8th International Conference on CFD in Oil & Gas, Metallurgical and Process Industries
SINTEF/NTNU, Trondheim NORWAY
21-23 June 2011

CFD11-117

CONSTRAINED REINITIALISATION OF THE CONSERVATIVE LEVEL SET METHOD

Claudio WALKER[1*], **Bernhard MÜLLER**[2†]

[1]NTNU Department of Energy and Process Engineering, 7491 Trondheim, NORWAY

[2]NTNU Department of Energy and Process Engineering, 7491 Trondheim, NORWAY

[*] E-mail: claudio.walker@ntnu.no
[†] E-mail: bernhard.muller@ntnu.no

ABSTRACT

In this paper it is shown that the reinitialisation of the conservative level function introduces spurious displacements of the interface. A method which prevents the interface in conventional level set methods from moving during the reinitialisation is adopted for the conservative level set method. It is shown that using the constrained reinitialisation with an adaptive distributed forcing retains the shape of the interface during reinitialisation. This eliminates the negative effects if too many reinitialisation steps are applied. However, our numerical experiments show that fixing of the interface during reinitialisation spoils the mass conservation of the conservative level set method.

Keywords: Multiphase flow, Level set method .

NOMENCLATURE

Greek Symbols

β	Forcing distribution factor.
Γ	Set of points addjacent to the interface.
δ	Least squares weights.
ε	Width parameter for the conservative level set function.
κ	Interface curvature.
ρ	Distance to domain center.
τ	Pseudo time.
ϕ	Conservative level set function.
Ψ	Stream function.
ψ	Signed distance function.

Latin Symbols

C	Set of points with nonzero foring term.
e	Error.
F	Forcing term.
f	Numerical flux.
h	Grid spacing.
$\mathbf{n}$	Interface normal.
m	Number of grid points.
t	Time.
S	Set of neighbouring points on opposit side of the interface.
$\mathbf{u}$	Velocity.
$\mathbf{x}$	Position.

Sub/superscripts

α	Index α.
i	Index i.
j	Index j.
n	Discrete time level.

INTRODUCTION

The level set method (LSM) is a popular method to describe the location of the interface in multiphase flow computations. It represents the interface with the help of a signed distance function which is advected by the fluid velocity. For a more thorough discussion of the LSM and its application to multiphase flow we refer the reader to the review by (Sethian and Smereka, 2003). This representation has the advantage of relatively simple calculations of interface normals and curvatures. Another often cited advantage of the LSM is that the parallelisation is straightforward. Due to the advection the level set function loses its signed distance property. Therefore it has to be reinitialised after a few advection steps. This is done by solving a reinitialisation equation. The deformation of the interface during this reinitialisation is a known problem and an explanation of the cause and a possible remedy was given by (Russo and Smereka, 2000). There exists a number of methods to reduce the displacement of the signed distance function during reinitialisation. Among these methods is the Constrained Reinitialisation (CR) (Hartmann *et al.*, 2008), which is minimising the displacement of the intersection points between the grid lines and the zero level set contour.

However, the LSM has an important disadvantage, it does not conserve the mass of the two fluids. Different approaches have been developed to satisfy the mass conservation of the level set method. Examples include the conservative level set method (CLSM) (Olsson and Kreiss, 2005) (Olsson *et al.*, 2007), the particle level set method (PLS) (Enright *et al.*, 2002) and the coupled level set/volume-of-fluid (CLSVOF) (Sussman and Puckett, 2000). The added complexity for both PLS and CLSVOF are significant. On the other hand the conservative level set method improves the mass conservation and keeps the simplicity of the original method.

We discovered that during the reinitialisation of the CLSM the interface is displaced considerably. In most applications of the CLSM this problem is not evident since typically only a few reinitialisation steps are conducted, and the deformation becomes only significant for high numbers of reinitialisation steps. There is no general rule on how frequent the CLSM has to be reinitialised and how many reinitialisation steps should be applied each time. It is therefore important

to make sure that too many reinitialisation steps do not compromise the accuracy of the CLSM. Recently Hartmann et al. published a technique called high-order constrained reinitialisation (HCR) (Hartmann *et al.*, 2010) for the conventional LSM. But its implementation through a source term allows the adoption of HCR to do a constrained reinitialisation of the CLSM.
In the present paper a short introduction to the conservative level set method and its discretisation is given. Then we present an overview of the HCR and show how it can be adopted for the CLSM. Finally we show some numerical experiments and explain why it is after all not advisable to fix the interface during the reinitialisation of the CLSM.

CONSERVATIVE LEVEL SET METHOD

In level set methods the interface is defined as the iso contour of a smooth function. For ordinary level set methods this function is the signed distance from the interface, and the interface location is where the distance function is zero. The conservative level set function replaces the distance function by a hyperbolic tangent function ϕ with values between zero and one (Olsson and Kreiss, 2005). The position of the interface is located at the $\phi = 0.5$ contour line. Since we have smooth functions which are defined in the entire computational domain in both cases, we can easily extract additional geometrical information about the interface. For example the interface normal $\mathbf{n}$ and the curvature κ are defined as

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \tag{1}$$

and

$$\kappa = \nabla \cdot \mathbf{n}. \tag{2}$$

The interface is transported simply by advecting the level set function using the advection equation $\phi_t = -\mathbf{u}\cdot(\nabla\phi)$. If we have a divergence free velocity field, as it is the case for incompressible flow, the interface transport can be written as a conservation law.

$$\frac{\partial\phi}{\partial t} + \nabla\cdot(\mathbf{u}\phi) = 0 \tag{3}$$

Since all numerical methods will introduce an error as ϕ is advected, it will lose its hyperbolic tangent shape. The diffusion of the advection schemes will increase the distance in which ϕ rises from zero to one. (Olsson *et al.*, 2007) propose the following reinitialisation equation to force ϕ back to its hyperbolic tangent shape, which is solved to steady state with respect to the pseudo time τ:

$$\frac{\partial\phi}{\partial\tau} + \nabla\cdot(\phi(1-\phi)\hat{\mathbf{n}} - \varepsilon((\nabla\phi\cdot\hat{\mathbf{n}})\hat{\mathbf{n}})) = 0, \tag{4}$$

where $\hat{\mathbf{n}}$ is the normal at the beginning of the reinitialisation, and ε determines the width of the hyperbolic tangent. It is important to note that also the reinitialisation equation is a conservation law. The first flux term causes a compression of the profile, whereas the second term is a diffusive flux. By multiplications with the normal $\hat{\mathbf{n}}$ there are only fluxes in the direction of the normal. This forced flux direction for both the compression and the diffusion term are essential to improve the mass conservation of the method. Indeed it was shown that the CLSM conserves mass as ε approaches 0. To illustrate the nature of Equation (4), we use a one-dimensional example. Suppose that the interface is located at $x = 0$. Then the normal reduces to $n = -1$ or $n = 1$. In the example we use the latter. In this case a steady state solution to Equation (4) is:

$$\phi = \frac{1}{2}\left(1 + \tanh\left(\frac{x}{2\varepsilon}\right)\right). \tag{5}$$

The solution is shown Figure 1, together with the compression and the diffusion terms. It is clearly visible that at steady state the compression and the diffusion are balanced. If ϕ would be too diffusive the compression term would outweigh the diffusive term and ϕ would be forced back to the steady state solution.

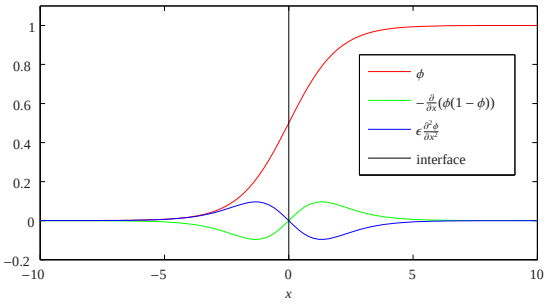

Figure 1: Illustration of the reinitialisation equation.

Discretisation

Advection

The divergence in the advection equation (3) is discretised by a 5th order finite difference weighted essentially non-oscillatory (WENO) scheme using a Lax-Friedrichs flux splitting (Shu, 1998). For time discretisation the 3rd order TVD Runge-Kutta method by (Shu and Osher, 1988) is applied.

Interface normal

In order to keep the stencils for the reinitialisation equation as small as possible (Desjardins *et al.*, 2008) proposed to compute the fluxes of the reinitialisation equation (4) at the cell faces. As a consequence the normal is required at the cell faces. It was also proposed by the authors to compute the normal from a signed distance function ψ which in turn is computed using a fast marching method (FMM). The reason for this is that the gradient of the conservative level set function ϕ becomes extremely small far away from the interface. In this region with small gradients small errors in ϕ can lead to a normal pointing in the wrong direction. Close to the interface, however, the gradient of ϕ is large enough such that a reliable normals can be computed directly form ϕ. Further away from the interface the accuracy of the normal is not so important anymore, but they should be continuous and not contain spurious oscillations. In the present work the normal are computed directly form the conservative level set function where $0.0001 \leq \phi \leq 0.9999$, otherwise they are computed from a signed distance function.
The gradient for the normal on the cell faces in x-direction at $x = x_{i-1/2,j}$ is approximated by the following stencil:

$$(\nabla\phi)_{x\,i-1/2,j} \approx \frac{\phi_{i,j} - \phi_{i-1,j}}{h} \tag{6}$$

$$(\nabla\phi)_{y\,i-1/2,j} \approx \frac{\phi_{i-1,j+1} + \phi_{i,j+1} - \phi_{i-1,j-1} - \phi_{i,j-1}}{4h}. \tag{7}$$

The gradient at the cell faces in y-direction $(\nabla\phi)_{i,j-1/2}$ is treated analogously. Finally the normal can be computed

from the gradient using equation (1). Note grid points where $|\nabla\phi| = 0$ have to be treated separately, here we chose to fill the normal at these points with a unit vector pointing in an arbitrary direction.

Reinitialisation

A central difference method on a staggered grid is used to discretise the reinitialisation equation (4). The fluxes in x-direction as well as those in y-direction are computed on the corresponding cell faces. First the convective fluxes are obtained at the cell centers from the level set function i.e. $f_{c\,i,j} = \phi_{i,j}(1-\phi_{i,j})$ and then they are interpolated to the cell faces using linear interpolation e.g. $f_{c\,i+1/2,j} = \left(f_{c\,i,j} + f_{c\,i+1,j}\right)/2$. The diffusive fluxes are computed directly at the cell faces to keep the stencil as small as possible. Accordingly the diffusive fluxes at the cell faces in x-direction read:

$$f_{d\,i-1/2,j} = \varepsilon\left((\nabla\phi)_{x\,i-1/2,j}\cdot n_{x\,i-1/2,j} + (\nabla\phi)_{y\,i-1/2,j}\cdot n_{y\,i-1/2,j}\right), \tag{8}$$

where $n_{x\,i-1/2,j}$ and $n_{y\,i-1/2,j}$ are the x- and y-components of the interface normal. This leads to the total fluxes in x- and y-directions:

$$f_{i-1/2,j} = \left(f_{c\,i-1/2,j} - f_{d\,i-1/2,j}\right)\cdot n_{x\,i-1/2,j} \tag{9}$$

$$g_{i,j-1/2} = \left(f_{c\,i,j-1/2} - f_{d\,i,j-1/2}\right)\cdot n_{y\,i,j-1/2}\,. \tag{10}$$

Eventually the total residual of the reinitialisation equation is:

$$\frac{\partial\phi}{\partial\tau} \approx -\frac{f_{i+1/2,j} - f_{i-1/2,j}}{h} - \frac{g_{i,j+1/2} - g_{i,j-1/2}}{h}. \tag{11}$$

The forward Euler scheme is used to advance the conservative level set function in the pseudo time τ. To ensure stability the time step must fulfill:

$$\Delta\tau\left(\frac{2}{h} + \frac{4}{h^2}\varepsilon\right) \leq 1. \tag{12}$$

CONSTRAINED REINITIALISATION

We observed that the convergence of the reinitialisation with the described discretisation was poor for simple cases. In Figure 2 the result of the reinitialisation of a circle is shown. During the first few iterations the residual drops fast and then the convergence is slowed down. During this period of small convergence rate the interface develops towards a rhombus. This tendency to deform the interface is not reduced as the grid is refined. In most applications this defect is not evident since typically only a few reinitialisation steps are conducted and the residual is not reduced until the numerical steady state.

Constrained reinitialisation for conventional level set methods

A similar problem appears in conventional level set methods. Recently (Hartmann *et al.*, 2010) proposed a method to reduce the displacement of the signed distance function ψ during reinitialisation. The idea is to add a source term to the residual of the differential equation, which minimises the displacement of the intersection points between the zero contour of ψ and the grid lines in a least squares sense.

If two grid points $\mathbf{x}_{i-1,j}$ and $\mathbf{x}_{i,j}$ which are located on opposite sides of the interface, the condition that the linear interpolation of the intersection point between the zero contour line and the grid line between those two points does not move during reinitialisation, can be reduced to $\frac{\tilde{\psi}_{i,j}}{\tilde{\psi}_{i-1,j}} = \frac{\psi_{i,j}}{\psi_{i-1,j}}$, where $\tilde{\psi}$ and ψ are the signed distance functions before and after reinitialisation, respectively. In general a grid point can have several neighbours which are on the opposite side of an interface. The previous condition cannot be fulfilled for all involved neighbours, since the problem is overdetermined. Let $S_{i,j}$ be the set of all neighbouring grid points of $\mathbf{x}_{i,j}$ which are on the opposite side of the interface, and $M_{i,j}$ the number of grid points in $S_{i,j}$. Further we denote an arbitrary point in $S_{i,j}$ by $\mathbf{x}_{(i,j)_\alpha}$ such that $\alpha = 1...M_{i,j}$. In the constrained reinitialisation CR-1 (Hartmann *et al.*, 2010) the following least squares function is minimised.

$$L_{i,j} = \sum_{\alpha=1}^{M_{i,j}} \delta_{i,j}^{\alpha}\left(\psi_{i,j} - \psi_{(i,j)_\alpha}\cdot r_{(i,j)_\alpha}^{i,j}\right)^2, \tag{13}$$

$$r_{(i,j)_\alpha}^{i,j} = \frac{\tilde{\psi}_{i,j}}{\tilde{\psi}_{i-1,j}} \tag{14}$$

(Hartmann *et al.*, 2010) chose the weights $\delta_{i,j}^{\alpha} = 1$. If $L_{i,j}$ is derived with respect to $\psi_{i,j}$ and this is set equal to zero we get the target value for the distorted signed distance function, such that it minimises the displacement of the interface.

$$T_{i,j} = \frac{\sum_{\alpha=1}^{M_{i,j}} \delta_{i,j}^{\alpha}\left(\psi_{(i,j)_\alpha}\cdot r_{(i,j)_\alpha}^{i,j}\right)}{\sum_{\alpha=1}^{M_{i,j}} \delta_{i,j}^{\alpha}} \tag{15}$$

Finally the CR-1 correction term at the nth reinitialisation time step is formulated as the difference between $T_{i,j}^n$ and $\psi_{i,j}^n$:

$$F_{i,j}^n = \frac{\beta_{i,j}}{h}\left(\psi_{i,j}^n - T_{i,j}^n\right). \tag{16}$$

Here $\beta_{i,j}$ is a coefficient which distributes the correction between neighbouring grid points. For consistency the sum $\beta_{i,j} + \beta_{(i,j)_\alpha}$ must always be equal to 1. Setting $\beta_{i,j} = 0.5$ corresponds to do half of the correction on either side of the interface. Since the forcing leads to an instability when one of points of the set $S_{i,j}$ changes its sign during reinitialisation,

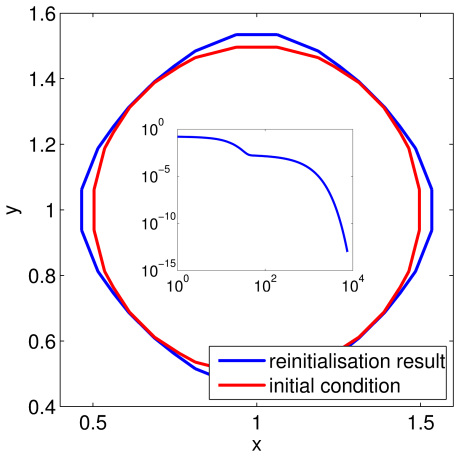

Figure 2: Reinitialisation result of a circle with 8 grid points per diameter, the small insert shows the evolution of the reinitialisation residual over the number of time levels.

the forcing is not applied at the grid points in this particular $S_{i,j}$.

Adaptation to the conservative level set method

The idea is to apply the CR-1 forcing to the reinitialisation of the CLSM to avert an undesirable movement of the interface during reinitialisation. In the CLSM the interface is given as the 0.5 contour line. Therefore ψ in equation (13) to (16) must be replaced by $\phi^- = \phi - 0.5$.

Stabilising CR-1

An essential difference between the signed distance function ϕ and the conservative level set function ψ is that the latter has larger gradients at the interface. As a consequence the forcing term for the CLSM tends to be stronger. For points where the distance to the interface approaches h the magnitude of the forcing term is increasing rapidly. It was observed in numerical experiments that if the distance between the intersection point and the grid point becomes smaller than $0.01h$ the constrained reinitialisation of ϕ becomes unstable. Imagine two bars which cross each other. Suppose one is only allowed to move one end of one bar to adjust the position of the crossing point. If now the crossing point is close to the fixed end of the moving bar, the free end has to be moved relatively far to obtain a given displacement of the crossing point. On the other hand if one could move the end which is closer to the crossing point a much smaller movement is required to obtain the same displacement of the crossing point. The correction can also be divided to both ends where the necessary corrections at each end are multiplied by factors which have to add to one. These factors are the distribution factors $\beta_{i,j}$.
In order to stabilise the constrained reinitialisation of the CLSM the forcing for pairs where the crossing point is close to one of the points should be distributed better, which means adjusting $\beta_{i,j}$ in equation (16). For pairs where both neighbours have $M_{i,j} = 1$, $\beta_{i,j}$ can be chosen such that the forcing term has the same magnitude on both sides. As an example if both $M_{i,j}$ and $M_{i-1,j}$ are equal to 1 and the crossing point is either close to $\mathbf{x}_{i,j}$ or $\mathbf{x}_{i-1,j}$ set

$$\beta_{i,j} = \left(\frac{\psi_{i,j} - T_{i,j}}{\psi_{i-1,j} - T_{i-1,j}} + 1 \right) \tag{17}$$

$$\beta_{i-1,j} = 1 - \beta_{i,j}. \tag{18}$$

This choice of distribution coefficients ensures that the forcing term is equal for both points.
In general $M_{i,j} > 1$ especially for points which are close to the interface. There $\beta_{i,j}$ cannot be adjusted to an arbitrary value since $\beta_{i,j} + \beta_{(i,j)_\alpha} = 1$, and the set $S_{i,j}$ often overlaps with the sets from the neighbouring points. Therefore setting $\beta_{i,j}$ gives a condition to a number of β from neighbouring grid points. A solution to keep the effect of choosing $\beta_{i,j}$ local is to use the weights $\delta^{\alpha}_{i,j}$ in the least squares function (13). If a grid point $\mathbf{x}_{i,j}$ is too close to an intersection point $\beta_{i,j}$ is set to 1. This would require to set all $\beta_{(i,j)_\alpha} = 0$. Instead we choose $\delta^{\hat{\alpha}}_{(i,j)_\alpha} = 0$ in all the neighbouring cells where $\hat{\alpha}$ points to cell (i,j). That is equivalent to do the complete forcing on the grid point close to the intersection and ignoring the constraint at all the neighbouring points.

Summary of the constrained reinitialisation

Only at points adjacent to the interface the operations for the level set forcing have to be conducted, we define therefore the set

$$\Gamma = \left\{ \mathbf{x}_{i,j} \colon \left(\phi^-_{i,j}\phi^-_{i',j} \le 0 \right) \vee \left(\phi^-_{i,j}\phi^-_{i,j'} \le 0 \right) \right\}, \tag{19}$$

where $i' \in \{i-1, i+1\}$ and $j' \in \{j-1, j+1\}$. For sets $S_{i,j}$ where ϕ at one of the points changes its sign during reinitialisation the forcing is not applied. To that effect the following set of grid points will have a forcing term $F_{i,j} \neq 0$:

$$C^n = \left\{ \mathbf{x}_{i,j} \in \Gamma \colon \phi^{-n}_{i,j}\phi^{-n}_{(i,j)_\alpha} < 0 \, \forall \alpha \in \{1, ..., M_{i,j}\} \right\}. \tag{20}$$

Using these definitions the steps for the constrained reinitialisation of the CLSM are:

1. Compute the interface normal. and (7))
2. Compute the shifted conservative level set function before reinitialisation $\tilde{\phi}^- = \tilde{\phi} - 0.5$.
3. Find all points in Γ.
4. Update the set $S_{i,j}$, and compute $r^{i,j}_{(i,j)_\alpha}$ for all points in Γ.
5. Find points where $r^{i,j}_{(i,j)_\alpha} < \text{TOL}$
   - If $M_{i,j} = M_{(i,j)_\alpha} = 1$, flag points for equal forcing
   - Else, set $r^{i,j}_{(i,j)_\alpha} = 0$, reduce $M_{i,j}$ by 1 and flag $\mathbf{x}_{i,j}$ as point with $\beta_{i,j} = 1$
6. For all points with $\beta_{i,j} = 1$, do for all (i',j') where $\mathbf{x}_{(i',j')_{\hat{\alpha}}} = \mathbf{x}_{i,j}$, $\delta^{\hat{\alpha}}_{i',j'} = 0$ and reduce $M_{i',j'}$ by 1 where $\mathbf{x}_{(i',j')_{\hat{\alpha}}} = \mathbf{x}_{i,j}$.
7. Remove points $\mathbf{x}_{i,j}$ where $M_{i,j} = 0$ from Γ.
8. Solve the constrained reinitialisation problem by performing the following steps for each iteration:
   (a) Compute the shifted conservative level set function at pseudo time step n: $\phi^{-n} = \phi^n - \frac{1}{2}$.
   (b) Update the set C^n
   (c) For all grid points in C^n compute the forcing terms given by equation (16), use $\beta_{i,j} = 0.5$ unless the point is flagged for $\beta_{i,j} = 1$ or equal correction.
   (d) Compute residual from the PDE (4).
   (e) Advance ϕ one pseudo time step.

NUMERICAL EXPERIMENTS

Reinitialisation

To test the constrained reinitialisation method described in the previous section we consider a circle with radius $R = \frac{1}{2}$ which is placed in the center of a square domain with unit length. The initial condition is $\tilde{\phi} = \left(1 + \exp\left(\frac{3(\rho - R)}{2\varepsilon}\right)\right)^{-1}$, where $\rho = \sqrt{\left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2}$ is the distance from the center of the domain. For this case the exact steady state solution of equation (4) is:

$$\phi_{\text{exact}} = \left(1 + \exp\left(\frac{\rho - R}{\varepsilon}\right)\right)^{-1}. \tag{21}$$

The reinitialisation equation was iterated until the 2-norm of the residual was below 10^{-13}. The error norm is computed as follows:

$$e = \frac{1}{m}\left(\sum_{i,j=1}^{m}\left(\phi_{\text{exact}\,i,j} - \phi_{i,j}\right)^2\right)^{\frac{1}{2}}, \qquad (22)$$

where m is the number of grid points in each direction. To assess the deformation of the interface the reinitialised ϕ was interpolated to a fine grid with 500 points in each direction using bi-cubic splines. On this fine grid the position of all intersection points between the grid lines and the 0.5 contour line were computed using a 1-dimensional cubic spline interpolation. The position error of the interface e_p is then given as the 2-norm of the distance between each interpolated intersection point and its correct position.

Analytic normal

The errors after the reinitialisation with analytic normal and a fixed $\varepsilon = 0.044$ are presented in Figure 3. The error norm e is reduced with second order which is consistent with the discretisation scheme. To find the position of the interface we need to interpolate it. This interpolation is the reason that the position error of the interface is one order higher than the order of the discretisation scheme. Special attention should be payed to the results with $m = 81$. At this grid resolution 16 grid points were marked since their distance to the interface was smaller than $0.01h$. These grid points where treated with the procedure explained previously. Without this improved distribution of the forcing term the reinitialisation is unstable for $m = 81$. As it can be seen in Figure 3, the convergence rate of the position error is slightly reduced due to the unequal distribution of the forcing term. For comparison the errors without the CR-1 are plotted in Figure 3 as well. Since the interface is developing a rhombus shape (see Figure 2) at all grid resolutions neither the error norm nor the position error are converging with grid refinement. In Table 1 we list the number of reinitialisation steps which were required to reduce the residual to the target value of 10^{-13}. The constrained reinitialisation accelerates the convergence of the residual significantly. For the unconstrained reinitialisation of the circle with $m = 64$ and $m = 81$ the target residual was not reached after 100000 reinitialisation steps. The residuals at $n = 100000$ were $1.55 \cdot 10^{-5}$ and $4.03 \cdot 10^{-5}$ for $m = 64$ and $m = 81$, respectively.

Table 1: Number of iterations required to reach a residual of 10^{-13} during the reinitialisation of a circle.

m	8	16	32	64	81
CR-1	165	318	943	3654	4760
without	372	7536	96637	> 100000	> 100000

Numeric normal

The same test as in the previous section is repeated, but the interface normal is computed numerically. For a constant $\varepsilon = 0.044$ the results are almost identical to the reinitialisation with the analytic interface normal. Since the mass conservation of the CLSM is improved as ε is decreased, the reinitialisation of the circle was repeated once with $\varepsilon = 0.2\sqrt{h}$ and with $\varepsilon = 0.6h$. As ε is decreased the width of tangent hyperbolic is also decreased, which means that there are less grid points resolving the area where ϕ changes from 0 to 1. The result is that the convergence rate is reduced if ε is decreasing together with the grid width. The errors for simultaneous reduction of h and ε are shown in Figure 4.

Advection and Reinitialisation

Initially a circle of radius $R = 0.1$ is placed at $(0.5, 0.7)$ in a square domain with unit length. The circle is advected with an external velocity field with the stream function

$$\Psi = -\pi\left(x^2 + y^2 - x - y\right). \qquad (23)$$

This velocity field is advecting the circle anticlockwise around the center of the computational domain. To keep the error contribution from the temporal discretisation negligible a small CFL number of 0.05 was chosen for all examples. After 100 time steps a reinitialisation was performed. During the reinitialisation the residual was reduced to 10^{-8} or a maximum of 500 iterations were done. The convergence of the errors for the constrained CLSM is plotted in Figure 5 for

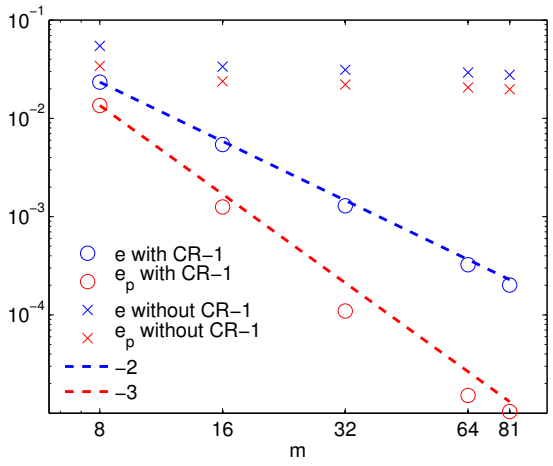

Figure 3: Error convergence for the reinitialisation of a circle with analytic interface normal and $\varepsilon = 0.044$.

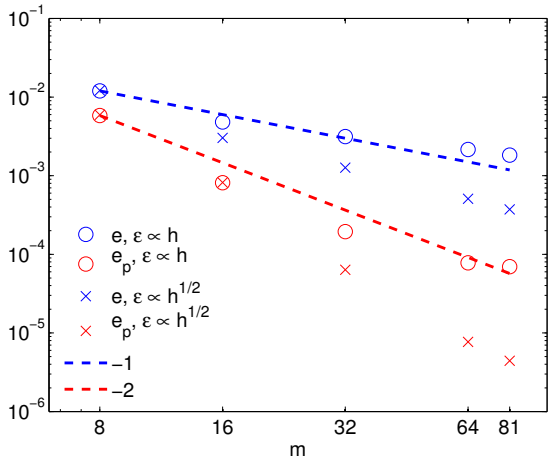

Figure 4: Error convergence for the reinitialisation of a circle with numeric interface normal and $\varepsilon \propto h$ or $\varepsilon \propto \sqrt{h}$.

$\varepsilon = 0.1 \cdot m^{-1/2}$. Figure 6 shows the interface position after one revolution for $m = 51$. For the case where the reinitialisation was done with CR-1 the interface retains a circular shape but the area is reduced significantly. On the other hand, if the unconstrained reinitialisation is used, the area is conserved but the interface loses its circular shape. Here we exaggerated this effect by the large number of reinitialisation steps. If e.g. every 100 advection steps only 10 reinitialisation steps are conducted, the interface position error is reduced by almost a factor of 16. If the interface is only advected and never reinitialised, the area loss is smaller than for the CR-1 reinitialisation and the circular shape is retained. The area loss for the three different reinitialisations is summarised in Table 2. The reduction of ε leads only for the CLSM without CR-1 to a reduction in the area change. For the constrained CLSM and if the circle is only advected, the area change decreases with increasing ε. This is due to the fact that the conservative level set function is less steep for larger ε, and therefore the diffusive error of the advection scheme is reduced.
The large mass loss for the constrained reinitialisation can be explained as follows. Initially the level set function can be compared with a cylinder with height 1 and the radius of the circle. If the circle has a radius of 1 then the volume of the cylinder is π. Since the numerical advection is diffusive, we assume the advected cylinder will transform to the shape of a cone with height of 1. The conservative advection will conserve the volume of the cylinder. For this reason the radius of the cone at a height of 0.5 will be only $\frac{\sqrt{3}}{2}$ instead of 1. During a constrained reinitialisation a new cylinder with this smaller cross section will be created and the cycle starts again.

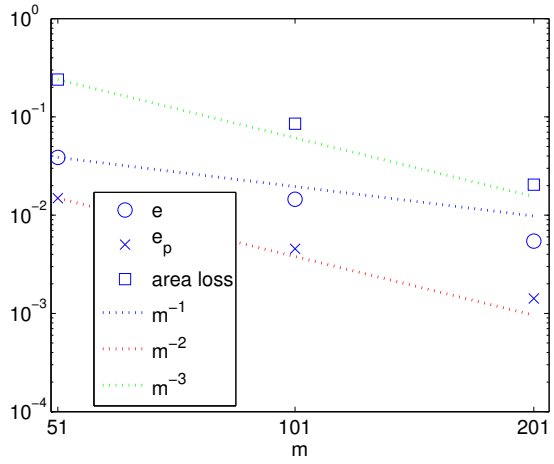

Figure 5: Errors for the advected circle after one revolution.

Table 2: Area change in % for the advected circle with $m = 51$.

ε	0.1	0.2	0.3
CLSM CR-1	−23.98	−11.14	−4.93
CLSM	0.10	0.60	0.65
Advection	14.49	5.60	1.43

CONCLUSION

We applied the constrained reinitialisation CR-1 by (Hartmann *et al.*, 2010) to the reinitialisation of the conservative level set function. The larger modulus of the gradient of the conservative level set function can cause instabilities during the constrained reinitialisation, if the interface is located close to a grid point. This instability can be avoided, if the forcing is not divided equally between two neighbouring points. Evidence was given that the constrained reinitialisation prevents spurious interface deformation during the reinitialisation independent of the number of reinitialisation steps, and that the convergence of the residual is accelerated. However, numerical experiments showed that preventing the interface from moving during reinitialisation spoils the mass conservation of the CLSM. It was also shown that care has to be taken that the number of reinitialisation steps of the unconstrained CLSM is not too large such that the reinitialisation introduces spurious interface deformations.

REFERENCES

DESJARDINS, O. *et al.* (2008). "An accurate conservative level set/ghost fluid method for simulating turbulent atomization". *Journal of Computational Physics*, **227(18)**, 8395–8416.

ENRIGHT, D. *et al.* (2002). "A hybrid particle level set method for improved interface capturing". *Journal of Computational Physics*, **183(1)**, 83–116.

HARTMANN, D. *et al.* (2008). "Differential equation based constrained reinitialization for level set methods". *Journal of Computational Physics*, **227(14)**, 6821 – 6845.

HARTMANN, D. *et al.* (2010). "The constrained reinitialization equation for level set methods". *Journal of Computational Physics*, **229(5)**, 1514 – 1535.

OLSSON, E. and KREISS, G. (2005). "A conservative level set method for two phase flow". *Journal of Computational Physics*, **210(1)**, 225–246.

OLSSON, E. *et al.* (2007). "A conservative level set method for two phase flow II". *Journal of Computational Physics*, **225(1)**, 785–807.

RUSSO, G. and SMEREKA, P. (2000). "A remark on computing distance functions". *Journal of Computational Physics*, **163(1)**, 51 – 67.

SETHIAN, J.A. and SMEREKA, P. (2003). "Level set

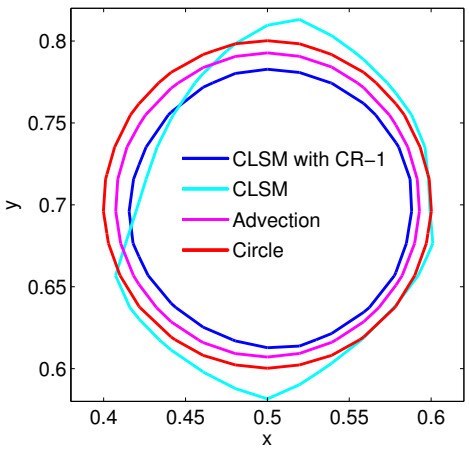

Figure 6: Advected circle, interface position after one revolution.

methods for fluid interfaces". *Annual Review of Fluid Mechanics*, **35(1)**, 341–372.

SHU, C.W. (1998). *Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws*, in: A. Quarteroni (ed.), Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, 325–432. Springer Berlin / Heidelberg.

SHU, C.W. and OSHER, S. (1988). "Efficient implementation of essentially non-oscillatory shock-capturing schemes". *Journal of Computational Physics*, **77(2)**, 439 – 471.

SUSSMAN, M. and PUCKETT, E.G. (2000). "A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows". *Journal of Computational Physics*, **162(2)**, 301–337.