INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **50**:437–468
Published online 1 September 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1067

Flow simulation on moving boundary-fitted grids and application to fluid–structure interaction problems

Martin Engel[*,†] and Michael Griebel[‡]

Institute of Numerical Simulation, University of Bonn, Wegelerstr. 6, 53115 Bonn, Germany

SUMMARY

We present a method for the parallel numerical simulation of transient three-dimensional fluid–structure interaction problems. Here, we consider the interaction of incompressible flow in the fluid domain and linear elastic deformation in the solid domain. The coupled problem is tackled by an approach based on the classical alternating Schwarz method with non-overlapping subdomains, the subproblems are solved alternatingly and the coupling conditions are realized via the exchange of boundary conditions. The elasticity problem is solved by a standard linear finite element method. A main issue is that the flow solver has to be able to handle time-dependent domains. To this end, we present a technique to solve the incompressible Navier–Stokes equation in three-dimensional domains with moving boundaries. This numerical method is a generalization of a finite volume discretization using curvilinear coordinates to time-dependent coordinate transformations. It corresponds to a discretization of the arbitrary Lagrangian–Eulerian formulation of the Navier–Stokes equations. Here the grid velocity is treated in such a way that the so-called Geometric Conservation Law is implicitly satisfied. Altogether, our approach results in a scheme which is an extension of the well-known MAC-method to a staggered mesh in moving boundary-fitted coordinates which uses grid-dependent velocity components as the primary variables.

To validate our method, we present some numerical results which show that second-order convergence in space is obtained on moving grids. Finally, we give the results of a fully coupled fluid–structure interaction problem. It turns out that already a simple explicit coupling with one iteration of the Schwarz method, i.e. one solution of the fluid problem and one solution of the elasticity problem per time step, yields a convergent, simple, yet efficient overall method for fluid–structure interaction problems. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: Navier–Stokes equations; finite volume discretization; fluid–structure interaction; moving boundary fitted grids; arbitrary Lagrangian–Eulerian method; geometric conservation law

[*] Correspondence to: Martin Engel, Institute of Numerical Simulation, University of Bonn, Wegelerstr. 6, 53115 Bonn, Germany.

[†] E-mail: engel@ins.uni-bonn.de

[‡] E-mail: griebel@ins.uni-bonn.de

Contract/grant sponsor: Deutsche Forschungsgemeinschaft

Received 22 November 2004
Revised 13 July 2005
Accepted 17 July 2005

1. INTRODUCTION

In many applications of computational fluid dynamics the problem domain is not constant but changes in time. Also its boundaries are moving. In particular, this is the case when interaction effects between the flow of a fluid and deformable structures are analysed. Problems of this type include flow through elastic pipes, for example in arteries or other blood vessels, moving pistons or the sloshing of fluids in elastic containers.

Small deformations of linear-elastic type pose no problems to most finite-element solvers. However this is not the case for the solution of flow problems. The reason is that structural mechanics problems are mostly formulated in a Lagrangian setting for which movement of the mesh together with the body under consideration is automatically obtained. A Lagrangian formulation of the Navier–Stokes equations is however not suitable for mesh-based discretization methods because of the complicated structure the 'deformations' of a fluid can exhibit. Here rotations or shearing of the fluid can lead to immediate entangling of grid cells. Therefore, special techniques for the treatment of flow problems with moving boundaries have been developed. Most existing methods are based on the Eulerian formulation of the Navier–Stokes equations and compute the solution on fixed grids. Additional 'markers' or property functions are used to describe the fluid domain. Examples are particle methods [1], the volume-of-fluid method [2, 3] and level-set-techniques [4, 5]. However with regard to fluid–structure interaction problems, these methods have some drawbacks. First, special techniques have to be used to prevent numerical diffusion and thus the smearing out of the interface where the interaction effects are localized. Second, the accurate prescription of boundary conditions is rather difficult and depends on a precise reconstruction of the interface which is only implicitly given in the above-mentioned Eulerian methods. But especially for the solution of fluid–structure interaction problems, a sharp representation of the interface is necessary. Arbitrary Lagrangian–Eulerian (ALE) methods [6, 7] provide the means to achieve this and at the same time avoid the disadvantages of a pure Lagrangian method. They introduce a frame of reference which is independent of both the fixed mesh of an Eulerian method and the particle-based reference frame of a Lagrangian method. Thus, the inner nodes of the mesh do not have to move with the local flow speed. There is a certain freedom for the mesh movement, which can be used to avoid large distortions or an entanglement of grid cells. However, in order to maintain a conservative numerical scheme, the mesh movement has to satisfy an additional condition known as the geometric conservation law [8, 9]. Altogether, ALE methods are well suited for the solution of the fluid subproblem in a coupled fluid–structure interaction analysis.

The fully coupled fluid–structure interaction problem is modelled by first decomposing it into the two subproblems of incompressible fluid flow and linear elasticity. Both subproblems are then solved separately in their respective domains. The interaction effects are modelled by coupling conditions which for each subproblem are incorporated in the solution process as boundary conditions on the moving interface. In particular, the fluid exerts a force on the elastic solid which is prescribed as a stress boundary condition for the elasticity problem. In turn, the influence of the elastic deformations of the solid on the fluid is given by the position and the velocity of the moving boundary. Thus, in each time step the fully coupled interaction problem is solved by an iterative method which corresponds to a block-Gauss–Seidel method applied to the system of the coupled equations [10]. This approach belongs to the family of the so-called partitioned or staggered algorithms, see References [11–13]. We will see that, for the test cases presented in this article, already one iteration of the coupled block-system per time step is sufficient to obtain a convergent, yet efficient method for the global coupled system.

The remainder of this paper is organized as follows. In the next section we formulate the coupled fluid–structure interaction problem composed of the subproblems for incompressible flow and linear elasticity. Then we discuss the solution process using the block-iterative approach. In Section 3, our method for the solution of flow problems in moving domains is presented. We briefly recall the most important properties of the basic finite-volume discretization as given in Reference [14]. Then we present a generalization to time-dependent grids. In Section 4 we give numerical results. First we discuss the convergence behaviour of the flow solver. Then we present simulations of a fluid–structure interaction problem, where we couple our flow solver with the elasticity problem. The discussion is completed by concluding remarks and an outlook to further developments.

2. THE FLUID–STRUCTURE INTERACTION PROBLEM

To set up a coupled fluid–structure interaction problem in a domain $\Omega \subset \mathbb{R}^3$, we first decompose Ω into the part of the domain which is filled by the fluid and the part which is occupied by the solid. To this end, we denote the fluid subdomain by Ω_{F} and the solid subdomain by Ω_{S} where $\Omega_{\mathrm{F}} \cap \Omega_{\mathrm{S}} = \emptyset$. The interface which separates the subdomains is given by $\Gamma = \bar{\Omega}_{\mathrm{F}} \cap \bar{\Omega}_{\mathrm{S}}$. A schematic view of the problem under consideration is depicted in Figure 1. Both subdomains and the interface are time-dependent, because the interface changes its position over time due to the interaction effects. The flow field in the domain $\Omega_{\mathrm{F}}(t)$ exerts a force onto the interface $\Gamma(t)$ which causes a deformation of the elastic structure $\Omega_{\mathrm{S}}(t)$. This deformation changes the shape of the fluid domain which in turn influences the flow itself.

In the fluid subdomain $\Omega_{\mathrm{F}}(t)$ we employ the Navier–Stokes equations in the ALE formulation given by

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_{\mathrm{F}}(t)} \rho_{\mathrm{F}} \, \mathrm{d}\Omega + \int_{\partial\Omega_{\mathrm{F}}(t)} \rho_{\mathrm{F}}(v - v_{\mathrm{g}}) \cdot n \, \mathrm{d}S = 0 \tag{1}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_{\mathrm{F}}(t)} \rho_{\mathrm{F}} v \, \mathrm{d}\Omega + \int_{\partial\Omega_{\mathrm{F}}(t)} [\rho_{\mathrm{F}} v (v - v_{\mathrm{g}}) - \sigma_{\mathrm{F}}] \cdot n \, \mathrm{d}S = \int_{\Omega_{\mathrm{F}}(t)} \rho_{\mathrm{F}} f \, \mathrm{d}S \tag{2}$$

Here ρ_{F} and v are the density and the velocity of the fluid, v_{g} is the boundary velocity of the volume $\Omega_{\mathrm{F}}(t)$, f are external volume forces acting on the fluid such as gravity and n is the outward pointing unit normal on the boundary of $\Omega_{\mathrm{F}}(t)$. The quantity σ_{F} is the stress tensor, which, for a viscous Newtonian fluid, is given by

$$\sigma_{ij}^{\mathrm{F}} = -p\delta_{ij} + \eta \left[\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right] - \frac{2}{3} \eta \left[\frac{\partial v_k}{\partial x_k} \right] \delta_{ij} \tag{3}$$

with dynamic viscosity η and hydrostatic pressure p. For incompressible flows, the density ρ_{F} is constant in space and time. Then, the expression for the stress tensor simplifies to

$$\sigma_{ij}^{\mathrm{F}} = -p\delta_{ij} + \eta \left[\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right]$$

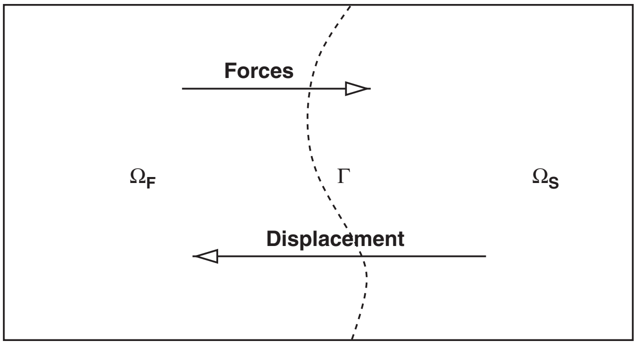

Figure 1. Schematic view of the fluid–structure interaction problem.

In the solid subdomain $\Omega_S(t)$ the time-dependent Lamé-equations

$$\rho_S \frac{\partial^2 u}{\partial t^2} - \mu\Delta u - (\lambda + \mu)\nabla\nabla \cdot u = \rho_S b \quad \text{in } \Omega_S(t) \tag{4}$$

serve as the mathematical formulation of the elasticity problem. Here, the quantity u denotes the displacement of the structure, λ and μ are the material-dependent Lamé constants,[§] ρ_S is the density of the solid material and b are external body forces acting on the structure. The Cauchy stress tensor of the solid, which is later needed in the coupling conditions, is given by

$$\sigma^S_{ij} = E_{ijkl}\varepsilon_{kl}$$

Here E denotes the fourth-order elasticity tensor which for a linear isotropic and homogeneous material is specified by the generalized Hooke's law and depends only on λ and μ. The quantity ε represents the linearized strains. Note that Equation (4) is obtained by employing the relations for linear stresses, strains and Hooke's law in the general equations of motion, see e.g. Reference [15].

The interaction of both subproblems is achieved by coupling conditions which state the continuity of the velocities and stresses at the moving interface $\Gamma(t)$. They are given by

$$v = \frac{\partial u}{\partial t} \quad \text{and} \quad \sigma^F \cdot n = \sigma^S \cdot n \quad \text{on } \Gamma(t) \tag{5}$$

For the prescription of further boundary conditions we proceed as follows: For each subdomain Ω_i we subdivide the boundary $\partial\Omega_i\backslash\Gamma$ into $\Gamma_{i,\mathrm{D}}$ and $\Gamma_{i,\mathrm{N}}$, i.e. $\partial\Omega_i\backslash\Gamma = \Gamma_{i,\mathrm{D}} \cup \Gamma_{i,\mathrm{N}}$. Here i denotes by F or S either the fluid or solid phase, respectively. Furthermore, D denotes the part of the boundary where Dirichlet conditions are applied, e.g. prescribed inflow for the fluid or fixed boundary parts for the solid, and N denotes the part of the boundary where Neumann conditions are applied, e.g. an outflow boundary for the fluid or a free boundary part for the solid.

[§] The Lamé constants can be computed from the more common elastic modulus E and the Poisson ratio v by the formulas $\mu = (E/2(1+v))$ and $\lambda = (vE/(1+v)(1-2v))$.

The algorithm which we use for the solution of the coupled problem treats the subproblems in an alternating fashion. Here, for each time step, the solution of one subproblem enters the other subproblem through the boundary conditions. This approach corresponds to an iterative solution of the global (abstract) coupled block-system by a block-Gauss–Seidel-type method in each time step.

For the solution of the elasticity subproblems we use the freely available package *tochnog* [16]. The implementation of *tochnog* is based on the velocity-stress formulation of the equations of motion. A standard linear finite element method is employed for the space discretization and the first-order in time systems are discretized using the implicit Euler method. The software provides a BiCGStab solver¶ for the solution of the arising system of linear equations. For further details on linear finite elements in elasticity theory we refer the reader to, e.g. References [17] or [18]. The flow subproblems are solved by a method which we will present in more detail in Section 3. For now, we assume that we are able to compute the flow field for time t^{n+1} from the field and data at time t^n, the grid velocity and the new fluid domain Ω^{n+1}. We denote by $V, P, U, \Gamma, V_{\mathrm{g}}$ the vectors containing the discrete values of the fluid velocity v, the pressure p, the displacements u, the nodes on the interface $\Gamma(t)$ and the mesh velocity v_{g}, respectively. For V^n, P^n, U^n and Γ^n given, our approach in the $(n+1)$th time step of the overall solution algorithm for the fluid–structure interaction problem reads as follows:

1. Set $\gamma = 1$ and for $\varphi \in \{V, P, U, \Gamma, V_{\mathrm{g}}\}$ set $\varphi^{n+1,\gamma-1} = \varphi^n$.
2. Compute from $V^{n+1,\gamma-1}$ and $P^{n+1,\gamma-1}$ the discrete values of the forces $f^{n+1,\gamma}$ on $\Gamma^{n+1,\gamma-1}$ which the fluid exerts on the structure according to the continuity condition for the stresses in (5).
3. Solve the elasticity subproblem (4) in Ω_{S} using the forces $f^{n+1,\gamma} = \sigma^{\mathrm{F}} \cdot n$ on the common interface $\Gamma^{n+1,\gamma-1}$ as Neumann boundary conditions according to (5).
4. Construct the mesh for the fluid domain $\Omega_{\mathrm{F}}^{n+1}$ by updating the boundary nodes according to the solution of the elasticity problem and then move the interior nodes accordingly.
5. Compute the grid velocity $V_{\mathrm{g}}^{n+1,\gamma}$ of the boundary $\Gamma^{n+1,\gamma}$ and prescribe the corresponding Dirichlet boundary conditions for the fluid subproblem according to the continuity condition for the velocities in (5).
6. Solve the flow subproblem in $\Omega_{\mathrm{F}}^{n+1,\gamma}$ with the algorithm described in Section 3.5.
7. If $\gamma < \gamma_{\max}$, increment γ and return to step 2, else set $\varphi^{n+1} = \varphi^{n+1,\gamma}$ for all $\varphi \in \{V, P, U, \Gamma, V_{\mathrm{g}}\}$ and proceed with the next time step.

The γ-cycle of this algorithm is depicted in Figure 2.

In step three of the algorithm, an arbitrary method to update the inner nodes according to the boundary movement can be chosen. The simplest way is to use transfinite interpolation to move the mesh. But also more elaborate methods to maintain the quality of the grid, e.g. smoothing techniques based on elliptic equations, could be used. A necessary condition here is that the grid velocity is computed from the mesh movement in order to satisfy the so-called Geometric Conservation Law [8]. This will be explained in detail in Section 3.3.

¶In principle, a CG method could be used since in our case the system of linear equations is symmetric, but the implementation of *tochnog* is not limited to linear elasticity.

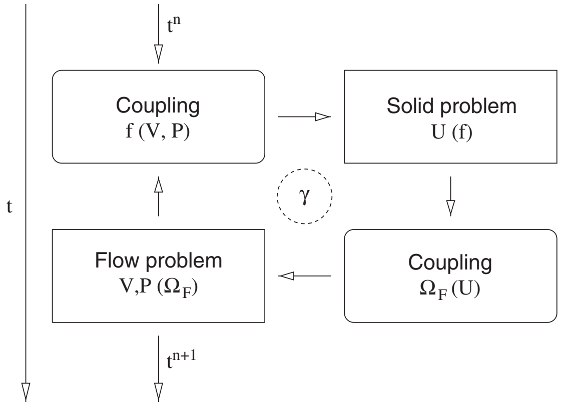

Figure 2. Coupling algorithm for one time step.

3. MOVING BOUNDARY-FITTED GRIDS

In this section we present a method to numerically solve the Navier–Stokes equations in time-dependent domains. It is based on a discretization in curvilinear coordinates proposed in Reference [14] which we generalize to time-dependent coordinate transformations. The basic idea is that the actual mesh which covers the deformable physical domain is mapped to a logical Cartesian grid which is fixed in time. The time derivative of this coordinate transformation enters the equations as the mesh velocity. Thus, the method corresponds to a discretization of the ALE formulation of the Navier–Stokes-equations as given in (1) and (2).

3.1. The coordinate transformation

Our general approach is based on a block-decomposition of the fluid domain $\Omega_F(t)$, i.e. the fluid domain is given by

$$\Omega_F(t) = \bigcup_{k=1}^{M} \Omega^k(t) \tag{6}$$

with disjoint subdomains $\Omega^k(t)$. This allows later for a straightforward parallelization based on the domain decomposition method. In this section we now describe the discretization for a *single* subdomain $\Omega^k(t)$. The coupling of the subdomains $\Omega^k(t)$ which form a cover of $\Omega_F(t)$ is straightforward and will be later explained in Section 3.6.

To discretize the Navier–Stokes-equations in general moving domains we employ a formulation which uses grid-oriented velocity components. Our approach is based on the method described in Reference [14]. We also refer the reader to Reference [19] for a similar approach in the context of isoparametric finite elements. In the following we present our extension of this method, which allows to handle time-dependent boundary-fitted grids. To this end, at each point in time and for each subdomain Ω_t^k a coordinate transformation maps the logical

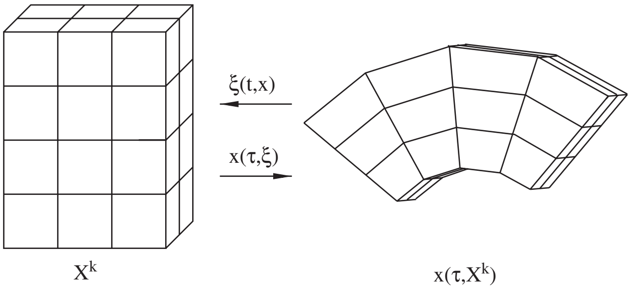

Figure 3. Transformation of the mesh for a single subdomain X^k.

subdomain, which is represented by a Cartesian grid and denoted by X^k, onto the deformed subdomain in physical space, see Figure 3.

For better readability from now on we omit the subdomain index k, i.e. Ω_t always represents a single subdomain in physical space and X its matching logical domain.

To describe the basics of the discretization we need to introduce some notation. Consider the time-dependent coordinate mapping

$$x(\tau,\xi):[0,T]\times X\rightarrow[0,T]\times\Omega_t,\quad \xi\in X,\ T\in\mathbb{R}^+ \tag{7}$$

with $\tau=t$. The logical domain is given by $X=[0,N_1]\times[0,N_2]\times[0,N_3]\subset\mathbb{R}^3$. For $0\leqslant n_\alpha<N_\alpha$, $n\in\mathbb{N}$ and $\alpha=1,2,3$, the centre of a unit cube X_j, i.e. a control volume in the logical space, is given by the multi-index

$$j=(j_1,j_2,j_3)=(n_1+\tfrac{1}{2},n_2+\tfrac{1}{2},n_3+\tfrac{1}{2})$$

If we denote by

$$e_1=(\tfrac{1}{2},0,0),\quad e_2=(0,\tfrac{1}{2},0),\quad e_3=(0,0,\tfrac{1}{2})$$

then the cell-face centres of X_j are given by $j\pm e_\alpha$, the edge centres by $j\pm e_\alpha\pm e_\beta$, and $j\pm e_1\pm e_2\pm e_3$ denotes the vertices. In this section we denote by α,β,γ indices which are cyclic in $1,2,3$.

The mapping $x(\tau,\xi)$ is now defined for the vertices of the unit cubes which build the logical domain. It is extended to the whole domain by trilinear interpolation. Figure 4 shows the mapping of a unit cube X_j onto a general hexahedral cell Ω_j, which forms a control volume in physical space.

We further require the mapping $x(\tau,\xi)$ to be boundary-fitted, that is, $x(\partial X)=\partial\Omega$ holds and ξ_α is constant everywhere on $\partial\Omega$ for an adequate $\alpha\in\{1,2,3\}$. Furthermore, the Jacobian

$$J=\det\left(\frac{\partial x_\alpha}{\partial\xi_\beta}\right)_{\alpha\beta},\quad \alpha,\beta=1,2,3$$

is assumed to be strictly positive, such that the inverse mapping $\xi(t,x)$ exists and $x(\tau,\xi)$ preserves orientation.

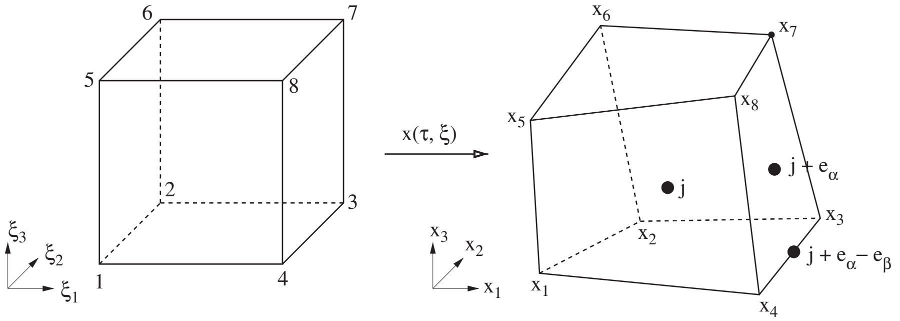

Figure 4. Notations for a control volume in logical and physical space.

The tangential base vectors are given by

$$a_{(\alpha)} = \frac{\partial x}{\partial \xi_\alpha}, \quad \alpha = 1, 2, 3 \tag{8}$$

and are easily computed from the vertices of a control volume. For example, using the notation $x_{ijkl} = x_i + x_j + x_k + x_l$, where x_i, x_j, x_k, x_l denote the coordinates of four corner points of the control volume, the tangential base vector $a_{(1)}$ in the centre of the cell is computed by $a_{(1)} = \frac{1}{4}(x_{3478} - x_{1256})$. Note that, since the mapping x is only piecewise defined, the tangential base vector $a_{(\alpha)}$ is not continuous at cell faces where ξ_α is constant.

The normal base vectors are defined by

$$a^{(\alpha)} = \nabla \xi_\alpha, \quad \alpha = 1, 2, 3 \tag{9}$$

Using the formula $J = a_{(1)} \cdot (a_{(2)} \times a_{(3)})$ and the relation $a^{(\alpha)} \cdot a_{(\beta)} = \delta^\alpha_\beta$ we obtain the so-called weighted normal base vectors

$$Ja^{(\alpha)} = a_{(\beta)} \times a_{(\gamma)} \tag{10}$$

In contrast to $a^{(\alpha)}$ or J, the weighted normal base vectors $Ja^{(\alpha)}$ are continuous at cell faces with constant ξ_α.

The weighted normal base vectors in the cell-face centres are computed by

$$Ja^{(1)}_{j+e_1} = s_{4378}, \quad Ja^{(2)}_{j+e_2} = s_{2673}, \quad Ja^{(3)}_{j+e_3} = s_{5876} \tag{11}$$

where $s_{ijkl} = \frac{1}{2}(x_j - x_l) \times (x_k - x_i)$. This expression is the average of the two weighted normal vectors for the two parallelograms constructed by adjacent sides of the cell face. For the computation of cell volumes, we use the formula (cf. Reference [14])

$$|\Omega_j| = \tfrac{1}{3}(b_1 \cdot (s_{1265} + s_{4378}) + b_2 \cdot (s_{1584} + s_{2673}) + b_3 \cdot (s_{1432} + s_{8765})) \tag{12}$$

where the vectors b_i are given by

$$b_1 = \tfrac{1}{8}(x_{3478} - x_{1256}), \quad b_2 = \tfrac{1}{8}(x_{2367} - x_{1458}), \quad b_3 = \tfrac{1}{8}(x_{5678} - x_{1234})$$

To avoid the well-known problem of pressure oscillations which can occur for projection methods on collocated grids, we will use a staggered mesh for the discretization. Then, the pressure and other scalar unknowns are stored at the cell centres and the velocity components are stored at the centres of cell faces. The continuity of $Ja^{(\alpha)}_{j+e_\alpha}$ motivates the choice of the weighted contravariant velocity components

$$V^{\alpha}_{j+e_\alpha} = Ja^{(\alpha)}_{j+e_\alpha} \cdot v \tag{13}$$

as the primary variables. Furthermore, the quantities $V^{\alpha}_{j+e_\alpha}$ have a physical meaning as the volume fluxes through the corresponding cell face centres.

For the transformation of the differential operators into a form which allows a conservative discretization, we need the basic identity

$$\sum_{\alpha=0}^{n} \frac{\partial}{\partial \xi_\alpha}\left(J \frac{\partial \xi_\alpha}{\partial x_\beta}\right) = 0, \quad \beta = 0, 1, \ldots, n \tag{14}$$

Using this identity (14) and the chain rule, the spatial derivative of a function $f(x)$ transforms in the following way:

$$\begin{aligned} \frac{\partial f}{\partial x_\alpha} &= \sum_{\beta=1}^{3} \frac{\partial \xi_\beta}{\partial x_\alpha} \frac{\partial f}{\partial \xi_\beta} + f \sum_{\beta=1}^{3} \frac{1}{J} \frac{\partial}{\partial \xi_\beta}\left(J \frac{\partial \xi_\beta}{\partial x_\alpha}\right) \\ &= \sum_{\beta=1}^{3} \frac{1}{J} \frac{\partial}{\partial \xi_\beta}\left(J \frac{\partial \xi_\beta}{\partial x_\alpha} f\right) \end{aligned} \tag{15}$$

Note that the application of the chain rule alone does not result in an expression which can be used as the basis for a conservative discretization, see, e.g. Reference [20].

The application of (15) to the divergence operator results in

$$\begin{aligned} \nabla_x u &= \sum_{\alpha=1}^{3} \frac{\partial u_\alpha}{\partial x_\alpha} = \sum_{\alpha,\beta} \frac{\partial \xi_\beta}{\partial x_\alpha} \frac{\partial u_\alpha}{\partial \xi_\beta} = \sum_{\beta=1}^{3} a^{(\beta)} \cdot \frac{\partial u}{\partial \xi_\beta} \\ &= \sum_{\alpha,\beta} \frac{1}{J} \frac{\partial}{\partial \xi_\beta}\left(J \frac{\partial \xi_\beta}{\partial x_\alpha} u_\alpha\right) = \sum_{\beta=1}^{3} \frac{1}{J} \frac{\partial}{\partial \xi_\beta} V^\beta \end{aligned} \tag{16}$$

At every point in time we can now apply the transformation rules (15) and (16) with respect to the spatial coordinates. However, the time derivative has to be transformed in the same manner. To this end we associate the time variable with ξ_0 and x_0 in the logical and the physical domain, respectively. From Equation (14) then follows for $\beta = 0$ the relation

$$0 = \frac{\partial}{\partial \tau}(J) + \sum_{\alpha=1}^{3} \frac{\partial}{\partial \xi_\alpha}\left(J \frac{\partial \xi_\alpha}{\partial t}\right) = \frac{\partial}{\partial \tau}(J) - \sum_{\alpha=1}^{3} \frac{\partial}{\partial \xi_\alpha}(Ja^{(\alpha)} \cdot v_g) \tag{17}$$

where $v_g = \partial x/\partial\tau$ denotes the grid velocity. The application of (17) to a time derivative results in

$$\begin{aligned}\frac{\partial f}{\partial t} &= \frac{\partial f}{\partial \tau} + \sum_{\alpha=1}^{3} \frac{\partial f}{\partial \xi_\alpha}\frac{\partial \xi_\alpha}{\partial t} \\ &= \frac{\partial f}{\partial \tau} - \sum_{\alpha=1}^{3} (a^{(\alpha)} \cdot v_g)\frac{\partial f}{\partial \xi_\alpha} + \frac{1}{J} f \left[\frac{\partial}{\partial \tau} J - \sum_{\alpha=1}^{3} \frac{\partial}{\xi_\alpha}(Ja^{(\alpha)} \cdot v_g)\right] \\ &= \frac{1}{J}\left(\frac{\partial}{\partial \tau}(Jf) - \sum_{\alpha=1}^{3} \frac{\partial}{\xi_\alpha}(Ja^{(\alpha)} \cdot v_g)\right) \qquad (18)\end{aligned}$$

In Equations (17) and (18) products of the grid velocity and the weighted normal base vector appear, therefore, analogous to (13), we define the weighted contravariant components of the grid velocity by

$$V_g^\alpha = Ja^{(\alpha)} \cdot v_g \quad \text{for } \alpha = 1, 2, 3 \qquad (19)$$

3.2. Discretization of the momentum equation

For the discretization based on a finite volume approach in logical coordinates we start out with the momentum equation in its fixed-grid, i.e. Eulerian, formulation. After application of (15) and (16) the momentum equation can be written as

$$\frac{\partial v}{\partial t} + \sum_{\alpha=1}^{3} \frac{1}{J}\frac{\partial}{\partial \xi_\alpha}(V^\alpha v) + \nabla p - \sum_{\alpha,\beta=1}^{3} v \frac{1}{J} a_\alpha^{(\beta)} e^{(\alpha)} = f \qquad (20)$$

with the kinematic viscosity $v = \eta/\rho_\infty$, ρ_∞ the constant density and

$$e^{(\alpha)} = \begin{pmatrix} \dfrac{\partial v_1}{\partial x_\alpha} + \dfrac{\partial v_\alpha}{\partial x_1} \\ \dfrac{\partial v_2}{\partial x_\alpha} + \dfrac{\partial v_\alpha}{\partial x_2} \\ \dfrac{\partial v_3}{\partial x_\alpha} + \dfrac{\partial v_\alpha}{\partial x_3} \end{pmatrix}$$

For the spatial discretization of the terms in the momentum equation we use the methods as described in Reference [14]. Due to the staggered mesh, Equation (20) is not integrated over the control volumes Ω_j, but, for each coordinate direction, the integration is carried out over the control volumes with the cell centre given by $j + e_\alpha$, i.e. the original volume Ω_j shifted by e_α, respectively. The gradients of both the pressure and the expressions e^α in the viscous term are handled by the so-called integration path method. Here, the gradient of a quantity is integrated over curves through surrounding nodes and a small associated system of linear equations is solved to obtain the discretized gradient. For the details, see Reference [21]. The convective term is handled by a simple hybrid scheme using a combination of first-order upwind and second-order central differences. Here, the blending of first- and second-order terms is controlled by the local mesh Peclet-number.

Note from (11) and (13) that due to the staggered mesh the velocity components and the weighted normal base vectors are defined only in the grid points x_{j+e_α}. For the evaluation of these quantities in other locations we use linear interpolation. This procedure is sufficient to guarantee that a constant velocity field is invariant under transformation to contravariant components and *vice versa*, see Reference [14].

What remains is the discretization of the time derivative in (20). The application of (18) yields

$$\frac{\partial v}{\partial t} = \frac{1}{J}\frac{\partial}{\partial \tau}(Jv) - \sum_{\alpha=1}^{3}\frac{1}{J}\frac{\partial}{\partial \xi_\alpha}(V_{\mathrm{g}}^{\alpha} v) \tag{21}$$

The first term on the right-hand side of (21) is integrated over the shifted control volume Ω_{j+e_α} and then approximated by the midpoint rule. This results in

$$\int_{\Omega_{j+e_\alpha}} \frac{1}{J}\frac{\partial Jv}{\partial t}\,\mathrm{d}\Omega = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_{j+e_\alpha}} v\,\mathrm{d}\Omega \approx \frac{\mathrm{d}}{\mathrm{d}t}|\Omega_{j+e_\alpha}|v \tag{22}$$

The total time derivative d/dt in (22) is treated automatically by the projection method, which is described later in Section 3.5. The second term of the right-hand side in (21) is combined with the convective term in (20) to

$$\sum_{\alpha=1}^{3}\frac{1}{J}\frac{\partial}{\partial \xi_\alpha}((V^{\alpha} - V_{\mathrm{g}}^{\alpha})v) \tag{23}$$

Expression (23) can be interpreted as a new convective velocity, i.e. the flow velocity in relation to the moving grid. This modified convective term is discretized in the same manner as the original convective term. We obtain

$$((V^{\alpha} - V_{\mathrm{g}}^{\alpha})v)|_{j}^{j+2e_\alpha} + ((V^{\beta} - V_{\mathrm{g}}^{\beta})v)|_{j+e_\alpha-e_\beta}^{j+e_\alpha+e_\beta} + ((V^{\gamma} - V_{\mathrm{g}}^{\gamma})v)|_{j+e_\alpha-e_\gamma}^{j+e_\alpha+e_\gamma} \tag{24}$$

The computation of the grid velocity V_{g} needed in (24) is the subject of the next section.

3.3. Discretization of the geometric conservation law

The system of conservation laws for momentum and mass balance (1) and (2) has to be completed by the so-called geometric conservation law (GCL) which reads (cf. Reference [8])

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\mathrm{d}\Omega - \int_{\partial\Omega} v_{\mathrm{g}}\cdot n\,\mathrm{d}S = 0 \tag{25}$$

In Reference [9] the GCL is derived from the mass conservation equation (1) by setting the flow velocity to zero. This is motivated by the requirement that the mesh movement is not allowed to have an effect on the flow field. It is essential that the discretization satisfies the GCL. In, e.g. Reference [22] it is shown that a violation of the GCL results in spurious oscillations due to artificial mass sources and sinks. Furthermore, if the GCL is fulfilled, then we can easily employ a standard projection method, because the equation for conservation of mass (1) reduces to the well-known divergence-free constraint for the velocity field. This can be seen immediately by using (25) in (1):

$$\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\mathrm{d}\Omega - \int_{\partial\Omega} v_{\mathrm{g}}\cdot n\,\mathrm{d}S}_{=0} + \int_{\partial\Omega} v\cdot n\,\mathrm{d}S = 0 \Rightarrow \nabla\cdot v = 0 \tag{26}$$

In References [9, 22] the GCL is used to compute the grid velocities from the grid node positions at different times. Thus, the GCL is automatically satisfied by the discretization scheme. Especially in three dimensions, this approach is appealing and will therefore be used in our method. However, in order to directly recover the weighted contravariant components of the grid velocity, it is convenient to look at the GCL in the following way: When we transform the differential operators into the logical coordinates (τ,ξ), we make use of the basic identity (14) to achieve a conservative discretization. This algebraic identity now has to be fulfilled on the discrete level as well. To this end, we integrate Equation (14) for $\beta=0$ over a logical control volume X_j. This results in

$$\begin{aligned} 0 &= \int_{X_j}\sum_{\alpha=0}^{3}\frac{\partial}{\partial\xi_\alpha}\left(J\frac{\partial\xi_\alpha}{\partial x_0}\right)\mathrm{d}\xi \\ &= \int_{\Omega_j}\frac{1}{J}\frac{\partial}{\partial\xi_0}J\frac{\partial\xi_0}{\partial x_0}\,\mathrm{d}x + \int_{\Omega_j}\frac{1}{J}\sum_{\alpha=1}^{3}\frac{\partial}{\partial\xi_\alpha}\left(J\frac{\partial\xi_\alpha}{\partial x_0}\right)\mathrm{d}x \end{aligned} \tag{27}$$

With the relation $\frac{\partial\xi_\alpha}{\partial x_0} = -a^{(\alpha)}v_{\mathrm{g}}$, $x_0=t$, $\xi_0=\tau$ as in Section 3.1 and with (27) we obtain

$$0 = \int_{\Omega_j}\frac{1}{J}\frac{\partial}{\partial\tau}J\,\mathrm{d}x - \int_{\Omega_j}\frac{1}{J}\sum_{\alpha=1}^{3}\frac{\partial}{\partial\xi_\alpha}(Ja^{(\alpha)}\cdot v_{\mathrm{g}})\,\mathrm{d}x \tag{28}$$

$$= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_j}\mathrm{d}x - \int_{\Omega_j}\sum_{\alpha=1}^{3}\frac{1}{J}\frac{\partial}{\partial\xi_\alpha}V_{\mathrm{g}}^{\alpha}\,\mathrm{d}x \tag{29}$$

$$= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_j}\mathrm{d}x - \int_{\Omega_j}\nabla\cdot v_{\mathrm{g}}\,\mathrm{d}x \tag{30}$$

Thus we see that the GCL corresponds to a finite volume discretization of the basic identity (14). Furthermore, we can use Equation (29) as a rule to compute the weighted contravariant components of the grid velocity as needed for the discretization of the convective terms (24). To avoid further interpolation of the grid velocity, we discretize the GCL for the shifted control volumes $\Omega_{j+e_\alpha}, \alpha=1,2,3$. Thus, we can compute the grid velocity directly at the points where it is needed, namely in the cell face centres of the shifted volumes instead of the centred ones. To this end, we consider Equation (29) for a shifted control volume Ω_{j+e_α}. We obtain

$$\int_{\Omega_{j+e_\alpha}}\sum_{i=1}^{3}\frac{1}{J}\frac{\partial}{\partial\xi_i}V_{\mathrm{g}}^{i}\,\mathrm{d}\Omega = \sum_{i=1}^{3}\int_{X_j+e_\alpha}\frac{\partial}{\partial\xi_i}V_{\mathrm{g}}^{i}\,\mathrm{d}\xi \tag{31}$$

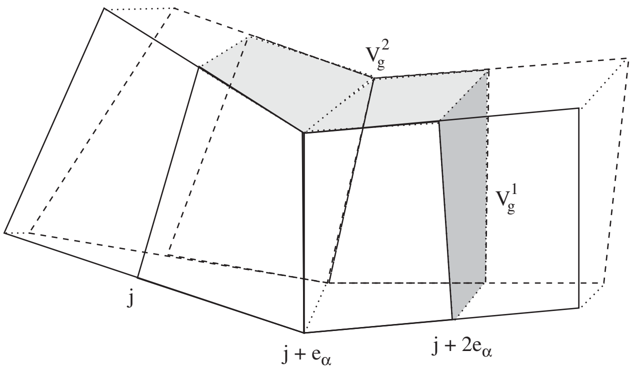

Figure 5. Calculation of the grid velocity from the node movement (two-dimensional case).

For the case $i=\alpha$ the integral is approximated by

$$\int_{j_\alpha}^{j_\alpha+1}\int_{j_\beta-(1/2)}^{j_\beta+(1/2)}\int_{j_\gamma-(1/2)}^{j_\gamma+(1/2)} \frac{\partial}{\partial\xi_\alpha} V_{\mathrm{g}}^{\alpha}\,\mathrm{d}\xi_\alpha\,\mathrm{d}\xi_\beta\,\mathrm{d}\xi_\gamma \tag{32}$$

$$= \int_{j_\beta-(1/2)}^{j_\beta+(1/2)}\int_{j_\gamma-(1/2)}^{j_\gamma+(1/2)} V_{\mathrm{g}}^{\alpha}\Big|_{j_\alpha}^{j_\alpha+1}\,\mathrm{d}\xi_\beta\,\mathrm{d}\xi_\gamma \approx V_{\mathrm{g}}^{\alpha}\Big|_{j}^{j+2e_\alpha} \tag{33}$$

where continuity of V_{g}^{α} in x_{j+e_α} is used for the integration in α-direction and the midpoint rule is used for the two remaining integrals. In a similar way we obtain for $i=\beta$ (and analogous for $i=\gamma$)

$$\int_{j_\alpha}^{j_\alpha+1}\int_{j_\beta-(1/2)}^{j_\beta+(1/2)}\int_{j_\gamma-(1/2)}^{j_\gamma+(1/2)} \frac{\partial}{\partial\xi_\beta} V_{\mathrm{g}}^{\beta}\,\mathrm{d}\xi_\alpha\,\mathrm{d}\xi_\beta\,\mathrm{d}\xi_\gamma \tag{34}$$

$$= \int_{j_\alpha}^{j_\alpha+1}\int_{j_\gamma-(1/2)}^{j_\gamma+(1/2)} V_{\mathrm{g}}^{\beta}\Big|_{j_\beta-(1/2)}^{j_\beta+(1/2)}\,\mathrm{d}\xi_\alpha\,\mathrm{d}\xi_\gamma \approx V_{\mathrm{g}}^{\beta}\Big|_{j+e_\alpha-e_\beta}^{j+e_\alpha+e_\beta} \tag{35}$$

Altogether, this results in

$$\int_{\Omega_{j+e_\alpha}} \sum_{i=1}^{3} \frac{1}{J}\frac{\partial}{\partial\xi_i} V_{\mathrm{g}}^{i}\,\mathrm{d}\Omega \approx \sum_{i=1}^{3} V_{\mathrm{g}}^{i}\Big|_{j+e_\alpha-e_i}^{j+e_\alpha+e_i} \tag{36}$$

In (36) the index α denotes the coordinate direction in which the control volume is shifted by e_α, the index i denotes the coordinate direction of the cell face location where the grid velocity is evaluated, see Figure 5. This expression can be interpreted as the grid velocity of the cell face centre in its normal direction weighted by the cell face area, which is analogous to the flow velocity. Now we split the change in volume given by $\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_{j+e_\alpha}}\mathrm{d}\Omega$, see (29), into the six contributions which are given by the movements of the individual cell faces and approximate the time derivative by the first order Euler method, i.e.

$$\frac{1}{\Delta t}(|\Omega^{n+1}_{j+e_\alpha}| - |\Omega^{n}_{j+e_\alpha}|) = \frac{1}{\Delta t}\sum_{i=1}^{3}[(\Delta\Omega)_{j+e_\alpha+e_i} - (\Delta\Omega)_{j+e_\alpha-e_i}] \tag{37}$$

Then we compute the grid velocity by

$$V^{i}_{\mathrm{g},j+e_\alpha+e_i} = \frac{(\Delta\Omega)_{j+e_\alpha+e_i}}{\Delta t}, \quad \alpha = 1,2,3, \; i = 1,2,3 \tag{38}$$

The right-hand side in Equation (38) is evaluated by formula (12) for the computation of cell volumes. The corner points of $(\Delta\Omega)_{j+e_\alpha+e_i}$ in the case $i=\alpha$ are given by

$$\begin{aligned}&\tfrac{1}{2}(x^k_{j+e_\alpha+e_\beta+e_\gamma} + x^k_{j+3e_\alpha+e_\beta+e_\gamma}), \quad \tfrac{1}{2}(x^k_{j+e_\alpha+e_\beta-e_\gamma} + x^k_{j+3e_\alpha+e_\beta-e_\gamma})\\ &\tfrac{1}{2}(x^k_{j+e_\alpha-e_\beta+e_\gamma} + x^k_{j+3e_\alpha-e_\beta+e_\gamma}), \quad \tfrac{1}{2}(x^k_{j+e_\alpha-e_\beta-e_\gamma} + x^k_{j+3e_\alpha-e_\beta-e_\gamma})\end{aligned} \tag{39}$$

where $k\in\{n,n+1\}$. In the case of $i\neq\alpha$, the volume $(\Delta\Omega)_{j+e_\alpha+e_i}$ is split into the two sub-volumes $(\Delta\Omega)_{j+e_i}$ and $(\Delta\Omega)_{j+2e_\alpha+e_i}$. The corner points of $(\Delta\Omega)_{j+e_i}$ are given by

$$\begin{aligned}&x^k_{j+e_\alpha+e_i+e_\beta}, \quad \tfrac{1}{2}(x^k_{j-e_\alpha+e_i+e_\beta} + x^k_{j+e_\alpha+e_i+e_\beta}), \quad \beta\neq i,\alpha\\ &x^k_{j+e_\alpha+e_i-e_\beta}, \quad \tfrac{1}{2}(x^k_{j-e_\alpha+e_i-e_\beta} + x^k_{j+e_\alpha+e_i-e_\beta}), \quad \beta\neq i,\alpha\end{aligned} \tag{40}$$

and the corner points of $(\Delta\Omega)_{j+2e_\alpha+e_i}$ are given by

$$\begin{aligned}&x^k_{j+e_\alpha+e_i+e_\beta}, \quad \tfrac{1}{2}(x^k_{j+e_\alpha+e_i+e_\beta} + x^k_{j+3e_\alpha+e_i+e_\beta}), \quad \beta\neq i,\alpha\\ &x^k_{j+e_\alpha+e_i-e_\beta}, \quad \tfrac{1}{2}(x^k_{j+e_\alpha+e_i-e_\beta} + x^k_{j+3e_\alpha+e_i-e_\beta}), \quad \beta\neq i,\alpha\end{aligned} \tag{41}$$

again with $k\in\{n,n+1\}$.

3.4. Discretization of the mass balance equation

For the discretization of the mass balance equation we integrate the divergence-free constraint (26) over a control volume Ω_j and employ the transformation rule (16) to obtain

$$0 = \int_{\Omega_j} \nabla\cdot v \,\mathrm{d}x = \int_{\Omega_j} \sum_{\alpha=1}^{3} \frac{1}{J}\frac{\partial}{\partial\xi_\alpha} V^\alpha \,\mathrm{d}x = \sum_{\alpha=1}^{3}\int_{X_j} \frac{\partial}{\partial\xi_\alpha} V^\alpha \,\mathrm{d}\xi \tag{42}$$

$$= \sum_{\alpha=1}^{3}\int_{j_1-(1/2)}^{j_1+(1/2)}\int_{j_2-(1/2)}^{j_2+(1/2)}\int_{j_3-(1/2)}^{j_3+(1/2)} \frac{\partial}{\partial\xi_\alpha} V^\alpha \,\mathrm{d}\xi_3\,\mathrm{d}\xi_2\,\mathrm{d}\xi_1$$

$$\approx \sum_{\alpha=1}^{3}(V^\alpha_{j+e_\alpha} - V^\alpha_{j-e_\alpha}) \tag{43}$$

Here, we have used the same approximations as in (36), applied to the case of the control volume Ω_j.

Finally, we arrive at the following set of semi-discretized equations:

$$\begin{aligned} DV &= 0 \\ \frac{\mathrm{d}}{\mathrm{d}t}(Qv) + M(V^{\alpha} - V_{\mathrm{g}}^{\alpha}, v) + QGp - QF &= 0 \end{aligned} \tag{44}$$

where D is the discrete divergence operator, G accounts for the discrete gradient, Q represents multiplication with the volume $|\Omega|$, and M denotes the difference of the discrete convective and the discrete diffusive terms.

3.5. The extended projection method

For the solution of the coupled system of Equations (1) and (2) we employ a Chorin-type projection method [23, 24] which basically consists of the following parts: In a first step, we compute an auxiliary velocity field from the momentum equations (2), neglecting the pressure gradient. This auxiliary velocity field will in general not satisfy the divergence-free constraint. In a second step, we use the condition for the conservation of mass (1) to derive a Poisson-like equation for the pressure. This equation has to be solved, usually by an iterative method. The computed pressure is then used in a correction step to project the auxiliary velocity field onto a divergence-free one. This well-known approach will now be generalized to the case of moving grids.

To this end, we assume that the shape Ω^{n+1} of the fluid domain at time t^{n+1} is known. Then the grid velocity V_{g} and the necessary grid-dependent base vectors $\sqrt{g}\mathbf{a}^{(\alpha),n+1}$ can be computed. In our test cases the domain Ω^{n+1} is either explicitly prescribed or computed as the solution of the structural mechanic subproblem. Then, using the notation from the previous section, the individual steps of the extended projection method for the moving-grid problem read as follows: First we have to discretize the set of equations (44) in time. In the current implementation we use the explicit Euler method for the velocities, which is convergent of first order provided that appropriate CFL-conditions are satisfied. The pressure is always evaluated at the new time t^{n+1}. The fully discretized system to be solved now reads

$$DV^{n+1} = DV^{n} = 0 \tag{45}$$

$$Q^{n+1}v^{n+1} - Q^{n}v^{n} + \Delta t[M(V^{\alpha,n} - V_{\mathrm{g}}^{\alpha,n}, v^{n}) + Q^{n+1}G^{n+1}p^{n+1} - Q^{n+1}F^{n+1}] = 0 \tag{46}$$

The discrete momentum equation (46) is still vector-valued. By multiplication with the corresponding weighted contravariant base vector we obtain a scalar equation for each velocity component. The intermediate velocity field is computed according to

$$V^{*} = \frac{Q^{n}}{Q^{n+1}}V^{n} - \Delta t\frac{1}{Q^{n+1}}M(V^{n} - V_{\mathrm{g}}^{n})V^{n} + \Delta t\frac{Q^{n}}{Q^{n+1}}F^{n} \tag{47}$$

Note that in Equation (47) the relation $V^{\alpha,*} = \sqrt{g}\mathbf{a}^{(\alpha),n} \cdot \tilde{v}^{n+1}$ holds with the Cartesian components $\tilde{v}^{n+1}$ of the intermediate velocity field. To obtain $\tilde{V} = \sqrt{g}\mathbf{a}^{(\alpha),n+1} \cdot \tilde{v}^{n+1}$ from V^{*} we change the basis by

$$\tilde{V} = T_{\mathscr{B}^{n}}^{\mathscr{B}^{n+1}}V^{*} = C_{n+1}C_{n}^{-1}V^{*} \tag{48}$$

with $C_{t} = \left(\sqrt{g}\mathbf{a}^{(1),t}, \sqrt{g}\mathbf{a}^{(2),t}, \sqrt{g}\mathbf{a}^{(3),t}\right)^{\mathrm{T}}$.

Now we apply the discrete divergence operator to the relation

$$\tilde{V} - V^{n+1} = \Delta t G p^{n+1} \tag{49}$$

and use the equation for mass balance (45) to obtain a linear system of equations for the pressure

$$DGp^{n+1} = \frac{1}{\Delta t} D\tilde{V} \tag{50}$$

We then solve (50) by a preconditioned BiCGStab method. Here, faster iterative methods such as algebraic multigrid can be employed as well, see e.g. References [25] or [26].

In the last step of the projection method the pressure is used to compute the divergence-free velocity field according to

$$V^{n+1} = \tilde{V} - \Delta t G p^{n+1} \tag{51}$$

To guarantee the convergence of this explicit discretization in time, the time step size must be limited properly. In the case of non-vanishing grid velocity the CFL-condition for the convective term reads (cf. Reference [9])

$$\Delta t \leqslant \min_{1 \leqslant \alpha \leqslant 3, \Omega_j} \frac{|\Omega_{j+e_\alpha}|}{|V^\alpha_{j+e_\alpha} - V^\alpha_{\mathrm{g},j+e_\alpha}|} \tag{52}$$

3.6. Boundary conditions and parallelization issues

The spatial discretization for each subdomain has to be completed by boundary conditions. This is necessary for parts of the boundary which coincide with the physical boundary as well as for artificial inner boundaries between adjacent subdomains which arise due to the block-decomposition of the physical domain. In both cases the implementation is based on a ghost cell technique. The computational grid of a single subdomain is extended by one layer of cells in each coordinate direction which is then used for the implementation of the appropriate boundary condition. In the case of a physical boundary, normal velocity components are directly set in the corresponding cell face centre whereas tangential components are extrapolated to the ghost cell layer in such a way that the discretization stencils can be directly applied. The boundary conditions for the fluid subproblem on the moving fluid–structure interface are treated as inhomogeneous Dirichlet boundary conditions, where the boundary values are computed from the movement of the boundary according to the GCL as described in Section 3.3.

The additional ghost cell layer of each subdomain is also used to set the necessary boundary values at artificial inner boundaries between subdomains. Here, the ghost cell values are updated with the computed unknowns in the inner cell layer of the adjacent subdomain. Since the domain decomposition is used for parallelization, the neighbouring subdomain can reside on a different processor. In this case a communication step between the corresponding processors is involved. Copying of variables between subdomains is necessary

- once in each time step to exchange V and $\tilde{V}$,
- for each matrix–vector product involving the application of the discrete pressure Poisson operator in the BiCGStab solver,

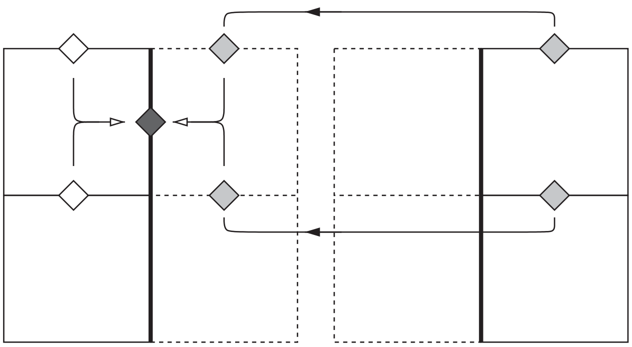

Figure 6. Exchange of data at boundaries, here shown for the basis transformation in 2D.

- for the basis transformation (48), see Figure 6.

All other computations can be performed independently for each single subdomain.

4. NUMERICAL RESULTS

In this section we present some numerical examples. We validate our proposed method for moving grids on two test cases and show that the convergence for the discretization in space is of second-order. Later we show the properties of our approach when applied to fluid–structure interaction problems.

4.1. Examples for flow on moving grids

We validate the proposed method and demonstrate that second-order convergence in space is obtained for moving grids. It can also be seen that the discretization scheme fulfills the discrete GCL.

4.1.1. Movement of inner grid nodes. In the first test case we consider the flow in a three-dimensional channel with the dimensions 3 × 1 × 1 and show that an initially given velocity profile is not disturbed by a prescribed movement of the inner grid nodes. The exact solution is given by the parabolic velocity profile

$$v_1(x_1,x_2,x_3,t) = 16\, v_{\max}\, x_2(1-x_2)x_3(1-x_3), \quad v_2 = 0, \quad v_3 = 0 \tag{53}$$

with a constant pressure gradient of $\nabla p = -1$ and applied volume forces specified by

$$f(x_1,x_2,x_3,t) = 32\, v_{\max}\, \nu(x_3 - x_3^2 + x_2 - x_2^2) + \nabla p \tag{54}$$

The problem setting is completed by homogeneous Dirichlet no-slip boundary conditions at all walls except at the inflow and outflow boundary, where the given parabolic velocity profile and homogeneous Neumann boundary conditions are used, respectively. The initial data for all simulations was a fully developed flow profile from a previous computation on a static uniform grid. The simulations were carried out with the values $v_{\max} = 1$ and $\nu = 0.01$, which results in a Reynolds number $Re = u_\infty L/\nu$ of about 44, based on the mean inflow velocity u_∞ and the channel height L.

We performed simulations on three different grids for this example. The main interest was to study the convergence behaviour of our approach for a moving grid. To this end, the grid is deformed from its original equidistant Cartesian configuration by the movement of the inner grid nodes during the simulation over a time interval of length T. Here, the displacement of a point with position x at time t is prescribed by

$$\operatorname{disp}(x,t) = \begin{cases} -\frac{1}{2}d(t)\cos\left(\pi x_1 - \frac{\pi}{2}\right) + \frac{1}{2} & \text{for } \frac{1}{2} \leqslant x_1 \leqslant \frac{5}{2} \\ 0 & \text{else} \end{cases} \tag{55}$$

with

$$d(t) = \frac{1}{2}d_{\max}\left(1 - \cos\left(2\pi\frac{t-t_0}{T}\right)\right), \quad d_{\max} = (0, 0.15, 0.2) \tag{56}$$

Figure 7 shows a view inside the three-dimensional mesh at time $T/2$; i.e. when the maximum deformation of the grid occurs.

The grid deformation was chosen in such a way that the predominant deformations are orthogonal to the direction of the main flow and not parallel to the main flow, since the solution is constant with respect to x_1. For comparison, a reference calculation on the original undeformed Cartesian grid and another reference calculation on the stationary, but maximally deformed mesh configuration of time $T/2$ were carried out. For all three simulations we measured the error of the computed solution with respect to the exact solution given by (53) in the grid-dependent norm

$$e_{L^2,h} := \left(\sum_{\Omega_j} |\Omega_j| |v(x_j) - v_j|^2\right)^{1/2} \tag{57}$$

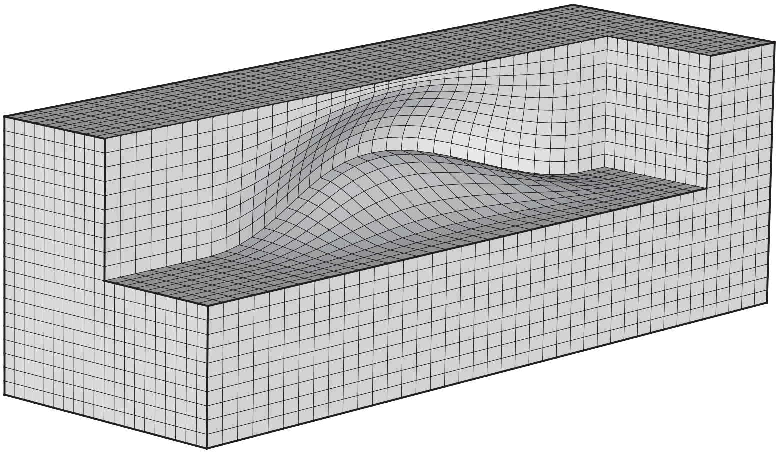

Figure 7. View inside the mesh at time $T/2$ when maximum deformation occurs.

Figure 8 shows the temporal development of $e_{L^2,h}$ for all three grids (on the left) and the divergence of the velocity field (for the moving grid calculation only, on the right side). As expected, the uniform Cartesian grid gives the best approximation and thus exhibits the smallest error. The approximation quality of the deformed fixed mesh of time $T/2$ is lower and thus results in the upper error curve. The error of the moving mesh computation is bounded by those two curves. This is to be expected, since the error cannot be lower than the error on the undeformed mesh at any given point in time. On the other hand, the error of the moving grid calculation should not be larger than that of the deformed mesh computation. The computations show further that essentially the divergence of the velocity field, computed by (42), and thus the mass balance is independent of the grid movement.

Table I shows the maximum error over the time interval $[0,T]$ of the moving grid calculation for different mesh resolutions and the corresponding numerical rate of convergence ρ. From these results we clearly observe a convergence rate of second-order. All simulations for this test case were carried out with the same time step size to minimize the influence of the time discretization on the error measurement. The size of the time step was chosen such that the CFL-condition for the finest mesh in its fully deformed state was fulfilled. This value was then used for all mesh resolutions. The mesh size for the coarsest mesh was $h=(h_x,h_y,h_z)=(0.15,0.1,0.1)$.

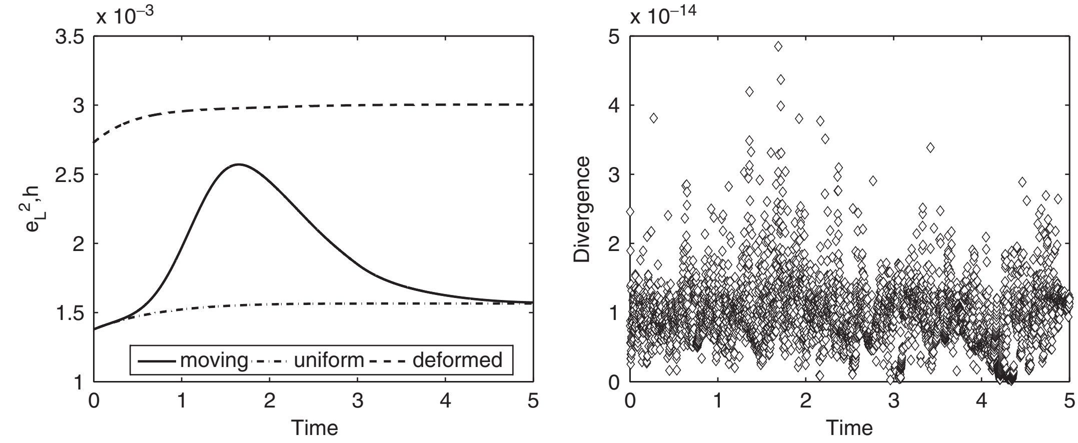

Figure 8. $e_{L^2,h}$ of computed solution over time for moving, uniform fixed in time and deformed fixed in time mesh (left), divergence of V for the moving grid (right).

Table I. Maximum error over time interval $[0,T]$ and numerical rate of convergence for different norms.

	$e_{L^1,h}$	$\rho_{e_{L^1,h}}$	$e_{\infty,h}$	$\rho_{e_{\infty,h}}$	$e_{L^2,h}$	$\rho_{e_{L^2,h}}$
h	1.30_{-2}	—	4.45_{-2}	—	9.96_{-3}	—
$h/2$	3.36_{-3}	1.95	6.70_{-3}	2.73	2.51_{-3}	1.98
$h/4$	8.32_{-4}	2.01	1.85_{-3}	1.85	6.25_{-4}	2.01
$h/8$	2.09_{-4}	1.99	4.81_{-4}	1.94	1.56_{-4}	2.00

4.1.2. Flow through a channel with a moving indentation. As a second test case we consider the simulation of a flow through a channel with a time-dependent indentation. This test has first been analysed experimentally in Reference [27] and was later numerically studied in References [28, 29]. The setting is sketched in Figure 9. The time-dependent indentation at the channel's bottom is given by

$$x_2(t,x_1)=\begin{cases}\beta(t) & \text{for } 0<x_1<a_1\\ 0.5\beta(t)(1-\tanh(\alpha(x_1-a_2))) & \text{for } a_1<x_1<a_2\\ 0 & \text{for } x_1>a_3\end{cases}\tag{58}$$

with $\alpha=4.14, a_1=4H, a_3=6.5H, a_2=0.5(a_1+a_3)$ and

$$\beta(t)=0.5\beta_{\max}(1-\cos(2\pi t^*)),\quad t^*=(t-t_0)/T$$

Here, H is the channel height, T is the length of the time interval, $\beta(t)$ is the size of the channel indentation at time t and $\beta_{\max}=0.38H$ is the maximum indentation at normalized time $t^*=0.5$.

The Reynolds number of the fluid based on the mean inflow velocity and the channel height is 507. The initial condition for the simulation was a fully developed parabolic flow profile. This profile is also prescribed as Dirichlet boundary condition at the inflow boundary. In Reference [29] two different mesh resolutions were considered, a coarse mesh with 91×20 control volumes and a finer mesh with 221×40 control volumes. It was experienced that severe oscillations occur for the coarse grid computations without upwinding scheme, the use of first-order upwinding however introduces excessive numerical diffusion in the discrete solution. Therefore, we restrict ourself to the case of the fine grid with a resolution of 221×40 control volumes and employ pure second-order central differences for the discretization of the convective term. The two-dimensionality of this test case is handled in our implementation by employing periodic boundary conditions in the x_3-direction.

Figure 10 shows the contours of the velocity component in the main flow direction for successive time values. Here, only the region of interest downstream of the indentation is shown. Blue colour represents a velocity of 2.75 m/s, red colour represents a value of −0.75 m/s.

Altogether, the results show a good qualitative agreement with the results published in Reference [29]. The velocity in the main flow direction reaches its peak value of 2.6603 m/s at time $t^*=0.4$, which is roughly a value of 77.3% higher than the constant inflow velocity of 1.5 m/s. The reference value in Reference [29] is given by 2.645 m/s, which is 76.3% higher than the inflow value. After time $t^*=0.5$, when the maximum indentation occurs, the flow rate decreases, until it reaches approximately the initial state at $t^*=1.0$. Between $t^*=0.2$ and 0.3 a first eddy detaches from the channel wall immediately behind the indentation. This is the beginning of a periodic evolution of eddies which form alternatingly at the upper and lower channel wall. The distance between the eddies in the main flow direction approximately equals the height of the channel. An observation made in Reference [28] is the breaking-up of the first eddy into two eddies at the upper channel wall. This flow feature can also be seen in our results, cf. Figure 10 for the time values t^* from 0.5 up to 0.8.

The overall flow behaviour is reflected by the evolution of the wall shear stress $\sigma_{\mathrm{w}}=\eta(\partial v_1/\partial x_3)$, which is shown in Figure 11 at the top and bottom channel walls for the dimensionless time values $t^*=0.5$ (on the left) and $t^*=0.7$ (on the right). Here we also observe a

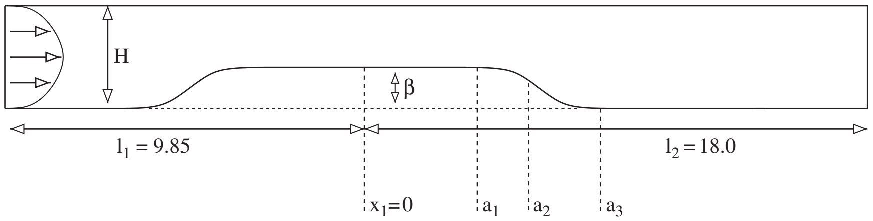

Figure 9. Channel with time-dependent indentation.

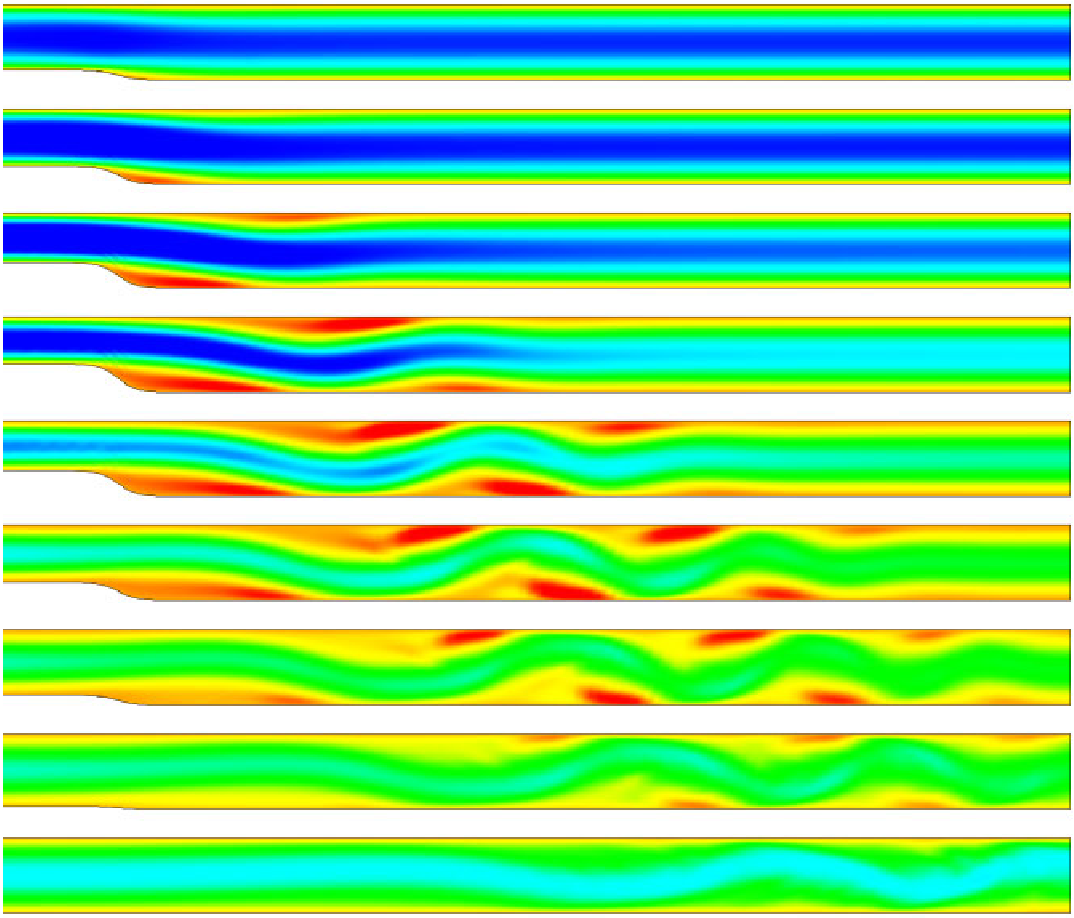

Figure 10. Contour of the v_1 velocity component at consecutive times t^* from 0.2 to 1.0.

good qualitative agreement with the results given in Reference [29]. Unfortunately no quantitative results are available but distinctive features of the plots confirm the good agreement. In particular positions along the channel at which peaks of the wall shear stress appear match

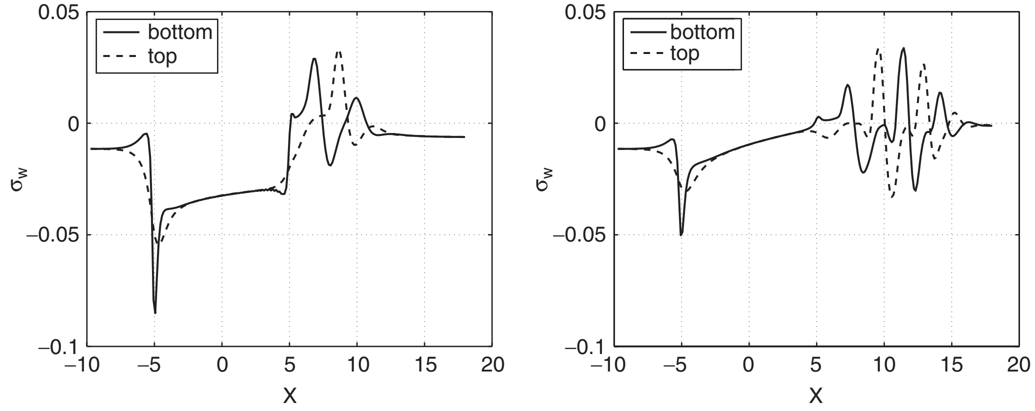

Figure 11. Shear stress σ_{W} at the lower and upper channel wall for times $t^* = 0.5$ (left) and $t^* = 0.7$ (right).

very well. Differences show only in the magnitude of the peaks. In our results the amplitude is slightly underestimated. These differences can be attributed to the different time discretization schemes used in both studies. In Reference [29] it was observed that different sizes of the time step employed in the same numerical scheme lead to a different prediction for the height of the wall shear stress peaks.

4.1.3. Parallel efficiency. In this section we present results regarding the parallel performance of our flow solver on moving grids. We used the setting of the testproblem of Section 4.1.1 to calculate speedups and scaleups. The fluid domain is partitioned in subdomains as described in Section 3.6 and each subdomain is mapped to a processor. For the scaleup calculations the resolution of each subdomain was fixed at $64 \times 32 \times 32 = 65\,536$ mesh cells and the number of subdomains was increased together with the number of used processors. For the speedups, the fluid domain was resolved with $192 \times 96 \times 96 = 1\,769\,472$ mesh cells.[‖] The domain was then divided in an appropriate number of subdomains and distributed to the different processors. All computations were performed on Pentium4 Xeon 3.2 GHz dual processor nodes each with 2 GB main memory and MyrinetXP interconnect.

Since the scaling behaviour of the employed linear solver dominates the parallel efficiency, we first present the dependence of the implemented BiCGStab(l) (cf. Reference [30]) solver on the mesh width h for two different preconditioners, namely a diagonal scaling and the lifting interpolet preconditioner presented in Reference [31].

In Table II the iteration numbers for a BiCGStabL(8) method are given for the two employed preconditioners. The iterations of the solver were stopped when the residual dropped below 1e − 10. As expected, we see that the iteration numbers grow with order h when the diagonal scaling is employed, whereas the iteration numbers of the lifting interpolet preconditioned solver grow only as the square root of h.

[‖] This resolution was chosen such that the whole domain fits into the memory of a single node.

Table II. Iteration numbers of preconditioned BiCGStabL(8) method on a single processor machine. Dependence on mesh size h.

$1/h$	Lifting interpolet	Diagonal scaling
8	40	48
16	56	96
32	64	184
64	80	384
128	112	744

Table III. Scaleup for example 1. Constant number of 65 536 mesh cells per processor. Executing times in seconds.

Processors		1st step	First 10 steps
	Iterations	72	504
1	t_R	3.68	36.61
	t_S	3.59	25.18
	Iterations	232	1552
8	t_R	3.77	37.72
	t_S	15.53	104.57
	Iterations	424	3088
64	t_R	3.81	37.92
	t_S	29.44	214.51

In Table III the iteration numbers of the linear solver, the executing time in seconds for the linear solver, denoted by t_S, and the time for the parts of main loop excluding the solver, denoted by t_R, are shown. The same stopping criterion as above was used. In the first column the respective values for the first time step and in the second column the accumulated values for the first ten time steps of the moving grid method are given. In the optimal case the executing time in seconds would be constant for each number of processors because the problem size grows proportionally to the number of employed processors. However, this is prevented by the h-dependence of the linear solver as well as the dependence of the preconditioner on the number of subdomains. All other parts of the moving grid algorithm show an optimal scaling behaviour which is expected. The executing time t_R is constant per time step and independent on the number of processors used.

In Table IV we present the analogue data for the case of the speedups. The global mesh resolution was fixed at 1 769 472 cells and the number of used processors was increased. Again we observe a perfect scaling behaviour for all parts of the algorithm subsumed by t_R. The dependence of the solver and in particular of the preconditioner on the number of subdomains prevents an optimal parallel performance. However, for a larger number of subdomains this effect diminishes. The number of iterations grows only moderately and therefore the executing time approaches a nearly linear behaviour with respect to the number of employed processors.

The scaleup and speedup experiments show that the solution of the linear system for the pressure is the part of our algorithm which gives room for improvement. In order to achieve

Table IV. Speedup for example 1. Constant mesh with $192 \times 96 \times 96 = 1\,769\,472$ cells. Executing times in seconds.

Processors		1st step	First 10 steps
	Iterations	112	808
1	t_R	94.78	950.29
	t_S	156.01	1129.03
	Iterations	136	928
2	t_R	49.26	493.36
	t_S	120.60	826.01
	Iterations	256	1704
4	t_R	24.96	248.82
	t_S	114.43	762.48
	Iterations	312	2112
8	t_R	12.41	124.33
	t_S	69.46	470.61
	Iterations	248	2160
16	t_R	6.25	62.69
	t_S	26.52	231.39
	Iterations	280	2392
32	t_R	3.19	32.02
	t_S	15.51	132.71
	Iterations	280	2544
64	t_R	1.68	16.44
	t_S	8.16	74.18

optimal parallel performance multigrid methods or specialized block-preconditioners for domain decomposition methods could be employed, see for instance Reference [32].

4.2. Examples for fluid–structure interaction

In this section we present numerical examples of simulations performed with our algorithm for the fully coupled fluid–structure interaction problem. To this end, we consider two test problems. First, we are concerned with the flow through a channel, where the movement of the elastic top wall is enforced by applying a load vector which varies sinusoidal in time. In the second example, the interaction behaviour between the elastic top wall without any external forces and the flow is studied. The common setting for both problems is sketched in Figure 12.

The total length of the channel is $L_x = 3.5$, the channel height is $L_z = 0.75$ and the width is $L_y = 1.0$. The fluid domain is decomposed into three subdomains, compare Figure 12. The top wall of the middle fluid block (the grey shaded part of the top boundary in Figure 12) represents the elastic interface where motion due to interaction effects can occur. All other boundary parts are fixed in time. The thin elastic solid domain has a height of 0.05 and the same width and length as the middle fluid block. At $\xi_1 = 0$ a parabolic flow profile is

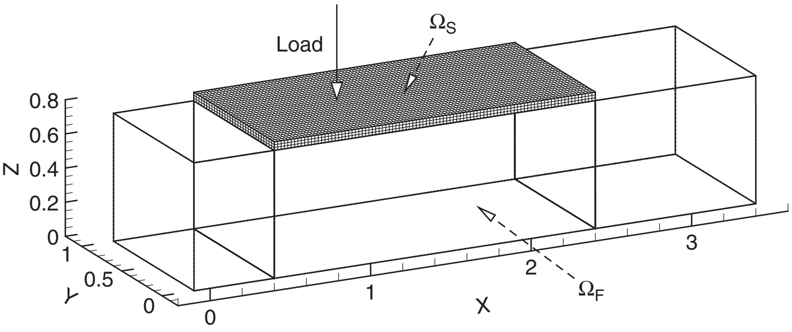

Figure 12. The general setting for the fluid–structure interaction problem.

prescribed, at $\xi_1 = \xi_{\max}$ homogeneous Neumann conditions are used. For the elastic part of the boundary the interface conditions (5) are used. On all other walls no-slip conditions are applied. Since the nodes of the fluid grid and the solid grid coincide at the interface, no additional specialized interpolation procedures for the exchange of fluid and solid quantities are required.

All the results in this section have been obtained with an iteration parameter $\gamma = 0$ in the solution algorithm (cf. Section 2). This results in a simple but efficient approach for the solution of a coupled problem since only one solution of each subproblem per time step is required. The method corresponds to a weakly coupled approach which is known to work well for fluid–structure interaction problems where the density of the solid material is larger than the density of the fluid. This is for example the case in the field of aeroelasticity, see, e.g. References [12] or [13].

4.2.1. Fluid–structure interaction governed by an external force. We now consider the fluid–structure interaction problem, where the movement of the upper boundary is enforced by an external load. The load is given by a force field which is oscillating sinusoidally in time. The maximum load is applied at the point $x_{\text{load}} = (1.0, 0.5, 0.75)$ and is smoothly distributed over a small region of three grid cells around x_{load}. The material parameters for the solid are chosen as follows: the elastic modulus is $E = 1.0e9$, the Poisson ratio is $\nu = 0.29$ and the density is $\rho_S = 1000$. The maximum load value applied was 100. The Reynolds number for the flow based on the mean inflow velocity, the channel height, the fluid density $\rho_F = 1.0$ and the kinematic viscosity of 0.001 is approximately 187.50. The mesh of the fluid domain consists of $(32+96+64)\times 32\times 32 = 196\,608$ control volumes, the elastic solid is discretized with $96\times 32\times 4 = 11\,776$ eight-noded finite elements.

Figure 13 shows the applied load versus time, the resulting displacement $d_{\text{centre}} = x_3(t) - x_3(0)$ measured at the centre of the interface as well as the total volume of the fluid domain $\Omega_F(t)$ (top row). In the bottom row of Figure 13 the rate of change of the fluid volume, the corresponding mass flow at the inflow and outflow boundaries as well as the error of the mass balance equation (1) is shown. The measured data shows the behaviour which is expected from the equation for conservation of mass (1). When the rate of change of the volume is zero, the inflow and outflow mass fluxes are balanced. In time intervals where the volume shrinks or expands, the mass flux at the outflow boundary rises above or falls below

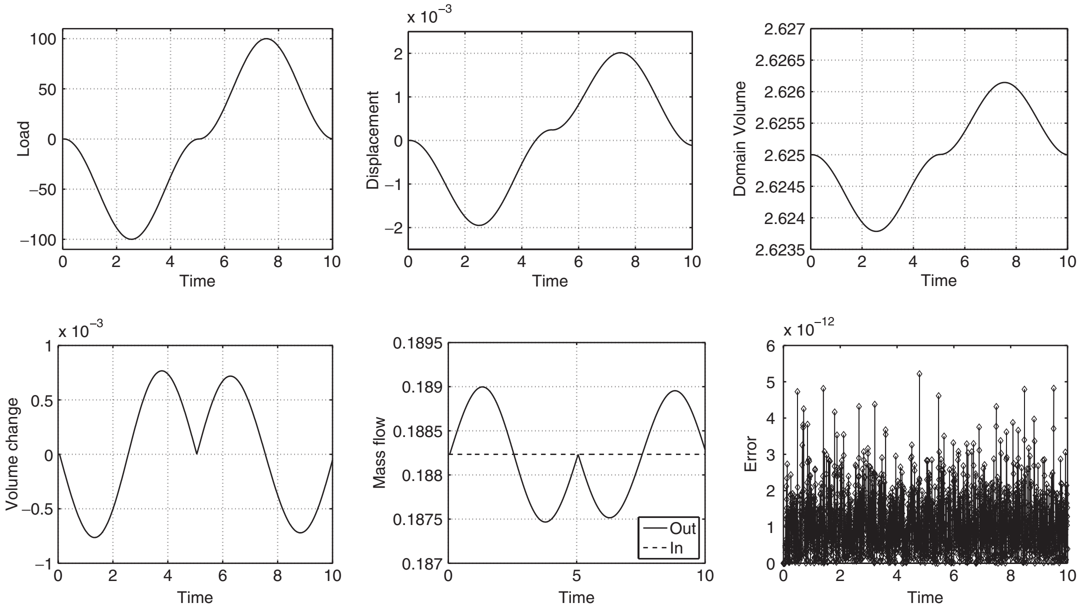

Figure 13. Load, displacement of interface centre and volume of fluid domain $\Omega_F(t)$ (top from left to right), volume change of $\Omega_F(t)$, mass fluxes and mass balance error (bottom from left to right).

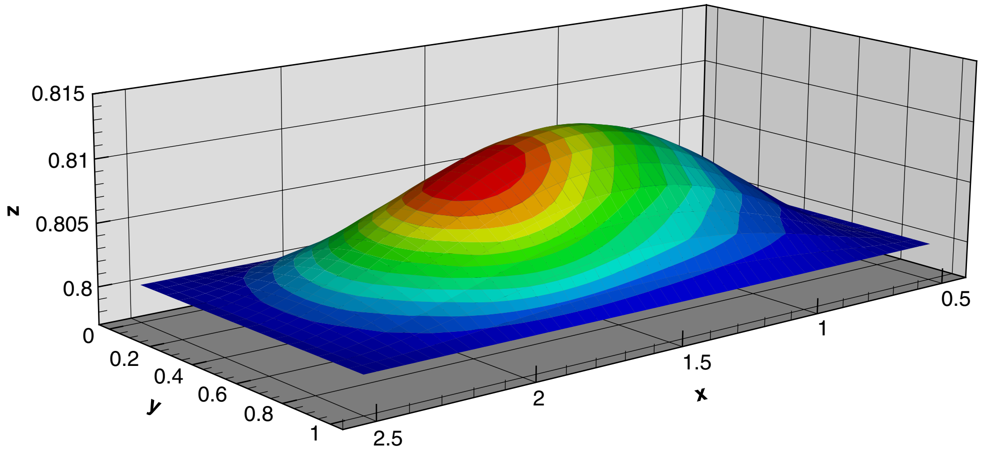

Figure 14. Displacement of the interface, magnified by a factor of 50 (only the middle fluid block is shown).

the constant flux value at the inflow boundary, respectively. Figure 14 shows the displacement of the section of the top wall which forms the elastic interface. Figure 15 shows the vertical velocity component on three slices in the fluid domain.

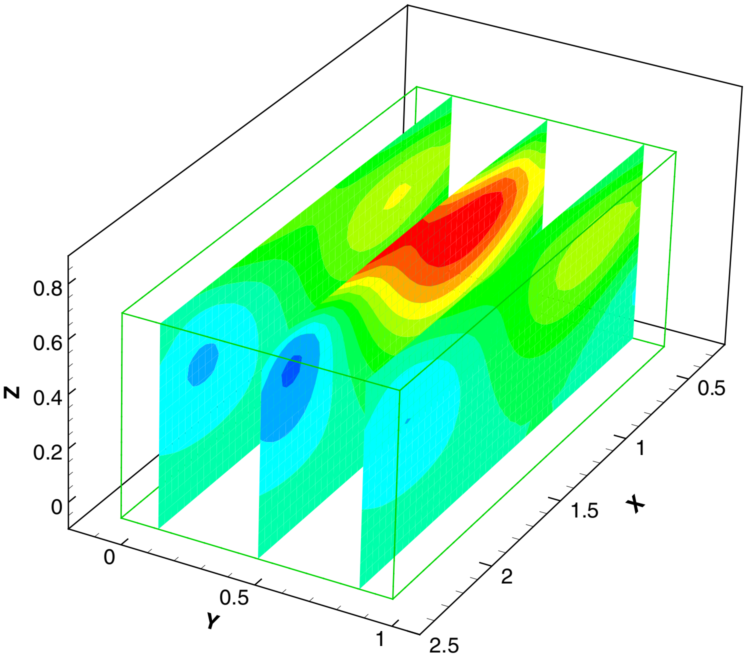

Figure 15. Vertical velocity v_3 in the middle fluid block.

4.2.2. Flow through a channel with an elastic wall. Finally we consider the case where only the forces exerted by the fluid are acting on the elastic structure. The general setting is the same as in the previous section, however this time no external load is applied. Furthermore, the elastic modulus of the structure is set to a value of $E = 1e2$. Thus the forces which the fluid exerts on the interface result in a noticeable displacement.

First we address the issue of convergence with respect to the time step size Δt. To this end we fix the mesh size h. All simulations in this section are performed on a spatial grid with a resolution of $16 \times 16 \times 16$ control volumes for the first fluid block, $32 \times 16 \times 16$ control volumes for the last block, $48 \times 16 \times 16$ control volumes in the middle block and $48 \times 16 \times 2$ control volumes for the solid. The time steps for the different simulations were fixed at five values from $1e-2$ to $6.25e-4$.

Figure 16 shows the volume of the fluid domain $\Omega_F(t)$ over time for the different time step sizes used. It reflects the dynamic response of the system. Since the initial condition is not an equilibrium of the coupled system, the interaction forces between fluid and solid lead to an oscillating behaviour with a period of about 19. The oscillations damp out over time and will eventually lead to an equilibrated state.

Tables V and VI show numerical rates of convergence for simulation runs with different time step sizes on the mesh of fixed spatial resolution given above. Because the exact solution is not known, we compute the numerical rate of convergence $\rho_{\Delta t}$ by the formula

$$\rho_{\Delta t} = \frac{\log|\phi_{\Delta t} - \phi_{\Delta t/2}| - \log|\phi_{\Delta t/2} - \phi_{\Delta t/4}|}{\log 2} \tag{59}$$

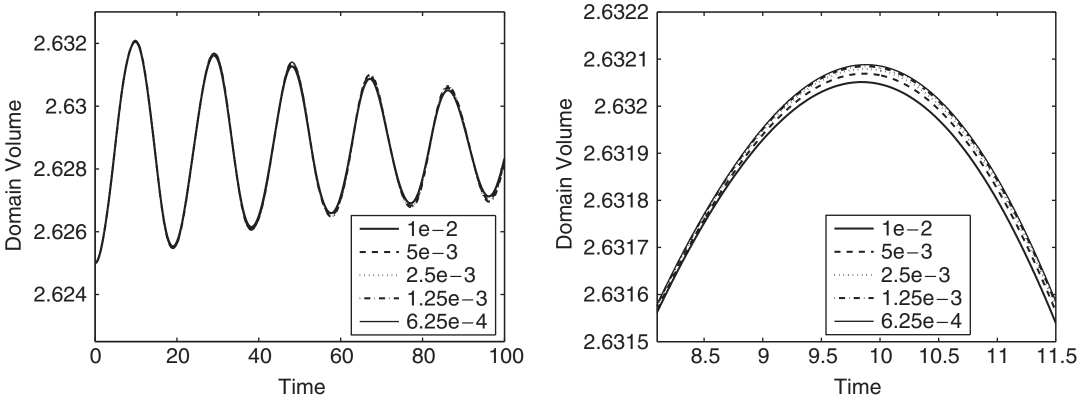

Figure 16. Volume of the fluid domain $\Omega_F(t)$ for simulations with different time step sizes.

Table V. Numerical rate of convergence based on the mass flux at the outflow boundary, computed at time $t = 5.0$, 15.0 and 32.0.

Δt	$m_{O\,\vert t=5.0}$	$\rho_{h\,\vert t=5.0}$	$m_{O\,\vert t=15.0}$	$\rho_{h\,\vert t=15.0}$	$m_{O\,\vert t=32.0}$	$\rho_{h\,\vert t=32.0}$
1.00_{-2}	0.1870821521	—	0.1893511121	—	0.1891208682	—
5.00_{-3}	0.1870775564	—	0.1893638886	—	0.1891419918	—
2.50_{-3}	0.1870750512	0.88180	0.1893697910	1.11410	0.1891544123	0.76613
1.25_{-3}	0.1870737138	0.85997	0.1893724036	1.17583	0.1891601207	1.12155
6.25_{-4}	0.1870730088	0.92601	0.1893735093	1.24061	0.1891627225	1.13359

Table VI. Numerical rate of convergence based on the domain volume, computed at times $t = 9.8$, 29.0 and 48.0.

Δt	$\vert\Omega_F\vert_{t=9.8}$	$\rho_{\Delta t\,\vert t=9.8}$	$\vert\Omega_F\vert_{t=29}$	$\rho_{\Delta t\,\vert t=29}$	$\vert\Omega_F\vert_{t=48}$	$\rho_{\Delta t\,\vert t=48}$
1.00_{-2}	2.632050812	—	2.631610186	—	2.631263961	—
5.00_{-3}	2.632068180	—	2.631652283	—	2.631336935	—
2.50_{-3}	2.632077938	0.83182	2.631671851	1.10520	2.631373272	1.00595
1.25_{-3}	2.632083552	0.79750	2.631680339	1.20501	2.631389684	1.14668
6.25_{-4}	2.632086924	0.73531	2.631685252	0.78877	2.631399299	0.77136

Here, Δt is the size of the time step. The rates in Table V were obtained for the quantity ϕ chosen as the mass flux at the outflow boundary of the fluid domain which is given by

$$m_O = \int_{\Gamma_{\text{out}}} v \cdot vecn \, dS \approx \sum_{j+e_\alpha \in \Gamma_{\text{out}}} V^{\alpha}_{j+e_\alpha} \tag{60}$$

The rates in Table VI are based on the quantity ϕ chosen as the volume of the fluid domain, which is given as the sum of the cell volumes, compare Equation (12). In both cases we observe rates of convergence between 0.73 and 1.2, which roughly reflects a convergence rate of first-order. This is to be expected due to the Euler discretization in time used in both

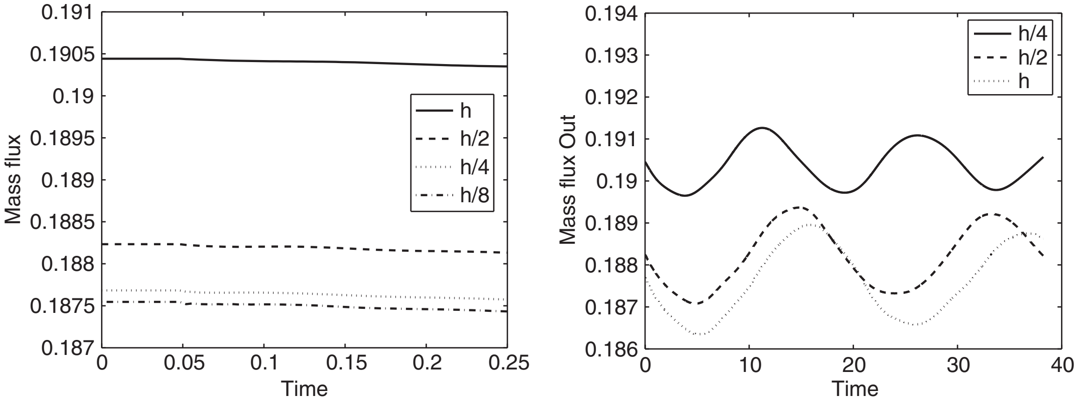

Figure 17. Mass flux at the outflow boundary, up to time $t = 0.25$ (left) and up to $t = 40$ (right).

subproblem solvers. Here the following remark is in order: Because the absolute values of both, the mass flow and the domain volume, are underestimated for computations with a larger time step, the curves intersect in between the peaks due to their oscillating behaviour. In the regions around these intersections convergence can of course not be measured in a meaningful way. To this end we chose time instants near the peaks of the curves for our measurements in Tables V and VI. For the mass flux case we therefore selected the time instants $t = 5$, 15 and 32. The curve for the domain volume is shifted by a half period with respect to the mass flux. This can also be seen from the equation of mass balance (1). Therefore we selected the time instants $t = 9.8$, 29.0 and 48.0.

Now we address convergence with respect to the mesh size h. Here, we have performed simulations with four different grid resolutions, starting from a coarse grid with $8 \times 8 \times 8$ control volumes in the first fluid block, $24 \times 8 \times 8$ control volumes in the middle fluid block, $16 \times 8 \times 8$ in the third block and $24 \times 8 \times 1$ elements for the solid. Each successive finer level is obtained by a uniform refinement with a factor of two. The time step for all four simulations was fixed at a value of $4.0e - 3$, which was sufficiently small to satisfy the CFL condition on the finest mesh.

To measure the numerical convergence rate, we again employed Equation (59), but using h instead of Δt. We selected the mass flux at the outflow boundary and the volume of the fluid domain as the quantity ϕ. Due to the nonlinear coupling of the subproblems the spatial discretization error does not only results in an amplitude change for the computed quantities, but also introduces a phase shift in time. This behaviour is shown for the mass flux on the right of Figure 17 and for the volume of the fluid domain on the left of Figure 18. This shift hampers the estimation of a convergence rate for larger times. Therefore we performed our measurements at a small time ($t = 0.2$), where the shift is not yet too significant. The results for the measurements based on the mass flux and the domain volume are given in Table VII and VIII, respectively.

From Table VII we observe a convergence rate of two. The same order of convergence was already shown in Section 4.1 for the flow solver. Thus, our approach for the solution of the coupled fluid–structure interaction problem seems to maintain the convergence rates of the individual subproblem solvers.

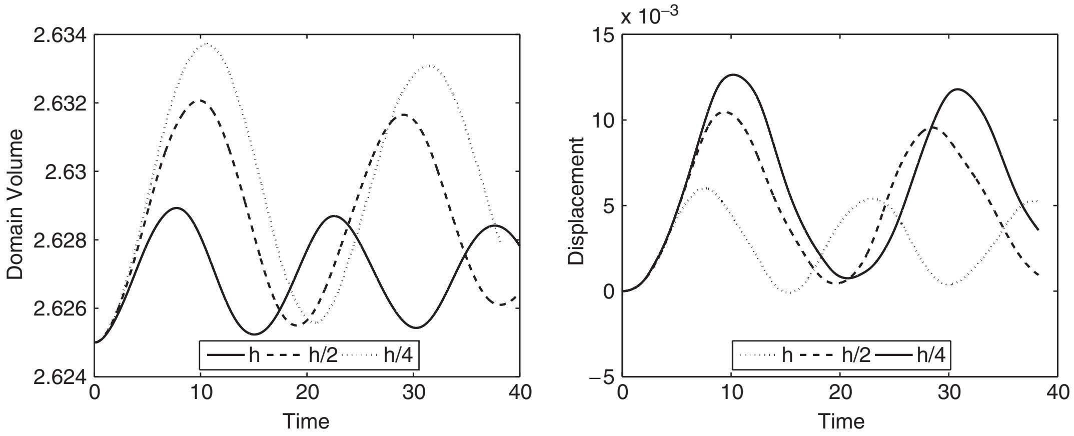

Figure 18. Volume of the fluid domain for simulations with different mesh resolutions (left) and the displacement in the centre of the interface (right).

Table VII. Numerical rate of convergence based on the mass flux at the outflow boundary, computed at $t = 0.2$.

h	m_{O}	$m_{\mathrm{O},h} - m_{\mathrm{O},h/2}$	Ratio	ρ_h
h	0.190371715	—	—	—
$h/2$	0.188151765	2.21994927_{-3}	—	—
$h/4$	0.187598599	5.53165999_{-4}	4.01317	2.00474
$h/8$	0.187459755	1.38843820_{-4}	3.98408	1.99424

Table VIII. Numerical rate of convergence based on the volume of the fluid domain, computed at $t = 0.2$.

h	$\lvert\Omega_{\mathrm{F},h}\rvert$	$\lvert\Omega_{\mathrm{F},h}\rvert - \lvert\Omega_{\mathrm{F},h/2}\rvert$	Ratio	ρ_h
h	2.625005403	—	—	
$h/2$	2.625006571	-1.1679_{-6}	—	
$h/4$	2.625007027	-4.5579_{-7}	2.56230	1.35744
$h/8$	2.625007283	-2.5659_{-7}	1.77630	0.82887

Note however that the theoretical understanding of the convergence rates of the overall method is not completely clear. It is well known that the discrete pressure in Chorin's projection method develops boundary layers with reduced convergence order [33–35]. Since pressure values near the boundary enter the stress tensor and thus the coupling conditions (5) to the solid part, it has to be expected that the convergence order of the overall method is reduced as well. How this affects the convergence rates especially for larger time instants and for finer mesh sizes is not yet fully understood and needs further investigation.

5. CONCLUSIONS AND OUTLOOK

In this paper we presented a numerical approach to the treatment of fluid–structure interaction problems in three space dimensions. The coupling of the elasticity problem to the fluid flow problem was done via an alternating Schwarz algorithm. Here we employed a standard finite element method for the elasticity problem and a novel finite volume discretization on moving boundary-fitted grids. The proposed finite volume scheme implicitly satisfies the geometric conservation law and therefore ensures mass conservation.

The algorithm for the solution of the coupled problem requires only one solution of each subproblem in each time step which results in an explicit and simple, but efficient solution strategy for fluid–structure interaction problems. In our experiments we observed a convergence rate slightly lower than first-order with respect to time and roughly a convergence rate of the order one for the domain volume and of the order two for the mass flux for a fully coupled problem. It is well known that the convergence rate of the pressure deteriorates for a projection scheme in the boundary layer near the fluid–solid interface. How this influences the convergence rate of the overall scheme needs further investigation.

Finally, it should be noted that the stability and convergence properties of explicit and implicit coupling strategies for fluid–structure interaction are not yet fully understood. Explicit coupling strategies so far proved to be a valuable tool in aeroelastic computations where the density of the fluid is low compared to the density of the solid. In Reference [36] it was reported that the stability of explicit coupled algorithms is also influenced by the geometry of the domain and the material properties, in particular the involved densities and their ratios. The study of these dependencies will be further work.

ACKNOWLEDGEMENTS

This work was supported in part by the *Sonderforschungsbereich 611* '*Singular Phenomena and Scaling in Mathematical Models*' of the *Deutsche Forschungsgemeinschaft* (DFG).

REFERENCES

1. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surfaces. *Physics of Fluids* 1965; **8**(12):2182–2189.
2. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
3. Lafaurie B, Nardone C, Scardovelli R, Zaleski S, Zanetti G. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics* 1994; **113**:134–147.
4. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12–49.
5. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**(1):146–159.
6. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics* 1974; **14**:227–253.
7. Donea J, Giuliani S, Halleux JP. An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering* 1982; **33**:689–723.
8. Thomas PD, Lombard CK. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal* 1979; **17**(10):1030–1037.
9. Zhang H, Reggio M, Trépanier J, Camarero R. Discrete form of the GCL for moving meshes and its implementation in CFD schemes. *Computers and Fluids* 1993; **22**(1):9–23.
10. Codina R, Cervera M. On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems. *Engineering Computations* 1996; **13**(6):4–30.

11. Felippa CA, Park KC. Staggered transient analysis procedures for coupled mechanical systems: formulation. *Computer Methods in Applied Mechanics and Engineering* 1980; **24**:61–111.
12. Piperno S, Farhat C, Larrouturou B. Partitioned procedures for the transient solution of coupled aeroelastic problems. *Computational Methods in Applied Mechanical and Engineering* 1995; **124**:79–112.
13. Farhat C, Lesoinne M, Maman N. Mixed explicit/implicit time integration of coupled aeroelastic problems: three-field formulation, geometric conservation and distributed solution. *International Journal for Numerical Methods in Fluids* **21**:807–835.
14. Wesseling P, Segal A, Kassels CGM. Computing flows on general three-dimensional nonsmooth staggered grids. *Journal of Computational Physics* 1999; **149**(2):333–362.
15. Marsden JE, Hughes TJR. *Mathematical Foundations of Elasticity*. Dover Publications Inc.: New York, 1983.
16. Roddeman D. Tochnog—a free explicit/implicit finite element program, 2003. URL: http://tochnog.sourceforge.net (19.08.2003).
17. Zienkiewicz OC, Taylor OC. *The Finite Element Method* (4th edn). Mc Graw-Hill: London, New York, 1994.
18. Braess D. *Finite Elements*: *Theory*, *Fast Solvers and Applications in Solid Mechanics*. Cambridge University Press: Cambridge, MA, 2001.
19. Hughes TJR. *The Finite Element Method*. Prentice-Hall: Englewood Cliffs, NJ, 1987.
20. Liseikin VD. *Grid Generation Methods*. Scientific Computation. Springer: Berlin, Heidelberg, 1999.
21. van Beek P, van Nooyen RRP, Wesseling P. Accurate discretization of gradients on non-uniform curvilinear staggered grids. *Journal of Computational Physics* 1995; **117**:364–367.
22. Demirdžić I, Perić M. Space conservation law in finite volume calculations of fluid flow. *International Journal for Numerical Methods in Fluids* 1988; **8**:1037–1050.
23. Chorin AJ. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation* 1968; **22**: 745–762.
24. Peyret R, Taylor TD. *Computational Methods for Fluid Flow*. Series in Computational Physics. Springer: Berlin, Heidelberg, 1983.
25. Webster R. An algebraic multigrid solver for Navier–Stokes problems. *International Journal for Numerical Methods in Fluids* 1994; **18**:761–780.
26. Griebel M, Neunhoeffer T, Regler H. Algebraic multigrid methods for the solution of the Navier–Stokes equations in complicated geometries. *International Journal for Numerical Methods in Fluids* 1998; **26**:281–301.
27. Pedley TJ, Stephanoff KD. Flow along a channel with a time-dependent indentation in one wall: the generation of vorticity waves. *Journal of Fluid Mechanics* 1985; **160**:337–367.
28. Ralph ME, Pedley TJ. Flow in a channel with a moving indentation. *Journal of Fluid Mechanics* 1988; **190**: 87–112.
29. Demirdžić I, Perić M. Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries. *International Journal for Numerical Methods in Fluids* 1990; **10**:771–790.
30. Sleijpen G, Fokkema D. BiCGStab(L) for linear equations involving unsymmetric matrices with complex spectrum. *Electronic Transaction on Numerical Analysis* 1993; **1**:11–32.
31. Koster F. Preconditioners for sparse grid discretizations. *Technical Report No. 746*, SFB 256, University of Bonn, Germany, 2001.
32. Smith B, Bjrstad P, Gropp W. *Domain Decomposition*: *Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press: Cambridge, MA, 1996.
33. Shen J. On error estimates of projection methods for Navier–Stokes equations: first order schemes. *SIAM Journal on Numerical Analysis* 1992; **29**(1):57–77.
34. E W, Liu JG. Projection method I: convergence and numerical boundary layers. *SIAM Journal on Numerical Analysis* 1995; **32**(4):1017–1057.
35. Rannacher R. On Chorin's projection method for the incompressible Navier–Stokes equations. In *The Navier–Stokes Equations II—Theory and Numerical Methods*, Heywood JG, Masuda K, Rautmann R (eds), Lecture Notes in Mathematics, vol. 1530. Springer: Berlin, 1992.
36. Causin P, Gerbeau JF, Nobile F. Added-mass effect in the design of partitioned algorithms for fluid–structure interaction problems. *Rapport de recherche 5084*, INRIA, Le Chesnay Cedex, 2004.