Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2011)
A. Bargteil and M. van de Panne (Editors)

Mathematical Foundation of the Optimization-based Fluid Animation Method

Kenny Erleben[1], Marek Krzysztof Misztal[2], and J. Andreas Bærentzen[2]

[1]Department of Computer Science, University of Copenhagen, Denmark
[2]DTU Informatics, Technical University of Denmark, Denmark.

Abstract

We present the mathematical foundation of a fluid animation method for unstructured meshes. Key contributions not previously treated are the extension to include diffusion forces and higher order terms of non-linear force approximations. In our discretization we apply a fractional step method to be able to handle advection in a numerically simple Lagrangian approach. Following this a finite element method is used for the remaining terms of the fractional step method. The key to deriving a discretization for the diffusion forces lies in restating the momentum equations in terms of a Newtonian stress tensor. Rather than applying a straightforward temporal finite difference method followed by a projection method to enforce incompressibility as done in the stable fluids method, the last step of the fractional step method is rewritten as an optimization problem to make it easy to incorporate non-linear force terms such as surface tension.

Keywords: Computational Fluid Dynamics, Unstructured Meshes, Finite Element Method, Optimization-based Fluid Animation, Diffusion Forces, Deformable Simplicial Complexes.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Physically-based modeling—Computer Graphics [I.3.7]: Animation—Mathematics of Computing [G.1.6]: Non-linear programming—

1. Introduction

The simulation of liquid surfaces is highly dependent on accurate treatment of surface tension forces and thus on the ability to track the free surface as it develops. This paper presents the foundation of a fluid animation method that can deal with such surface phenomena. This work applies to fluid dynamics in a finite element method type simulation where the liquid and vacuum (other phase) is represented as a simplicial complex where each simplex contains either one phase or the other but not both. Thus, the free surface is given as the subcomplex of faces separating the two phases.

The deformable simplicial complexes (DSC) method is an interface tracking method where the domain is represented as an unstructured simplicial complex (triangles in 2D, tetrahedra in 3D). Each simplex must be completely on one side of the interface. Thus, the interface itself is precisely the faces (line segments in 2D, triangles in 3D) separating interior from exterior. Since DSC uses an unstructured grid as its underlying representation, we can use it in an optimization-based fluid animation method for unstructured meshes [MBE*10] something which originates from ideas of the variational fluid method [BBB07, BB08]. The difference is that our work takes a rigorous finite element method approach and makes a connection to an optimization problem through first order optimality conditions. A 2D fluid animation example is shown in Figure 1. The strength of the optimization-based reformulation is the wealth of optimization methods that can be applied and the performance–quality tradeoff one obtains through direct control of the stopping criteria. For instance in all our examples volume loss is in the order of 0.01%.

In a simulation loop one specifies the displacement field as input to the DSC method which then will update the mesh connectivity and vertex positions. Upon completion the fluid solver can query the new mesh state of the simulation. The DSC method finds the new state using an iterative relaxation scheme that

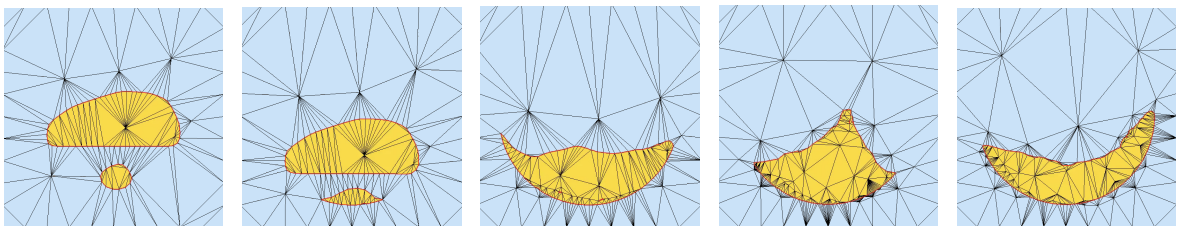


Figure 1: *A 2D optimization-based fluid animation using deformable simplicial complexes. Observe that ambient space (blue) and fluid domain (yellow) are both tessellated and the fluid surface are accurately tracked. Notice how the tessellation changes as simulation progress and that volume loss are kept at a minimum.*

continues to iterative over all mesh elements one-by-one and make local adjustments to improve both tessellation as well as numerical conditioning. We refer to [KBAE09, Mis10, MBE*10] for all meshing details.

One of the major benefits of the optimization-based fluid animation method is the direct way in which surface tension forces can be accounted for. In this paper, we derive the diffusion term for the fluid animation method. The main result is given by the modified optimization problem that minimize the objective function

$$\frac{1}{2}\tilde{\mathbf{u}}^T\left(\mathbf{A}\tilde{\mathbf{u}}+2\mathbf{b}\right) \tag{1}$$

subject to constraints that enforce incompressibility. Here $\mathbf{u}$ is the unknown velocity field of the fluid that we wish to find. As we will show the $\mathbf{b}$ vector models the result of the advection whereas $\mathbf{A}$ contains inertia and diffusion force terms. In particular we will show that $\mathbf{A}$ is a symmetric positive definite block matrix that can be written as $\mathbf{A}=\mathbf{M}+\Delta t\mathbf{D}$ where $\mathbf{M}$ is a mass matrix and $\mathbf{D}$ is a diffusion matrix and Δt is the time step. The term $\Delta t\mathbf{D}$ accounts for the major contribution of this work.

1.1. Our Contributions

The focus in this paper lies on the mathematical formulation for the finite element method that forms the foundation for the fluid animation method. In this context, we analyze the problem of solving the equations of fluid dynamics and derive the formulas for a numerical scheme. The present work extends that of [MBE*10]. In particular, we make the following contributions:

- Include diffusion (ie. viscosity) in the scheme allowing for a wider range of types of fluid.
- Derive the finite element method equations from the continuous fluid flow equations.
- Analyze the optimization process which arises in the second part of the time step.

In Section 2 we cover previous work on fluid animation with focus on unstructured meshes. Following this we present the mathematical model for incompressible flows in Section 3. We present full details of the discretization process in Section 4. Finally, we show 3D examples in Section 5 before summarizing the key mathematical points of our derivations in Section 6.

2. Previous Work about Fluid Animation on Unstructured Meshes

Many works are based on regular grids and we refer to [FM96, FM97, Sta99, FF01, FSJ01, BMF07, Bri08, BBB10] for details. In comparison with most previous work our work focusses on unstructured meshes. The work in computer graphics on fluid solvers based on unstructured meshes is sparse. Early work used static unstructured meshes [FOK05]. Dynamic meshes where the deformation is limited to preserve mesh quality where introduced in [FOKG05]. Local topological operations have been explored in [MBE*10, WRK*10]. Remeshing using visual clues to generate a high resolution mesh in visual important regions have also been explored [KFCO06] and variational meshing [BWHT07]. Other re-meshing approaches are based on extraction of the free surface [CFL*07] or subdivision of elongated elements [WT08].

Solid boundaries and two-way coupling have been touched upon [FOK05, KFCO06]. The preferred method for dealing with advection has been the semi-Lagrangian advection method [FOK05] and its generalization to deforming meshes [FOKG05] which have been applied in many works [KFCO06, CGFO06, CFL*07, WBOL07].

The finite volume method is a popular choice for unstructured meshes [FOK05, FOKG05, KFCO06, ETK*07, WBOL07]. The finite element method have been used in [BWHT07, WT08, WRK*10] for plastic and elastic objects. However, its application for fluids is sparse [MBE*10].

Many schemes are based on staggered grid layouts [FOK05,FOKG05,ETK*07,WBOL07]. Here the face centers often store the normal velocities and volume centers store pressure values. These schemes often suffer from the problem of having to reconstruct the full velocity field to deal with advection and diffusion.

In summary past work is based on staggered meshes using a face centered velocity grid layout. Most work on unstructured meshes deal with free surfaces using contouring and complete re-meshing. Deforming meshes has been considered to control visual quality but in a deformation rate limited manner. We use a moving and deforming tetrahedral mesh and we store the full velocity vector at the vertices. Our approach follows the physical simulation and has no limitations. Further, our work uses a finite element method for fluid simulation whereas previous work on fluid simulation on unstructured meshes mostly use finite volume methods. Our work currently does not address two-way coupling.

3. The Mathematical Model of Incompressible Fluids

We have a volumetric domain $V \subseteq \mathbb{R}^3$ containing a fluid of volume $V_{\text{fluid}} \subseteq V$, solids $V_{\text{solid}} \subseteq V$ and the remaining non-fluid and non-solid part $V_{\text{air}} \subseteq V$. Here we assume V_{air} to be vacuum as we do not have two phase flows. The surface boundary of the fluid are given by $\partial V_{\text{fluid}}$ and can be divided into two disjoint parts, one being the contact at solid walls $\partial V_{\text{solid}} = V_{\text{fluid}} \cap V_{\text{solid}}$ and the other being the free surface between fluid and air $\partial V_{\text{free}} = V_{\text{fluid}} \cap V_{\text{air}}$.

The motion of a Newtonian fluid is given by the Navier–Stokes equation,

$$\rho \dot{\mathbf{u}} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - \nabla p + \mu\nabla^2\mathbf{u} + \mathbf{f}, \qquad \mathbf{x} \in V_{\text{fluid}}. \tag{2}$$

where ρ is the mass density, $\mathbf{u}$ is the unknown velocity field, p is the pressure field, μ is the dynamic viscosity coefficient, $\mathbf{x}$ is the spatial position and $\mathbf{f}$ is a force term including external forces like gravity and surface tension etc.. Assuming a constant mass density then mass conservation given by the continuity equation models incompressibility of the fluid,

$$\nabla\cdot\mathbf{u} = 0, \qquad \mathbf{x} \in V_{\text{fluid}}. \tag{3}$$

If we did not assume constant mass density then we would need a constitutive law relating pressure field and mass density field. The ideal gas law is such an example.

Further, we specify the boundary conditions for the fluid model. The surface tension forces act to minimize the fluid surface and may be included in our model through the force term $\mathbf{f}$. Thus, one may define the surface potential energy of the whole fluid surface as

$$U = \gamma A, \tag{4}$$

where A is the area of the fluid surface. The surface tension force is then given as,

$$-\nabla U(\mathbf{x}) = -\gamma\nabla A(\mathbf{x}), \qquad \mathbf{x} \in \partial V_{\text{fluid}}, \tag{5}$$

At solid walls we have,

$$\mathbf{u}\cdot\mathbf{n} = 0, \qquad \text{for} \qquad \mathbf{x} \in \partial V_{\text{solid}}, \tag{6a}$$
$$\Delta p = 0, \qquad \text{for} \qquad \mathbf{x} \in \partial V_{\text{solid}}, \tag{6b}$$

These equations state that the fluid and solid velocities must be the same in the normal direction and that the difference in the pressure field Δp is zero across the solid boundary. At the free surface of the fluid we have the boundary conditions

$$p = 0, \qquad \text{for} \qquad \mathbf{x} \in V_{\text{air}}, \tag{7a}$$
$$\Delta u = 0, \qquad \text{for} \qquad \mathbf{x} \in \partial V_{\text{free}}. \tag{7b}$$

The first equation states that pressure in vacuum is zero. The second equation is a slightly pseudo physical condition. Because we only consider single phase flow then the air region of our simulation is vacuum. Thus, no matter is present in the air region and it can be argued what value the velocity of empty space has.

We have now stated our idealized model of the fluid motion problem and may move on to show how we discretize our model. This process includes four ingredients, fractional step method, semi-Lagrangian implicit time integration, finite element method, and constrained optimization.

4. The Optimization-based Fluid Animation Method

We will now discretize the mathematical model into a numerical scheme. The finite element method is developed in Section 4.1, time discretization is done in Section 4.2 and finally the optimization-based reformulation is detailed in Section 4.3.

4.1. Finite Element Discretization

We are given a tetrahedral mesh with N vertices and K elements. We use a staggered grid layout where fluid velocities $\hat{\mathbf{u}}_i \in \mathbb{R}^3$ are stored at the vertices $\mathbf{x}_i \in \mathbb{R}^3$ and pressure values $p_k \in \mathbb{R}$ are stored at the tetrahedra centers. Using the shape functions $\phi_i(\mathbf{x}) : \mathbb{R}^3 \mapsto \mathbb{R}$ we may write an approximation $\hat{\mathbf{u}}$ to the true velocity $\mathbf{u}(\mathbf{x})$ at any given point $\mathbf{x}$ as

$$\mathbf{u}(\mathbf{x}) \approx \hat{\mathbf{u}}(\mathbf{x}) \equiv \sum_i \phi_i(\mathbf{x})\hat{\mathbf{u}}_i. \tag{8}$$

Here the hat-notation is used to distinguish between true velocities and the discrete velocities. We will omit

a method for interpolating the pressure field values because as we show later the pressure field ends up acting as Lagrange multipliers that enforce the discrete incompressibility constraint so we do not need any interpolating functionality for the pressure field.

The advection term causes the non-linear behavior of the model. However, it is easy to solve using a Lagrangian representation. Therefore, the time derivatives of the equation of motion are solved using a fractional step method [FP02], where the equation of motion is split into a first step where advection is dealt with and a second step where the remaining force terms are handled. The first step of the method is handled by using a generalized semi-Lagrangian implicit time integration method [Sta99, KFCO06]. The idea is to think of the vertices of the mesh as particles and trace their motion back in time. The past velocity values are then copied to the current location to account for the advection. That is given the initial discrete velocity field $\hat{\mathbf{u}}_i^t$ then the advection of the velocity field is given by

$$\hat{\mathbf{u}}_i^{t+\frac{1}{2}} = \begin{cases} \hat{\mathbf{u}}^t(\mathbf{x}_i^{t-\Delta t}) & ;\mathbf{x}_i^{t-\Delta t} \in V_{\text{fluid}} \\ \hat{\mathbf{u}}^t(P(\mathbf{x}_i^{t-\Delta t})) & ;\mathbf{x}_i^{t-\Delta t} \notin V_{\text{fluid}} \end{cases}, \tag{9}$$

where

$$\mathbf{x}_i^{t-\Delta t} = \mathbf{x}_i^t - \Delta t \hat{\mathbf{u}}_i^t \tag{10}$$

and $P(\mathbf{x}_i^{t-\Delta t}))$ is the projection of the point $\mathbf{x}_i^{t-\Delta t}$ onto the closest point on the fluid volume V_{fluid}. As an alternative one could use an explicit Lagrangian approach moving vertices forward in time according to the current velocity while the vertices keep their velocities. This has the advantage that the DSC method can handle the advection while tracking the fluid surface. One specifies the current velocity field multiplied by the time step as the displacement field for the DSC method. This is termed a co-moving mesh in computational fluid dynamics [FP02] and has the benefit that the advection term can be dropped. The explicit approach restricts the time step size as a rule of thumb we have observed that time step should be chosen such that the maximum displacement for DSC is proportional to the average edge length in the initial mesh. The implicit approach allows for large time step sizes but suffers from more numerical dissipation.

Finally the second step of the fractional step method is reformulated using a finite element method discretization [BW00, ZT00]. We will start by restating the momentum equation of the Navier–Stokes equations in an alternative equivalent form,

$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \mathbf{T} + \mathbf{f}, \tag{11}$$

where $\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}$ is the full derivative and $\mathbf{T}$ is the Newtonian fluid stress tensor given as,

$$\mathbf{T} = -p\mathbf{I}_{3\times 3} + \mu\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right), \tag{12}$$

where $\mathbf{I}_{3\times 3}$ is the 3×3 identity matrix. Observe this is by definition a symmetric stress tensor. Since the first step was dealt with using the semi-Lagrangian implicit time integration what remains to be solved in the second step is

$$\rho \frac{\partial \mathbf{u}}{\partial t} = \nabla \cdot \mathbf{T} + \mathbf{f}. \tag{13}$$

Next we multiply the above partial differential equation by an admissible test function $\mathbf{w}$([†]) and take the volume integral over the fluid volume V_{fluid},

$$\int_{V_{\text{fluid}}} \mathbf{w}^T \rho \frac{\partial \mathbf{u}}{\partial t} dV - \int_{V_{\text{fluid}}} \mathbf{w}^T (\nabla \cdot \mathbf{T}) dV - \int_{V_{\text{fluid}}} \mathbf{w}^T \mathbf{f} dV = 0. \tag{14}$$

We will apply a Galerkin method which means that the test function is written in terms of the shape functions we defined in (8),

$$\mathbf{w} = \sum_i \phi_i(\mathbf{x}) \hat{\mathbf{w}}_i. \tag{15}$$

In the first term of the strong form formulation (14) we substitute the interpolation formular given by the shape functions in (8) and the test function in (15),

$$P_\rho = \int_{V_{\text{fluid}}} \mathbf{w}^T \rho \frac{\partial \mathbf{u}}{\partial t} dV, \tag{16a}$$

$$= \sum_j \sum_i \hat{\mathbf{w}}_j^T \left(\int_{V_{\text{fluid}}} \rho \phi_j \phi_i dV \right) \frac{\partial \hat{\mathbf{u}}_i}{\partial t}. \tag{16b}$$

Similar the third term gives

$$P_{\mathbf{f}} = \int_{V_{\text{fluid}}} \mathbf{w}^T \mathbf{f} dV, \tag{17a}$$

$$= \sum_j \sum_i \hat{\mathbf{w}}_j^T \left(\int_{V_{\text{fluid}}} \phi_j \phi_i dV \right) \hat{\mathbf{f}}_i. \tag{17b}$$

This term gives the body forces which is usually gravity. The stress tensor term is rewritten into a weak form using the symmetry of the stress tensor and the identity $\nabla \cdot (\mathbf{T}\mathbf{w}) = (\nabla \cdot \mathbf{T}) \cdot \mathbf{w} + \mathbf{T} : \nabla \mathbf{w}$ ([‡]),

$$\int_{V_{\text{fluid}}} \mathbf{w}^T (\nabla \cdot \mathbf{T}) dV = \int_{V_{\text{fluid}}} \nabla \cdot (\mathbf{T}\mathbf{w}) dV - \int_{V_{\text{fluid}}} \mathbf{T} : \nabla \mathbf{w} dV, \tag{18}$$

and using Gauss divergence theorem we rewrite the

[†] In our case $\mathbf{w}$ can have any arbitrary value as long as it is zero on the boundary and continuous differentiable on the domain

[‡] The double contraction between two second order tensors $\mathbf{A}$ and $\mathbf{B}$ is defined as $\mathbf{A} : \mathbf{B} = \sum_j \sum_i \mathbf{A}_{ij} \mathbf{B}_{ij}$

first term on the right hand side into a surface integral,

$$\int_{V_{\text{fluid}}} \mathbf{w}^T (\nabla \cdot \mathbf{T})\, dV = \int_{\partial V_{\text{fluid}}} \mathbf{w}^T \mathbf{T}\mathbf{n} dS - \int_{V_{\text{fluid}}} \mathbf{T} : \nabla \mathbf{w} dV. \tag{19}$$

The boundary integral term $\int_{\partial V_{\text{fluid}}} \mathbf{T}\mathbf{n} dS$ are the integral of prescribed surface traction and are given by appropriate boundary conditions. In our model no such surface traction is used and the term vanishes. Inserting the Galerkin type test function yields

$$P_{\mathbf{T}} = \int_V \mathbf{T} : \left(\sum_j \nabla \left(\phi_j \hat{\mathbf{w}}_j \right) \right) dV. \tag{20a}$$

Next we use $\nabla \left(\phi_j \hat{\mathbf{w}}_j \right) = \hat{\mathbf{w}}_j \otimes \nabla \phi_j$ ([§]) and that $\mathbf{T} : \hat{\mathbf{w}}_j \otimes \nabla \phi_j = \hat{\mathbf{w}}_j^T \mathbf{T} \nabla \phi_j$ so

$$P_{\mathbf{T}} = \sum_j \left(\hat{\mathbf{w}}_j^T \int_V \mathbf{T} \nabla \phi_j dV \right) \tag{21}$$

which we split into normal and shear stress terms, $P_{\mathbf{T}} = P_{\mathbf{D}} - P_p$,

$$P_p = \sum_j \left(\hat{\mathbf{w}}_j^T \int_V -p \mathbf{I}_{3\times 3} \nabla \phi_j dV \right), \tag{22a}$$

$$P_{\mathbf{D}} = \sum_j \sum_i \left(\hat{\mathbf{w}}_j^T \int_{V_{\text{fluid}}} \mu (\hat{\mathbf{u}}_i \otimes \nabla \phi_i + \nabla \phi_i \otimes \hat{\mathbf{u}}_i) \nabla \phi_j dV \right). \tag{22b}$$

A last rewrite gives

$$(\hat{\mathbf{u}}_i \otimes \nabla \phi_i + \nabla \phi_i \otimes \hat{\mathbf{u}}_i) \nabla \phi_j = \left(\nabla \phi_i^T \nabla \phi_j \mathbf{I}_{3\times 3} + \nabla \phi_i \nabla \phi_j^T \right) \hat{\mathbf{u}}_i, \tag{23}$$

so the shear stress term becomes

$$P_{\mathbf{D}} = \sum_j \sum_i \left(\hat{\mathbf{w}}_j^T \int_{V_{\text{fluid}}} \mu \left(\nabla \phi_i^T \nabla \phi_j \mathbf{I}_{3\times 3} + \nabla \phi_i \nabla \phi_j^T \right) \hat{\mathbf{u}}_i dV \right). \tag{24}$$

Our weak form formulation now reads

$$P_{\rho} - P_{\mathbf{f}} - P_p + P_{\mathbf{D}} = 0, \tag{25}$$

and it must hold for all values of $\hat{\mathbf{w}}_j$ which means we end up with the ordinary differential equation,

$$\mathbf{M} \frac{\partial \tilde{\mathbf{u}}}{\partial t} - \mathbf{B}\tilde{\mathbf{f}} - \mathbf{P}\tilde{\mathbf{p}} + \mathbf{D}\tilde{\mathbf{u}} = \mathbf{0}, \tag{26}$$

where

$$\tilde{\mathbf{u}} = \begin{bmatrix} \hat{\mathbf{u}}_1^T & \cdots & \hat{\mathbf{u}}_N^T \end{bmatrix}^T, \tag{27a}$$

$$\tilde{\mathbf{f}} = \begin{bmatrix} \hat{\mathbf{f}}_1^T & \cdots & \hat{\mathbf{f}}_N^T \end{bmatrix}^T, \tag{27b}$$

$$\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_1^T & \cdots & \mathbf{x}_N^T \end{bmatrix}^T, \tag{27c}$$

$$\tilde{\mathbf{p}} = \begin{bmatrix} p_1 & \cdots & p_K \end{bmatrix}^T, \tag{27d}$$

and

$$\mathbf{M}_{ij} = \mathbf{I}_{3\times 3} \int_{V_{\text{fluid}}} \rho \phi_j \phi_i dV, \tag{28a}$$

$$\mathbf{B}_{ij} = \mathbf{I}_{3\times 3} \int_{V_{\text{fluid}}} \phi_j \phi_i dV, \tag{28b}$$

$$\mathbf{D}_{ij} = \int_{V_{\text{fluid}}} \mu \left(\nabla \phi_i^T \nabla \phi_j \mathbf{I}_{3\times 3} + \nabla \phi_i \nabla \phi_j^T \right) dV. \tag{28c}$$

These equations reveal the block and symmetric properties of the matrices. One may apply a lumped matrix approach in which case $\mathbf{M}$ and $\mathbf{B}$ simplify to diagonal block matrices. Observe, that the shape functions are linear polynomials when using linear shape functions (ie. barycentric coordinates). In this particular case simple closed form solutions exist for the blocks $\mathbf{M}_{ij}$ and $\mathbf{B}_{ij}$. Further, in this case the block $\mathbf{D}_{ij}$ becomes very simple as the spatial gradients of the shape functions are constant. The number of nonzeros in $\mathbf{A}$ depends on the mesh connectivity. A block row corresponds to one node in the mesh and the number of non-zero column blocks is equal to the number of edges incident to that node. In general no bound on the number of neighbors of a node can be given. However, in practice the maximum number of neighbors is observed to be bounded by a constant c. Therefore the number of nonzero blocks in any row is less than or equal c and the number of nonzeros scale as $\mathcal{O}(cN)$. In practice $c \ll N$ so matrix vector products involving $\mathbf{A}$ can be done in $\mathcal{O}(N)$.

The pressure term derivation is not quite complete yet. However, when using linear shape functions then the deformation gradient is constant over a tetrahedral element. This means the normal stress tensor is constant per tetrahedral element. From this it follows that the pressure values are constant over the tetrahedral elements and we have

$$\mathbf{P}_{jk} = \int_{V^k} \nabla \phi_j dV = V^k \nabla \phi_j, \tag{29}$$

where V^k is the volume of the k^{th} tetrahedron. The last equality follows only if linear shape functions are used. The final step in the finite element discretization is to process the continuity equation that is

$$0 = \int_{V_{\text{fluid}}} (\nabla \cdot \mathbf{u})\, dV. \tag{30}$$

[§] The tensor product $\otimes$ is defined as $(\mathbf{a} \otimes \mathbf{b})\mathbf{x} = (\mathbf{b} \cdot \mathbf{x})\mathbf{a}$ given the three vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{x}$.

Substitution of the shape function interpolation yields

$$0 = \sum_i \int_{V_{\text{fluid}}} \nabla \phi_i^T \hat{\mathbf{u}}_i dV, \tag{31a}$$

$$= \sum_i \sum_k \left(\int_{V^k} \nabla \phi_i^T dV \right) \hat{\mathbf{u}}_i, \tag{31b}$$

$$= \mathbf{P}^T \tilde{\mathbf{u}}. \tag{31c}$$

What remains to be considered is the final time discretization of the resulting ordinary differential equations. We will embark on this in the following section.

4.2. Time Discretization

Using the finite element method we have derived the ordinary differential equations,

$$\mathbf{M} \frac{\partial \tilde{\mathbf{u}}}{\partial t} - \mathbf{P}\tilde{\mathbf{p}} + \mathbf{D}\tilde{\mathbf{u}} = \mathbf{F}, \tag{32}$$

$$\mathbf{P}^T \tilde{\mathbf{u}} = \mathbf{0}, \tag{33}$$

where we introduced the notation $\mathbf{F} = \mathbf{B}\hat{\mathbf{f}}$. Using the result $\tilde{\mathbf{u}}^{t+\frac{1}{2}}$ from the first fractional step as initial value we have an initial value problem for our second step. We may now apply finite differences to obtain the first order approximation,

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} \approx \frac{\tilde{\mathbf{u}}^{t+\Delta t} - \tilde{\mathbf{u}}^{t+\frac{1}{2}}}{\Delta t}. \tag{34}$$

This is an advantage from a computational cost view point. Choosing an implicit scheme for stability yields the equations,

$$(\mathbf{M} + \Delta t \mathbf{D})\, \tilde{\mathbf{u}}^{t+\Delta t} - \mathbf{M}\tilde{\mathbf{u}}^{t+\frac{1}{2}} - \Delta t \mathbf{F} - \Delta t \mathbf{P}\tilde{\mathbf{p}} = \mathbf{0}, \tag{35a}$$

$$\mathbf{P}^T \tilde{\mathbf{u}}^{t+\Delta t} = \mathbf{0}. \tag{35b}$$

Defining $\mathbf{A} = (\mathbf{M} + \Delta t \mathbf{D})$, $\mathbf{b} = -\mathbf{M}\tilde{\mathbf{u}}^{t+\frac{1}{2}} - \Delta t \mathbf{F}$, and $\tilde{\mathbf{p}}' = \Delta t \tilde{\mathbf{p}}$ we have

$$\mathbf{A}\tilde{\mathbf{u}}^{t+\Delta t} + \mathbf{b} - \mathbf{P}\tilde{\mathbf{p}}' = \mathbf{0}, \tag{36a}$$

$$\mathbf{P}^T \tilde{\mathbf{u}}^{t+\Delta t} = \mathbf{0}. \tag{36b}$$

These equations conclude what could be considered the "classical" finite element method for the fluid scheme we have presented. If no external force terms or only simple constant force terms are needed in the simulation then one does not need to develop the numerical method further. However, in some cases one may wish to add complex non-linear force types to the model. Surface tension is such an example. In the next section we will briefly discuss how all such force types can be dealt with by recasting the above equations as the first order optimality conditions of an optimization problem.

4.3. The Optimization-based Reformulation

Consider the optimization problem

$$\tilde{\mathbf{u}}^* = \arg\min_{\tilde{\mathbf{u}}} \frac{1}{2} \tilde{\mathbf{u}}^T (\mathbf{A}\tilde{\mathbf{u}} + 2\mathbf{b}) \tag{37}$$

subject to

$$\mathbf{P}^T \tilde{\mathbf{u}} = \mathbf{0}. \tag{38}$$

The first order optimality conditions (the Karush–Kuhn–Tucker (KKT) conditions) of this convex constrained quadratic minimization problem result in the KKT system [NW99]:

$$\mathbf{A}\tilde{\mathbf{u}}^* + \mathbf{b} - \mathbf{P}\lambda^* = \mathbf{0}, \tag{39a}$$

$$\mathbf{P}^T \tilde{\mathbf{u}}^* = \mathbf{0}. \tag{39b}$$

By comparison with our finite difference approximation equations we observe that $\tilde{\mathbf{u}}^* = \tilde{\mathbf{u}}^{t+\Delta t}$ and that $\lambda^* = \Delta t \tilde{\mathbf{p}}$.

The optimization problem as it stands give us some insight into whether there are any solutions for our discrete fluid simulation problem. From its definition one observe that $\mathbf{A}$ is a block symmetric positive definite matrix. Thus, we have a strict convex quadratic programming problem subject to linear constraints. In essence that means constraint qualifications are fulfilled and that the unconstrained objective has one unique global minimizer [NW99].

Our current approach for solving the quadratic programming problem is to apply a direct method based on a Shur method and/or factorization. We have not explored any iterative methods for solving the optimization problem because the high accuracy of our direct approach results in a volume loss in the order of 0.01% in all our test cases.

Since $\mathbf{A}^{-1}$ is non-singular it can be inverted and the KKT system can be solved efficiently using a Shur method [NW99, Saa03]. A Shur method results in the Shur system $\mathbf{P}^T\mathbf{A}^{-1}\mathbf{P}\lambda^* = \mathbf{P}^T\mathbf{A}^{-1}\mathbf{b}$. This has the advantage of reducing the number of variables. Observe the Shur matrix is non-singular if $\mathbf{P}$ has full column rank. Locking may occur for instance near solid boundaries or at non-manifold fluid surface points (happens at droplet collisions). In case of locking $\mathbf{P}$ does not have full column rank and the Shur system becomes singular. To circumvent this numerical problem we apply a numerical damping strategy and add a stabilization term to the second equation in the KKT system, $\mathbf{P}^T\tilde{\mathbf{u}}^* + \mathbf{S}\lambda^* = \mathbf{0}$. Here $\mathbf{S}$ is a symmetric block matrix. In [MBE*10] an area weighted strategy is used that allows $\mathbf{S}$ to be interpreted as Laplacian smoothing of the resulting pressure field and has the property of provable global volume conservation. However, from a numerical viewpoint one may simply use a small valued positive diagonal matrix.

Currently our implementation uses a sparse Cholesky factorization for solving the Shur system [DH11]. When we add second order approximations later we apply the solver to the full KKT system. However, it should be noted that the KKT matrix is a sparse symmetric non-singular matrix and a Conjugate Gradient method may be considered as a computational effective alternative. We speculate that one may use a mass-matrix like (using a block diagonal matrix of $\mathbf{M}$ and $\mathbf{S}$) pre-conditioner for such an iterative scheme. We have not explored this further as we are content with the current performance of our direct factorization method.

However, it does not appear as though we have gained much from restating our time discretized equations as first order optimality conditions for this optimization problem. In fact the optimization problem seems to have complicated matters. The added benefit comes when other force contributions are considered such as the surface tension forces. The surface tension energy potential should be minimized as much as possible this suggest we should apply a minimization problem with the objective

$$\frac{1}{2}\tilde{\mathbf{u}}^T\left(\mathbf{A}\tilde{\mathbf{u}}+2\mathbf{b}\right)+U(\tilde{\mathbf{x}}). \tag{40}$$

The problem is that we wish to minimize with respect to velocity $\tilde{\mathbf{u}}$ and not position $\tilde{\mathbf{x}}$. To get around this we make a second order Taylor series approximation for $U(\tilde{\mathbf{x}}) = U(\mathbf{x}^t + \Delta t\tilde{\mathbf{u}})$,

$$\begin{aligned} U(\mathbf{x}^t+\Delta t\tilde{\mathbf{u}}) \approx U(\mathbf{x}^t) &+ \Delta t\nabla U(\mathbf{x}^t)\tilde{\mathbf{u}} \\ &+ \Delta t^2\tilde{\mathbf{u}}^2\nabla^2 U(x^t)\tilde{\mathbf{u}}. \end{aligned} \tag{41}$$

Observe the approximation is now only a function of the velocities. When minimizing the above expression the $U(\mathbf{x}^t)$ term is a constant and can therefore be ignored. Substitution of this simplified second order approximation leads to a new objective function

$$\frac{1}{2}\tilde{\mathbf{u}}^T\left(\left(\underbrace{\mathbf{A}+2\Delta t^2\nabla^2 U}_{\mathbf{A}'}\right)\tilde{\mathbf{u}}+\underbrace{2\mathbf{b}+2\Delta t\nabla U}_{2\mathbf{b}'}\right) \tag{42}$$

where we for readability have dropped explicitly writing the $\mathbf{x}^t$ dependency of the U terms. Notice that the optimization problem is still a strict convex quadratic problem given by the objective

$$\frac{1}{2}\tilde{\mathbf{u}}^T\left(\mathbf{A}'\tilde{\mathbf{u}}+2\mathbf{b}'\right). \tag{43}$$

We replaced the position dependent force term with an implicit second order approximation which will introduce a small amount of discretization error. However, the added benefit is that we can add complex external forces to the scheme in a consistent manner without breaking the discrete divergence free constraint.

5. Results and Examples

In the 2D water splash in Figure 1 we optimized the Delaunay property as 2D Delaunay meshes have good numerical properties from the finite element method point of view. In the 3D case we optimize the volume to root mean square edge length quality measure [PGH94].

The improvement loop in DSC which runs over all mesh elements continues to iterate either until all vertices have moved to their final positions given by the specified displacement field or a maximum iteration count is exceeded. In all our cases the DSC loop stops after at most 3 iterations (normally, 1 or 2) so we use a maximum of 5 iterations. In all the test examples shown we apply the explicit Lagrangian advection method and we used the sparse Cholesky factorization to solve the full KKT system. For the experiments in this paper we also applied lumped matrix approximation due to its simplicity.

In the examples we used the area weighted damping matrix $\mathbf{S}$ from [MBE*10]. The δ parameter used for creating the $\mathbf{S}$ matrix was chosen such that the absolute values of the $\mathbf{S}$ matrix entries are at least 100 times smaller than the entries of any other matrices. This has worked well in practice for our experiments.

The implementation is in C++ and single threaded. We have done no attempts at optimizing this. The platform used for Figure 2 to 5 is based on 64-bit Intel® Core® i7 CPU X980 @ 3.33 GHz, 24 GB RAM. Fine renderings are in the supplementary movies.

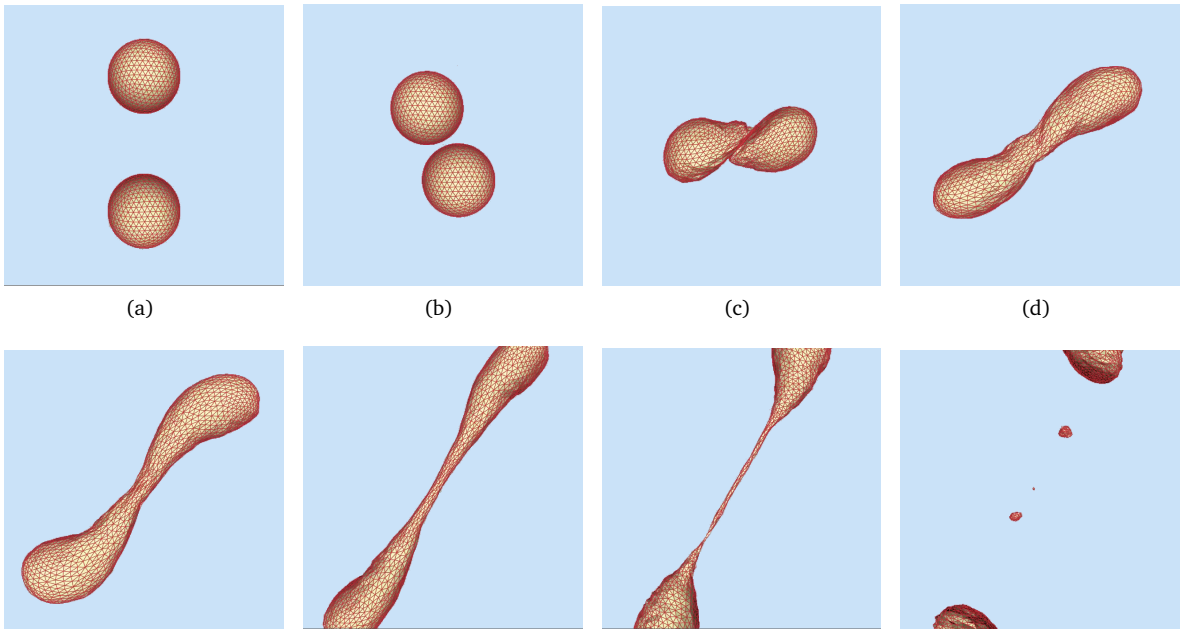
Frame computing time depends on the number of DSC iterations per frame, However, it is only a fraction of the frame time that is used for DSC. Frame times without rendering are in the range of 5-18 seconds on average in the droplets examples shown below, and 60-80 seconds in the bunny examples shown below.

We have observed that $\Delta t = 0.02$ seconds or smaller seems to work fine for the droplets examples. In all examples the surface tension values were chosen such that $\cos(\theta) = (\gamma_{SA} - \gamma_{SF})/(\gamma_{FA}) = -0.9$ where θ is the contact angle, γ_{SA} is the surface tension between solid and air, γ_{SF} between solid and fluid, and γ_{FA} is between fluid and air.

In Figure 2 a symmetrical collision between two water droplets in zero gravity is shown. The surface area and mesh statistics change a lot during the simulation. An non-symmetrical droplet collision is shown in Figure 3. Here the mesh statistics also varies a lot. A water Stanford bunny is left in a zero gravitational field in Figure 4 . Figure 5 shows the bunny splashing against a spherical container. We present detailed measurements of the droplet collision simulations in Table 1. The table reveal that the KKT solving is currently our bottleneck. The results also show that matrices are very

(a) (b) (c) (d) (e)

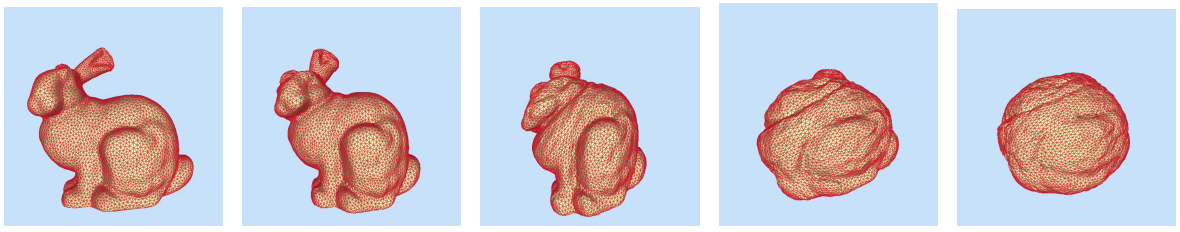
Figure 2: *A 3D optimization-based fluid animation using deformable simplicial complexes. Two water droplets in a symmetric collision in zero gravity. Observe the paper thin structure that develops and ruptures due to surface tension forces.*



(a) (b) (c) (d)

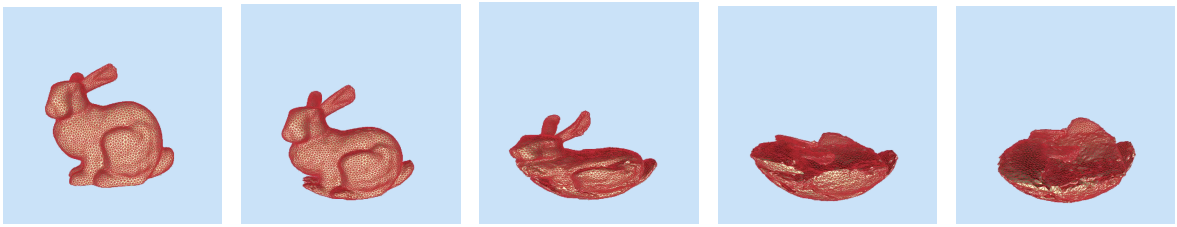
(e) (f) (g) (h)

Figure 3: *A 3D optimization-based fluid animation using deformable simplicial complexes. Two water droplets during a non-symmetric collision in zero gravity. Observe how the thin water tail after the collision breaks into small droplets*



(a) (b) (c) (d) (e)

Figure 4: *A 3D optimization-based fluid animation using deformable simplicial complexes. The stanford water bunny in outer space. Observe how surface tension forces pulls the bunny into a near perfect spherical form.*



(a) (b) (c) (d) (e)

Figure 5: *A 3D optimization-based fluid animation using deformable simplicial complexes. The stanford water bunny is falling under gravity in a spherical container.*

Scene	Unknowns (#)/Nonzeros of $\mathbf{A}$ (%)				Fluid Tet./Surface Tri. (#)				Volume Loss (%)			
	Avg.	Min.	Max.	Std.	Avg.	Min.	Max.	Std.	Avg.	Min.	Max.	Std.
Sym. Droplet	31K /0.002	12K/0.001	67K/0.003	19K/0.001	16K/ 9K	6K/3K	34K/22K	9K/6K	0.008	-0.024	0.358	0.038
Non-Sym. Droplet	14K/0.003	12K/0.002	17K/0.003	1K/0.0003	7K/4K	6K/3K	8K/5K	0.5K/0.6K	0.008	-0.017	0.045	0.005

Scene	Total Wall Clock (secs)				Advection Step (%)				Matrix Assembly (%)				KKT Solver (%)			
	Avg.	Min.	Max.	Std.	Avg.	Min.	Max.	Std.	Avg.	Min.	Max.	Std.	Avg.	Min.	Max.	Std.
Sym. Droplet	18.1	4.8	127.9	13	14.1	4.9	43.7	4	13.5	5.6	35.5	4.3	59	27.1	79.2	11.0
Non-Sym. Droplet	5.9	3.5	11.4	2.0	20.7	9.2	45.9	6.8	14.8	6.2	25.3	5.1	45.4	15.8	75.3	14.2

Table 1: *Statistics on performance measurements. All numbers have been rounded up. The platform used is based on 64-bit Intel® Core™ i5 CPU M520 @ 2.40 GHz, 4GB RAM. Windows 7. Observe that KKT solver takes more than half the time and the matrices are very sparse even though mesh statistics vary greatly.*

sparse throughout simulation and that the DSC iterations done in the advection step only takes a smaller fraction of the total time.

6. Discussion and Summary

In this paper we derived closed form formulas for the finite element matrices (28) used by the optimization-based fluid animation method. In particular we extended past work to include the diffusion term resulting in the diffusion matrix given by (28c). Finally, we showed how the time discretization of the finite element equations were equivalent to a constrained minimization problem shown in (37). This reformulation proved to be particular useful when considering non-linear force terms as one continues to have a quadratic programming problem given by the objective in (43).

As demonstrated by our derivations and examples the fluid animation method in this paper is capable of handling complex non-linear force terms, the optimization-based setting makes it easy to set tolerances for what the end-user considers as acceptable volume loss. Although linear elements are applied in our work, it is clear from derivations how to extend the finite element method to higher order shape functions simply by applying quadrature rules to the integrals in (28). There are limitations to our work:

- The explicit semi-Lagrangian approach for advection causes the time step to be bounded by the average edge size in the mesh. This is tightly coupled to the workings of DSC.
- Mixing of fluids is not obvious. It is clear that each simplex can have only one phase, but we do not have to have discrete phases. We could have a color or other continuous property in each simplex and allow the color of neighboring simplices to influence each other. However, it raises some difficult questions. For instance when the connectivity of the mesh is changed in the DSC method then it is not obvious what to do. We may also need to directly model some exchange/mixing of material between simplices which calls for additional governing equations in our mathematical model.
- Moving obstacles and non-mixing two phase flows are not supported. We believe it is not too hard to extend to include these.
- The optimization-based method needs to assemble the matrices in each time step as they depend on the current spatial position of the mesh. This has an added computational cost compared to matrix-free methods.
- Real-life computational fluid dynamics applications is not yet within our grasp. More formal error analysis and validation are needed. However, this is not a limitation for computer animation applications as our examples demonstrate.

Acknowledgements

We would like to acknowledge Robert Bridson for conceiving the original idea for the optimization-based fluid method. Thanks to Jeppe Revall Frisvad for providing us with a fluid render.

References

[BB08] BATTY C., BRIDSON R.: Accurate viscous free surfaces for buckling, coiling, and rotating liquids. In *Proceedings of the 2008 ACM/Eurographics Symposium on Computer Animation* (July 2008), pp. 219–228. 1

[BBB07] BATTY C., BERTAILS F., BRIDSON R.: A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. 26*, 3 (2007), 100. 1

[BBB10] BROCHU T., BATTY C., BRIDSON R.: Matching fluid simulation elements to surface geometry and topology. In *ACM SIGGRAPH 2010 papers* (2010), ACM, p. XX. 2

[BMF07] BRIDSON R., MÜLLER-FISCHER M.: Fluid simulation: Siggraph 2007 course notes. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses* (New York, NY, USA, 2007), ACM, pp. 1–81. 2

[Bri08] BRIDSON R.: *Fluid Simulation*. A. K. Peters, Ltd., Natick, MA, USA, 2008. 2

[BW00] BONET J., WOOD R. D.: *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, 2000. 4

[BWHT07] BARGTEIL A. W., WOJTAN C., HODGINS J. K., TURK G.: A finite element method for animating large viscoplastic flow. *ACM Trans. Graph. 26* (July 2007). 2

[CFL*07] CHENTANEZ N., FELDMAN B. E., LABELLE F., O'BRIEN J. F., SHEWCHUK J. R.: Liquid simulation on lattice-based tetrahedral meshes. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2007), Eurographics Association, pp. 219–228. 2

[CGFO06] CHENTANEZ N., GOKTEKIN T. G., FELDMAN B. E., O'BRIEN J. F.: Simultaneous coupling of fluids and deformable bodies. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2006), Eurographics Association, pp. 83–89. 2

[DH11] DAVIS T. A., HAGER W.: Cholmod. Open source software, GNU LGPL license. Accessed online http://www.cise.ufl.edu/research/sparse/cholmod/, May 2011. 7

[ETK*07] ELCOTT S., TONG Y., KANSO E., SCHRÖDER P., DESBRUN M.: Stable, circulation-preserving, simplicial fluids. *ACM Trans. Graph. 26*, 1 (2007), 4. 2, 3

[FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 23–30. 2

[FM96] FOSTER N., METAXAS D.: Realistic animation of liquids. *Graph. Models Image Process. 58*, 5 (1996), 471–483. 2

[FM97] FOSTER N., METAXAS D.: Modeling the motion of a hot, turbulent gas. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 181–188. 2

[FOK05] FELDMAN B. E., O'BRIEN J. F., KLINGNER B. M.: Animating gases with hybrid meshes. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), ACM, pp. 904–909. 2, 3

[FOKG05] FELDMAN B. E., O'BRIEN J. F., KLINGNER B. M., GOKTEKIN T. G.: Fluids in deforming meshes. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM, pp. 255–259. 2, 3

[FP02] FERZIGER J. H., PERIC M.: *Computational Methods for Fluid Dynamics*, 3rd ed. Springer, 2002. 4

[FSJ01] FEDKIW R., STAM J., JENSEN H. W.: Visual simulation of smoke. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 15–22. 2

[KBAE09] KRZYSZTOF M. M., BÆRENTZEN J. A., ANTON F., ERLEBEN K.: Tetrahedral mesh improvement using multiface retriangulation. In *Proocedings of the 18th International Meshing Roundtable* (Salt Lake City, 2009), pp. 539–556. 2

[KFCO06] KLINGNER B. M., FELDMAN B. E., CHENTANEZ N., O'BRIEN J. F.: Fluid animation with dynamic meshes. *ACM Trans. Graph. 25*, 3 (2006), 820–825. 2, 4

[MBE*10] MISZTAL M. K., BRIDSON R., ERLEBEN K., BÆRENTZEN J. A., ANTON F.: Optimization-based fluid simulation on unstructured meshes. In *Proceedings of the Seventh Workshop on Virtual Reality Interactions and Physical Simulations, VRIPHYS 2010, Copenhagen, Denmark, 2010* (2010), pp. 11–20. 1, 2, 6, 7

[Mis10] MISZTAL M. K.: *Deformable Simplicial Complexes*. PhD thesis, Technical University of Denmark (DTU), Denmark, 2010. 2

[NW99] NOCEDAL J., WRIGHT S. J.: *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999. 6

[PGH94] PARTHASARATHY V. N., GRAICHEN C. M., HATHAWAY A. F.: A comparison of tetrahedron quality measures. *Finite Elements in Analysis and Design 15*, 3 (1994), 255–261. 7

[Saa03] SAAD Y.: *Iterative methods for sparse linear systems*. Society for Industrial Mathematics, 2003. 6

[Sta99] STAM J.: Stable fluids. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 121–128. 2, 4

[WBOL07] WENDT J. D., BAXTER W., OGUZ I., LIN M. C.: Finite volume flow simulations on arbitrary domains. *Graph. Models 69*, 1 (2007), 19–32. 2, 3

[WRK*10] WICKE M., RITCHIE D., KLINGNER B. M., BURKE S., SHEWCHUK J. R., O'BRIEN J. F.: Dynamic local remeshing for elastoplastic simulation. *ACM Trans. Graph. 29* (July 2010), 49:1–49:11. 2

[WT08] WOJTAN C., TURK G.: Fast viscoelastic behavior with thin features. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), ACM, pp. 1–8. 2

[ZT00] ZIENKIEWICZ O. C., TAYLOR R. L.: *The Finite Element Method Volume 1: The Basis*, fifth edition ed. Butterworth-Heinemann, 2000. 4