A Level Set Discontinuous Galerkin Method for Free Surface Flows[1]

J. Grooss[a], J.S. Hesthaven[b]

[a] *Informatics and Mathematical Modeling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

[b] *Division of Applied Mathematics, Brown University, Providence, RI 02912, USA*

Abstract

We present a discontinuous Galerkin method on a fully unstructured grid for the modeling of unsteady incompressible fluid flows with free surfaces. The surface is modeled by embedding and represented by a levelset. We discuss the discretization of the flow equations and the level set equation as well a various ways of advancing the equations in time using velocity projection techniques. The efficacy of the method for the representation of the levelset and its reinitialization is discussed and several numerical tests confirm the robustness and versatility of the proposed scheme.

Key words: Level set, Discontinuous Galerkin, Free Surface Flow, Spectral, High Order, Incompressible Navier Stokes, Fluid-Structure interaction, Surface Tension

1 Introduction

In maritime engineering, as in numerous other fluid driven applications, one is often interested in the modeling and prediction of phenomena involving free surfaces, e.g., water wave load or current load on structures. Such modeling efforts are not only complicated due to the free surface but also by the need to accurately account for the fluid-structure interaction problems, often in geometrically complex domains. Furthermore, such types of problems often involve a variety of length scales, from an incoming wave length of several

Email addresses: `jg@imm.dtu.dk` (J. Grooss), `Jan.Hesthaven@Brown.edu` (J.S. Hesthaven).

[1] The work of JSH was partly supported by NSF Career Award DMS0132967 and by the Alfred P. Sloan Foundation through a Sloan Research Fellowship.

meters to the flow at microscopic scale around the structures, putting rather severe requirements on suitable computational techniques.

Classical strategies for incorporating a free surface in the flow simulation include moving (Lagrangian) grid techniques (22), marker and cell (MAC) (10), volume of fluid (VOF) (14), and level set methods.

In this work we shall pursue the latter technique, level set methods, (18; 23; 20) to define the surface as the zero contour of a function. This function is subsequently advected with the flow as an implicitly defined identifier of the interface.

We shall discuss the development of a high order accurate discontinuous Galerkin (DG) method for solving the incompressible unsteady two-fluid Navier-Stokes equations. The DG method utilizes a fully unstructured grid based on nodal triangular elements, thus enabling the treatment of general geometries. A high-order nodal basis is used to enable local high-order and is well suited for problems with many scales.

We shall furthermore discuss the use of high-order schemes for advancing the unsteady equations. For computational reasons, a common approach is to decouple the evaluation of the pressure and the velocities, and treat the diffusive term implicitly and the non-linear terms explicitly. As is well known, this introduces time splitting errors. In an attempt to address this, we explore the use of a semi implicit spectral deferred correction method, following the work in (6; 17), although recast to work on the physical variables and therefor suited to free surface flow problems. The study exposes errors associated with low stage-order, impacting the accuracy adversely for such methods when used for initial-boundary-value problems. However, the examples also show the method to be very flexible and with potential to achieve high temporal order.

In Sec. 2 we describe the equations for the two-dimensional incompressible two-fluid flows. Section 3 is devoted to the DG method in general and how to apply it to an elliptic problem and the Navier-Stokes equations. This sets the stage for Sec. 4 where we discuss different ways of achieving high temporal order, while Sec. 5 describes in detail the level set method, its reinitialization, boundary conditions, and solution. In Sec. 6 we present a few numerical examples while Sec. 7 concludes with a few remarks.

2 The Two-Fluid Navier-Stokes Equations

We shall consider the dynamics of a two-dimensional incompressible non-reacting two-fluid, described by the two sets of Navier-Stokes equations

$$\forall \boldsymbol{x} \in \Omega_l \ : \ \rho_l \left(\frac{\partial \boldsymbol{u}_l}{\partial t} + (\boldsymbol{u}_l \cdot \nabla)\boldsymbol{u}_l \right) = -\nabla p_l + \mu_l \nabla^2 \boldsymbol{u}_l \ , \ \ \nabla \cdot \boldsymbol{u}_l = 0 \ , \tag{1}$$

$$\forall \boldsymbol{x} \in \Omega_g \ : \ \rho_g \left(\frac{\partial \boldsymbol{u}_g}{\partial t} + (\boldsymbol{u}_g \cdot \nabla)\boldsymbol{u}_g \right) = -\nabla p_g + \mu_g \nabla^2 \boldsymbol{u}_g \ , \ \ \nabla \cdot \boldsymbol{u}_g = 0 \ . \tag{2}$$

In both fluids, ρ, and μ, represent the constant density and dynamic viscosity, respectively. In each fluid we also have the velocity field, $\boldsymbol{u}$, and the pressure field, p. The full computational domain, $\Omega = \Omega_l \cup \Omega_g$ is assumed fixed in time while both Ω_l and Ω_g are time-dependent. We shall also call the boundary of Ω for $\partial\Omega$ while $\Gamma = \Omega_l \cap \Omega_g$ represents the interface between the two fluids.

At the interface between the two fluids, we have the continuity and kinematic condition

$$\forall \boldsymbol{x} \in \Gamma \ : \ \boldsymbol{u}_l = \boldsymbol{u}_g \ , \quad (\mu_l \nabla \boldsymbol{u}_l - \mu_g \nabla \boldsymbol{u}_g) \cdot \boldsymbol{n}_\Gamma = (p_l - p_g + \sigma\kappa)\, \boldsymbol{n}_\Gamma \ ,$$

where $\boldsymbol{n}_\Gamma$ is the normal along Γ, $\kappa = \nabla \cdot \boldsymbol{n}_\Gamma$ is the local curvature of the interface, and σ is the coefficient of surface tension.

Let us introduce the scalar level set function, ϕ, defined as

$$\phi(\boldsymbol{x}, t) = \begin{cases} > 0 & \boldsymbol{x} \in \Omega_l \\ 0 & \boldsymbol{x} \in \Gamma \\ < 0 & \boldsymbol{x} \in \Omega_g \end{cases} \ , \tag{3}$$

with which we can now define

$$\begin{aligned} \rho(\phi) &= \rho_g + (\rho_l - \rho_g)H(\phi) \ , \\ \mu(\phi) &= \mu_g + (\mu_l - \mu_g)H(\phi) \ , \end{aligned}$$

where $H(x)$ is the classic Heaviside function. We shall also define the global quantities

$$\boldsymbol{u} = \begin{cases} \boldsymbol{u}_l, & \boldsymbol{x} \in \Omega_l \\ \boldsymbol{u}_g, & \boldsymbol{x} \in \Omega_g \end{cases} \ ,$$

and likewise for the pressure, p.

Following (21), one easily shows that we can now combine all the pieces to arrive at a formulation

$$\rho(\phi)\left(\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right)=-\nabla p+\mu(\phi)\nabla\cdot\nabla\boldsymbol{u}-\sigma\delta(\phi)\kappa\boldsymbol{n}_\Gamma \ , \quad \nabla\cdot\boldsymbol{u}=0 \ , \quad (4)$$

where $\delta(\phi)=\frac{\partial}{\partial\phi}H(\phi)$ is the Dirac delta function. Note that the correct way of writing the diffusion term for varying viscosity is $\nabla\cdot\mu(\phi)\nabla\boldsymbol{u}$. The version in Eq. (4) is correct up until the interface Γ. Having in mind the approximations at the interface that we introduce in the coming sections, we assume this approximation to be of same magnitude and to simplify the scheme.

Assuming ϕ is differentiable, we can move ϕ with the flow $\boldsymbol{u}$

$$\frac{\partial\phi}{\partial t}+\boldsymbol{u}\cdot\nabla\phi=0. \quad (5)$$

If we further seek a non-dimensional form using

$$\boldsymbol{x}=L\tilde{\boldsymbol{x}} \ , \quad \boldsymbol{u}=U\tilde{\boldsymbol{u}} \ , \quad t=(L/U)\tilde{t} \ , \quad p=\rho_l U^2\tilde{p} \ , \quad \rho=\rho_l\tilde{\rho} \ , \quad \mu=\mu_l\tilde{\mu} \ ,$$

where $\tilde{}$-variables refer to the dimensionless variable, we recover the general form

$$\rho(\phi)\left(\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right)=-\nabla p+\frac{1}{Re}\mu(\phi)\nabla\cdot\nabla\boldsymbol{u}-\frac{1}{We}\delta(\phi)\kappa(\phi)\frac{\nabla\phi}{|\nabla\phi|} \ ,$$
$$\nabla\cdot\boldsymbol{u}=0 \ , \quad (6)$$

where we have further utilized the properties of the level set function, ϕ, that

$$\boldsymbol{n}_\Gamma=\left.\frac{\nabla\phi}{|\nabla\phi|}\right|_{\phi=0} \ .$$

In Eq.(6), which is the general form we shall consider subsequently, we now have

$$Re=\frac{\rho_l LU}{\mu_l} \ , \quad We=\frac{\rho_l LU^2}{\sigma} \ ,$$

as the Reynolds and the Weber number, respectively, being measures of the dynamics of the equations.

We shall generally assume that $\partial\Omega=\partial\Omega_W\cup\partial\Omega_O$ where $\partial\Omega_W$ refers to hard walls where we impose a no-slip condition

$$\boldsymbol{u}=0 \ , \quad \boldsymbol{x}\in\partial\Omega_W \ ,$$

while we, at open boundaries, $\partial\Omega_O$, shall impose

$$(\boldsymbol{n}\cdot\nabla)\boldsymbol{u}=0 \ , \quad \boldsymbol{x}\in\partial\Omega_O \ .$$

3 The Spatial Discretization

We begin describing the spatial discretization for the conservation law $\frac{\partial q}{\partial t} + \nabla \cdot \boldsymbol{f}(q) = 0$, and subsequently extend it to the incompressible Navier Stokes.

The computational domain, Ω, is split into K non-overlapping triangular elements D^k, such that $\cup_k \mathsf{D}^k = \Omega$, and we define a standard triangular element I,

$$\mathsf{I} = \{(r, s) \in \mathsf{R}^2 \mid r, s \geq -1 \,;\, r + s \leq 0\}. \tag{7}$$

Operations on I are trivially related to those on D^k by standard curvilinear mappings.

In the standard element the value of a function q is represented by a Lagrange interpolating polynomial based on nodes $\boldsymbol{r}_i$

$$q(\boldsymbol{r}, t) \approx \tilde{q}(\boldsymbol{r}, t) = \sum_{i=1}^{N} \hat{q}_i(t) L_i(\boldsymbol{r}) \ , \tag{8}$$

where N is the number of nodes in the element, $\hat{q}_i(t) = q(\boldsymbol{r}_i, t)$, and $L_i(\boldsymbol{r})$ is the ith Lagrange interpolating polynomial.

Consider the weak elementwise formulation of the conservation law

$$\forall\ i \in [0 \ldots N] \ : \quad \int_{\mathsf{D}} \left(\frac{\partial \tilde{q}}{\partial t} + \nabla \cdot \tilde{\boldsymbol{f}} \right) L_i(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x} = 0. \tag{9}$$

Integration by parts yields

$$\forall\ i \in [0 \ldots N] \ : \quad \int_{\mathsf{D}} \frac{\partial \tilde{q}}{\partial t} L_i(\boldsymbol{x}) - \tilde{\boldsymbol{f}} \cdot \nabla L_i(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x} = -\oint_{\partial \mathsf{D}} \boldsymbol{n} \cdot \boldsymbol{f}^* L_i(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x}, \tag{10}$$

where $\boldsymbol{n}$ is the outward pointing normal vector on $\partial\mathsf{D}$. On the right hand side, the numerical flux $\boldsymbol{f}^*$ is used for imposing boundary conditions on each element, thereby specifying how information passes between adjacent elements. This is the classical Discontinuous Galerkin (DG) method in weak form. Integrating by parts again, we end up with the strong formulation of the DG method,

$$\forall\ i \in [0 \ldots N] \ : \quad \int_{\mathsf{D}} \left(\frac{\partial \tilde{q}}{\partial t} + \nabla \cdot \tilde{\boldsymbol{f}} \right) L_i(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x} = \oint_{\partial \mathsf{D}} \boldsymbol{n} \cdot (\tilde{\boldsymbol{f}} - \boldsymbol{f}^*) L_i(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x}. \tag{11}$$

In what remains we shall focus on this form, using the weak form only for illustrative purposes. Introducing the vector $\hat{\boldsymbol{q}} = \{\hat{q}_1, ..., \hat{q}_N\}$ and similar for

$\hat{\boldsymbol{f}}$, and defining the operators $\hat{\mathsf{M}}$, $\hat{\mathsf{S}}$ and $\hat{\mathsf{F}}$ with the elements

$$\hat{M}_{ij} = \int_{\mathsf{D}} L_j L_i \, \mathrm{d}\boldsymbol{x} \ \ , \ \ \hat{S}_{ij} = \int_{\mathsf{D}} \nabla L_j L_i \, \mathrm{d}\boldsymbol{x} \ \ , \ \ \hat{F}_{ij} = \oint_{\partial\mathsf{D}} L_j L_i \, \mathrm{d}\boldsymbol{x},$$

the DG method in the strong formulation translates into

$$\hat{\mathsf{M}} \frac{\partial \hat{\boldsymbol{q}}}{\partial t} + \hat{\mathsf{S}} \cdot \hat{\boldsymbol{f}} = \hat{\mathsf{F}}(\boldsymbol{n} \cdot (\hat{\boldsymbol{f}} - \hat{\boldsymbol{f}}^*)) \ \ . \tag{12}$$

The numerical flux $\hat{\boldsymbol{f}}^*$ is a function of $\boldsymbol{q}$ on the boundary of the local element $\boldsymbol{q}^-$ and the neighboring element $\boldsymbol{q}^+$. Three different fluxes are utilized here, a central flux, the Lax Friedrich flux and a flux for applying boundary conditions:

$$\begin{aligned} \hat{\boldsymbol{f}}^*_C(\boldsymbol{q}^-, \boldsymbol{q}^+) &= \frac{1}{2}(\boldsymbol{f}(\boldsymbol{q}^-) + \boldsymbol{f}(\boldsymbol{q}^+)) \ \ , \\ \hat{\boldsymbol{f}}^*_{LF}(\boldsymbol{q}^-, \boldsymbol{q}^+) &= \frac{1}{2}(\boldsymbol{f}(\boldsymbol{q}^-) + \boldsymbol{f}(\boldsymbol{q}^+)) - \frac{c}{2}(\boldsymbol{q}^+ - \boldsymbol{q}^-) \ \ , \\ \hat{\boldsymbol{f}}^*_{BC}(\boldsymbol{q}^-, \boldsymbol{q}^+) &= \hat{\boldsymbol{f}}(\boldsymbol{q}_{BC}) \ \ , \end{aligned}$$

where c is the maximum wave-speed, $c = \max |\lambda(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}})|$. If not otherwise stated, we will use the central flux for linear terms and the Lax-Friedrich flux for nonlinear terms, i.e., for the momentum equation in Navier-Stokes equation and in the levelset reinitialization equation to be discussed later.

The accuracy of the interpolation depends strongly on the distribution of nodes within the standard element. In (11) it is discussed how to distribute the nodes to ensure wellbehaved interpolations up till 16th order. The nodal distribution takes into account that a boundary integral is a vital part of the method, hence nodes are placed on the element boundary such that the boundary integral can be easily evaluated. The nodal distribution for elements of order 3, 5, 7 and 9 are illustrated in Fig. 1. Nodes on the edges are Legendre-Gauss-Lobatto quadrature nodes, hence the boundary integral operator $\hat{\mathsf{F}}$ can be expressed as

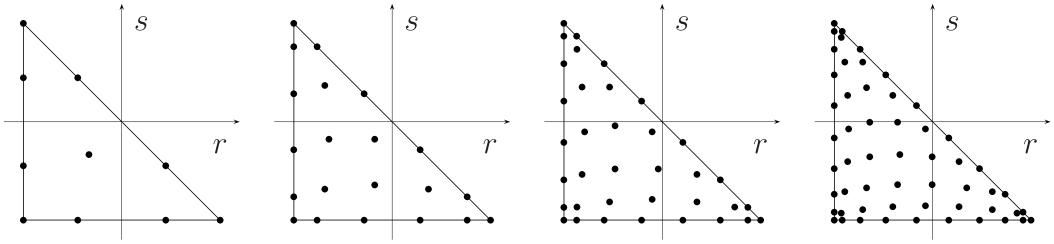

Fig. 1. Nodes in standard element of order 3, 5, 7 and 9

$$\hat{\mathsf{F}} = \mathsf{R}^T \ \hat{\mathsf{F}}^e \ \mathsf{R} \tag{13}$$

where R extracts the nodes on the element edges, and $\hat{\mathsf{F}}^e$ performs the 1D integration over the edges,

$$\hat{F}^e_{ij} = \int_{edge} l_j \; l_i \; \mathrm{d}\boldsymbol{x}, \tag{14}$$

with l_i being the 1D Lagrangian interpolation polynomial defined by the nodes on an edge. This formulation simplifies the evaluation of the boundary integral, only needing values on the nodes on the edge of the neighboring element, e.g.

$$\hat{\mathsf{F}}(\boldsymbol{q}^- + \boldsymbol{q}^+) = \mathsf{R}^T \; \hat{\mathsf{F}}^e \; (\mathsf{R}^+\boldsymbol{q}^+ + \mathsf{R}^-\boldsymbol{q}^-) \tag{15}$$

where R^+ and R^- extract matching nodes on the edge from the local and neighboring element respectively.

In (13) it is discussed in detail how to compute the operators, $\hat{\mathsf{M}}, \hat{\mathsf{S}}, \hat{\mathsf{F}}$, for the standard element efficiently and accurately.

3.1 Elliptic problems

When solving the incompressible Navier Stokes equations as subproblem a density weighted Poisson problem with Neumann boundary conditions,

$$\nabla \cdot \frac{1}{\rho} \nabla p = f \quad \text{in } \Omega \tag{16a}$$

$$n \cdot \nabla p = 0 \quad \text{on } \partial\Omega, \tag{16b}$$

where ρ is a function of space $\boldsymbol{x}$. This is not a problem for which the DG method was originally intended.

A characteristic of this pure Neumann problem is that it is singular and needs a consistent right hand side f to ensure solvability. When solving the incompressible Navier Stokes equations, the right hand side f is not in general consistent, giving the system no solution. Thus, to find a solution, we need to ensure that the right hand side is consistent.

Define $\mathfrak{L} = \nabla \cdot \frac{1}{\rho} \nabla$, let $R(\mathfrak{L})$ and $N(\mathfrak{L})$ denote the range and null space of $\mathfrak{L}$. The right hand side is consistent when $f \in R(\mathfrak{L})$. Also $R(\mathfrak{L}) = N(\mathfrak{L}^T)^\perp$. If $\mathfrak{L}$ is symmetric, then $R(\mathfrak{L}) = N(\mathfrak{L})^\perp$. The null space of $\mathfrak{L}$ is known to be the constant vector, $N(\mathfrak{L}) = \mathbf{1}$. Hence, in the symmetric case we remove from f the part existing in $N(\mathfrak{L})$

$$f^* = f - \mathbf{1} \; (\mathbf{1} \cdot f)/(\mathbf{1} \cdot \mathbf{1}),$$

to ensure solvability.

The standard procedure in DG methods (4; 1) when discretizing higher order operators such as the Laplace operator is to split the system into two first order equations

$$\nabla \cdot \boldsymbol{q} = f, \tag{17a}$$
$$\nabla p = \rho\, \boldsymbol{q}. \tag{17b}$$

If we approximate the first in the weak DG sense and the second in the strong sense, using the notation $(\cdot,\cdot)_{\mathsf{D}}$ and $(\cdot,\cdot)_{\partial\mathsf{D}}$ for the integrals we obtain

$$-(\boldsymbol{q}, \nabla L_j)_{\mathsf{D}} = (f, L_j)_{\mathsf{D}} - (\boldsymbol{n}\cdot\boldsymbol{q}^*, L_j)_{\partial\mathsf{D}} \tag{18a}$$
$$(\nabla p, L_j)_{\mathsf{D}} = (\rho\, \boldsymbol{q}, L_j)_{\mathsf{D}} + (\boldsymbol{n}(p - p^*), L_j)_{\partial\mathsf{D}} \tag{18b}$$

where p^* is the Dirichlet conditions and $\boldsymbol{n}\cdot\boldsymbol{q}^*$ is the Neumann conditions. Since we have only Neumann condition on the global boundary, the $(p-p^*)$-part disappears by taking $p^* = p^-$. Using the discrete operators on the standard element, we get

$$-\hat{\mathsf{S}}^T \boldsymbol{q} = \hat{\mathsf{M}}\boldsymbol{f} - \hat{\mathsf{F}}(\boldsymbol{n}\cdot\boldsymbol{q}^*) \tag{19a}$$
$$\hat{\mathsf{S}}\boldsymbol{p} = \hat{\mathsf{M}}(\boldsymbol{\rho}\, \boldsymbol{q}) + \hat{\mathsf{F}}(\boldsymbol{n}(\boldsymbol{p} - \boldsymbol{p}^*)) \tag{19b}$$

Consider the simplified special case where ρ is constant and 1, and consider only one element, hence having Neumann conditions on all boundaries so the $\hat{\mathsf{F}}(\boldsymbol{n}(\boldsymbol{p}-\boldsymbol{p}^*))$ term drops out, then combining Eqs. (19a) and (19b) yields

$$-\hat{\mathsf{S}}^T\ \hat{\mathsf{M}}^{-1}\ \hat{\mathsf{S}}\,\boldsymbol{p} = \hat{\mathsf{M}}\boldsymbol{f} - \hat{\mathsf{F}}(\boldsymbol{n}\cdot\boldsymbol{q}^*). \tag{20}$$

Since $\hat{\mathsf{M}}$ is symmetric, then so is inverse, hence $\hat{\mathsf{S}}^T\ \hat{\mathsf{M}}^{-1}\ \hat{\mathsf{S}}$ forms a symmetric system. In a multi-element setup, we require weak continuity between the elements, hence keeping the $\hat{\mathsf{F}}(\boldsymbol{n}(\boldsymbol{p}-\boldsymbol{p}^*))$, but the resulting system will still be symmetric as long as the numerical fluxes $\boldsymbol{p}^*$ and $\boldsymbol{q}^*$ are symmetric over element boundaries, i.e. the same flux function is used in neighboring elements. In this work we use central fluxes for $\boldsymbol{p}^*$ and $\boldsymbol{q}^*$, i.e.

$$\boldsymbol{p}^*(\boldsymbol{p}^-, \boldsymbol{p}^+) = \frac{1}{2}(\boldsymbol{p}^- + \boldsymbol{p}^+)\ ,$$

and similar for $\boldsymbol{q}^*$. Internal penalty fluxes (1) are equally efficient, although often leading to a slightly worse conditioning.

If solving this Poisson equation with Dirichlet or periodic boundary conditions, a stabilizing term controlling the null space is needed. In this case we use a standard penalty penalization technique (1).

We now present the scheme used to solve the incompressible Navier Stokes. Let us first introduce the notation

$$(f,g)_{\mathsf{D}^k} = \int_{\mathsf{D}^k} f\ g\ \mathrm{d}\boldsymbol{x}\ , \qquad (f,g)_{\partial\mathsf{D}^k} = \int_{\partial\mathsf{D}^k\setminus\partial\Omega} f\ g\ \mathrm{d}\boldsymbol{x}\ ,$$
$$(f,g)_{\partial\Omega} = \int_{\partial\mathsf{D}^k\cap\partial\Omega} f\ g\ \mathrm{d}\boldsymbol{x}\ ,$$

thereby separating inner element boundaries from the global domain boundaries. Rewriting Eq. (6), introducing the variable g, and leaving out the surface tension for now, we obtain

$$\frac{\partial u_i}{\partial t} + \nabla\cdot(\boldsymbol{u}u_i) = -\frac{1}{\rho}\nabla_i p + \frac{1}{Re}\frac{\mu}{\rho}\nabla\cdot\boldsymbol{g}_i\ , \tag{21a}$$
$$\nabla u_i = \boldsymbol{g}_i\ . \tag{21b}$$

Define the following:

$$N^k(u_i) = (\nabla\cdot(\boldsymbol{u}u_i), L_j)_{\mathsf{D}^k} - (\boldsymbol{n}\cdot(\boldsymbol{u}u_i - (\boldsymbol{u}u_i)^*), L_j)_{\partial\mathsf{D}^k}\ , \tag{22a}$$
$$L^k(\boldsymbol{g}) = (\nabla\cdot\boldsymbol{g}, L_j)_{\mathsf{D}^k} - (\boldsymbol{n}\cdot(\boldsymbol{g}-\boldsymbol{g}^*), L_j)_{\partial\mathsf{D}^k}\ , \tag{22b}$$
$$\boldsymbol{P}^k(p) = (\nabla p, L_j)_{\mathsf{D}^k} - (\boldsymbol{n}(p-p^*), L_j)_{\partial\mathsf{D}^k}\ , \tag{22c}$$

to obtain the DG formulation

$$\left(\frac{\partial\boldsymbol{u}}{\partial t}, L_j\right)_{\mathsf{D}^k} + N^k(\boldsymbol{u}) = -\boldsymbol{P}^k(p) + \frac{1}{Re}\frac{\nu}{\rho}L^k(\boldsymbol{g}) - (\boldsymbol{u}-\boldsymbol{u}^*, L_j)_{\partial\Omega} \tag{23a}$$
$$(\nabla u_i, L_j)_{\mathsf{D}^k} = (\boldsymbol{g}(u_i), L_j)_{\mathsf{D}^k} + (\boldsymbol{n}(u_i - u_i^*), L_j)_{\partial\mathsf{D}^k} \tag{23b}$$

where the $(\cdot,\cdot)_{\partial\Omega}$ accounts for the global boundary conditions. Defining discrete versions of the above

$$\mathsf{N}^k(\boldsymbol{u}_i) = \mathsf{S}^k\ (\boldsymbol{u}\boldsymbol{u}_i) - \mathsf{F}^k(\boldsymbol{n}\cdot(\boldsymbol{u}\boldsymbol{u}_i - (\boldsymbol{u}\boldsymbol{u}_i)^*)) \tag{24a}$$
$$\mathsf{L}^k(\boldsymbol{g}) = \mathsf{S}^k\ \boldsymbol{g} - \mathsf{F}^k(\boldsymbol{n}\cdot(\boldsymbol{g}-\boldsymbol{g}^*)) \tag{24b}$$
$$\mathsf{P}^k(\boldsymbol{p}) = \mathsf{S}^k\ \boldsymbol{p} - \mathsf{F}^k(\boldsymbol{n}(\boldsymbol{p}-\boldsymbol{p}^*)) \tag{24c}$$

we end up with the following locally defined nodal scheme

$$\mathsf{M}^k\frac{\partial\boldsymbol{u}}{\partial t} + \mathsf{N}^k(\boldsymbol{u}) = -\mathsf{P}^k(\boldsymbol{p}) + \frac{1}{Re}\frac{\nu}{\rho}\mathsf{L}^k(\boldsymbol{g}) - \mathsf{F}^k_{\partial\Omega}(\boldsymbol{u}-\boldsymbol{u}^*_{\partial\Omega}), \tag{25a}$$
$$\mathsf{S}^k\boldsymbol{u} = \mathsf{M}^k\boldsymbol{g} + \mathsf{F}^k(\boldsymbol{n}(\boldsymbol{u}_i - \boldsymbol{u}_i^*)), \tag{25b}$$

where again $\boldsymbol{g}$ can be eliminated locally. For the fluxes in Eq. (24) and for the flux in Eq. (24a) we use a Lax-Friedrich flux, while for the remaining we use a central flux.

4 The Temporal Scheme

The incompressible Navier Stokes (INS) equations are a combination of nonlinear advection, linear diffusion, and an algebraic constraint. On top of the INS, we shall need to solve the level set equation, Eq. (5).

An often used approach for solving the INS is an explicit-implicit splitting, treating the diffusion part implicitly, the nonlinear part explicitly, and furthermore decoupling the calculation of the pressure from the velocities, thus introducing the potential for time splitting errors. This is equally true when solving the INS and the level set equation as we would like to decouple the evaluation of the level set from the velocities and the pressure, hence we have to be careful not to produce lower order time splitting errors.

4.1 Velocity Projection

We will now return to the incompressible Navier Stokes equations, and illustrate the standard velocity projection technique using a semi-implicit Euler approximation. We consider the following

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\frac{1}{\rho}\nabla p + \mathsf{N}(\boldsymbol{u}) + \mathsf{L}(\boldsymbol{u}) + \boldsymbol{f}, \quad \boldsymbol{u}|_{\partial\Omega} = \boldsymbol{u}_b \tag{26a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{26b}$$

with $\mathsf{N}(\boldsymbol{u})$ being the nonlinear advection and $\mathsf{L}(\boldsymbol{u})$ the diffusion. The purpose of the pressure variable is to assure no divergence in the new velocities. Consider a semi-implicit approximation of the time derivative and remove the implicit diffusive part

$$\boldsymbol{u}_{n+1} = \bar{\boldsymbol{u}}_n + h\Big(-\frac{1}{\rho}\nabla p_{n+1} + \mathsf{N}(\bar{\boldsymbol{u}}_n) + \boldsymbol{f}_{n+1}\Big), \tag{27a}$$

$$0 = \nabla \cdot \boldsymbol{u}_{n+1}, \tag{27b}$$

yields an equation for the pressure

$$\nabla \cdot \frac{1}{\rho}\nabla p_{n+1} = \nabla \cdot \Big(\frac{1}{h}\bar{\boldsymbol{u}}_n + \mathsf{N}(\bar{\boldsymbol{u}}_n) + \boldsymbol{f}_{n+1}\Big). \tag{28}$$

This is a Poisson problem, and Neumann boundary conditions can be found from the original equation

$$\boldsymbol{n} \cdot \nabla p = \boldsymbol{n} \cdot \rho \left(-\frac{\partial \boldsymbol{u}}{\partial t} + \mathsf{N}(\boldsymbol{u}) + \mathsf{L}(\boldsymbol{u}) + \boldsymbol{f}\right), \quad \text{on } \partial\Omega \tag{29}$$

which must be approximated to the same order as the time derivative. For a higher order method, this implies extrapolation of the non-linear as well as the linear term (16; 9) Having the pressure, we can insert it into Eq. (27a) and as the last step we solve for the new velocities.

$$\bar{\boldsymbol{u}}_{n+1} - h\mathsf{L}(\bar{\boldsymbol{u}}_{n+1}) = \boldsymbol{u}_{n+1} \tag{30}$$

Velocity projection is often used with an Adams-Moulton or BDF-type approximation of the time derivative and extrapolation of the nonlinear part, hence a k'th order BDF(k) method:

$$\frac{1}{h}\sum_{i=-1}^{k-1} \alpha_i \boldsymbol{u}_{n-i} = -\nabla p_{n+1} + \sum_{i=0}^{k} \beta_i \mathsf{N}(\boldsymbol{u}_{n-i}) + \mathsf{L}(\boldsymbol{u}_{n+1}) + \boldsymbol{f}_{n+1}, \tag{31}$$

where the α_i's are the BDF coefficients and the β_i's are extrapolation coefficients. This formulation is 1st order using a BDF(1) method, 2nd order using a BDF(2), and gives $O((\Delta t)^{5/2})$ order using a BDF(3). For a thorough analysis, see (16; 9; 8)

The velocity projection approach may yield $O((\Delta t)^{5/2})$, but being able to discretize in space to basically any order, on would like to mimic this in time. Furthermore, the BDF method in its original form is fully implicit, making it difficult to include more equations, e.g. the level set equation, in the solution process without increasing the amount of work significantly, or by extensive use of extrapolation like for the non-linear term above. Finally, how to apply Neumann type of boundary conditions for the velocities and retain the order is remains unknown (9; 8).

4.2 SISDC

Minion (17) presents a Semi Implicit Spectral Deferred Correction (SISDC) method to time integrate the incompressible Navier Stokes to arbitrary order of accuracy. It is based on low order time integration methods, which are corrected iteratively. The time splitting errors of the low order methods are corrected as any other part of the error, hence the final method does not suffer from order reduction due to the splitting. We can split and decouple terms as needed – an attractive and flexible feature. Hence we should be able to include the level set variable in a simple and computationally inexpensive way, and still retain high order.

In (17) an alternative formulation using $\boldsymbol{u} = \boldsymbol{m} + \nabla\chi$ is proposed, thereby eliminating the pressure variable. This, however, makes it difficult to apply velocity boundary conditions and to obtain and use the pressure, which is required in the free surface flow formulation. Here we will present a variant, which uses the velocity and pressure variables directly.

4.2.1 SISDC for ODEs

Let us first introduce the SISDC for ODEs for which we shall follow the original development in (6). Assume we want to solve the ODE

$$u'(t) = f\Big(u(t), t\Big), \qquad u(a) = u_a, \qquad t \in [a, b], \tag{32}$$

Before we continue, some definitions are appropriate. Let t_i be a grid in the interval, $a = t_0 < t_1 < ... < t_M = b$, and let $\boldsymbol{v} = (v_0, ..., v_M)$ be the values of $v(t)$ at the nodes, $v_i = v(t_i)$. Define the Lagrange interpolant as

$$\tilde{v}(t) = L_M(\boldsymbol{v}, t) = \sum_{i=0}^{M} v_i \ \pi_i(t), \tag{33}$$

where the fundamental polynomials $\pi_i(t)$ are defined by

$$\pi_i(t) = \prod_{\substack{j=0 \\ j \neq i}}^{M} \frac{t - t_j}{t_i - t_j}. \tag{34}$$

The Lagrange interpolant, Eq. (33), is a continuous function, hence we can differentiate and integrate it.

We define the integration operator I_j^{j+1} as

$$I_j^{j+1} \boldsymbol{v} = \int_{t_j}^{t_{j+1}} \tilde{v}(\tau) \ \mathrm{d}\tau, \quad j = 0, ..., M-1 \tag{35}$$

The grid, $\{t_i\}$, is typically chosen as teh nodes of a Gauss quadrature formula to enable high accuracy integration.

The spectral deferred correction method is based on the Picard solution of the ODE, Eq. (32),

$$u(t) = \int_a^t f\Big(u(\tau), \tau\Big) \ \mathrm{d}\tau + u_a. \tag{36}$$

Given an approximation of the solution $v(t)$, let the error be $\delta(t)$, such that $u(t) = v(t) + \delta(t)$. Substituting this into Eq. (36) gives

$$v(t) + \delta(t) = \int_a^t f\Big(v(\tau) + \delta(\tau), \tau\Big) \ \mathrm{d}\tau + u_a. \tag{37}$$

A measure of the error can be found by considering the error equation. Inserting $v(t)$ for $u(t)$ in Eq. (36) we have

$$\epsilon(t) = \int_a^t f\Big(v(\tau), \tau\Big)\ \mathrm{d}\tau + u_a - v(t). \tag{38}$$

Subtracting Eq. (38) from Eq. (37) gives a correction equation

$$\delta(t) = \int_a^t f\Big(v(\tau) + \delta(\tau), \tau\Big) - f\Big(v(\tau), \tau\Big)\ \mathrm{d}\tau + \epsilon(t). \tag{39}$$

which is on the same form as Eq. (36). We can solve the correction equation as we solved the original ODE, and the result can be used for updating the solution. We simply need to specify how to evaluate the integrals.

Consider Eq. (36) for the i'th interval, $h_i = t_{i+1} - t_i$. Using an explicit Euler approximation for the integral we recover the approximation

$$v_{i+1} = h_i f(v_i, t_i) + u_i. \tag{40}$$

Similarly, using an explicit Euler approximation for Eq. (39) gives

$$\delta_{i+1} = \delta_i + h_i\left(f(v_i + \delta_i, t_i) - f(v_i, t_i)\right) + \epsilon_{i+1} - \epsilon_i. \tag{41}$$

Subtract the residual, Eq. (38), at time t_i from that at time t_{i+1}, to get

$$\epsilon_{i+1} - \epsilon_i = \int_{t_i}^{t_{i+1}} f(v(\tau), \tau)\ \mathrm{d}\tau - v_{i+1} + v_i. \tag{42}$$

Combined with Eq. (41) yields a direct equation for the updated solution $v^*_{i+1} = v_{i+1} + \delta_{i+1}$

$$v^*_{i+1} = v^*_i + h_i\left(f(v^*_i, t_i) - f(v_i, t_i)\right) + \int_{t_i}^{t_{i+1}} f(v(\tau), \tau)\ \mathrm{d}\tau. \tag{43}$$

To complete the correction procedure, we need to specify how to calculate the integral in Eq. (43). This is where the spectral integration is needed. Let $f_i = f(v_i, t_i)$, and let $\tilde{f}$ be the Lagrange interpolant of the f_i's. We can now integrate using Eq. (35)

$$\int_{t_i}^{t_{i+1}} f(v(\tau), \tau)\ \mathrm{d}\tau \approx \int_{t_i}^{t_{i+1}} \tilde{f}\ \mathrm{d}\tau = I_i^{i+1}\ \boldsymbol{f}, \tag{44}$$

which completes the scheme. A similar expression exists for the implicit Euler approximation

$$v^*_{i+1} = v^*_i + h_i\Big(f(v^*_{i+1}, t_{i+1}) - f(v_{i+1}, t_{i+1})\Big) + I_i^{i+1}\ \boldsymbol{f}. \tag{45}$$

and a right hand side f can be split into different parts treated explicitly and implicitly in a straight forward manner.

One can repeat the process with another correction step, using the new v_i^* as v_i, which can be shown to raise the order of the approximation by one. This can continue as long as the integral, Eq. (44), is evaluated sufficiently accurately. Using $M+1$ points in the Lagrange interpolant provides an $O(h^{M+2})$ accuracy for the integral, and an $O(h^{M+1})$ global accuracy for the solution. A complete error analysis is provided in (6).

4.2.2 SISDC and INS

Consider the integral form of the momentum equation:

$$\boldsymbol{u}(t_{i+1}, x) = \boldsymbol{u}(t_i, x) + \int_{t_i}^{t_{i+1}} \left(-\nabla p + \mathsf{N}(\boldsymbol{u}) + \mathsf{L}(\boldsymbol{u}) + \boldsymbol{f}\right)\, \mathrm{d}t \tag{46}$$

As we want to remove the pressure from the spectral integration, we evaluate the pressure part of the integral using the mean value theorem,

$$\boldsymbol{u}(t_{i+1}, x) = \boldsymbol{u}(t_i, x) - \Delta t_i \nabla p(\xi_i) + \int_{t_i}^{t_{i+1}} \left(\mathsf{N}(\boldsymbol{u}) + \mathsf{L}(\boldsymbol{u}) + \boldsymbol{f}\right)\, \mathrm{d}t \tag{47}$$

where ξ_i is unknown, $t_i < \xi_i < t_{i+1}$. This allows the construction of the SISDC method, treating the linear term implicitly and the nonlinear explicitly, on the form

$$\begin{aligned} \boldsymbol{u}_{i+1}^* = \boldsymbol{u}_i^* + \Delta t \Big[&-\nabla p_{\xi_i}^* + \mathsf{N}(\boldsymbol{u}_i^*) - \mathsf{N}(\boldsymbol{u}_i) + \\ &\mathsf{L}(\boldsymbol{u}_{i+1}^*) - \mathsf{L}(\boldsymbol{u}_{i+1})\Big] + I_i^{i+1} F(\boldsymbol{u}), \qquad (48a) \\ 0 = \nabla \cdot \boldsymbol{u}_{i+1}^*. \qquad (48b) \end{aligned}$$

where $F(\boldsymbol{u})$ is the integrand of Eq. (47). This formulation can be combined with either a pressure projection or velocity projection approach.

4.2.3 Test of SISDC on INS

We shall test the velocity projection SISDC method on the following traveling wave solution of the incompressible Navier Stokes equations

$$u = \frac{3}{4} + \frac{1}{4}\cos(2\pi(x-t))\sin(2\pi(y-t))e^{-\alpha 8\pi^2 t}, \tag{49a}$$

$$v = \frac{3}{4} - \frac{1}{4}\sin(2\pi(x-t))\cos(2\pi(y-t))e^{-\alpha 8\pi^2 t}, \tag{49b}$$

$$p = \frac{1}{64}\Big(\cos(4\pi(x-t)) + \cos(4\pi(y-t))\Big)e^{-\alpha 16\pi^2 t}. \tag{49c}$$

Here α determines the decay of the waves, e.g., $\alpha = 0$ implies a traveling wave. The right hand side forcing $\boldsymbol{f}$ in the momentum equation is found by inserting the solution into the Navier Stokes equation. The solution is periodic, is set up in a $[0;1] \times [0;1]$ box and the solution is calculated up till $T = 4$.

The problem is solved on a grid having 226 elements of order 4. Figure 2 shows error plots for the test problem set up using periodic boundary conditions. The velocities all show more or less optimal order. The 4th order method seems to display some kind of order reduction, reducing to 3rd order for small time steps. However the 5th order method does not show alike. The 5th order method hits the limit of the spatial discretization at around 10^{-6} and does not improve beyond this point.

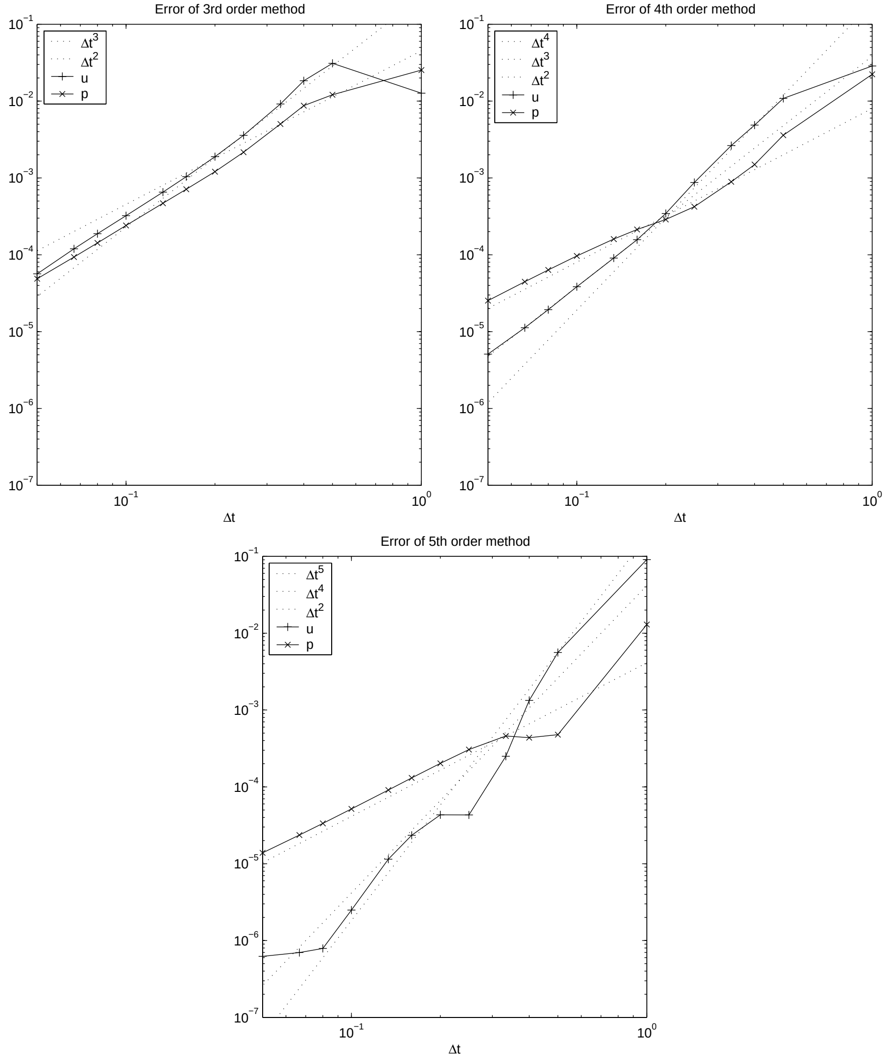

Fig. 2. Order Results 3rd to 5th order

The pressure shows only second order in all plots. This is casued by the fact that we do not know the pressure time, $t_{i+\xi}$, hence the pressure is compared with the exact solution at time $t_{i+1/2}$, which is only second order accurate. However since the velocities are high order, we would expect the pressure to be of equally high order.

4.3 Boundary Conditions in SISDC

So far we have not discussed the impact of boundary conditions. While boundary conditions for the velocities usually are available, conditions for the pressure are not and these must be approximated from the flow. In (16) it is argued that the Neumann condition for the pressure needs to be approximated to the same order in time as the time stepping method to avoid order reduction.

Applying velocity conditions to the SISDC method is not straight forward. Imposing prescribed boundary conditions on the implicit step even for simple problems like the heat equations will cause an order reduction similar to that of Runge Kutta methods, i.e., it is caused by the low stage-order, which is determined by the order of the internal scheme in the SISDC scheme. Using a high-order internal scheme in the SISDC may help alleviate this problem, although no analysis is available. Further discussions along this line can be found in (17).

Furthermore, it is not known how to approximate the pressure Neumann condition to sufficient order in a useful way. So far we have only been able to obtain 1st order accuracy for the pressure, limiting the order of accuracy for the velocities to 2nd order for the SISDC method with complex boundary conditions.

4.4 Time Stepping INS and the Level Set Equation

The level set equation, as the nonlinear part of the Navier-Stokes equation, has no need for implicit time stepping. Hence we seek to include it in a way which requires as little extra work as possible, e.g., in the form

$$\begin{aligned} \boldsymbol{u}_{i+1} &= \boldsymbol{F}(\boldsymbol{u}_{i+1},\, \boldsymbol{u}_i,\, p_{i+1},\, \phi_i), \\ \nabla \cdot \boldsymbol{u}_{i+1} &= 0, \\ \phi_{i+1} &= \boldsymbol{G}(\boldsymbol{u}_{i+1},\, \boldsymbol{u}_i,\, \phi_i), \end{aligned}$$

where subscript $i + 1$ refers to implicit and i to an explicit dependence. We solve for the velocities and the pressure as before, with the only difference that viscosity and density may vary in time. Afterwards we solve the level set equation, using the newly calculated velocity $\boldsymbol{u}_{i+1}$.

The above formulation fits directly into the SISDC projection method, using explicit or implicit approximations as appropriate. Unfortunately, as discussed above, it remains unknown how to apply pressure boundary conditions without impacting the accuracy adversely.

For problems requiring such types of boundary conditions, e.g., nonperiodic problems, we may consider other methods more appropriate and we shall use the classic BDF-type scheme described previously.

In the equation for the velocities, we need the level set at time t_{i+1}, hence we need to extrapolate, which we do with an explicit Adams method of the same order as the BDF method. When the pressure and the velocities are calculated, we will calculate the level set explicitly based on a BDF scheme

$$\frac{1}{h}\sum_{i=-1}^{k-1}\alpha_i\phi_{n-i} = -\boldsymbol{u}_{n+1}\cdot\nabla\bar{\phi}_{n+1} \tag{50}$$

where $\bar{\phi}_{n+1}$ is the extrapolated level set, and $\boldsymbol{u}_{n+1}$ the newly calculated velocity. Applying the BDF scheme for the level set, and simply using the value from the Adams method, minimizes the time splitting errors.

5 The Level set Modeling of the Free Surface

In the continuous case in Sec. 2 the choice of the level set function is arbitrary as long as it is differentiable and fulfills Eq.(3).

We will choose ϕ to be the signed distance to the interface. To avoid handling discontinuities in e.g. $\mu(\phi)$ and $\rho(\phi)$, the interface is given a thickness of size ϵ, i.e., we use a smooth Heaviside function

$$H(\phi,\epsilon) = \begin{cases} 0 & \phi < -\epsilon \\ \frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi}\sin\left(\frac{\pi\phi}{\epsilon}\right) & -\epsilon \le \phi \le \epsilon \\ 1 & \phi > \epsilon \end{cases} \tag{51}$$

This is only one of many possible definitions of a smooth Heaviside function (24). Using a high order method to enable high precision, it is important that the function is as smooth as possible. The above Heaviside function has continuous first and second derivative, while the third is discontinuous at $\phi = \pm\epsilon$. It is possible to design higher order polynomials or combinations of polynomials and trigonometric functions which are as smooth as required. However, the coefficients and the size of the derivatives grow larger with the degree of smoothness, which also makes the differentiation less accurate. When a very smooth function is needed, and the thickness is less important,

$$H_s(\phi, \epsilon) = \frac{1}{2} + \frac{\phi}{2\sqrt{\phi^2 + \epsilon^2}} \tag{52}$$

can be used.

The interface thickness ϵ should be chosen as small as possible for accuracy but big enough to stabilize the system. A typical choice is ϵ of the local size of an element. Note that ϵ does not have to be a global constant, but may vary throughout the domain.

The smooth Heaviside function only depends on the level set for $|\phi| < \epsilon$. In order to update the level set in Eq.(5) with good precision within the distance ϵ from the interface, the level set need to be accurate in a distance slightly larger, say 1.5ϵ. Hence the level set need only to be a signed distance function within 1.5ϵ of the interface, as elsewhere only the sign is needed. This lowers the computational cost of handling the level set, since we need only consider the level set in elements within 1.5ϵ of the interface, e.g., the gray elements in Fig. 3.

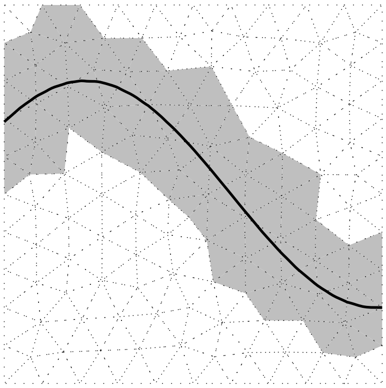

Fig. 3. Elements within 1.5ϵ of the interface

5.1 Reinitialization of the level set

A fluid particle at a certain distance from the interface will, as time progresses, seldom remain at that distance. Hence, even if the initial level set is a signed distance function, it will not remain so and to retain the level set as a signed distance function, it must be reinitialized. If the level set is not reinitialized, areas of small and large gradients will form, changing the thickness of the interface and eventually degrading the accuracy and stability of the method.

The reinitialization procedure must not move the interface, i.e., the zero contour must remain unchanged, and ϕ must be found such that $|\nabla\phi| = 1$. In

(23) the following reinitializing equation was proposed

$$\phi_\tau + \operatorname{sign}(\phi_0)(|\nabla\phi| - 1) = 0, \tag{53}$$

where ϕ_0 is the initial level set to be reinitialized and τ is an artificial time. Evolving this equation to steady state will produce a signed distance function ϕ. The characteristic lines of the system are normals to the interface, and the characteristic speed is 1, hence the level set will be reinitialized from the interface along the interface normals. As we seek the level set reinitialized in a distance 1.5ϵ from the interface, it is sufficient to run the reinitialization to time $\tau = 1.5\epsilon$. However, if the initial level set is already close to a signed distance function, fewer iterations are often sufficient.

The reinitialization equation, Eq. (53), is a Hamilton-Jacobi equation and the signed distance function is a C^0 function. Figure 4 is a 1D example showing how two surface points result in a level set function with a peak. A standard high-order method does not handle such discontinuities easily, especially when differentiating as in Eq.(53), and the method may turn inaccurate, difficult to solve, or even unstable. The use of WENO methods has been proposed (26) to overcome such problems but remains a challenge at general unstructured grids.

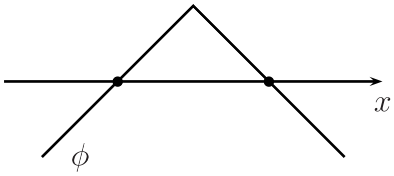

Fig. 4. Two surface points produces a level set function with a sharp peak

We will solve a slightly modified reinitialization equation, which will only make the level set a distance function within $|\phi| < \epsilon$. Outside we will make it approach the value $\frac{5}{4}\epsilon$:

$$\frac{\partial\phi}{\partial t} = \operatorname{sign}(\phi_0,\epsilon)(l(\phi,\epsilon) - |\nabla\phi|) + \nu_w\frac{\partial^2\phi}{\partial w^2} + \gamma(1 - l(\phi,\epsilon))(\frac{5}{4}\epsilon\ \operatorname{sign}(\phi) - \phi) \tag{54}$$

The first term is like the original reinitialization equation apart from the function

$$l(\phi,\epsilon) = \begin{cases} 1, & |\phi| < \epsilon, \\ 0, & |\phi| > \frac{5}{4}\epsilon, \\ 1 - 4\frac{|\phi|-\epsilon}{\epsilon}, & \text{in between,} \end{cases}$$

which, for $|\phi| < \epsilon$, is like the original, but for $|\phi| > \frac{5}{4}\epsilon$ forces the level set to be flat with a smooth transition in between. Figure 5 shows how this modification

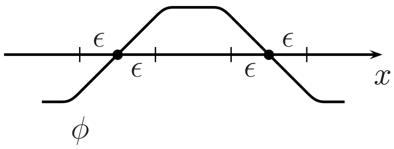

Fig. 5. Peak is removed

will reinitialize the example from Figure 4, and how peaks outside $|\phi| > \frac{5}{4}\epsilon$ are removed.

The second term is diffusion along the characteristics $\boldsymbol{w} = \frac{\nabla\phi}{|\nabla\phi|}$, ν_w being a diffusion constant. For a distance function the gradient along a characteristic is constant almost everywhere, $\boldsymbol{w} \cdot \nabla\phi = 1$, hence the term has no effect there. However, where peaks exists in the level set function, this term will smoothen the peaks and make the level set infinitely smooth. This effectively controls peaks within $|\phi| < \frac{5}{4}\epsilon$.

The last term is forcing the level set outside $|\phi| > \frac{5}{4}\epsilon$ toward the value $\frac{5}{4}\epsilon$. This is useful if ϵ depends on $\boldsymbol{x}$, to make sure the level set value far away from the interface is close to $\frac{5}{4}\epsilon$. For constant ϵ it may be omitted.

5.1.1 Boundary conditions for the reinitialization

Boundary conditions for the original reinitialization equation, Eq. (53), is needed when the interface intersects the domain boundary $\partial\Omega$. The top dotted part of the right boundary in Fig. 6 intersects characteristics of the level set, while the bottom part does not, i.e., boundary conditions are needed for the bottom part. In the modified reinitialization equation, Eq. (54), we have introduced diffusion, implying also that the top part requires boundary conditions. The boundary conditions can be found by reinitializing the boundary using

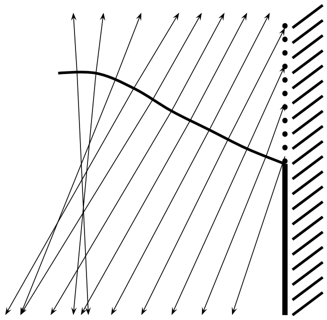

Fig. 6.

$$\left.\frac{\partial\phi}{\partial\tau}\right|_{\boldsymbol{x}\in\partial\Omega} = \operatorname{sign}(\phi_0, \epsilon)(\beta\ l(\phi, \epsilon) - |\nabla\phi|)|_{\boldsymbol{x}\in\partial\Omega}$$

where $\beta = |\nabla\phi - \boldsymbol{n}(\boldsymbol{n}\cdot\nabla\phi)|$ is the slope of the level set on the boundary.

When restricting attention to a band of elements within a certain distance of the interface, also boundary conditions on the outermost elements must be given. Using the modified reinitialization equation, the value of the level set far away from the interface is $\frac{5}{4}\epsilon$, and this may be used as boundary value on the outermost element boundaries.

5.1.2 Discretization of the Reinitialization Equation

The discretization of Eq. (54) follows the previous discussion directly. Rewrite the equation

$$\begin{aligned}
\frac{\partial\phi}{\partial t} &= \operatorname{sign}(\phi_0,\epsilon)(l(\phi,\epsilon) - |\boldsymbol{h}|) + \nu_w \boldsymbol{w}\cdot\boldsymbol{g} + f \ , && (55a)\\
\nabla\phi &= \boldsymbol{h} \ , && (55b)\\
\nabla(\boldsymbol{w}\cdot\boldsymbol{h}) &= \boldsymbol{g} \ , && (55c)
\end{aligned}$$

We discretize this as

$$\begin{aligned}
\left(\frac{\partial\phi}{\partial t}, L_j\right)_{\mathsf{D}^k} &= (\operatorname{sign}(l - |\boldsymbol{h}|) + \nu_w \boldsymbol{w}\cdot\boldsymbol{g} + f, L_j)_{\mathsf{D}^k} \ , && (56a)\\
(\nabla\phi, L_j)_{\mathsf{D}^k} &= (\boldsymbol{h}, L_j)_{\mathsf{D}^k} + (\boldsymbol{n}(\phi - \phi^*), L_j)_{\partial\mathsf{D}^k} \ , && (56b)\\
(\nabla(\boldsymbol{w}\cdot\boldsymbol{h}), L_j)_{\mathsf{D}^k} &= (\boldsymbol{g}, L_j)_{\mathsf{D}^k} + (\boldsymbol{n}\cdot\boldsymbol{w}(\boldsymbol{h} - \boldsymbol{h}^*), L_j)_{\mathsf{D}^k} && (56c)
\end{aligned}$$

or in discrete operators

$$\begin{aligned}
\frac{\partial\boldsymbol{\phi}}{\partial t} &= \boldsymbol{s}(\boldsymbol{l} - |\boldsymbol{h}|) + \nu_w \boldsymbol{w}\cdot\boldsymbol{g} + \boldsymbol{f} \ , && (57a)\\
\mathsf{S}^k\boldsymbol{\phi} &= \mathsf{M}^k\boldsymbol{h} + \mathsf{F}^k(\boldsymbol{n}(\boldsymbol{\phi} - \boldsymbol{\phi}^*)), && (57b)\\
\mathsf{S}^k(\boldsymbol{w}\cdot\boldsymbol{h}) &= \boldsymbol{M}^k\boldsymbol{g} + \mathsf{M}^k(\boldsymbol{n}\cdot\boldsymbol{w}(\boldsymbol{h} - \boldsymbol{h}^*)) && (57c)
\end{aligned}$$

If global boundary conditions are needed, a $\mathsf{F}^k_{\partial\Omega}(\boldsymbol{\phi} - \boldsymbol{\phi}^*_{\partial\Omega})$ term can be included in Eq. (57a).

5.1.3 Filtering the reinitialization process

It is necessary to use a filtering technique for stabilizing the reinitialization process and retain high order accuracy. Even though the modified reinitialization equation makes the system much easier to handle, it is still necessary to filter. What we basically use is a low-pass exponential filter, the precise description can be found in (7).

Consider the filtered result $\mathsf{T}\boldsymbol{u}$, where $\mathsf{T} = \mathsf{I} - \mathsf{H}$, I being the identity operator and H pick out the high modes. Applying an explicit method to solve Eq. (53)

and the exponential filter this corresponds to

$$\tilde{\phi}_{i+1} = \phi_i - \Delta t \ \mathrm{sign}(\phi_0)(|\nabla \phi_i| - 1) \tag{58a}$$

$$\phi_{i+1} = \Big(\mathsf{I} - \mathsf{H}\Big)\tilde{\phi}_{i+1} \tag{58b}$$

Iterating the original reinitialization equation, Eq. (53), to steady state implies eventually $|\nabla \phi| - 1 = 0$. When adding the filter, the steady state solution will emerge when the filter and the forcing balances, $\Delta t \ \mathrm{sign}(\phi_0)(|\nabla \phi_i| - 1) \approx \mathsf{H}\phi_i$. The errors due to filtering becomes a part of the solution. Filtering of steady state problems must be designed with care. We have designed the reinitialization method such that filtering only takes place when necessary, and then only as little as possible, and preferably never in elements containing the interface. This is applied adaptively, by calculating a measure of the "need for filtering" and using this measure to determine the degree of filtering in each element. We have used a very simple measure, $\eta = ||\nabla \phi_i| - 1|$ in each element, and we apply more filtering in elements with large η.

Kanevsky et al. (15) have recently developed time-consistent filters especially designed for time-integrating solutions of DG methods to steady state. Such filters would be interesting to test as part of the reinitialization process.

5.1.4 Reinitialization test

Figure 7 shows the upper right part of the reinitialization of a circular interface in a 1 × 1 domain, centered at $(0.5; 0.5)$ and with a radius 0.3. The domain is discretized using 894 elements of 3rd order. We are using a big ϵ such that the level set will be a distance function in the whole domain. On the right plot, the true and exact solution is indistinguishable apart from close to the center of the circle at $(0.5; 0.5)$ due to the local loss of smoothness. Oscillations coming from the discontinuity is controlled by the filtering with the price for

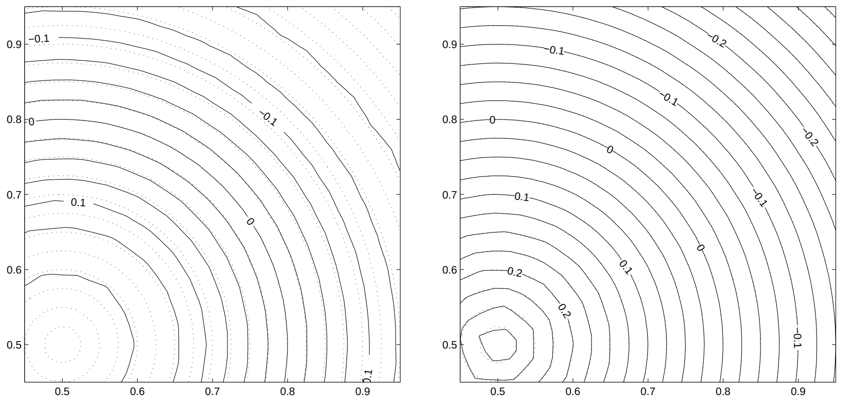

Fig. 7. Reinitialization of circle after time $\tau = 0.155$ (left) and $\tau = 0.5$ (right). Dotted lines are exact distance contours, solid lines are level set contours.

stability being some accuracy, but only close to the discontinuity. The left plot shows the level set contours at $\tau = 0.155$, and illustrates that the level set is first reinitialized at the interface and the reinitialization progresses along the characteristic lines, almost with speed one. The speed is not exactly one due to the smoothed sign function. At this point of $\tau = 0.155$, the level set would be accurate in a band wide enough for further use.

6 A Few Numerical Tests

In the following we shall show a few examples of the performance of the proposed scheme. In doing these we have not made any attempt to optimize the grids and elements in all tests are of same size in the entire domain. However, in all tests there are regions which could be resolved using much larger elements, hence decreasing the total number of elements and the computing time while maintaining accuracy.

6.1 Zalesaks problem

Zalesak (25) proposed a test to evaluate how well a method transports an interface. The Zalesak disk is rotated one revolution around the center of the domain using $\phi_t - \boldsymbol{u} \cdot \nabla\phi = 0$. The Zalesak disk, including the grid used, is depicted in Fig. 8. The rotation is a shape preserving transformation, hence reinitialization is not necessary. The solution method uses a small amount of filtering to control oscillations coming from the discontinuities. Figure 8 shows the initial disk and the disk after one full revolution using 3rd and 5th order elements respectively. The grid has 894 elements. For comparison with other methods, counting the number of unknown, the 3rd order element has 10 and

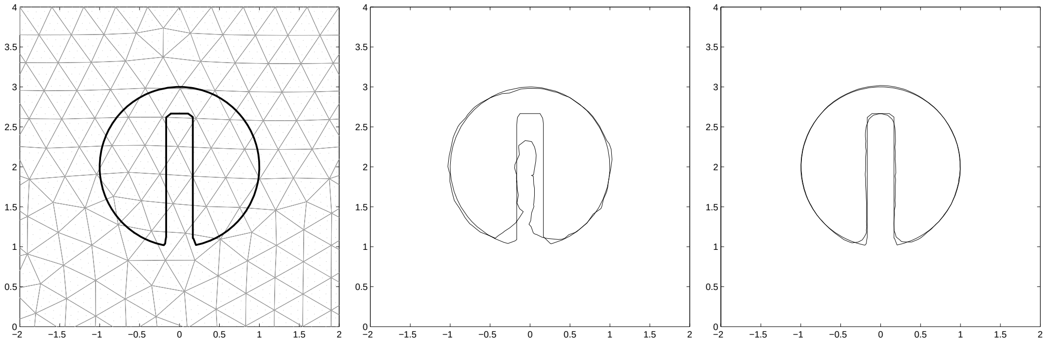

Fig. 8. Zalesaks problem, grid (left) and solution after one revolution for 3rd (middle) and 5th (right) order elements. Entire domain is $[-4; 4]x[-4, 4]$ hence only a part of the domain is shown

the 5th has 21 unknowns, in total 8940 and 18774 unknowns respectively, corresponding approximately to rectangular grids of size 95x95 and 137x137.

Results using the 5th order elements show that the DG method can evolve the Zalesak disk well and solve it accurately without the need for any special techniques or tricks. Note that the disk itself is only resolved with about 30 elements, i.e., a relatively coarse grid.

Results using 3rd order elements show a typical degradation of the solution, when the method have difficulties representing the smallest scales of the problem.

The DG method works very well where the level set is smooth, while where the surface has sharp corners and where two surfaces are very close, it is less accurate. Compared to lower order methods using a 137x137 grid, the results are excellent.

To show the effect of reinitialization, we present in Fig. 9 the results after one full revolution where also reinitialization is applied. The test use 3rd order elements and should be compared with the middle plot in Fig. 8. Plots (a) and (b) applies two different levels of filtering and diffusion, (a) as little as possible and in (b) somewhat more, emphasizing that dissipation must be applied with care. Plot (c) is a preliminary result using a technique for local mass conservation presented in (20), showing superior preformance for this particular example.

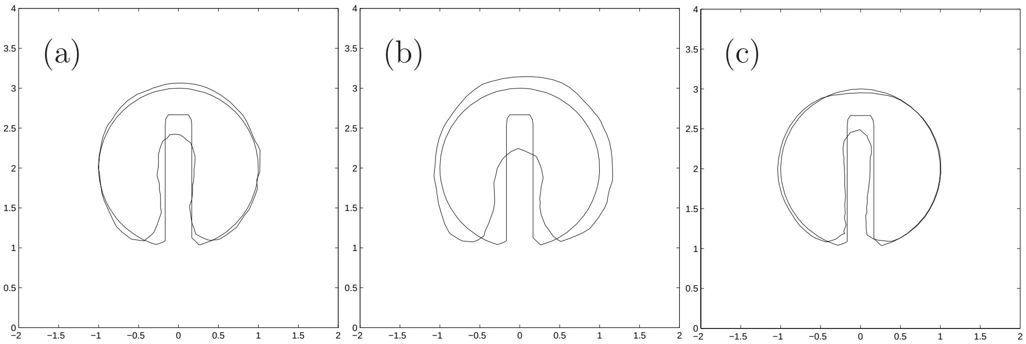

Fig. 9. Zalesaks problem including reinitialization. On left (a) is used minimal dissipation, in the middle (b) a bit more while the right (c) is a prelimenary result using a scheme with improved local mass conservation.

6.2 *Standing Wave*

In a box of size 1 × 2, the water in rest fills half the box and the initial surface is set to $1 + 0.2\sin(\pi x)$, see Fig. 10. The density ratio used are true values for water and air 1000:1, while the viscosity for both fluids are set to 1. Slip boundary conditions, i.e., no stress conditions (2), are set at all boundaries, apart from at the 4 corners, where homogeneous Dirichlet conditions are used. Velocities are set to zero, and at time 0 the water is "released" and it will start to move from side to side. Due to viscosity, the wave-height will decrease. The integration is carried out until $T = 2$, where the wave has moved back and forth almost 2 times.

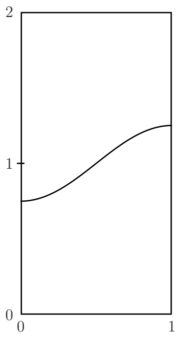

Fig. 10. Standing wave initial conditions

The problem was solved on a grid having 62 elements of order 4. Figure 11 (left) shows the area of the water over time for different time steps. It shows that mass is quite well conserved even for a longer integration time on this very coarse grid. Figure 11 (right) shows the energy over time for different time steps. Energy is lost due to viscosity. If integrating once with very small time step and defining that as the true solution, the energy converges toward the true solution in $O(\Delta t^{3/2})$ for a 2nd order BDF/Adams combination.

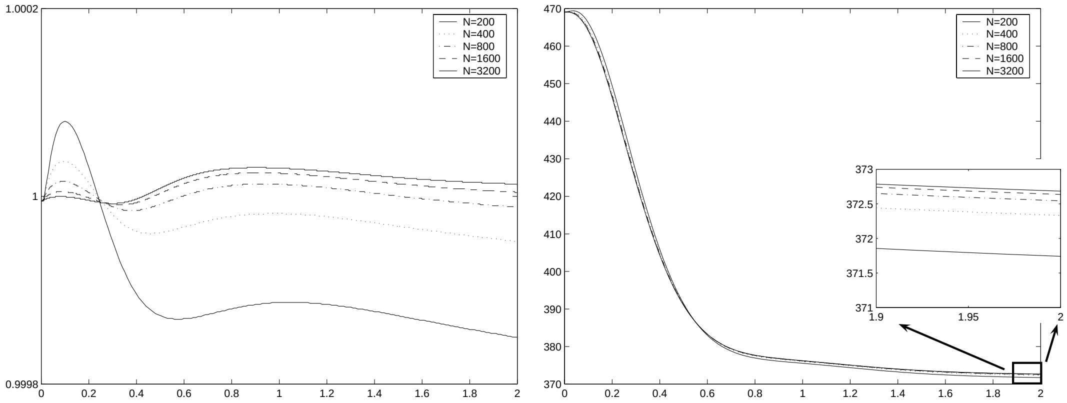

Fig. 11. Standing Wave mass conservation (left) and energy (right) as a function of the timestep, $\Delta = 2/N$.

The wobbling bubble test displays an example of a surface tension driven flow. A bubble formed initially as an ellipse and without a gravitational force, will oscillate due to the difference in curvature and the induced surface tension.

The test is set up in the domain $(x, y) \in [-\frac{1}{2}; \frac{1}{2}] \times [-\frac{1}{2}; \frac{1}{2}]$. Initial and boundary conditions are

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}, 0) &= \mathbf{0}, \\ \boldsymbol{u}(\boldsymbol{x}, t) &= \mathbf{0}, \qquad \text{on } \partial\Omega \end{aligned}$$

The surface is initially an ellipse with semi axis 0.2 and $\frac{\sqrt{2}}{10}$. Density ratio is 1000:10 while viscosity ratio is $\frac{1}{2} : \frac{1}{2}$. Using the bubble rest diameter $L = 0.2$, $U = 0.1$, $\rho_l = 1000$, $\mu_l = 0.5$ gives a Reynolds number of $Re = 40$. Surface tension is of size $\sigma = 50$ giving a Weber number of $We = 0.4$.

The problem is solved on a grid having 894 elements of order 3. The level set uses $\epsilon = 0.075$, which is a bit bigger than the side-length of the biggest element. Figure 12 shows 8 still pictures of the wobbling bubble over almost one period and Fig.13 illustrates the y-diameter of the bubble as a function of time. As expected, the oscillations decrease over time due to viscosity. The right plot shows the relative area as a function of time, and illustrates a mass loss of about 3.5% after time $T = 2$.

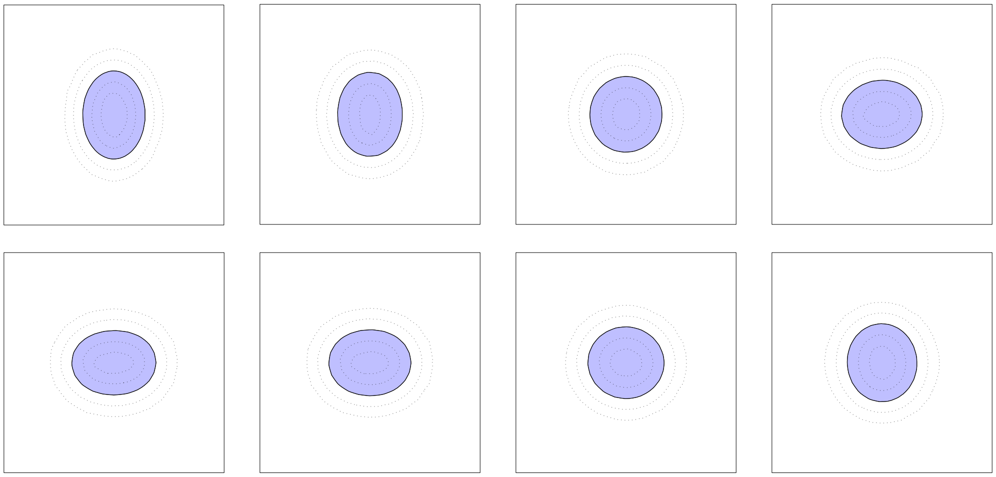

Fig. 12. Surface at times $t = 0, 0.1, ..., 0.7$. Dotted lines are level set contours at $-0.1, -0.05, ..., 0.1$

The mass loss is primarily due to effects depending on the surface thickness, described by the level set parameter ϵ. Decreasing ϵ decreases the mass loss, i.e., it is simply a question of resolution.

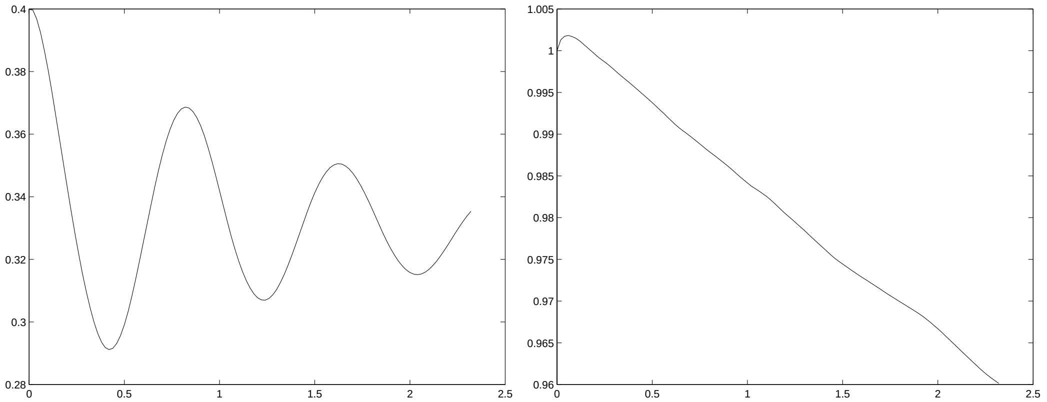

Fig. 13. Bubble diameter (left) and area (right)

6.4 Bubble falling

The falling bubble test shows that the method has no problem handling the merge of water volumes.

The test is set up in the domain $(x, y) \in [-\frac{1}{2}; \frac{1}{2}] \times [-\frac{1}{2}; \frac{1}{2}]$. Initial conditions are $\boldsymbol{u}(\boldsymbol{x}, 0) = \mathbf{0}$. Boundary conditions are periodic in the horizontal direction and $\boldsymbol{u}(\boldsymbol{x}, t) = \mathbf{0}$ elsewhere. The bubble is initially an ellipse with semi axis 0.2 and $\frac{\sqrt{2}}{10}$ and center $(0.1; 0.2)$, the surface is at hight -0.3. Density ratio is 1000:10 while viscosity ratio is $4 : \frac{1}{2}$. Using the bubble rest diameter $L = 0.2$, $U = 0.1$, $\rho_l = 1000$, $\mu_l = 4$ gives a Reynolds number of $Re = 25$. Surface tension is not included. The problem is solved on a grid having 894 elements of order 3. The level set uses $\epsilon = 0.05$.

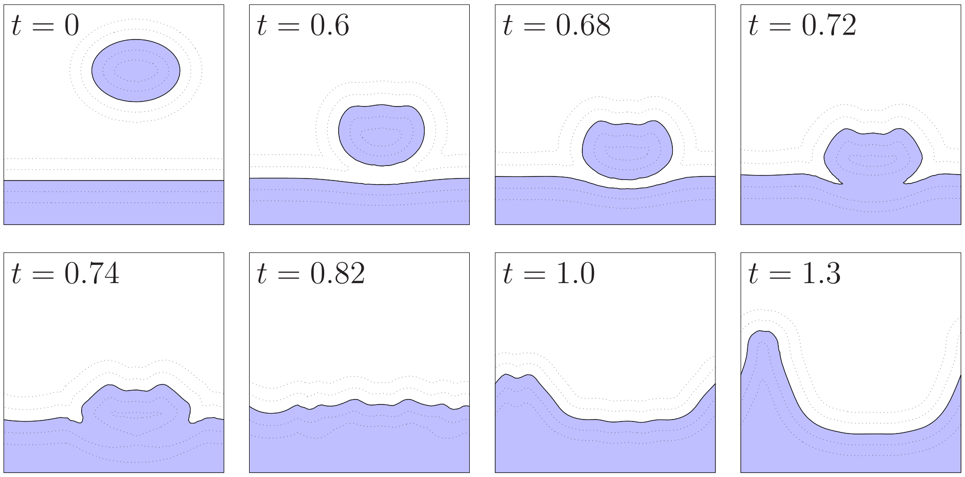

Fig. 14. Surface at different times. Dotted lines are level set contours at $-0.1, -0.05, ..., 0.1$

Figure 14 shows the surface at 8 different times, focusing on times where the bubble hits the surface and merges with the remaining water. The viscosity in the air is so big that the surface is seen to sink slightly before the bubble impacts. Figure 15 plots the mass change in time, and documents about 4% added mass at bubble impact. Reducing the surface thickness reduces this change in mass.

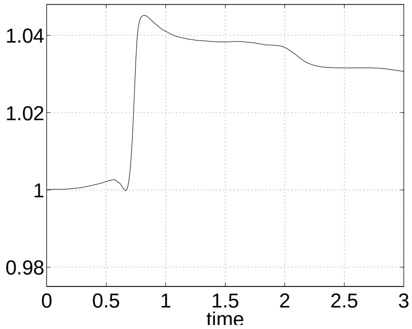

Fig. 15. Area of water (relative) shows small changes in time

6.5 Flow Generated Surface Waves

This test highlights surface waves generated by an obstacle below the surface. The domain is $(x, y) \in [-1; 1] \times [-\frac{1}{2}; \frac{1}{2}]$ and in the middle a rigid box of size 0.15×0.2. Initial and boundary conditions are

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}, 0) &= (1, 0), && \\ \boldsymbol{u}(\boldsymbol{x}, t) &= (1, 0), && \text{at inflow, top and bottom boundaries} \\ \boldsymbol{u}(\boldsymbol{x}, t) &= \boldsymbol{0}, && \text{on box boundaries} \\ (\boldsymbol{n} \cdot \nabla)\boldsymbol{u}(\boldsymbol{x}, t) &= \boldsymbol{0}, && \text{at outflow boundary} \end{aligned}$$

The surface is initially at rest at height $\frac{1}{4}$, and at inflow set at height $\frac{1}{4}$ for all time. The density ratio is 1000:10 while the viscosity ratio is 1:1. Using the box height, $L = 0.2$, and the inflow condition, $U = 1$, $\rho_l = 1000$, $\mu_l = 1$, gives a Reynolds number of $Re = 200$. Surface tension is neglected. The problem

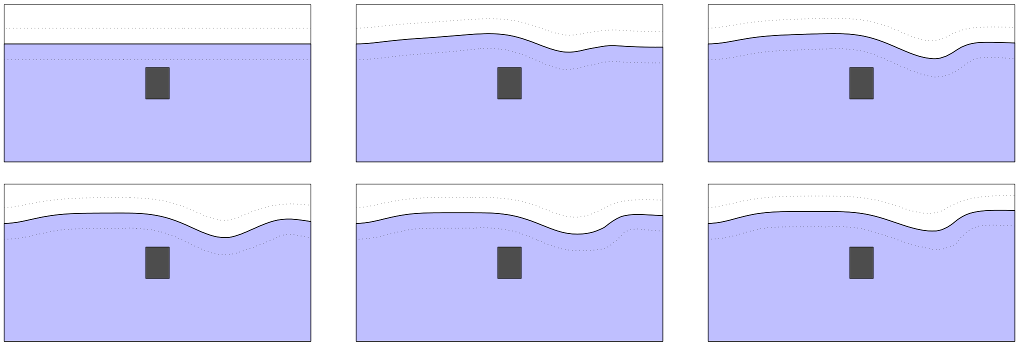

Fig. 16. Surface at times $t = 0, 0.5, ..., 2.5$

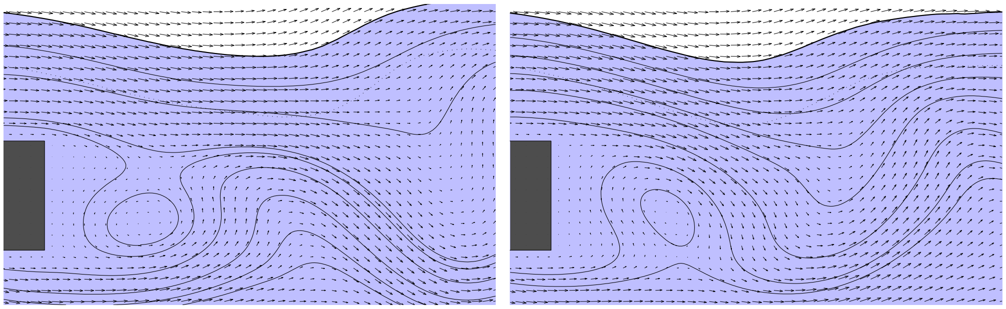

Fig. 17. Detailed flow with streamlines behind box at time 3.5 and 4.0

is solved on a grid having 304 elements of order 5. Figure 16 shows 6 still pictures of the surface, while Fig. 17 shows 2 still pictures of the flow rear of the obstacle. There is no steady state solution, the surface will oscillate due to the shedding of Von Karman vortices. We can calculate the force on the box by integrating up the pressure at the box boundaries. In Fig. 18, the force is split up into its x and y component. The y component of the force corresponds to the force if the box had been massless. However also gravity pulls on the box, hence the actual lift applied to the box will be less, and depend on its weight. We see that after a transition phase of about 4 time units, the flow enters a quasi-stationary state, where we can read the period to be slightly less than one time unit.

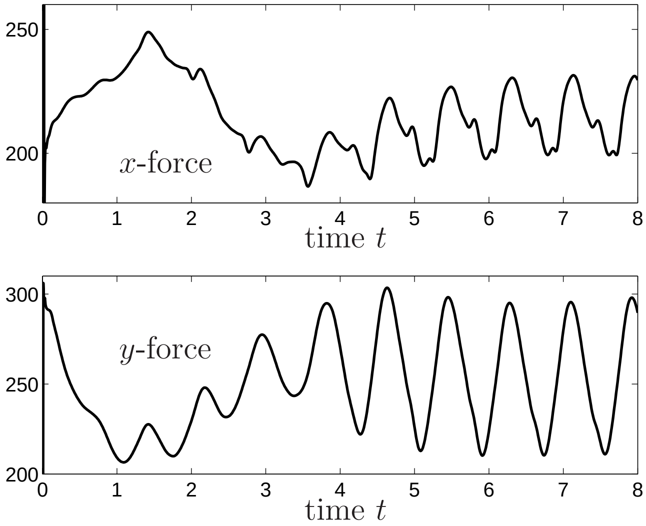

Fig. 18. Force on box in x and y direction as a function of time

7 Concluding Remarks

We have discussed the development of a levelset based discontinuous Galerkin methods as an approach to modeling the incompressible Navier-Stokes equations with free surfaces. The scheme, solving the problem as a two-fluid problem with the interface described by a levelset, utilizes a nodal high-order discontinuous Galerkin discretization on a fully unstructured grid and a velocity projection scheme in combination with a semi-implicit approach in time to advance the unsteady equations. As an alternative we propose the use of a semi-implicit spectral deferred correction projection (SISDC) formulation which has a number of attractive properties, although at present it remains unknown how to use it in finite domains without impacting the accuracy adversely.

A modified level set equation was introduced to enable the robust and simple representation of the free surface and we discussed appropriate boundary conditions for the level set at inflow and outflow boundaries. A few tests confirm the accuracy, robustness, and versatility of the proposed scheme.

However, the work also suggests a number of open questions. In particular, the correct treatment of boundaries in the SISDC approach is one of interest as the formulation appears to have the potential to reach higher accuracy than standard projection methods. Furthermore, the introduction of the levelset raises a number of questions in relation to its treatment around no-slip boundaries and multiply connected surfaces, i.e., since there is zero velocity on no-slip boundaries, the zero level set will never intersect or move on a no-slip boundary, e.g., for a wave hitting the box, the level set would wrap around the box. Whether this “wrapping” is acceptable, or a heuristic should be applied for updating the level set in cells on no-slip boundaries newly filled with/emptied for water, or if slip conditions are more appropriate, is a topic for further investigation.

References

[1] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.* 39:1749–1779, 2002.

[2] E. Bänsch and B. Höhn. Numerical treatment of the Navier-Stokes equations with slip boundary condition. *SIAM J. Sci. Comput.*, 21(6):2144–2162 (electronic), 2000.

[3] David L. Chopp. Some improvements of the fast marching method. *SIAM J. Sci. Comput.* 23(1):230–244, 2001

[4] B. Cockburn and C.-W. Shu. Runge-Kutta Discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.* 16:173–261, 2001.

[5] Douglas Enright, Ronald Fedkiw, Joel Ferziger and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.* 183(1):83–116, 2002.

[6] A. Duut, L. Greengard, and V. Rokhlin. Spectral Deferred Correction Methods for Ordinary Differential Equations. *BIT* 40(2):241-266, 2000.

[7] D. Gottlieb and J. S. Hesthaven. Spectral methods for hyperbolic problems. *J. Comput. Appl. Math.*, 128(1-2):83–131, 2001.

[8] J. L. Guermond and J. Shen. A new class of truly consisten splitting schemes for incompressible flows. *To appear*, 2003.

[9] J. L. Guermond and J. Shen. Velocity-correction projection methods for incompressible flows. *SIAM J. Numer. Anal.*, 41(1):112–134 (electronic), 2003.

[10] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8:2182–2189, 1965.

[11] J. S. Hesthaven. From Electrostatics to Almost Optimal Nodal Sets for Polynomial Interpolation in a Simplex. *SIAM J. Numer. Anal.* 35(2):655–676, 1998.

[12] J. S. Hesthaven and D. Gottlieb. Stable spectral methods for conservation laws on triangles with unstructured grids. *Comput. Methods Appl. Mech. Engrg.*, 175(3-4):361–381, 1999.

[13] J. S. Hesthaven and T. Warburton. Nodal high-order methods on unstructured grids. I. Time-domain solution of Maxwell's equations. *J. Comput. Phys.*, 181(1):186–221, 2002.

[14] C.W. Hirt and B.D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *J. of Computational Physics*, 39:201–225, 1981.

[15] A. Kanevsky, M. H. Carpenter, and J. S. Hesthaven. Time-consistent filtering in spectral and spectral element methods. *J. Comput. Phys.* 2004 – submitted.

[16] G.E. Karniadakis, M. Israeli, and S. A. Orszag. High-order splitting methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 97(2):414–443, 1991.

[17] M.L. Minion. Higher-order semi-implicit projection methods. In *Numerical simulations of incompressible flows (Half Moon Bay, CA, 2001)*, pages 126–140. World Sci. Publishing, River Edge, NJ, 2003.

[18] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[19] Seungwon Shin and Damir Juric Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking with connectivity. *J. Comput. Phus.*, 180:427–470, 2002.

[20] M. Sussman and E. Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. Sci. Comput.*, 20(4):1165–1191 (electronic), 1999.

[21] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. An improved level set method for incompressible two-phase flows. *Computers and Fluids*, 27(5-6):663–680, 1998.

[22] Mark Sussman and M.Y. Hussaini. A discontinuous spectral element

method for the level set equation. *J. Sci. Comput.* 19(1-3):479–500, 2003.

[23] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two phase flow. *J. Comput. Phys.*, 114(1):149–159, 1994.

[24] A.-K. Tornberg and B. Engquist. Regularization Techniques for Numerical Approximation of PDEs with Singularities *J. Sci. Comput.* 19(1-3):527–552, 2003.

[25] S.T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.*, 31(3):335–362, 1979.

[26] Y.-T. Zhang and C.-W. Shu. High order WENO schemes for Hamilton-Jacobi equations on triangular meshes. *SIAM J. Sci. Comp.* 24:1005–1030, 2003.