Monotone level-sets on arbitrary meshes without redistancing

Akkerman, Ido

DOI
10.1016/j.compfluid.2017.01.007

Publication date
2017

Document Version
Accepted author manuscript

Published in
Computers & Fluids

Citation (APA)
Akkerman, I. (2017). Monotone level-sets on arbitrary meshes without redistancing. *Computers & Fluids*, *146*, 74-85. https://doi.org/10.1016/j.compfluid.2017.01.007

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Monotone level-sets on arbitrary meshes without redistancing

Ido Akkerman

Mechanical, Maritime and Materials Engineering Department
Delft University of Technology

Abstract

In this paper we present a level-set approach that addresses two issues that can occur when the level-set approach is used to simulate two-fluid flows in engineering practice.

The first issues is that of smoothing of the Heaviside on arbitrary meshes. It is shown that the Heaviside can be non-smooth on non-uniform meshes. Alternative definitions of smoothing, that are indeed smooth and monotonic, are defined. These new definitions lead to smooth Heavisides by taking the changing local meshsize into account.

The second issue is the computational cost and fragility caused by the necessity of redistancing the level-set field. In [1, 2], it is shown that strongly coupling the level-set convection with the flow solver gives robustness and potentially efficiency and accuracy advantages. The next step would be to include redistancing within the strong coupling. Four alternative approaches for circumventing the expensive redistancing step are proposed.

Some benchmark cases used to show the efficacy of the proposed approaches. These includes the standard test case of the vortex in a box. Based on these results the most favourable redistancing approach is selected.

Keywords: level-set, arbitrary meshes, smooth Heaviside, implicit redistancing

1. Introduction

Level-sets are a very powerful approach for solving interface problems [3]. By disconnecting the interface description from the underlying mesh, topology changes of the interface are handled with ease. For numerical reasons, such as numerical quadrature or finite differencing, the interface is often given a finite mesh dependent width. This is done by adopting a so-called smoothed Heaviside, where the transition from zero to unity does not occur instantaneously but over a finite band. This smoothing , however, causes two problems. On irregular meshes – such as that might occur in the analysis of engineering artefacts – the smoothed Heaviside can become non-smooth or even non-monotone. Additionally, in order to control the thickness of the interface the need level-set is required to be distance function. There as level-set evolves it needs to be redistanced.

In [1, 2] the level-set method is used to model a rigid-body floating on a water surface. This water surface is handled using a level-set. In this research we adopted a strongly coupled modelling approach. Meaning that both the flow; interface evolution; body motion and mesh deformation were solved at the same time. In agreement with [4, 5, 6] – and other FSI literature– this approach resulted in a robust solver. In a pure two-fluid problem, that is without a floating object, the strongly coupled approach proved to be more robust that traditional where interface evolution is solved after the flow problem is solved.

However, the redistancing of the level-set was not included in the strongly coupled part of the solver, this was done at the end of each time step. Due to the redistribution of mass associate with redistancing, momentum and energy conservation are virtually impossible to control. Certainly, the energy errors are troublesome as these might trigger instabilities in the solver. Therefor there is a desire to be able to include the redistancing step in the strongly coupled part of the method. This should open the way to construct methods that are either energy conservative or guaranteed to be energy dissipative. This should lead to more robust, and hopefully more accurate, methods.

In this paper we present four alternative approach to arrive at an approximate distance field without directly solving a redistancing problem. These approaches are formulated in such a way that they potentially solve the non-smoothness on irregular grids.

The paper is structured in the following way. In section 2 we give a short introduction to the level-set approach and the smooth Heaviside. In section 3 we show that a naive definition of the smooth Heaviside can actually be non-smooth. A potential solution to this problem is presented.

In section 4 we build on the result from the previous section and introduce four alternative approaches to redistance the level-set field. All these approaches have in common that the difficult non-linear Eikonal equation is not solved directly. The Eikonal equation is translated to simple projection problems that should achieve the same: obtaining a distance field suitable for defining a smooth Heaviside.

In section 5 we give a short description of the numerical formulations and the Finite Element and Isogeometric discretisations used to solve the convection and redistancing problems in the next section. In this section, section 6, we first solve a simple redistancing problem on irregular grids and then solve the vortex in a box problem using different discretisations and different redistancing approaches. Using these test cased we identify which methods work best, and what are the appropriate numerical parameters. For this method a mesh convergence study is performed.

In section 7 we conclude and sketch a perspective for further use of the proposed methods

2. The level set method

In the level-set method an surface of lower dimension, such as an interface between two distinct materials is indirectly parametrised using a variable defined in the entire domain. This variable is denoted as ϕ and defines a surface as follows,

$$\Gamma_i = \{\boldsymbol{x} \in \Omega : \phi = 0\} \tag{1}$$

This automatically leads to the following distinct subdomains,

$$\begin{aligned} \Omega^- =& \{\boldsymbol{x} \in \Omega : \phi < 0\} \\ \Omega^+ =& \{\boldsymbol{x} \in \Omega : \phi > 0\} \end{aligned} \tag{2}$$

allowing for instance the prescription of different physical parameters in each subdomain. For instance the density

$$\rho = \begin{cases} \rho_0 & \text{if} \quad \boldsymbol{x} \in \Omega^- \\ \rho_1 & \text{if} \quad \boldsymbol{x} \in \Omega^+ \end{cases} \tag{3}$$

An alternative view would be to introduce the Heaviside

$$H(\phi) = \begin{cases} 0 & \text{if} \quad \phi < 0 \\ \frac{1}{2} & \text{if} \quad \phi = 0 \\ 1 & \text{if} \quad \phi > 0 \end{cases} \tag{4}$$

and use convex interpolation to define a density

$$\rho = \rho_0(1 - H(\phi)) + \rho_1 H(\phi) \tag{5}$$

this has the advantage of automatically handling the interface itself in a natural way.

2.1. Smoothed Heaviside

In numerical methods the sharp interfaces as defined in the previous section can lead to problems. For instance when determining a mass matrix,

$$M_{ab} = \int_\Omega \rho N_a N_b d\Omega \approx \sum_{i=1..n_{ip}} \rho(\boldsymbol{x}_i) N_a(\boldsymbol{x}_i) N_b(\boldsymbol{x}_i) w_i \tag{6}$$

when the density changes instantly the quadrature is a very bad approximation of the intended integral.

In order to alleviate this problem the sharp Heaviside, as defined in eq (4) , a smoothed Heaviside is used instead. This smoothed Heaviside is often defined as,

$$H_\epsilon(\phi) = \begin{cases} 0 & \text{if} \quad \phi < -\epsilon \\ \frac{1}{2}(1 + \sin(\frac{\pi\phi}{2\epsilon})) & \text{if} \quad \phi = 0 \\ 1 & \text{if} \quad \phi > \epsilon \end{cases} \tag{7}$$

where ϵ is the smoothing distance. Instead on instant switch the switch is spread over an finite layer around the interface. In order to have strict control over the width of this interface layer, we require ϕ, the level-set variable, to be a signed distance function. This means it needs to satisfy,

$$\|\nabla\phi\| = 1 \tag{8}$$

this is the Eikonal equation. As the smoothing is introduced to deal with numerical issues, such as quadrature, it is natural to specify the finite interface layer in terms of mesh size h,

$$\epsilon = \alpha h \tag{9}$$

where α is an $O(1)$ parameter. In structured equidistant meshes the meshsize can be defined unambiguously. However, in arbitrary meshes this is not always straightforward. In [7, 8, 1] we have chosen to use

$$h = \frac{\|\nabla\phi\|}{\sqrt{\nabla\phi \cdot \boldsymbol{G}\nabla\phi}} \tag{10}$$

where $\boldsymbol{G}$ is the metric-tensor,

$$\boldsymbol{G} = \left(\frac{\partial\xi}{\partial\boldsymbol{x}}\right)^T \frac{\partial\xi}{\partial\boldsymbol{x}} \tag{11}$$

this definition tries to incorporate directional information in its definition. Effectively, the direction $\frac{\nabla\phi}{\|\nabla\phi\|}$ is used to extract a length scale out of the metric-tensor.

3. Monotonicity on arbitrary meshes

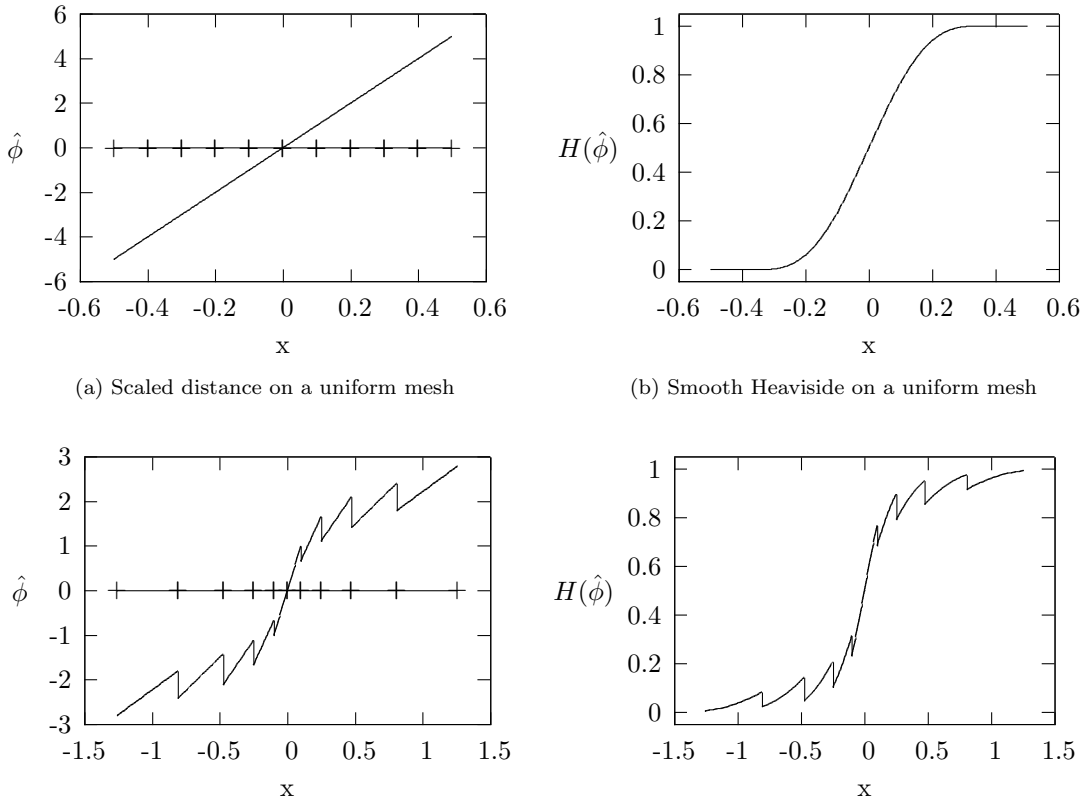
In this section we further discuss the smoothing of the Heaviside on arbitrary meshes. For this exposition it is useful to rewrite the smoothed Heaviside slightly,

$$\hat{H}(\hat{\phi}) = \begin{cases} 0 & \text{if} \quad \hat{\phi} < -\alpha \\ \frac{1}{2}(1 + \sin(\frac{\pi}{2}\frac{\hat{\phi}}{\alpha})) & \text{if} \quad \hat{\phi} = 0 \\ 1 & \text{if} \quad \hat{\phi} > \alpha \end{cases} \tag{12}$$

where

$$\hat{\phi} = \frac{\phi}{h} \tag{13}$$

is the scaled distance. In other words if ϕ is the actual distance to the interface then $\hat{\phi}$ can be thought of as the same distance but expressed in elements. However, the scaling occurs locally, the variation of h along the path is not taken into account. Therefore this rescaled distance is only an effective estimate if h only varies mildly (or not at all).



(a) Scaled distance on a uniform mesh

(b) Smooth Heaviside on a uniform mesh

(c) Scaled distance on a non-uniform mesh

(d) Smooth Heaviside on a non-uniform mesh

Figure 1: Original scaled distance and smooth Heaviside on a uniform and non-uniform mesh

In figure 1 a scaled distance and the resulting "smooth' Heaviside are plot. The top plots show the results on a uniform mesh. In this case the scaled distance is a straight line and the smooth Heaviside is indeed smooth. The bottom plots , however, show the results obtained on a non-uniform mesh. In this case the scaled distance and more importantly the "smooth" Heaviside show a saw-tooth behaviour. This defeats the purpose of the smoothing. Although, one could argue that the quadrature on each element is well behaved as the jumps occur at element interfaces. This is true, however, this will result in an inconsistent numerical method, unless the density jumps at the interface are handelled correctly [9].

3.1. Path dependent rescaling

In order to solve the problems as indicated in the previous section. We should translate eq (13) to a pure local relation,

$$d\hat{\phi} = \frac{d\phi}{h} \tag{14}$$

where the total scaled distance can– in principle – be found by path integration,

$$\hat{\phi} = \int \frac{d\phi}{h} \tag{15}$$

which is not trivial.

Instead we take another route to obtain $\hat{\phi}$. Alternative to eq (14) we can also state,

$$\nabla\hat{\phi} = \frac{\nabla\phi}{h} \tag{16}$$

which immediately results in an alternative Eikonal equation for the scaled distance,

$$\|\nabla\hat{\phi}\| = \frac{\|\nabla\phi\|}{h} = \frac{1}{h} \tag{17}$$

where we used $\|\nabla\phi\| = 1$. In principle any reasonable selection of h should work.

However, inspired on the definition for h as given in eq (10), we can define h in such a way that eq (17) gets replaced with an attractive alternative. We choose

$$h = \frac{\sqrt{\nabla\hat{\phi} \cdot \boldsymbol{G}^{-1}\nabla\hat{\phi}}}{\|\nabla\hat{\phi}\|} \tag{18}$$

where

$$\boldsymbol{G}^{-1} = \frac{\partial \boldsymbol{x}}{\partial \xi}\left(\frac{\partial \boldsymbol{x}}{\partial \xi}\right)^T \tag{19}$$

is the inverse of the metric-tensor defined in eq (11). The same rational behind eq (10) is behind eq (18). We use the direction $\frac{\nabla\hat{\phi}}{\|\nabla\hat{\phi}\|}$ to obtain the relevant mesh size out of the inverse metric tensor, which parametrises the rotation, scaling and distortion of the element at hand.

Defining ∇_ξ as the gradient in the reference direction, we can use the chain rule to arrive at,

$$\nabla_\xi\hat{\phi} \equiv \frac{\partial\hat{\phi}}{\partial\xi} = \frac{\partial\hat{\phi}}{\partial\boldsymbol{x}}\frac{\partial\boldsymbol{x}}{\partial\xi} = \nabla\hat{\phi}\frac{\partial\boldsymbol{x}}{\partial\xi} \tag{20}$$

which can then be substituted in eq (18) to obtain the following definition for h,

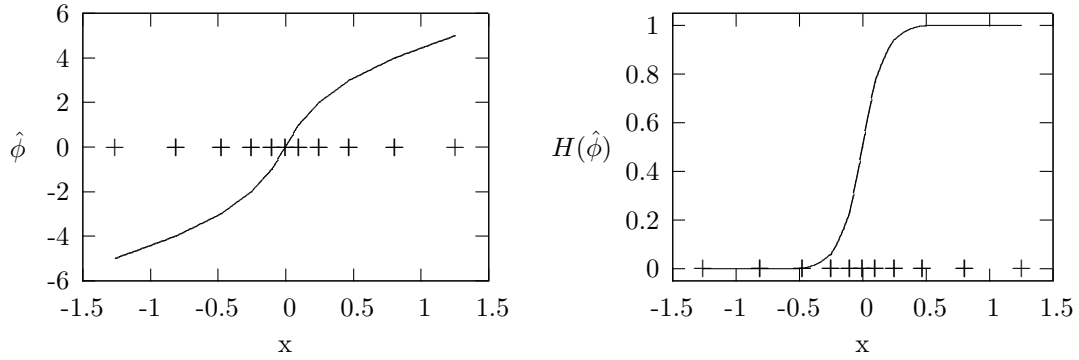
$$h = \frac{\|\nabla_\xi\hat{\phi}\|}{\|\nabla\hat{\phi}\|} \tag{21}$$

Combining this with the scaled Eikonal equation (eq (17)) we get,

$$\|\nabla_\xi\hat{\phi}\| = 1 \tag{22}$$

which is again the Eikonal equation, but in terms of the parametrised space. Which states $\hat{\phi}$ is a distance in terms of mesh edges, which was exactly what we where looking for in the first place.

Figure 2 shows the resulting scaled distance and smooth Heaviside when the alternative scaling is used. The mesh is the same as in fig 1. Both the distance and Heaviside do not show a saw-tooth behaviour and the Heaviside is actually smooth, as intended. The scaled distance is not a straight line, which is in fact correct, because of the metric defined by the non-uniform mesh.



(a) Corrected scaled distance on a non-uniform mesh (b) Corrected smooth Heaviside on a non-uniform mesh

Figure 2: Modified scaled distance and smooth Heaviside on a uniform and non-uniform mesh

4. Circumventing explicit redistancing

In the previous section we addressed the issue of monotone smoothing of the Heaviside. An equally important issue is that of the distance property of ϕ or better $\hat{\phi}$. As the levelset is evolved in time using equations such as,

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0 \tag{23}$$

or equivalent it is not a given that the distance property is maintained during the simulation. In order to keep explicit control over the interface layer, we need to restore the distance property. Even if ϕ magically remains a distance function the path dependent scaling of the distance function still requires us to solve an Eikonal equation, namely eq (22),

$$\|\nabla_\xi \hat{\phi}\| = 1 \qquad \text{on} \qquad \Omega^- \cup \Omega^+ \tag{24}$$

this equation is augmented with the following embedded boundary condition,

$$\hat{\phi} = 0 \qquad \text{on} \qquad \Gamma_i = \{\boldsymbol{x} \in \Omega : \phi = 0\} \tag{25}$$

which forces the interface to remain fixed in the scaling/redistancing step.

Al kinds of different approaches to solve the Eikonal equation have been developed, see for instance [10] and references there in. Most methods introduce a pseudo time τ and march in time using,

$$\frac{\partial \hat{\phi}}{\partial \tau} + \boldsymbol{a} \nabla \hat{\phi} = S \tag{26}$$

where $\boldsymbol{a}$ is an effective convection velocity and S is a source. Both $\boldsymbol{a}$ and S are carefully chosen to guarantee convergence towards the desired solution.

Note: as shown in [1] the source term needs to include an penalty term penalizing deviation of the interface in order to guarantee a steady state solution. This issue is often circumvented by limiting the pseudotime steps.

This redistancing procedure has several drawbacks. For starters it is quite costly, even when limited in space (only redistance near the interface) or time (only redistance every so many timesteps). Additionally, careful attention needs to be paid to the interplay of different parameters associated with the time marching algorithm making the procedure somewhat fragile. Last but not least it is difficult to find the right balance

between redistancing on the one hand and maintaining the interface on the other hand. Therefor it ideally the explicit redistancing is perhaps bast to be circumvented.

4.1. General idea

In the following sections we present 4 alternative procedures to achieve this. All 4 alternatives hinge on the idea that we relate the unknown $\hat{\phi}$ with the known ϕ using,

$$\hat{\phi} = \phi\epsilon \tag{27}$$

where ϵ is a unknown multiplicative scaling, which is required to be strictly positive. It is easily seen that $\hat{\phi}$ is vanishes when ϕ vanishes. This means that the interface location is trivially conserved. There is no need to balance redistancing with maintaining the interface.

The multiplicative scaling can be found by enforcing parametrised Eikonal equation eq (22),

$$\|\nabla_\xi \hat{\phi}\| = 1 \qquad \text{on} \qquad \Omega^- \cup \Omega^+ \tag{28}$$

and applying the chain rule. As we are only interested in what happens near the interface we can assume $\phi \approx 0$ which results in the following relation for the multiplicative scaling,

$$\epsilon = \frac{1}{\|\nabla_\xi \phi\|} \tag{29}$$

In the following section 4 different alternative approaches based on this general idea are presented.

4.2. First alternative: Direct redistancing

The first option is straightforward pointwise scaling,

$$\hat{\phi} = \frac{\phi}{\|\nabla_\xi \phi\|} \tag{30}$$

which has the risk of resulting in divisions by zero in some occasions. Additionally, on standard C_0 continuous discretisations this will likely lead to a discontinuous $\hat{\phi}$ field and consequently a discontinuous $H(\hat{\phi})$ field.

4.3. Second alternative: Projected redistancing

We can obtain a continuous distance fields from the discontinuous $\hat{\phi}$ of the previous section, by projecting it on a C_0 discretisation. This can be done by solving the following projection problem,

$$\hat{\phi} - \nabla_\xi \kappa_d \nabla_\xi \hat{\phi} = \frac{\phi}{\|\nabla_\xi \phi\|} \tag{31}$$

or by solving the following equivalent weak form,

Find $\hat{\phi} \in \mathcal{V}^h$ *such that:*

$$(w, \hat{\phi})_\Omega + (\nabla_\xi w, \kappa_d \nabla_\xi \hat{\phi}) = \left(w, \frac{\phi}{\|\nabla_\xi \phi\|}\right)_\Omega \qquad \forall w \in \mathcal{V}^h \tag{32}$$

where κ_d is a smoothing parameter and $\mathcal{V}^h$ is an appropriate discrete space. Due to the projection and the associated discretisation errors the interface location only remains in the same location in an approximate sense. This means the interface is not necessarily maintained during the redistancing step.

4.4. Third alternative: Projected scaling

The properties of continuity and interface conservation can be achieved simultaneously, by projecting the scaling coefficient on a continuous discretisation. This can be done with the following projection,

$$\epsilon - \nabla_\xi \kappa_d \nabla_\xi \epsilon = \frac{1}{\|\nabla_\xi \phi\|} \tag{33}$$

or by solving the following equivalent weak form,

$$\boxed{\begin{aligned} &\textit{Find } \epsilon \in \mathcal{V}^h \textit{ such that:} \\ &\quad (w, \epsilon)_\Omega + (\nabla_\xi w, \kappa_d \nabla_\xi \epsilon) = \left(w, \frac{1}{\|\nabla_\xi \phi\|} \right)_\Omega \qquad \forall w \in \mathcal{V}^h \end{aligned}} \tag{34}$$

again κ_d is a smoothing parameter and $\mathcal{V}^h$ is an appropriate discrete space. The sought after distance field can easily be computed as,

$$\hat{\phi} = \phi \epsilon \tag{35}$$

4.5. Fourth alternative: Projected inverse scaling

The last alternative approach is a small modification on the previous alternative. We are not projecting the multiplicative correction, but its reciprocal value. This can be done with the following projection,

$$\epsilon - \nabla_\xi \kappa_d \nabla_\xi \epsilon = \|\nabla_\xi \phi\| \tag{36}$$

or by solving the following equivalent weak form,

$$\boxed{\begin{aligned} &\textit{Find } \epsilon \in \mathcal{V}^h \textit{ such that:} \\ &\quad (w, \epsilon)_\Omega + (\nabla_\xi w, \kappa_d \nabla_\xi \epsilon) = (w, \|\nabla_\xi \phi\|)_\Omega \qquad \forall w \in \mathcal{V}^h \end{aligned}} \tag{37}$$

again κ_d is a smoothing parameter and $\mathcal{V}^h$ is an appropriate discrete space. The sought after distance field can easily be computed as,

$$\hat{\phi} = \frac{\phi}{\epsilon} \tag{38}$$

5. Numerical algorithm

In order to solve the convection problem given in eq (23) we choose to use finite elements and Isogeometric analysis. For completeness we describe the weak formulation and associated numerical parameters adopted. Then we briefly discuss the Isogeometric analysis shape functions that are used. Adoption of Isogeometric analysis allows the selection of C_1 continous discretisations, which wil turn out to be beneficial.

5.1. SUPG formulation

The convective problem from eq (23) is solved using a SUPG formulation [11]. This problem is state as follows:

$$\boxed{\begin{aligned} &\textit{Find } \phi \in \mathcal{V}^h \textit{ such that:} \\ &\quad (w + \tau \boldsymbol{u} \cdot \nabla w, \phi_t + \boldsymbol{u} \cdot \nabla \phi)_\Omega + (\nabla_\xi w, \kappa_c \nabla_\xi \phi)_\Omega = 0 \qquad \forall w \in \mathcal{V}^h \end{aligned}} \tag{39}$$

where $\mathcal{V}^h$ is an appropriate discrete space, τ is the stabilisation parameter computed as

$$\tau = \left(\Delta t^2 + \boldsymbol{u} \cdot \boldsymbol{G}\boldsymbol{u}\right)^{-1/2} \tag{40}$$

and κ_c is the discontinuity capturing parameter computed as

$$\kappa_c = C \left|\phi_t + \boldsymbol{u} \cdot \nabla\phi\right|. \tag{41}$$

Note that the residual-based nature choice of the discontinuity capturing parameter makes the total method strongly consistent. This discontinuity capturing is inspired on similar discontinuity capturing reported [12] and is dimensionally consistent with more traditional discontinuity capturing parameter such as reported in for instance [13, 14].

The equations are integrated in time using Crank-Nicolson, where for convenience we introduce the mass correction in the time derivative,

$$\phi_t = \frac{\phi^{n+1} - (\phi^n + \phi'^n)}{\Delta t} \tag{42}$$

in this way ϕ'^{n+1} as an instantaneous correction necessary for maintaining mass conservation. The resulting nonlinear system is solved using PETSc [15, 16].

5.2. Volume conservation

Often, for instance when dealing with incompressible fluids, it is *a priori* know that the volume of each subdomain is conserved. That is

$$\begin{aligned} V_0 &= \int_\Omega 1 - H(\hat{\phi}) d\Omega \\ V_1 &= \int_\Omega H(\hat{\phi}) d\Omega \end{aligned} \tag{43}$$

should be both conserved during the simulation. This mass conservation is not guaranteed in the discrete setting and it can happen that due to accumalation of volume errors one constituent completely disappears. To prevent this from happening we add a global constant to our convected level-set field ϕ in order to restore global conservation. This global constant is found by solving the following non-linear equation,

$$\int_\Omega H(\hat{\phi}(\phi^{n+1} + \phi'^{n+1})) d\Omega = \int_\Omega H(\hat{\phi}(\phi^n + \phi'^n)) d\Omega \tag{44}$$

this is done using a Newton-Raphson procedure. As this is a scalar equation, this routine is quite fast.

5.3. Isogeometric analysis

Isogeometric analysis, introduced in [17], is the idea to use shape functions used in CAD directly in the simulation. Since its introduction a large body of research is directed towards this general idea, see for instance [18] and references there in. One of the more popular choices for CAD shape functions are the so called NURBS, see for instance [19] for more information, which are also adopted in this work.

In [20] it has been has been shown that for NURBS similar approximation properties can be proofed as for standard finite element. While in [21] it has been demonstrated in a fundamental level that show superior convergence properties in terms of accuracy per degree of freedom. A large body of work exist conforming the benefits of adopting NURBS shapefunctions – or isogeometric analysis in general.

6. Numerical experiments

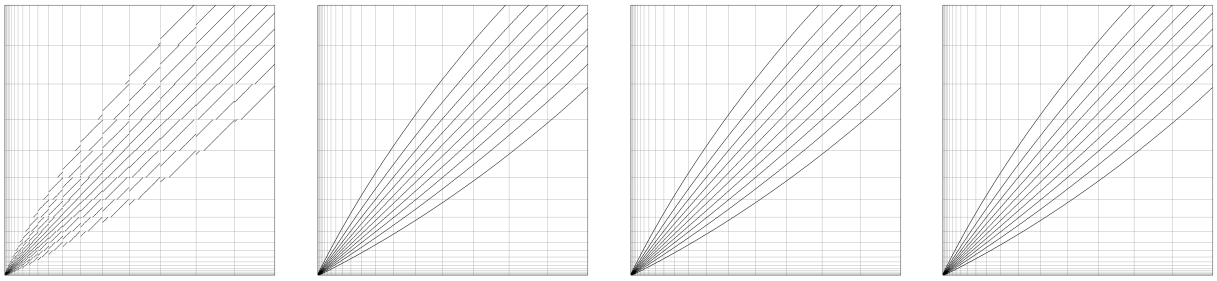
6.1. Distortion test

In order to assess the four alternative approaches for redistancing we apply them to distorted meshes, where the meshsize is uniformly reduced in toward both the x and y axis. The underlying level-set functions is given as,

$$\phi = x - y \tag{45}$$

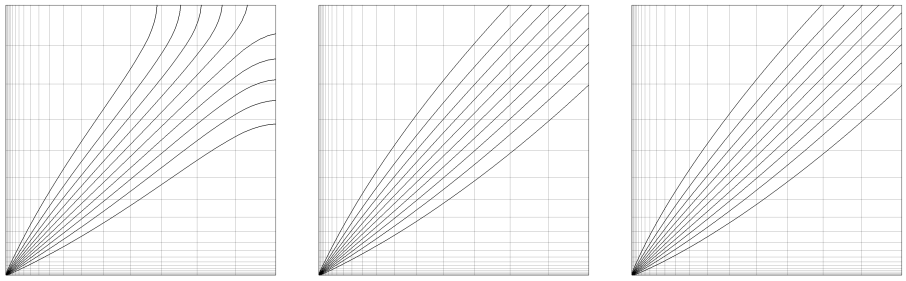
which results in an interface running from the bottom left to the top right.

The contour lines of the Heaviside due to the redistancing in terms of mesh length is given in figures 3, 4,5 and 6. To exaggerate the results we have selected a relatively wide interface width, namely $\alpha = 3$.



(a) *Direct redistancing* (b) *Projected redistancing* (c) *Projected scaling* (d) *Projected inverse scaling*

Figure 3: Contour lines for the different redistancing approaches, all results are on de $C_1 - Q_2$ mesh without any smoothing.

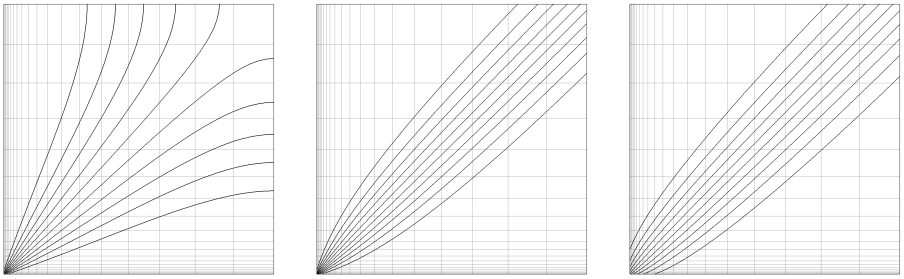


(a) *Projected redistancing* (b) *Projected scaling* (c) *Projected inverse scaling*

Figure 4: Contour lines for the different projection approaches, all results are on de $C_1 - Q_2$ mesh with smoothing $\kappa_d = 1$.

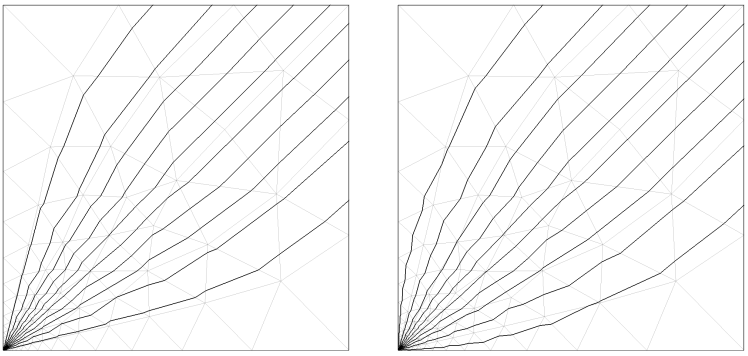
In figure 3 the contours for all four methods on a $C_1 - Q_2$ mesh are presented. From this it can be clearly seen that the *direct redistancing* results in a discontinuous Heaviside, illustrated by the contour lines that terminate at element edges. This shows that even on C_1 meshes the direct scaling does not necessarily provide a smooth Heaviside. For the other three approaches the contour lines are nicely continuous and there is hardly any distinction between the three projection based methods. On a $C_0 - Q_1$ mesh the result for all four approaches will look analogous.

In figure 4 the effect of moderate smoothing is presented. *Projected redistancing* clearly suffers from this inclusion of smoothing. While the two scaled projection methods are largely unaffected, although for *projected inverse scaling* the contour lines converge a bit less at the origin due to the smoothing. With increased smoothing *projected inverse scaling* will suffer more from this effect, as shown in figure 5.



(a) (b) (c)

Figure 5: Contour lines for the scaling approaches, all results are on de $C_1 - Q_2$ mesh with high smoothing $\kappa_d = 10$.



(a) *Projected scaling* (b) *Projected inverse scaling*

Figure 6: Contour lines for the different redistancing approaches on a triangular mesh with smoothing $\kappa_d = 1$.

In figure 6 the two scaled projection methods on a triangular mesh are presented. The triangular meshes are generated with gmsh [22] In this case the irregularity of the triangular meshes highly favours the inclusion of some modest smoothing.

6.2. Problem description: Vortex in a box

To assess the efficacy of the proposed redistancing approaches we apply all methods on the vortex in a box problem. Given the conclusions of the previous section we focus only on the three projection based alternatives, as the direct approach failed to deliver smooth Heaviside functions.

The problem setup and parameters are taken from [23]. The computational domain is a unit square and ϕ is initialised such that its zero level set is a circular disc of radius 0.15 centered at $(0.50, 0.75)$. The flow velocity is time-dependent and is prescribed to be $\boldsymbol{u} = (u, v)$, where

$$\begin{aligned} u &= \cos\left(\frac{\pi t}{8}\right) \sin(2\pi y) \sin(\pi x)^2 \\ v &= -\cos\left(\frac{\pi t}{8}\right) \sin(2\pi x) \sin(\pi y)^2 \end{aligned} \tag{46}$$

Until $t = 4$ this velocity field stretches the disc into a very thin spiral, after this time the flow reverses and the spiral is deformed back into a perfect circle at t = 8.

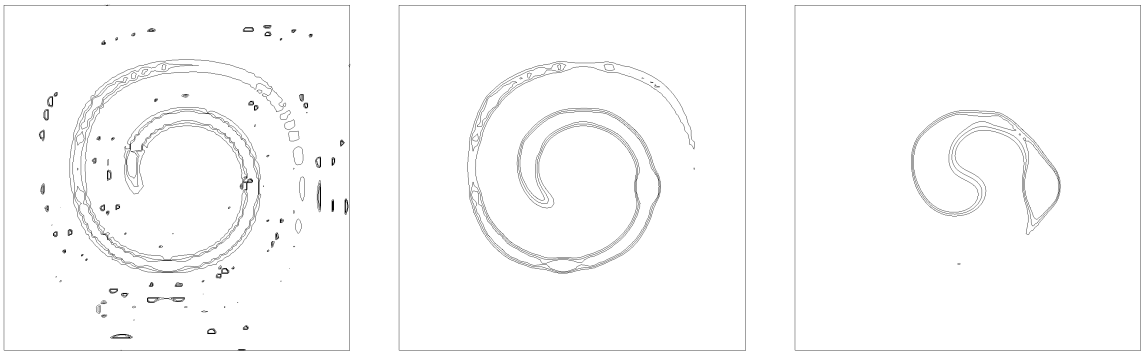
We solve this problem on different meshes. On the one hand we have a sequence of quadrilateral NURBS

elements with linear-C_0 or quadratic-C_1 tensor product based shape functions. Additionally we solve the problem on triangular elements. Which should indicate the generality of the approach.

Initially we solve select an 80×80 element mesh for the NURBS meshes and a equivalent triangular mesh with 16794 linear elements 8558 nodes. In this triangular mesh the target edge length is identical to the edge length in the NURBS meshes. This mesh size is selected because it is sufficiently accurate to draw meaningful conclusion, while on the other hand it is still coarse enough to see the numerical errors and judge the behaviour of the methods with the naked eye.

6.3. Projected redistancing

Figure 7 shows snapshots of the smooth Heaviside at maximum distortion for the different smoothing parameters κ for a 80x80 C_0 linear mesh.



(a) $\kappa_d = 0.1$ (b) $\kappa_d = 1$ (c) $\kappa_d = 10$

Figure 7: Contour plots of the smooth Heaviside for *projected redistancing* on a 80x80 C_0 linear mesh

It can be seen there is a delicate balance between not enough smoothing and to much smoothing. When κ is small there is not enough smoothing and the distance field shows oscillations, while for the higher values of κ the diffusion caused by the smoothing moves the interface more then desired. For the other discretisations – P_1 triangles and $C_1 - Q_2$ quadrilaterals – we get similar results, leading to the same conclusion.

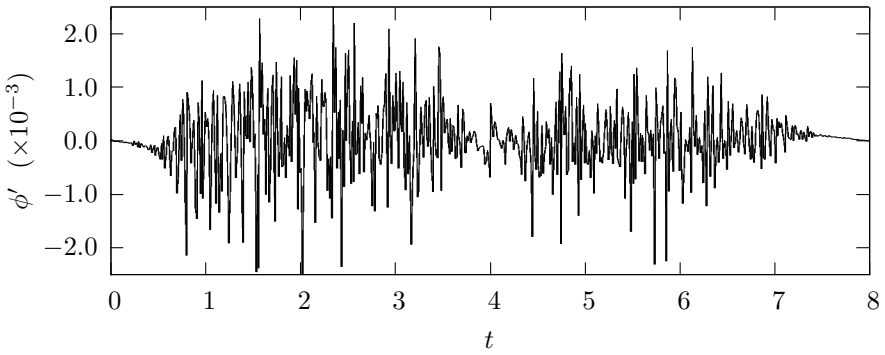
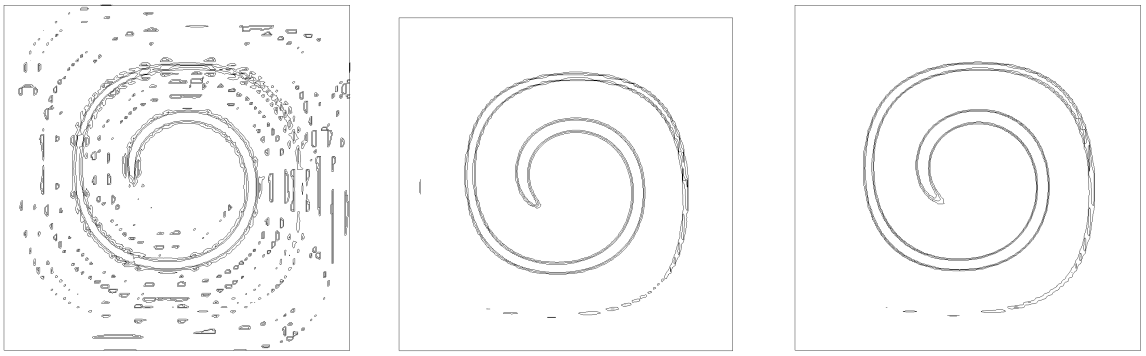


Figure 8: Correction required for volume conservation for the 80×80 linear mesh with $\kappa_d = 1$.

For all simulations the volume is conserved up to machine precision, even when the solution is highly deformed due to the smoothing. This demonstrates effectivity of the volume correction method presented in section 5.2. The correction required for volume conservation is of the order 10^{-3} per time step. A typical time trace of this correction is given in figure 8.

6.4. Projected scaling

Figure 9 shows snapshots of the smooth Heaviside at maximum distortion for the different smoothing parameters κ for a 80x80 C_0 linear mesh.



(a) $\kappa_d = 0.1$ (b) $\kappa_d = 1$ (c) $\kappa_d = 10$

Figure 9: Contour plots of the smooth Heaviside for *projected scaling* on a 80x80 C_0 linear mesh

It can be seen that for this approach there is no problem when a large smoothing parameter is used. However, when the smoothing parameter is too small the solution becomes spotty and irregular similar as in the previous section. This monotoon behaviour with respect to the smoothing parameter makes this a viable approach. For the other discretisations – P_1 triangles and $C_1 - Q_2$ quadrilaterals – we get similar results, leading to the same conclusion.

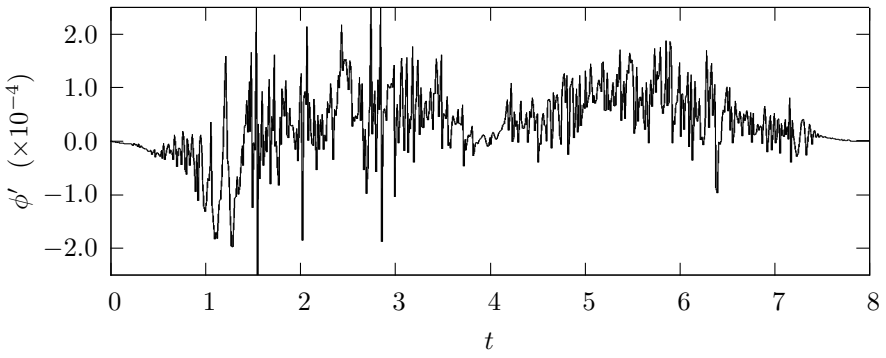
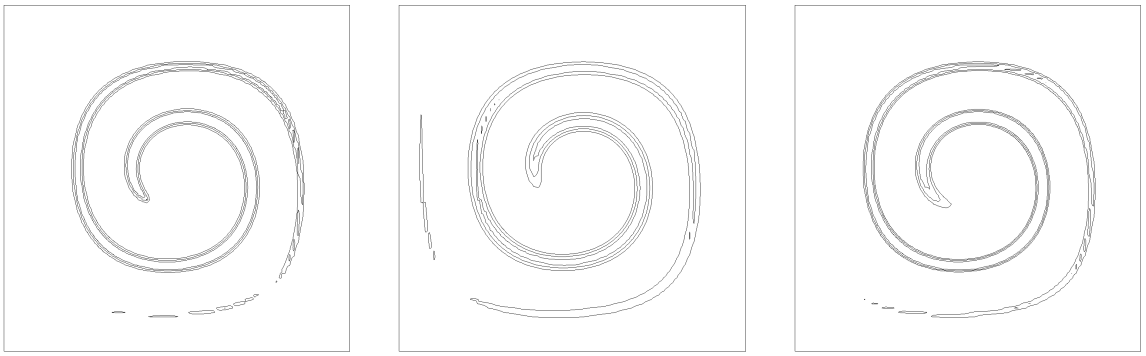


Figure 10: Correction required for volume conservation for the 80×80 linear mesh with $\kappa_d = 1$.

The interface conservative nature of the redistancing approach is also apparent in the time trace of the correction parameter required to conserve volume. A typical time trace is depicted in figure 10. When comparing with the time trace of the previous section, in figure 8, we see that the required correction is roughly an order of magnitude less.

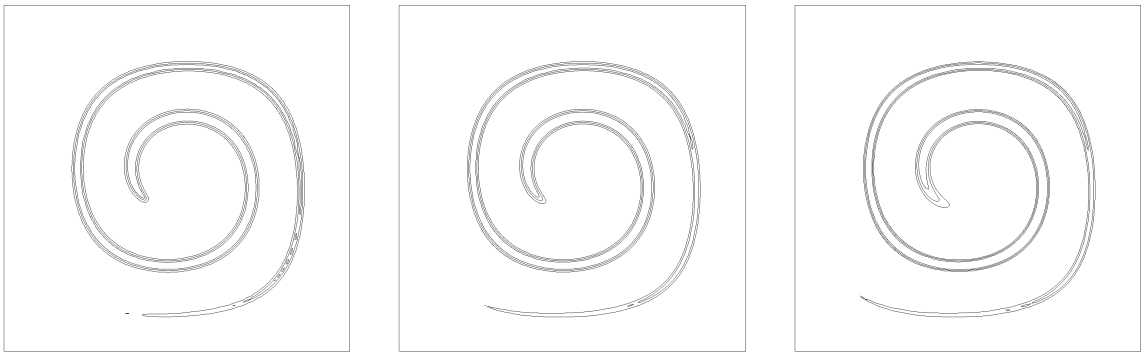
6.5. Projected inverse scaling

Figure 11 shows snapshots of the smooth Heaviside at maximum distortion for the different smoothing parameters κ for a 80x80 C_0 linear mesh. For this approach both the smoothing does not have a significant influence on the answers, both high and low smoothing limit give very acceptable answers.



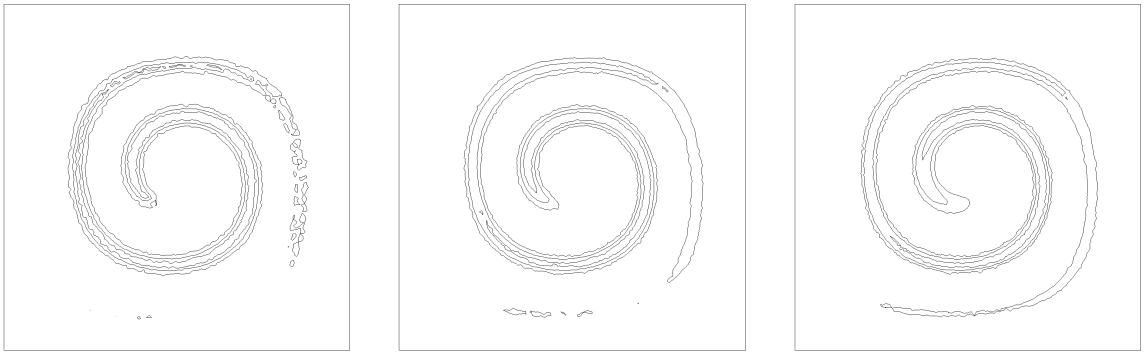
(a) $\kappa_d = 0.01$ (b) $\kappa_d = 1$ (c) $\kappa_d = 100$

Figure 11: Contour plots of the smooth Heaviside for *projected inverse scaling* on a 80x80 C_0 linear mesh



(a) $\kappa_d = 0$ (b) $\kappa_d = 1$ (c) $\kappa_d = 100$

Figure 12: Contour plots of the smooth Heaviside for *projected inverse scaling* on a 80x80 C_1 quadratic mesh



(a) $\kappa_d = 0.01$ (b) $\kappa_d = 1$ (c) $\kappa_d = 100$

Figure 13: Contour plots of the smooth Heaviside for *projected inverse scaling* on a triangular mesh with 4541 elements

Figure 12 shows the contours for the same smoothing range but now on the $C_1 - Q_2$ mesh. The results

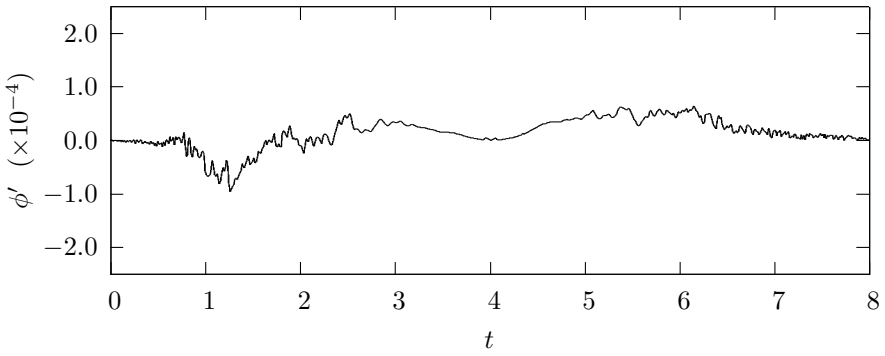


Figure 14: Correction required for volume conservation for the 80×80 linear mesh with $\kappa_d = 1$.

look very similar. For the triangular mesh the results, as shown in figure 13 , are less smooth due to unstructured nature of the mesh. Therefor smoothing has a more pronounced effect for this mesh, where the higher smoothing results in better and smoother answers. However, as shown in section 6.1 there is also an upper limit to the desired smoothing.

The better behaviour of the *projected inverse scaling* compared with the *projected scaling* is also apparent when we look at the time trace of the correction required for volume conservation, see figure14. This correction value is smaller, but also varies significantly less violently over time.

6.6. Convergence study

Using the results form the previous sections we conclude focus our attention only on the approaches that showed the most potential. This are is *projected inverse scaling* which is suitable for all mesh types. For the quadrilateral meshes, there is smoothing is absent $\kappa_d =$, while for the triangular mesh we use a relatively high smoothing of$\kappa_d = 10$. Results form the previous paragraph indicate these smoothing values lead to satisfying results.

In this section we will investigate the convergence of this approach. We look at two different measures for the correctness of the solution at the final time $T = 8$. First, we look at the area mismatch between actual and anticipated area indicated by the level-set. The anticipated value is the initial area at $T = 0$. This measure is computed as,

$$L_1(\hat{H}) = \int_\Omega |H(\hat{\phi}_T) - H(\hat{\phi}_0)| d\Omega \tag{47}$$

and is physically the most relevant in future application.

Additionally, we look how far the local distance field meandered from the anticipated value. Again the initial solution is used as reference. For this we use the following norm.

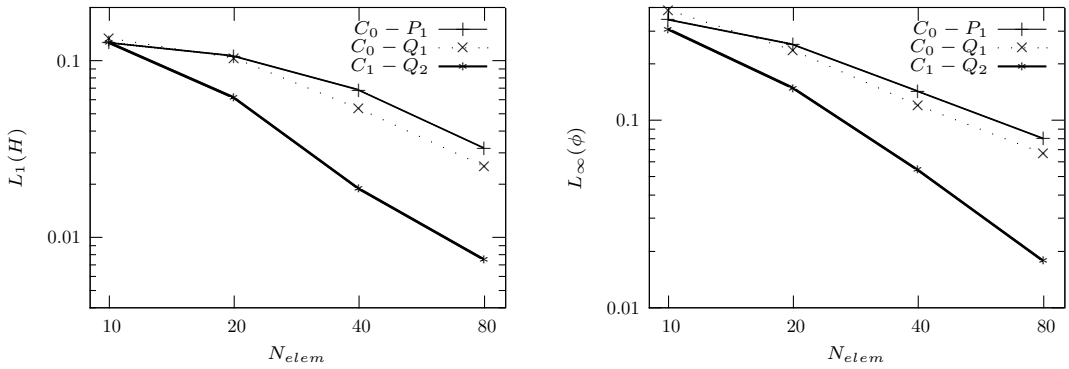
$$L_\infty(\phi) = \max_{\boldsymbol{x} \in \Omega} |\phi_T - \phi_0| \tag{48}$$

The convergence data for a sequence of meshes is give in table 1 and plotted in figure 15. For the triangular mesh the indicated element count is the target element count for the mesh generator. The resulting meshes have 242, 1054, 4262 and 16794 elements and 141, 568, 2212 and 8558 nodes, respectively.

The convergence results are slightly disappointing, for the linear meshes the convergence approaches first order for both norms, while for the quadratic NURBS mesh both norms converge with a slightly higher right of around 1.5. This is less than is reported in [7]. This might be explained by the local volume conservation routine used in that work, while in this paper a global volume conservation is employed.

	Linear triangle				Linear quad				Quadratic quad			
Elems	$L_1(\hat{H})$	Rate	$L_\infty(\phi)$	Rate	$L_1(\hat{H})$	Rate	$L_\infty(\phi)$	Rate	$L_1(\hat{H})$	Rate	$L_\infty(\phi)$	Rate
10	1.26e-1		3.44e-1		1.34e-1		3.88e-1		1.26e-1		3.06e-1	
20	1.06e-1	0.26	2.54e-1	0.44	1.03e-1	0.38	2.37e-1	0.71	6.19e-2	1.03	1.49e-1	1.04
40	6.80e-2	0.64	1.42e-1	0.84	5.37e-2	0.94	1.20e-1	0.98	1.89e-2	1.72	5.44e-2	1.45
80	3.20e-2	1.09	7.95e-2	0.84	2.52e-2	1.09	6.68e-2	0.85	7.46e-3	1.34	1.78e-2	1.61

Table 1: Convergence of the level-set method



(a) Mesh convergence in the $L_1(H)$-norm

(b) Mesh convergence in the $L_\infty(\phi)$-norm

Figure 15: Mesh convergences plots of the *projected inverse scaling* approach.

7. Conclusions

In this paper four approach to get a distance field in terms of element lengths are presented. These approaches attempted to solve two issues at the same time. To get truly smooth heavisides on arbitrary meshes, and to get a distance field without solving the difficult Eikonal equations directly.

In order to get smooth Heavisides on arbitrary meshes the standard Eikonal equation has been recast in terms of a mesh-metric, thereby taking the local meshsize into account when computing a distance field in terms of element lengths. Solving the Eikonal equation is circumvented by introducing a multiplicative scaling of the level-set. In this paper four different approaches for performing this scaling are investigated. This is done for linear triangular meshes, linear quadrilateral meshes and C_1-continuous quadratic meshes. Form this the following conclusions can be drawn:

- *Direct redistancing.*In this approach the original levelset function is corrected using local derivative information, resulting in a simple relation for the scaled distance field. However, this approach fails to deliver a smooth Heaviside, even on C_1 meshes. Therefore, despite its simplicity this approach has to be discarded.

- *Projected redistancing* In this approach the discontinuous scaled distance field from the *direct redistancing* approach is projected on a continous discretisation. This does corrects the smoothness issues but introduces other problems. On the distorted mesh the use of smoothing resulted in spuriously diverging contour lines at the boundary. Additionally, in the vortex problem insufficient smoothing results in a irregular solution while increasing the smoothing leads to excessive problems to conserve the interface location. Therefore, this approach has to be discarded as well.

- *Projected scaling* In this approach not the scaled distance field itself, but only the scaling is projected. This solves the problem of excessive interface movement at higher smoothing numbers. Unfortunately, low smoothing can still result in irregular solutions. Therefore we conclude that this is in principle a viable approach, but not the most desirable.

- *Projected inverse scaling* In this approach not the scaling but its reciprocal value is projected. In this approach the presence of smoothing plays a less prominent roll. In the absence of smoothing the approach still gives high quality solutions. Similar to the *projected scaling* approach the interface is precluded from moving. Therefore we conclude that this is approach has all the desired properties and is the approach of choice.

A convergence study using the selected approach is performed looking at the L_1 convergence of the Heaviside, this is effectively the mismatched area. Additionally we look at the L_∞ norm of the level-set. It is shown that the method is first order on when using linear triangles and quadrilaterals While on quadratic quadrilaterals the method is slightly higher order, approaching an order of 1.5. This slightly reduced order of accuracy might be due to the global volume conservation.

In the future the presented level-set approach will be to solve two-fluid flow including (flexible) floating objects. By avoiding a solution of the redistancing problem, the new approaches should facilitate stong coupling of the entire problem. Opening a way for strict energy control at the two-fluid interface.

Acknowledgements

The author is gratefully acknowledging support from Delft University of Technology and Durham University.

References

[1] I. Akkerman, Y. Bazilevs, D.J. Benson, and M.W. Farthing C.E.Kees. Free-surface flow and fluid-object interaction modeling with emphasis on ship hydrodynamics. *Journal of Applied Mechanics*, 2012.

[2] I. Akkerman, J. Dunaway, J. Kvandal, J. Spinks, and Y. Bazilevs. Toward free-surface modeling of planing vessels: Simulation of the Fridsma hull using ALE-VMS. *Computational Mechanics*, 50(6):719–727, 2012.

[3] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[4] C.A. Felippa, K.C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3247–3270, 2001.

[5] T.E. Tezduyar. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering*, 8(2):83–130, 2001.

[6] E. Walhorn, A. Kölke, B. Hübner, and D. Dinkler. Fluid-structure coupling within a monolithic model involving free surface flows. *Computers & Structures*, 83(25-26):2100 – 2111, 2005.

[7] I. Akkerman, Y. Bazilevs, C. Kees, and M. Farthing. Isogeometric analysis of free-surface flow. *Journal of Computational Physics*, 2011. Published online. doi:10.1016/j.jcp.2010.11.044.

[8] C. Kees, I. Akkerman, M. Farthing, and Y. Bazilevs. A conservative level set method suitable for variable-order approximations and unstructured meshes. *Journal of Computational Physics*, 2011. Published online. doi:10.1016/j.jcp.2011.02.030.

[9] E. Marchandise and J.-F Remacle. A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows. *Journal of Computational Physics*, 219(2):780 – 800, 2006.

[10] J.A. Sethian. Evolution, implementation, and application of level set and fast marching methods for advancing fronts. *Journal of Computational Physics*, 169:503–555, 2001.

[11] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[12] J.-L. Guermond and M. Nazarov. A maximum-principle preserving finite element method for scalar conservation equations. *Computer Methods in Applied Mechanics and Engineering*, 272:198 – 213, 2014.

[13] V. John and P. Knobloch. On spurious oscillations at layers diminishing (sold) methods for convection-diffusion equations: Part I - a review. *Computer Methods in Applied Mechanics and Engineering*, 196(17-20):2197 – 2215, 2007.

[14] J.-L. Guermond, R. Pasquetti, and B. Popov. Entropy viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 230(11):4248 – 4267, 2011. Special issue High Order Methods for {CFD} Problems.

[15] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W.D. Gropp, Dinesh Kaushik, M. G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, and H. Zhang. PETSc Web page. http://www.mcs.anl.gov/petsc.

[16] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P.Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L. Curfman McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.6, Argonne National Laboratory, 2015.

[17] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[18] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, Chichester, 2009.

[19] L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication), 2nd ed.* Springer-Verlag, New York, 1997.

[20] Y. Bazilevs, L. Beirao da Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16:1031–1090, 2006.

[21] J.A. Evans, Y. Bazilevs, I. Babuška, and T.J.R. Hughes. N-widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 198:1726–1741, 2009.

[22] Ch. Geuzaine and J.-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[23] W.J. Rider and D.B. Kothe. Reconstructing volume tracking. *Journal of Computational Physics*, 141:112–152, 1998.