VII International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2017
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

COUPLING FREE-SURFACE FLOW AND MESH DEFORMATION IN AN ISOGEOMETRIC SETTING

STEFANIE ELGETI*, FLORIAN ZWICKE*AND SEBASTIAN EUSTERHOLZ*

*Chair for Computational Analysis of Technical Systems (CATS)
CCES, Schinkelstr. 2, 52062 Aachen, Germany
e-mail: elgeti@cats.rwth-aachen.de, web page: http://www.cats.rwth-aachen.de/

Key words: Isogeometric Analysis, Interface Tracking, Free Surface Flow

Abstract. The simulation of certain flow problems requires a means for modeling a free fluid surface; examples being viscoelastic die swell or fluid sloshing in tanks. In a finite-element context, this type of problem can, among many other options, be dealt with using an interface-tracking approach with the Deforming-Spatial-Domain/Stabilized-Space-Time (DSD/SST) formulation [1]. A difficult issue that is connected with this type of approach is the determination of a suitable coupling mechanism between the fluid velocity at the boundary and the displacement of the boundary mesh nodes. In order to avoid large mesh distortions, one goal is to keep the nodal movements as small as possible; but of course still compliant with the no-penetration boundary condition. One common choice of displacement that fulfills both requirements is the displacement with the normal component of the fluid velocity. However, when using finite-element basis functions of Lagrange type for the spatial discretization, the normal vector is not uniquely defined at the mesh nodes. This can create problems for the coupling, e.g., making it difficult to ensure mass conservation. In contrast, NURBS basis functions of quadratic or higher order are not subject to this limitation. These types of basis functions have already been used in the context of free-surface boundaries, in connection with the NURBS-enhanced finite-element method (NEFEM) [2]. However, this method presents some difficulties due to the fact that it does not adhere to the isoparametric concept. As an alternative, we investigate the suitability of using the method of isogeometric analysis for the spatial discretization. If NURBS basis functions of sufficient order are used for both the geometry and the solution, both a well-defined normal vector as well as the velocity are available on the entire boundary. This circumstance allows the weak imposition of the no-penetration boundary condition. We compare this option with a number of alternatives. Furthermore, we examine several coupling methods between the fluid equations, boundary conditions, and equations for the adjustment of interior control point positions.

1 INTRODUCTION

This paper can be placed in the field of free boundary problems. More specifically, it considers fluid flow problems where the computational domain is part of the solution — for example a computational domain that contains a free surface. Examples are sloshing tanks — under a seismic load, the liquid stored in a tank begins to slosh — and rising bubbles — a liquid bubble enclosed in a second fluid rises or falls due to the buoyancy force.

2 GOVERNING EQUATIONS FOR THE FREE-BOUNDARY VALUE PROBLEM OF FREE-SURFACE FLOW

For free-surface flow, three types of equations are relevant: (1) the Navier-Stokes equations in combination with an appropriate constitutive equation govern the fluid flow, (2) the displacement of the free surface, which is governed by the no-penetration boundary condition, and (3) the equations governing the possible displacement of interior parts of the domain in order to maintain validity of the mesh. This section will discuss the relevant aspects of these equations.

2.1 Governing equations for fluid flow: The Navier-Stokes equations

In the generic incompressible and isothermal fluid flow problem the unkowns are the velocity, $\mathbf{u}(\mathbf{x},t)$, and the pressure, $p(\mathbf{x},t)$. The computational domain at each instant in time, denoted by Ω_t, is a subset of $\mathbb{R}^{nsd}$, where nsd is the number of space dimensions. Then, at each point in time $t \in [0,T]$, the flow problem is governed by the Navier-Stokes equations, which in our notation read:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \mathbf{f}\right) - \nabla\cdot\boldsymbol{\sigma} = \mathbf{0} \quad \text{on} \quad \Omega_t \ \forall\ t \in [0,T]\,, \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{on} \quad \Omega_t\ \forall t \in [0,T]\,, \tag{2}$$

with ρ as the density of the fluid. We consider only Newtonian fluids, meaning that the stress tensor σ is defined as

$$\boldsymbol{\sigma}(\mathbf{u},p) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) \quad \text{on} \quad \Omega_t\,, \tag{3}$$

with

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\,, \tag{4}$$

where μ denotes the dynamic viscosity. $\mathbf{f}$ includes all external body forces with respect to the unit mass of fluid. Note that the spatial domain is time-dependent, which is indicated by subscript t.

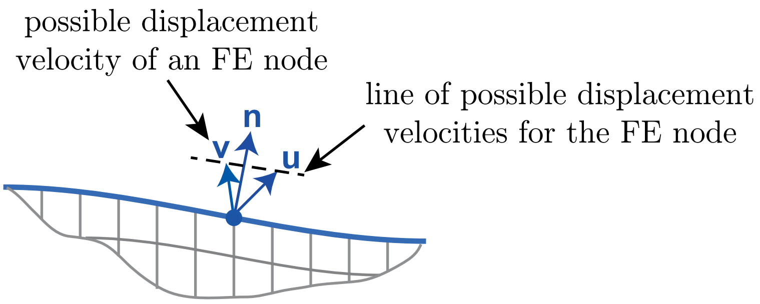
possible displacement velocity of an FE node
line of possible displacement velocities for the FE node
n
v
u

Figure 1: The figure shows an excerpt of a finite element mesh with the free surface indicated in blue. For each FE node, the fluid velocity $\mathbf{u}$ is given. Together with the normal vector $\mathbf{n}$, it serves as a basis for the displacement velocity $\mathbf{v}$.

2.2 Displacement of the free surface: The no-penetration boundary condition

We consider test cases in which the position of the full boundary Γ, or a portion thereof, is not known in advance, but part of the flow solution. The boundary is then defined as a free surface Γ_{free}. In principle, there are two general methods to determine the exact position of the free surface: interface capturing and interface tracking [1]. Interface capturing means that in addition to the finite element mesh for the flow solution, a separate indicator — e.g., a level-set function or particles — is employed to indicate the position of the free surface. In interface tracking — as it is used here —, the computational domain will adapt to the displacement of the free surface. This displacement in turn is governed by the kinematic boundary condition

$$\mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = \mathbf{u}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})\,. \tag{5}$$

It contains the fluid velocity $\mathbf{u}$ and the displacement velocity $\mathbf{v}$ of the free surface. Notice that several choices for $\mathbf{v}$ are valid (cf. Figure 1); the straightforward one being

$$\mathbf{v}(\mathbf{x}) = \mathbf{u}(\mathbf{x})\,. \tag{6}$$

However, there exist other choices that still comply with Equation (5), but suppress certain components of the full velocity for the benefit of a higher mesh quality. Behr [11] details two common alternatives: displacement with the normal velocity component with $\mathbf{v} = (\mathbf{u} \cdot \mathbf{n})\mathbf{n}$ and displacement only in a specific coordinate direction $\mathbf{d}$ (e.g., y-direction), i.e., $\mathbf{v} = \frac{(\mathbf{u}\cdot\mathbf{n})\mathbf{d}}{\mathbf{n}\cdot\mathbf{d}}$.

2.3 Retaining mesh quality: Mesh update for inner nodes

As the boundary Γ_{free} is modified, usually the discretization of the interior of the domain needs to be adapted as well. For this purpose, we employ the Elastic Mesh

Update Method (EMUM) [12]. In this method, the computational mesh is treated as an elastic body reacting to the boundary deformation applied to it. The linear elasticity equation is solved for the mesh displacement $\boldsymbol{\upsilon}$, which relates to the mesh velocity $\mathbf{v}$ as $\boldsymbol{\upsilon} = \mathbf{v}\Delta t$:

$$\nabla \cdot \boldsymbol{\sigma}_{\text{mesh}} = 0\,, \tag{7}$$

$$\boldsymbol{\sigma}_{\text{mesh}}(\boldsymbol{\upsilon}) = \lambda_{mesh}\left(\operatorname{tr}\boldsymbol{\varepsilon}_{\text{mesh}}(\boldsymbol{\upsilon})\right)\mathbf{I} + 2\mu_{mesh}\boldsymbol{\varepsilon}_{\text{mesh}}(\boldsymbol{\upsilon}), \tag{8}$$

$$\boldsymbol{\varepsilon}_{\text{mesh}}(\boldsymbol{\upsilon}) = \frac{1}{2}\left(\nabla\boldsymbol{\upsilon} + (\nabla\boldsymbol{\upsilon})^T\right). \tag{9}$$

λ_{mesh} and μ_{mesh} — in structural mechanics the Lamé-parameters — have no physical meaning within the mesh deformation. They can be chosen freely for each element in order to control its respective stiffness.

3 ISOGEOMETRIC ANALYSIS

Initiated by Hughes at al. [3], a recent trend in the finite element analysis is the use of isogeometric methods. The key idea is to use Non-Uniform Rational B-splines (NURBS) as finite element shape functions. Compared with classic Lagrange polynomial shape functions, this concept has distinct advantages — ranging from higher geometrical accuracy to higher stability of the numerical solution. Details of Isogeometric Analysis can be found in a vast number of sources, e.g., [4]. It has been applied to a variety of applications, e.g., in the context of the phase field method for the Cahn-Hillard equations [5], brittle fracture [6], and topology optimization [7]. In connection with level-set, IGA has been utilized to compute the dam break problem [8, 16] or as a boundary indicator within the finite cell method [9]. Another free-surface-related application is the computation of wave resistance on a ship hull using the isogeometric boundary element method [17]. To the knowledge of the authors, IGA has not been utilized in conjunction with interface tracking of free surfaces.

To achieve an application to interface tracking, one difference between IGA and the standard finite element method becomes crucial: In IGA, the unkown velocities are not stored at points, which lie on the free surface. Instead, the unknown values — termed control variables — can be associated with the control points of the spline.

4 DISPLACEMENT OF THE FREE SURFACE

In Section 2.2 we discussed the importance of displacement conditions that do not involve the full velocity vector, but only a portion of it, pointing in a certain direction. The section named displacement in normal and in vertical direction as examples. Bear in mind that both the vertical and the normal direction vector — but also any other directional vector one could imagine — are all associated with a specific point on the surface. As indicated in the previous section, however, IGA stores the unknowns — in

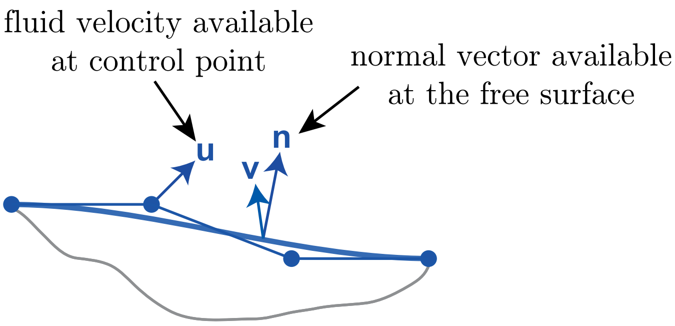
fluid velocity available at control point
normal vector available at the free surface
u
v
n

Figure 2: The figure shows a free surface represented by a NURBS curve. The fluid velocity is stored as a control variable, whereas the normal vector is computed on the free surface.

our case velocity information — at the control points. This entails that the kinematic boundary condition, Equation (5), now contains two components — the velocity unknowns and the directional vector — which are evaluated at different locations. It can no longer be directly evaluated and much less directly fulfilled (cf. Figure 2). As a remedy, we propose two displacement methods for the free surface; one based on strong imposition and one on weak imposition of the kinematic boundary condition. The first option entails moving the control points with normal vectors that are computed at the Greville abscissae. The second option considers the weak fulfillment of the no-penetration boundary condition — meaning that an additional equation system will need to be solved.

4.1 Displacement based on the normal vector at the Greville abscissae

As detailed before, there is no notion of a normal vector at a control point. One point on the spline, whose normal vector might come close to a vector that might be considered a normal vector at a specific control point, is the normal vector at the corresponding Greville abscissa. The Greville abscissa is the point on the surface the control point converges to in case of refinement. The Greville abscissa can be computed as the average of the knot values relevant for a control point, excluding the first and last value [18]. Typically, this is also close to the point where the associated basis function is maximal (this is exactly fulfilled for uniform knot vectors). The normal vector of, e.g., a NURBS curve, at any parametric coordinate θ can then be computed as [19]:

$$\boldsymbol{n}(\theta) = \begin{pmatrix} -t_y \\ t_x \end{pmatrix}, \qquad \text{with } \boldsymbol{t}(\theta) = \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \frac{\boldsymbol{C}'(\theta)}{|\boldsymbol{C}'(\theta)|}. \tag{10}$$

The formula requires $\boldsymbol{C}'(\theta)$, the first derivative of the curve with respect to the local parameter θ, for the definition of which we refer to [13].

Based on the fluid velocity at any given control point $\mathbf{P}_i$, we define its displacement (or control point coordinate increment) $\Delta\mathbf{P}_i$ as:

$$\Delta\mathbf{P}_i = (\bar{\mathbf{u}}_i \cdot \mathbf{n})\,\mathbf{n} \cdot \Delta t\,, \tag{11}$$

with $\bar{\mathbf{u}}_i$ the fluid velocity connected to control point i averaged over the time step and $\mathbf{n}$ the time-averaged normal vector evaluated at the i^{th} Greville abscissa.

4.2 Displacement based on a weak formulation of the kinematic boundary condition

In a finite element context, the kinematic boundary condition can also be imposed weakly. The respective formulation reads:

$$\int_{\Gamma_{free}} \mathbf{w}\,\mathbf{F}\left(\mathbf{v},\mathbf{u}\right)\,d\mathbf{x} = 0 \quad \forall \mathbf{w} \in \mathcal{V}, \tag{12}$$

with Γ_{free} as the spatial free surface boundary, the test functions $\mathbf{w}$ in a suitable test space $\mathcal{V}$, surface displacement velocity $\mathbf{v}(\theta)$ and the fluid velocity $\mathbf{u}(\theta)$ both at a given surface point . After discretization, this yields a system of equations that can be solved for new control point coordinates $\mathbf{P}_i$. The fulfillment of the kinematic boundary condition, and thus mass conservation, now depends on how closely the correct boundary $\mathbf{x}(\theta)$ can be interpolated by the spline basis functions. In addition, the exact choice of $\mathbf{F}$ influences mass conservation when using discrete basis functions. $\mathbf{F}\left(\mathbf{v},\mathbf{u}\right)$ can contain any expression that fulfills the kinematic boundary condition.

5 ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of DFG under the collaborative research projects CoE 128 “Integrative Production Technology for High-Wage Countries” and SFB 1120 (subproject B2). Computing resources were provided by the AICES graduate school, RWTH Aachen University Center for Computing and Communication and by the Forschungszentrum Jülich.

REFERENCES

[1] S. Elgeti, H. Sauerland, Deforming Fluid Domains Within the Finite Element Method: Five Mesh-Based Tracking Methods in Comparison, Archives of Computational Methods in Engineering 23 (2) (2015) 323–361.

[2] T. E. Tezduyar, M. Behr, J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces–the deforming-spatial-domain/space-time procedure: I. the concept and the preliminary numerical tests 94 (3) (1992) 339 – 351.

[3] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement 194 (2005) 4135–4195.

[4] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, Ltd, 2009.

[5] H. Gómez, V. Calo, Y. Bazilevs, T. Hughes, Isogeometric analysis of the Cahn–Hilliard phase-field model 197 (49) (2008) 4333–4352.

[6] M. Borden, J. Michael, C. Verhoosel, M. Scott, T. J. Hughes, C. Landis, A phase-field description of dynamic brittle fracture, Computer Methods in Applied Mechanics and Engineering 217 (2012) 77–95.

[7] L. Dedè, M. Borden, T. Hughes, Isogeometric Analysis for Topology Optimization with a Phase Field Model 19 (2012) 427–465.

[8] I. Akkerman, Y. Bazilevs, C. E. Kees, M. W. Farthing, Isogeometric analysis of free-surface flow, Journal of Computational Physics 11 (2011) 4137–4152.

[9] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, A. Düster, Geometric modeling, isogeometric analysis and the finite cell method, Computer Methods in Applied Mechanics and Engineering 249 (2012) 104–115.

[10] J. Ferziger, M. Perić, Computational Methods for Fluid Dynamics, Springer, 1999.

[11] M. Behr, Stabilized Finite Element Methods for Incompressible Flows with Emphasis on Moving Boundaries and Interfaces, Ph.D. thesis, University of Minnesota, Department of Aerospace Engineering and Mechanics (1992).

[12] A. Johnson, T. Tezduyar, Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces 119 (1994) 73 – 94.

[13] L. Piegel, W. Tiller, The NURBS Book, Springer, Berlin, Germany, 1997.

[14] D. Rogers, An Introduction to NURBS with Historical Perspective, Morgan Kaufmann Publishers, 2001.

[15] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, T. Sederberg, Isogeometric analysis using T-splines 199 (2010) 229–263.

[16] R. Amini, R. Maghsoodi, N. Z. Moghaddamog, Simulating free surface problem using isogeometric analysis, Journal of the Brazilian Society of Mechanical Sciences and Engineering 38 (2) (2016) 413–421.

[17] A. Ginnisa, K. Kostasb, C. Politisb, P. Kaklisa, K. Belibassakisa, T. Gerostathisb, M. Scott, T. Hughes, Isogeometric Boundary-Element Analysis for the Wave-Resistance Problem using T-splines, Computer Methods in Applied Mechanics and Engineering 279 (2014) 425–439.

[18] G. Farin, Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide, Elsevier, 2014.

[19] A. Gray, Modern differential geometry of curves and surfaces with Mathematica, 2nd Edition, CRC Press, Boca Raton, 1998.