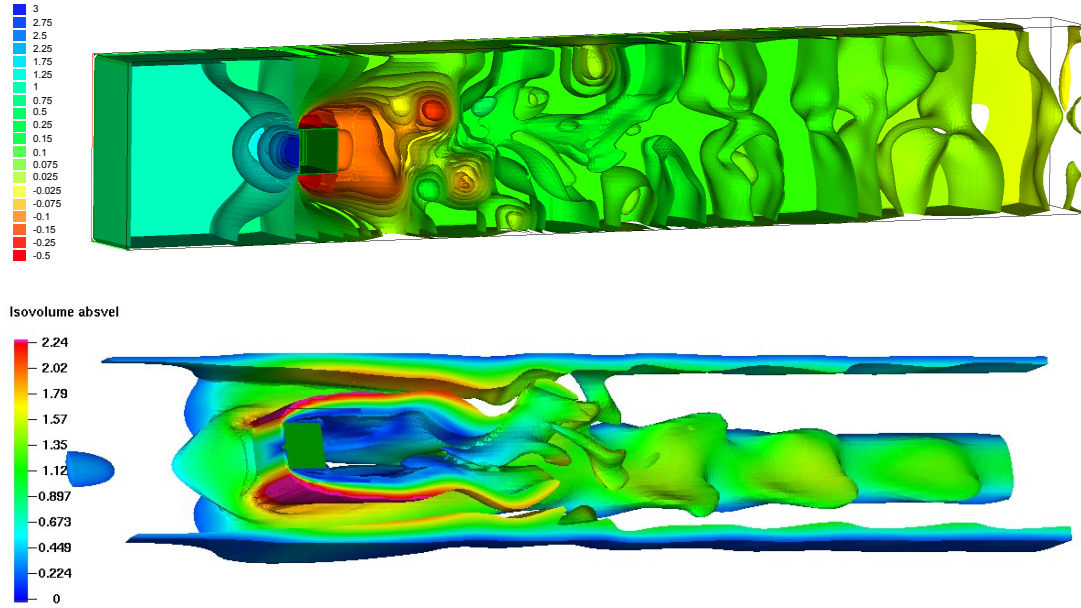
24

Figure 15: Problem Q2 (Re=100): Pressure isosurfaces and vorticity isosurface $|\mathbf{w}| = 20$ colored by the absolute velocity. The plots illustrate solution computed with $h_{max} = 1/256$ and $h_{min} = 1/1024$ at $t = 16$.

We conclude that for turbulent flows the numerical dissipation produced by the present method can be sufficient to diffuse the energy of resolved structures, although an additional modeling is likely required to simulate the effect of unresolved scales in a proper way and recover meaningful averaged statistics.

6. Conclusions

Octree cartesian grids are super-convenient for fast mesh adaptation, reconstruction and data access. Finite difference and finite volume methods on octree grids provide a cost effective alternative to discontinuous Galerkin methods. This efficiency, however, comes at a price: local refinement does not lead automatically to better accuracy and higher order discretizations require large nodal stencils and higher order interpolation. In this paper, we introduced an extension of staggered grid MAC scheme such that specific instabilities stemming from local grid refinement are suppressed. The discretization is second order accurate and stable. It involves the construction of only planar second order polynomials and linear interpolation. The performance of the scheme was studied for a set of smooth and non-smooth benchmark solutions. The method produces stable low dissipative second order accurate solutions and compares favorably to a scheme built on semi-Lagrangian treatment of advection. The scheme was found to be numerically stable also for certain high Reynolds number flows.

References

[1] J. E. Flaherty, R. M. Loy, M. S. Shephard, B. K. Szymanski, J. D. Teresco, L. H. Ziantz, Adaptive local refinement with octree load balancing for the parallel solution of three-dimensional conservation laws, Journal of
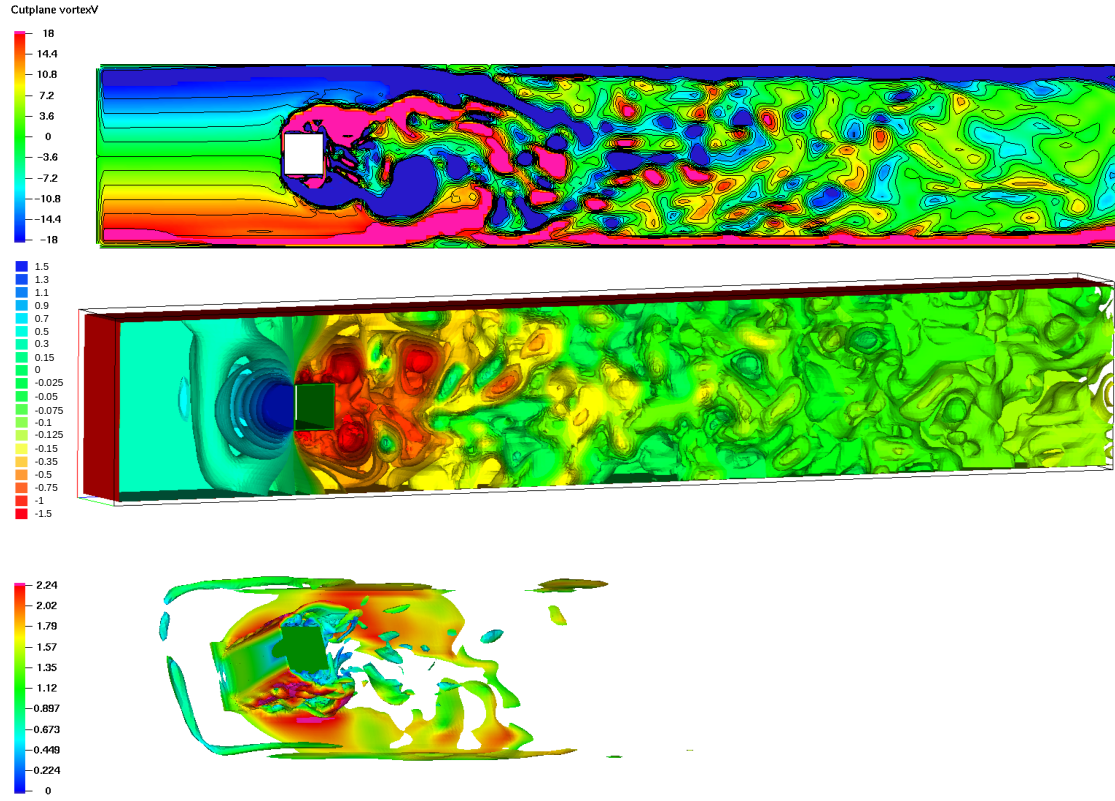
Figure 16: Channel flow around a square cylinder at Re=1000: Spanwise vorticity for the midplane $y = 0.205$. Pressure isosurfaces and vorticity isosurface $|\mathbf{w}| = 100$ colored by the absolute velocity. The plots illustrate solution computed with $h_{\max} = 1/256$ and $h_{\min} = 1/1024$ at $t = 8$.

Parallel and Distributed Computing 47 (1997) 139–152.

[2] J.-F. Remacle, J. E. Flaherty, M. S. Shephard, An adaptive discontinuous galerkin technique with an orthogonal basis applied to compressible flow problems, SIAM Review 45 (2003) 53–72.

[3] S. M. Murman, Compact upwind schemes on adaptive octrees, J. Comput. Phys. 229 (2010) 167–1180.

[4] J. Strain, Tree methods for moving interfaces, J. Comput. Phys. 151 (1999) 616–648.

[5] V. Sochnikov, S. Efrima, Level set calculations of the evolution of boundaries on a dynamically adaptive grid, Int. J. Numer. Meth. Engng. 56 (2003) 1913–1929.

[6] F. Losasso, F. Gibou, R. Fedkiw, Simulating water and smoke with an octree data structure, ACM Transactions on Graphics (TOG) 23.

[7] F. Losasso, R. Fedkiw, S. Osher, Spatially adaptive techniques for level set methods and incompressible flow, Computers & Fluids 35 (2006) 995–1010.

[8] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, J. Comput. Phys. 228 (2009) 5838–5866.

[9] D. Fuster, G. Agbaglah, C. Josserand, S. Popinet, S. Zaleski, Numerical simulation of droplets, bubbles and waves: state of the art, Fluid Dyn. Res. 41 (2006) 065001.

[10] K. D. Nikitin, M. A. Olshanskii, K. M. Terekhov, Y. V. Vassilevski, A numerical method for the simulation of free surface flows of viscoplastic fluid in 3D, J. Comput. Math. 29 (2011) 605–622.

[11] K. Nikitin, Y. V. Vassilevski, Free surface flow modelling on dynamically refined hexahedral meshes, Rus. J. Numer. Anal. Math. Model. 23 (2008) 469–485.

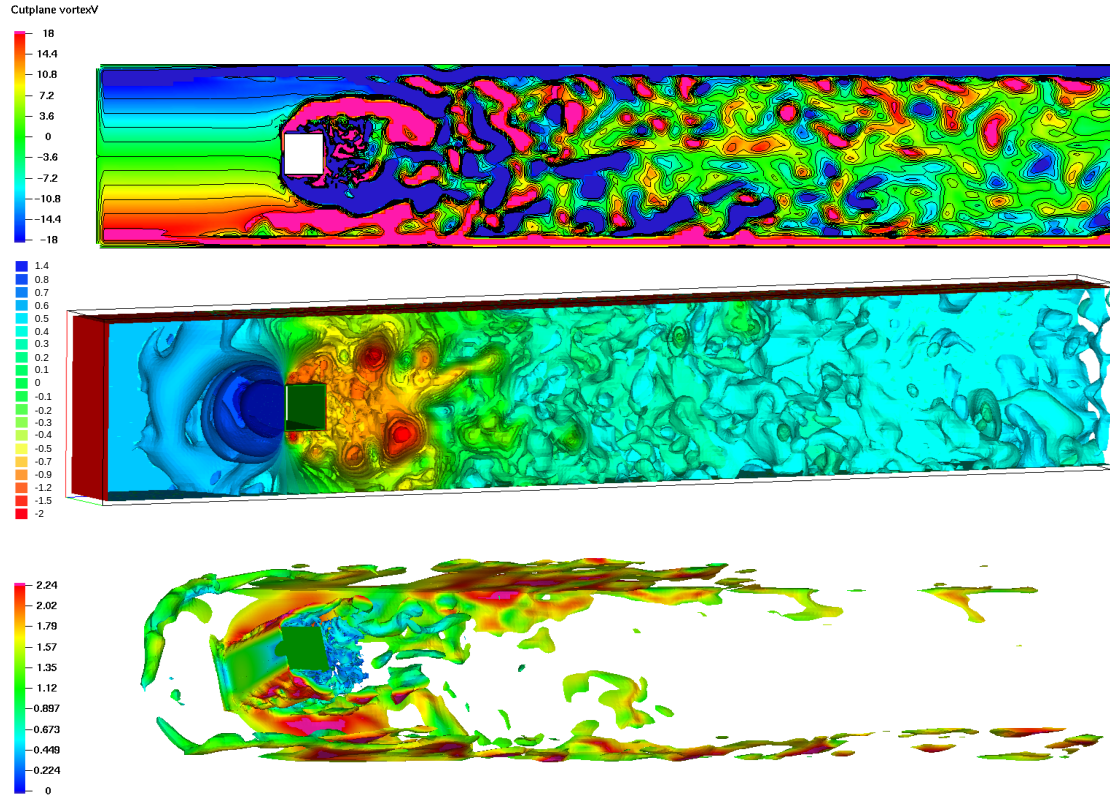[12] R. Szeliski, Rapid octree construction from image sequences, CVGIP: Image Understanding 58 (1993) 23–32.

Figure 17: Channel flow around a square cylinder at Re=10000: Spanwise vorticity for the midplane $y = 0.205$. Pressure isosurfaces and vorticity isosurface $|\mathbf{w}| = 100$ colored by the absolute velocity. The plots illustrate solution computed with $h_{\max} = 1/256$ and $h_{\min} = 1/1024$ at $t = 8$.

[13] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries, J. Comput. Phys. 190 (2003) 572–600.

[14] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive cartesian grids, J. Comput. Phys. 225 (2007) 300–321.

[15] F. Gibou, C. Min, H. Ceniceros, Finite difference schemes for incompressible flows on fully adaptive grids, International Series of Numerical Mathematics 154 (2006) 199–208.

[16] V. Lebedev, Difference analogues of orthogonal decompositions, basic differential operators and some boundary problems of mathematical physics. I, U.S.S.R. Comput. Math. Math. Phys. 4 (3) (1964) 69–92.

[17] V. Lebedev, Difference analogues of orthogonal decompositions, basic differential operators and some boundary problems of mathematical physics. II, U.S.S.R. Comput. Math. Math. Phys. 4 (4) (1964) 36–50.

[18] F. Harlow, J. Welch., Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.

[19] R. A. Nicolaides, Analysis and convergence of the mac scheme i. the linear problem, SIAM J. Numer. Anal. 29 (1992) 1579–1591.

[20] J. Mullen, P. Fischer, Filtering techniques for complex geometry fluid flows, Commun. Numer. Meth. Engng. 15 (1999) 9–18.

[21] A. Chorin, Numerical solution of the Navier-Stokes equations, Math. Comp. 22 (1968) 745–762.

[22] J. L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Comput. Meth. Appl. Mech. Engrn. 195 (2006) 6011–6045.

[23] A. Prohl, Projection and quasi-compressibility methods for solving the incompressible Navier-Stokes equations,

B.G. Teubner (Stuttgart), 1997.

[24] J. L. Guermond, P. Minev, J. Shen, Error analysis of pressure-correction schemes for the time-dependent stokes equations with open boundary conditions, SIAM J. Numer. Anal. 42 (2005) 239258.

[25] P. Angot, R. Cheaytou, Vector penalty-projection method for incompressible fluid flows with open boundary conditions, in: Proceedings of 19th Conference on Scientific Computing, Algoritmy 2012, 2012, pp. 219–229.

[26] D. Xiu, G. E. Karniadakis, A semi-lagrangian high-order method for Navier–Stokes equations, J. Comput. Phys. 172 (2001) 658–684.

[27] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids, J. Comput. Phys. 219 (2006) 912–929.

[28] J. Fürst, A weighted least square scheme for compressible flows, Flow, turbulence and combustion 76 (2006) 331–342.

[29] S. Turek, Efficient solvers for incompressible flow problems: An algorithmic approach in view of computational aspects, Lect. Notes in Computational Science and Engineering 6, Springer, 1999.

[30] J. V. Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, SIAM J. Sci. Statist. Comput. 7 (1986) 870–891.

[31] C. Ethier, D. Steinman, Exact fully 3d Navier-Stokes solutions for benchmarking, Int. J. Numer. Meth. Fluids 19 (1994) 369–375.

[32] D. A. Kay, P. M. Gresho, D. F. Griffiths, D. J. Silvester, Adaptive time-stepping for incompressible flow part ii: Navier-stokes equations, SIAM J. Sci. Comp. 32 (2010) 111–128.

[33] K.L.Wong, A.J.Baker, A 3d incompressible Navier-Stokes velocity-vorticity weak form finite element algorithm, Int. J. Numer. Meth. Fluids 38 (2002) 99–123.

[34] P. N. Shankar, M. D. Deshpande, Fluid mechanics in the driven cavity, Annu. Rev. Fluid Mech 32 (2000) 93–136.

[35] Z. Zunic, M. Hribersek, L. Skerget, J. Ravnik, 3d driven cavity flow by mixed boundary and finite element method, in: E. O. P. Wesseling, J. Periaux (Eds.), European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2006, 2006.

[36] M. Schäfer, S. Turek, Benchmark computations of laminar flow around a cylinder 52 (1996) 547–566.

[37] M. Braack, T. Richter, Solutions of 3d NavierStokes benchmark problems with adaptive finite elements, Computers & Fluids 35 (2006) 372–392.

[38] A. L. Bowers, L. G. Rebholz, A. Takhirov, C. Trenchea, Improved accuracy in regularization models of incompressible flow via adaptive nonlinear filtering, Int. J. Numer. Meth. Fluids 70 (2012) 805–828.

[39] V. John, Higher order finite element methods and multigrid solvers in a benchmark problem for 3d NavierStokes equations, Int. J. Numer. Meth. Fluids 40 (2002) 775–98.

[40] V. E. Henson, U. M. Yang, Boomeramg: A parallel algebraic multigrid solver and preconditioner, Applied Numerical Mathematics 41 (2002) 155–177.

[41] M. Dauge, Stationary Stokes and Navier-Stokes systems on two- or three-dimensional domains with corners. Part I: linearized equations, SIAM J. Math. Anal. 20 (1989) 74–97.