Analysis of the continuum surface force method

Delft University of Technology

Shang-Jen Wang (4215974)

Supervisors:
Dr. D.R. van der Heul and Dr. S. Kenjeres

2015

Abstract

The subject of this report is the analysis of the original Continuous Surface Force (CSF) model introduced by Brackbill for modelling surface tension between two different fluids. The method replaces the infinitely small interface by a transition region, which alleviates the interface boundary conditions and replaces the surface force on the interface by a body force in the interface region. This method is widely used in models for liquid-liquid and liquid-gas two-phase flow. In the two-phase flow models developed in the Scientific Computing group of the Delft Institute for Applied Mathematics (DIAM) the strict application of this method was identified as one of the causes for an observed discrepancy between the experimentally observed and theoretically predicted rise speed of a single gas bubble. In this report the mathematical correctness of the derivation of the CSF model and some of its assumptions are verified.

Different interpretations of the model reported in literature are discussed that differ from the formulation proposed by Brackbill in two aspects: In many models Brackbill's density weighting is not applied. Often Brackbill's assumption that the interface curvature can be replaced by the curvature of the local indicator isosurface is not made.

The density weighting that Brackbill introduces will make sure that the neighbouring isosurfaces of the indicator function remain equidistant from the interface when the surface force is applied. Here it is shown that for an interface with finite thickness this results in a dependence on the density difference between the two adjacent fluids for the jump in the pressure across the interface, which contradicts the Young-Laplace equation.

Brackbill states the curvature of the interface in the body force formulation can be replaced by the curvature of a local isosurface. It is shown that contrary to this the use of the local curvature can result in an error that is 10 times larger than when a suitable interpolation technique is applied to obtain the interface curvature.

It is recommended not to use the density weighting and the use of the local curvature in the DIAM two-phase flow model is replaced by the techniques proposed in this report.

Contents

List of Figures

Chapter 1

Introduction

In many industrial processes, the liquid-liquid two-phase flow play a fundamental role. For instance bubbles in a pipeline experience the liquid-liquid two-phase flow. The simulation of the two-phase flow is still actively researched and in Delft they have researched this for over 15 years and are still researching this.

Surface tension is a very important factor in two-phase flow for computing the ascent speed of a bubble. Only if the effects of the surface tension is modelled accurately will the bubble have the correct shape and the right ascent speed. Surface tension arises when two different kind of fluids make contact, this is due to the unbalance created by the difference in molecular forces at the interface. Surface tension also arises when the fluid is contact with a wall, for instance the wall of a pipeline.

On the interface a number of conditions hold of which the most important is the pressure being discontinuous on the interface. This makes modelling the effects due to surface tension quite complex, because the model has to satisfy the condition on the interface. The original CSF model introduced by Brackbill interprets the surface tension as a continuous volume force in the direct vicinity of the interface. This method alleviates the interface condition, thus making this method widely used.

The MCLS algorithm is a method to simulate two phase flow which includes the original CSF model and is being developed in the Scientific computing group of DIAM. The Scientific group of DIAM is working on modelling rising bubbles in a vertical pipe, but the right ascent speed of the bubble is not achieved. A reason could be the currently applied original CSF model is not correct. So the correctness of the original CSF model has to be investigated.

The research questions are:

1. Is the derivation of the CSF model mathematically correct? If not how can it be corrected mathematically?

2. In what ways does the current implementation of the original CSF model differ from what Brackbill proposes?

The following steps will be taken to answer the reasearch questions:

1. Analysing the original CSF model from Brackbill.

2. If needed, making the CSF model mathematically correct.

3. Determining and researching the differences between the implementation of the original CSF model and the original CSF model proposed by Brackbill.

This report only addresses the cases where the edge of the domain don't play a role. So the interest lies only in the surface tension that arises from the liquid-liquid two-phase flow. The analysis of the CSF model will be done in 3 dimensions and the adjustments will also be based on 3 dimensions, but the application of the new method will be used on 2 dimensions objects to make things easier. Extending the application from 2 dimension to 3 dimensions won't change the conclusions made in this report.

Chapter 2

Analysis of the original Continuous Surface Force model by Brackbill

First of all an analysis of the CSF model will be made, this will be done by looking at every step of the derivation and steps that are unclear or mathematically incorrect will be commented on in italic writing. If the steps need more attention, then these steps will be addressed in the next chapter of this report. In this report and in the article of Brackbill the Einstein summation convention is used. For example given the vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$, the dot product is defined as $\boldsymbol{u} \cdot \boldsymbol{v} = \sum_\alpha u_\alpha v_\alpha$. This is the summation of all the components with index α and according to the Einstein summation convention this can be written as $\sum_\alpha u_\alpha v_\alpha = u_\alpha v_\alpha$.

The article starts of with the equation for the interface condition according to Brackbill et al. [1]:

$$(p_1 - p_2 + \sigma\kappa)\hat{n}_i = (\tau_{1,ik} - \tau_{2,ik})\hat{n}_k + \frac{\partial \sigma}{\partial x_i} \tag{2.1}$$

where p_α is pressure in fluid α, with $\alpha = 1, 2$, σ is the fluid surface tension coefficient which is influenced by the cohesion of the molecules, $\tau_{\alpha,ik}$ is the viscous stress tensor in fluid α, $\hat{n}_i$ is the unit normal at the interface and κ is the local surface curvature. In 2 dimensions $\kappa = \frac{1}{R}$ with R being the radius of curvature and in 3 dimensions it is twice the mean curvature $\kappa = (\frac{1}{R_1} + \frac{1}{R_2})$, where R_1 and R_2 are the maximal and minimal radii of curvature. A representation of Equation (2.1) can be seen in Figure 2.1 and Figure 2.2

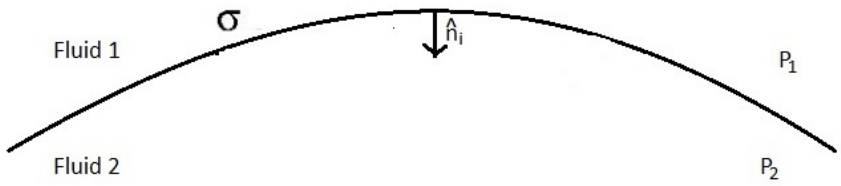

Figure 2.1: Two different fluids in contact with each other creating an interface. There is a pressure difference across the interface and there is a fluid surface tension on the interface.

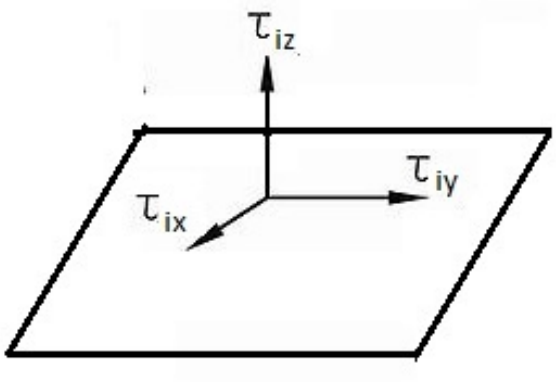

Figure 2.2: The square is the interface between the two liquids, here $\tau_{\beta\gamma}$ is the viscous stress tensor, which works in the γ direction on a surface, of which the normal is in the β direction.

This is indeed the surface boundary.
It is stated that σ is only defined on the interfacial surface. The interfacial surface has no thickness, so there won't be a change in surface tension in the normal direction. So $\frac{\partial \sigma}{\partial x_i}$ can now be written as $(\delta_{ik} - \hat{n}_i \hat{n}_k)\frac{\partial \sigma}{\partial x_k}$ where δ_{ik} is the Kronecker delta, this will make sure that only the change in surface tension is taken into account which is in the tangential direction.

An expression for the gradient along the normal direction with respect to the interface is according to Brackbill et al. [1]:

$$\nabla_N = \hat{n}\,(\hat{n} \cdot \nabla)\,. \tag{2.2}$$

This is correct to get the direction of the gradient in the normal direction,

the part of the gradient in the normal direction needs to be taken.

From (2.2) the vector projection of the gradient in the tangential direction can be expressed according to Brackbill et al. [1]:

$$\nabla_S = \nabla - \nabla_N. \tag{2.3}$$

The total stress tensor can be expressed by:

$$\sigma_{\alpha,ik} = -P\delta_{ij} + \mu_\alpha \left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} \right)_\alpha - \frac{2}{3}\delta_{ik}\frac{\partial u_l}{\partial x_l}, \qquad \alpha = 1, 2,$$

where μ_α is the molecular viscosity in fluid α and $\boldsymbol{u}$ is the fluid velocity. This equation can be divided in a deviatoric and nondeviatoric part. The deviatoric part:

$$s_{\alpha,ik} = -P\delta_{ij} + \mu_\alpha \left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} \right)_\alpha, \qquad \alpha = 1, 2.$$

The nondeviatoric part:

$$-\frac{2}{3}\delta_{ik}\frac{\partial u_l}{\partial x_l},$$

When both fluids are incompressible the viscous stress tensor can be expressed according to Brackbill et al. [1] as:

$$\tau_{\alpha,ik} = \mu_\alpha \left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} \right)_\alpha, \qquad \alpha = 1, 2. \tag{2.4}$$

The projection in the normal direction of (2.1) can be expressed by substituting (2.4) in (2.1), this looks like according to Brackbill et al. [1]:

$$(p_1 - p_2 + \sigma\kappa) = 2\mu_1\hat{n}_k \left(\frac{\partial u_k}{\partial n} \right)_1 - 2\mu_2\hat{n}_k \left(\frac{\partial u_k}{\partial n} \right)_2. \tag{2.5}$$

This is correct, this is the projection in the normal direction so as earlier stated the change in surface tension from Equation (2.1) $\frac{\partial\sigma}{\partial x_i} = 0$.

The projection of the tangential direction of (2.1) is according to Brackbill et al. [1]:

$$\mu_2 \left(\hat{t}_i\frac{\partial u_i}{\partial n} + \hat{n}_k\frac{\partial u_k}{\partial s} \right)_2 - \mu_1 \left(\hat{t}_i\frac{\partial u_i}{\partial n} + \hat{n}_k\frac{\partial u_k}{\partial s} \right)_1 = \frac{\partial\sigma}{\partial s}. \tag{2.6}$$

In the projection of the tangential direction there is no pressure difference and there is no curvature, so the left side of Equation (2.1) $(p_1 - p_2 + \sigma\kappa) = 0$ *and as stated earlier the change in surface tension from Equation (2.1)* $\frac{\partial \sigma}{\partial x_i} \neq 0$.

The surface derivative can be defined according to Brackbill et al. [1] as:

$$\frac{\partial}{\partial s} = \hat{t} \cdot \nabla, \tag{2.7}$$

and a normal derivative according to Brackbill et al. [1] as:

$$\frac{\partial}{\partial n} = \hat{n} \cdot \nabla. \tag{2.8}$$

If the fluid is inviscid and incompressible (2.5) can expressed according to Brackbill et al. [1] as:

$$p_s \equiv p_2 - p_1 = \sigma\kappa, \tag{2.9}$$

because for an inviscid fluid $\mu = 0$.

From (2.9) a stress can be defined according to Brackbill et al. [1]:

$$\boldsymbol{F}_{sa}(\boldsymbol{x}_s) = \sigma\kappa(\boldsymbol{x}_s)\hat{n}(\boldsymbol{x}_s). \tag{2.10}$$

The assumption was made that the fluid is inviscid and assuming that the surface tension coefficient is constant between inviscid fluids the following can be achieved $\boldsymbol{F}_{sa} = \boldsymbol{F}_{\text{sa}}^{(\text{n})} + \boldsymbol{F}_{\text{sa}}^{(\text{t})} = \boldsymbol{F}_{\text{sa}}^{(\text{n})}$and $\boldsymbol{F}_{sa}$ has $\frac{N}{m^2}$ as units.

If there are two fluids that can be distinguished by a characteristic function $c(\boldsymbol{x})$ it can be defined as followed according to Brackbill et al. [1]:

$$c(\boldsymbol{x}) = \begin{cases} c_1, & \text{in fluid 1.} \\ c_2, & \text{in fluid 2.} \\ < c > \equiv (c_1 + c_2)/2 & \text{at the interface.} \end{cases} \tag{2.11}$$

Each fluid has a distinguishable characteristic and on the interface the characteristic changes discontinuously. It is quite arbitrary what the $c(\boldsymbol{x})$ *is at the interface, but for ease* $< c > \equiv (c_1 + c_2)/2$ *can be chosen.*

For example if the characteristic function is given by density at the interface the following holds according to Brackbill et al. [1]:

$$\rho(\boldsymbol{x}_s) = < \rho > . \tag{2.12}$$

The discontinuity at the interface will make simulating several processes very complex. A solution would be to introduce a mollified characteristic function $\tilde{c}(\boldsymbol{x})$ which satisfies the following:

1. The first and second derivative of the mollified function have to exist.

2. The mollified function and first derivative have to be continuous.

The $\tilde{c}(\boldsymbol{x})$ function will smooth the discontinuity between c_1 and c_2 over a distance h. This will result in a transition region instead of an interface and this also means that the surface force works on the whole transition region and must be seen as a volume force. The volume force $\boldsymbol{F}_{sv}(\boldsymbol{x})$ has to satisfy according to Brackbill et al. [1]:

$$\lim_{h\to 0}\int_{\Delta V}\boldsymbol{F}_{sv}(\boldsymbol{x})\mathrm{d}^3x = \int_{\Delta A}\boldsymbol{F}_{sa}(\boldsymbol{x}_s)\mathrm{d}A_s. \tag{2.13}$$

The area integral is over a portion ΔA of the interface which lies within a small volume of integration ΔV. ΔV is constructed in such a way that the edges are normal to the surface, and the thickness is very small compared to the radius of curvature of A

The volume force $\boldsymbol{F}_{sv}(\boldsymbol{x})$ also has to satisfy according to Brackbill et al. [1]:

$$\boldsymbol{F}_{sv}(\boldsymbol{x}) = 0 \quad \text{for} \quad |\hat{n}(\boldsymbol{x}_s)\cdot(\boldsymbol{x}-\boldsymbol{x}_s)| \geq h, \tag{2.14}$$

This is correct, because the surface force only operates on the interface, so the volume force should only operate on the newly introduced transition region with width h. This should be for $|\hat{n}(\boldsymbol{x}_s)\cdot(\boldsymbol{x}-\boldsymbol{x}_s)| \geq h/2$, because the distance from the middle to the edge of the transition region is h/2 and not h as can be seen in Equation (2.14).

The Langrangian method of fluid momentum conservation for an inviscid fluid becomes according to Brackbill et al. [1]:

$$\rho\frac{d\boldsymbol{u}}{dt} = -\nabla p + \boldsymbol{F}_{sv}, \tag{2.15}$$

where ρ is the density, $\boldsymbol{u}$ is the velocity and p is the pressure, this equation can be found from the Navier-Stokes equation with the boundary condition that there is no change in velocity in the spatial direction $\nabla\boldsymbol{u} = 0$.

The mollified function $\tilde{c}(\boldsymbol{x})$ can be defined as convolution of the color function $c(\boldsymbol{x})$ with an interpolation function $\mathcal{J}$ according to Brackbill et al. [1]:

$$\tilde{c}(\boldsymbol{x}) = \frac{1}{h^3}\int_V c(\boldsymbol{x}')\mathcal{J}(\boldsymbol{x}'-\boldsymbol{x})\mathrm{d}^3x', \tag{2.16}$$

where $\mathcal{J}$ has to satisfy the following conditions according to Brackbill et al. [1]:

1.
$$\int_V \mathcal{J}(\boldsymbol{x})\mathrm{d}^3x = h^3. \tag{2.17}$$

2.
$$\mathcal{J}(\boldsymbol{x}) = 0 \quad \text{for} \quad |\boldsymbol{x}| \geq h/2. \tag{2.18}$$

3. $\mathcal{J}$ is differentiable and deceases monotonically with increasing $|\boldsymbol{x}|$.

These conditions will make sure that the mollified function only operates on the transition region, where it is needed.

If $h \to 0$ then the following holds according to Brackbill et al. [1]:

$$\lim_{h\to 0} \tilde{c}(\boldsymbol{x}) = c(\boldsymbol{x}). \tag{2.19}$$

This is correct, because the $\tilde{c}(\boldsymbol{x})$ must equal $c(\boldsymbol{x})$ when $h \to 0$. The integral is the same so $\int \tilde{c}(\boldsymbol{x})d^3x = \int c(\boldsymbol{x})d^3x$, the discontinuity in this problem does not effect the integral, this will be further addressed in chapter 3.

If the gradient of $\tilde{c}(\boldsymbol{x})$ is taken Equation (2.16) becomes according to Brackbill et al. [1]:

$$\nabla\tilde{c}(\boldsymbol{x}) = \frac{1}{h^3}\int_V c(\boldsymbol{x}')\nabla\left(\mathcal{J}(\boldsymbol{x}' - \boldsymbol{x})\right)\mathrm{d}^3x'. \tag{2.20}$$

This is true, because $\mathcal{J}$ is the only term which is a function of $\boldsymbol{x}$.

Since c($\boldsymbol{x}$) is constant within each fluid and with Gauss' divergence theorem the volume integral can be substituted for a surface integral according to Brackbill et al. [1]:

$$\nabla\tilde{c}(\boldsymbol{x}) = \frac{1}{h^3}\int_V c(\boldsymbol{x}')\nabla\mathcal{J}(\boldsymbol{x}' - \boldsymbol{x})\mathrm{d}^3x' = \frac{[c]}{h^3}\int_A \hat{n}(\boldsymbol{x}_s)\mathcal{J}(\boldsymbol{x} - \boldsymbol{x}_s)\mathrm{d}A_s, \tag{2.21}$$

where $[c] = [c_2 - c_1]$.

It should be noted that the domain V is divided in two domains on either side of the interface, because $c(\boldsymbol{x}')$ is discontinuous. Gauss can not be applied on discontinuous functions.

Define $\boldsymbol{x}_{s0}$ as the point on A from which the normal passes through $\boldsymbol{x}$. The integral from (2.21) is according to Brackbill et al. [1]:

$$\frac{1}{h^3}\int_A \hat{n}(\boldsymbol{x}_s)\mathcal{J}(\boldsymbol{x} - \boldsymbol{x}_s)\mathrm{d}A_s \simeq \frac{1}{h^3}\hat{n}(\boldsymbol{x}_{s0})\int_A \mathcal{J}(\boldsymbol{x} - \boldsymbol{x}_s)\mathrm{d}A_s + O\left(\left(\frac{h}{R}\right)^2\right). \tag{2.22}$$

This can be true if a Taylor series is used on the normal vector. The Taylor series for the normal vector is $\hat{n}(\boldsymbol{x}_s) \approx \hat{n}(\boldsymbol{x}_{s0}) + \nabla\hat{n}(\boldsymbol{x}_{s0}) \cdot (\boldsymbol{x}_s - \boldsymbol{x}_{s0})$*. The first term gives the expression* $\frac{1}{h^3}\hat{n}(\boldsymbol{x}_{s0}) \int_A \mathcal{J}(\boldsymbol{x}-\boldsymbol{x}_s)dA_s$ *and the second term gives the expression* $\frac{1}{h^3} \int_A \nabla\hat{n}(\boldsymbol{x}_{s0}) \cdot (\boldsymbol{x}_s - \boldsymbol{x}_{s0}) \cdot \mathcal{J}(\boldsymbol{x} - \boldsymbol{x}_s)dA_s$*. The second expression can be approximated by* $\nabla\hat{n}(\boldsymbol{x}_{s0}) \approx \kappa$*,* $(\boldsymbol{x}_s - \boldsymbol{x}_{s0}) \approx h$ *and the second expression can be approximated by* $O(\kappa)$

The integral in (2.22) can be bounded as followed according to Brackbill et al. [1]:

$$\frac{1}{h^2} \int_A \mathcal{J}(\boldsymbol{x} - \boldsymbol{x}_s)\mathrm{d}A_s \leq \mathcal{J}(\boldsymbol{x} - \boldsymbol{x}_{s0}). \tag{2.23}$$

This is correct, because $\mathcal{J}(\boldsymbol{x})$ *decreases monotonically with increasing* $|\boldsymbol{x}|$ *and the normalization factor is in front of the integral, this means that* $\mathcal{J}(\boldsymbol{x})$ *has a local maximum at* $\boldsymbol{x} = 0$*. Lets define that* $\mathcal{J}(\boldsymbol{x})$ *attains a maximum when* $\boldsymbol{x} = \boldsymbol{x_{s0}}$ *then* $\mathcal{J}(\boldsymbol{x} - \boldsymbol{x}_{s0})$ *has the highest value and is larger than any normalized surface integral at* $\boldsymbol{x} = \boldsymbol{x_s}$*.*

The limit of the integral of $\nabla\tilde{c}(\boldsymbol{x})$ becomes according to Brackbill et al. [1]:

$$\lim_{h\to 0} \int \hat{n}(\boldsymbol{x}_{s0}) \cdot \nabla\tilde{c}(\boldsymbol{x})\mathrm{d}^3x = [c], \tag{2.24}$$

this can be derived from (2.21) and (2.22).

In the article the limit $h \to 0$ of $\nabla\tilde{c}(\boldsymbol{x})$ is written according to Brackbill et al. [1] as:

$$\lim_{h\to 0} \nabla\tilde{c}(\boldsymbol{x}) = \hat{n}[c]\delta(\hat{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_s)) = \nabla c(\boldsymbol{x}). \tag{2.25}$$

When trying to get (2.25) from Equation (2.24) it is not possible to get the Dirac delta function into the equation without making mathematical errors so this would mean Equation (2.25) is wrong. Having a Dirac delta without an integral is also something that is not correct. Another peculiar thing is that $\nabla c(\boldsymbol{x})$ *is not defined in strong form, it is not possible to take the gradient from a discontinuous function. This will be further addressed in chapter 3*

Expressing $\boldsymbol{F}_{sa}(\boldsymbol{x}_s)$ using a $\delta(\boldsymbol{x})$ gives according to Brackbill et al. [1]:

$$\begin{aligned}\int_A \boldsymbol{F}_{sa}(\boldsymbol{x}_s)\mathrm{d}A_s & \\ &= \int_V \boldsymbol{F}_{sa}(\boldsymbol{x})\delta(\hat{n}(\boldsymbol{x}_s)\cdot(\boldsymbol{x}-\boldsymbol{x}_s))\mathrm{d}^3x, \\ &= \int_V \sigma\kappa(\boldsymbol{x})\hat{n}(\boldsymbol{x})\delta(\hat{n}(\boldsymbol{x}_s)\cdot(\boldsymbol{x}-\boldsymbol{x}_s))\mathrm{d}^3x.\end{aligned} \tag{2.26}$$

This derivation is correct.

Assuming (2.25) is correct δ can be substituted in (2.26) to get according to Brackbill et al. [1]:

$$\int_{\Delta A} \boldsymbol{F}_{sa}(\boldsymbol{x}_s)\mathrm{d}A_s = \lim_{h\to 0}\int_{\Delta V}\sigma\kappa(\boldsymbol{x})\frac{\nabla\tilde{c}(\boldsymbol{x})}{[c]}\mathrm{d}^3x. \tag{2.27}$$

It should be noted that the exact location of the interface is not needed anymore with this expression.

Thus $\boldsymbol{F}_{sv}(\boldsymbol{x})$ is according to Brackbill et al. [1]:

$$\boldsymbol{F}_{sv}(\boldsymbol{x}) = \sigma\kappa(\boldsymbol{x})\frac{\nabla\tilde{c}(\boldsymbol{x})}{[c]}. \tag{2.28}$$

The first important property of $\boldsymbol{F}_{sv}(\boldsymbol{x})$ is according to Brackbill et al. [1]:

$$\int_{p_1}^{p_2}\boldsymbol{F}_{sv}(\boldsymbol{x})\mathrm{d}(\hat{n}\cdot\boldsymbol{x}) = \int_{c_1}^{c_2}\sigma\kappa(\boldsymbol{x})\hat{n}(\boldsymbol{x})\frac{\mathrm{d}\tilde{c}(\boldsymbol{x})}{[c]} \simeq \sigma\kappa(\boldsymbol{x})\hat{n}(\boldsymbol{x}) \quad \text{for} \quad h>0. \tag{2.29}$$

Since $h\to 0$ is taken this property contradicts $\lim_{h\to 0}$, this will be further addressed in chapter 3

The second property is according to Brackbill et al. [1]:

$$\lim_{h\to 0}\boldsymbol{F}_{sv}(\boldsymbol{x}) = \boldsymbol{F}_{sa}(\boldsymbol{x})\delta(\hat{n}(\boldsymbol{x}_s)\cdot(\boldsymbol{x}-\boldsymbol{x}_s)). \tag{2.30}$$

This is correct if limit and integral may be interchanged, this may only be done if Lebesgue's dominated convergence theorem holds.

The density weighting is introduced by Brackbill. The density weighting is to ensure that neighbouring contours in the transition region remain a constant distance from the interface if the surface force is applied. Many

implementations of the original CSF model don't use the density weighting. This subject is important and will be addressed in the next chapter.

Brackbill also justifies the use of a local curvature, since in Equation (2.28) the exact location of the interface is not needed. This local curvature is the curvature in each point of the corresponding level set function. This would apply if the curvature of the points on the interface are needed. It is justified by Brackbill, because the transition region is small enough so that the error margin is not worse than the other terms. Many implementations of the original CSF model use the interface curvature instead of using the local curvature, because the local curvature is not always available. The models that implement the interface curvature have a better correlation with respect to the ascent speed of the bubble. A cause for the better correlation could be the use of the interface curvature instead of the local curvature. The use of local curvature instead of calculating the interface curvature will be investigated in the next chapter.

So to sum it all up there are three steps that are unclear or mathematically wrong and two subjects that will be addressed in the next chapter:

1. Equation (2.19) $\lim_{h\to 0} \tilde{c}(\boldsymbol{x}) = c(\boldsymbol{x})$. This is unclear, because the integral of a discontinuous function is not always finite.

2. Equation (2.25) $\lim_{h\to 0} \nabla\tilde{c}(\boldsymbol{x}) = \hat{n}[c]\delta(\hat{n}\cdot(\boldsymbol{x}-\boldsymbol{x}_s)) = \nabla c(\boldsymbol{x})$. This is wrong, because the the gradient of a discontinuous function does not exist.

3. Equation (2.29)
   $\int_{p_1}^{p_2} \boldsymbol{F}_{sv}(\boldsymbol{x})\mathrm{d}(\hat{n}\cdot\boldsymbol{x}) = \int_{c_1}^{c_2} \sigma\kappa(\boldsymbol{x})\hat{n}(\boldsymbol{x})\frac{\mathrm{d}\tilde{c}(\boldsymbol{x})}{[c]} \simeq \sigma\kappa(\boldsymbol{x})\hat{n}(\boldsymbol{x}) \quad \text{for} \quad h > 0.$
   This is wrong, because this property contradicts $\lim_{h\to 0}$.

4. The density weighting. Many implementations of the original CSF model don't use this.

5. The use of interface curvature, instead of using the local curvature.

Chapter 3

Correcting the original Continuous Surface Force model

In this chapter the unclear or mathematically wrong steps and the two subjects are going to be addressed and complemented on:

1. Equation (2.19) $\lim_{h\to 0} \tilde{c}(\boldsymbol{x}) = c(\boldsymbol{x})$. This is unclear, because the integral of a discontinuous function is not always finite.

2. Equation (2.25) $\lim_{h\to 0} \nabla\tilde{c}(\boldsymbol{x}) = \hat{n}[c]\delta(\hat{n}\cdot(\boldsymbol{x}-\boldsymbol{x}_s)) = \nabla c(\boldsymbol{x})$. This is wrong, because the the gradient of a discontinuous function does not exist.

3. Equation (2.29)
$\int_{p_1}^{p_2} \boldsymbol{F}_{sv}(\boldsymbol{x})\mathrm{d}(\hat{n}\cdot\boldsymbol{x}) = \int_{c_1}^{c_2} \sigma\kappa(\boldsymbol{x})\hat{n}(\boldsymbol{x})\frac{\mathrm{d}\tilde{c}(\boldsymbol{x})}{[c]} \simeq \sigma\kappa(\boldsymbol{x})\hat{n}(\boldsymbol{x}) \quad \text{for} \quad h > 0.$
This is wrong, because this property contradicts $\lim_{h\to 0}$

4. The density weighting. Many implementations of the original CSF model don't use this.

5. The use of interface curvature, instead of using the local curvature.

3.1 Improvements on the Continuous Surface Force model

Looking at Equation (2.19), the characteristic function is finite in both fluids an example is shown in Figure 3.1.

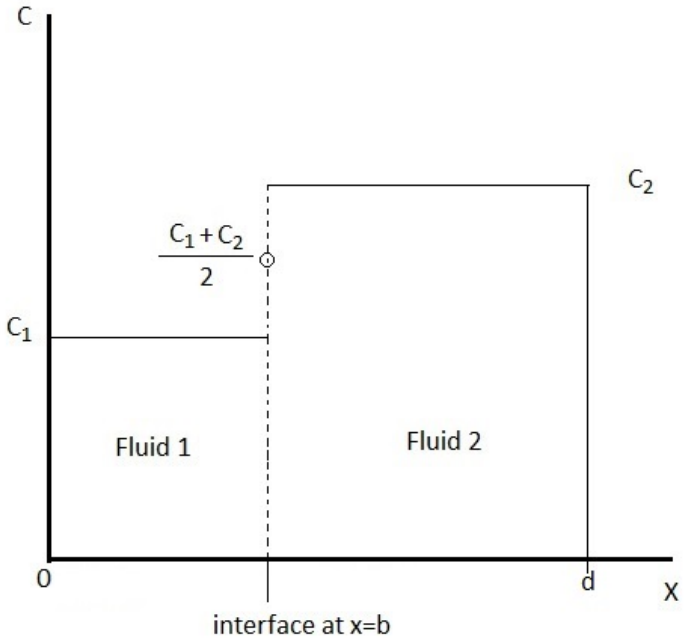

Figure 3.1: Fluid 1 occupies the region from x=0 to x=b and fluid 2 occupies the region from x=b to x=d, both fluids have a finite characteristic value. The interface is located at x=b and presents a discontinuity of the characteristic function.

The integral of the function that can be seen in Figure 3.1 can be expressed as:

$$\int_0^d c(\boldsymbol{x})\mathrm{d}x, \tag{3.1}$$

the integral can be expressed as the sum of two integrals:

$$\int_0^d c(\boldsymbol{x})\mathrm{d}x = \int_0^b c(\boldsymbol{x})\mathrm{d}x + \int_b^d c(\boldsymbol{x})\mathrm{d}x. \tag{3.2}$$

Due to the discontinuity at the interface it has to be verified if the following exists and is finite.

1. $\lim_{t\to b^-} \int_0^t c(\boldsymbol{x})\mathrm{d}x$.
2. $\lim_{s\to b^+} \int_s^d c(\boldsymbol{x})\mathrm{d}x$.

From (2.11) 1 can be solved to get:

$$\lim_{t\to b^-} \int_0^t c(\boldsymbol{x})\mathrm{d}x = \lim_{t\to b^-} (t \cdot c_1) = b \cdot c_1. \tag{3.3}$$

The same can be done for 2, so it can be seen that the integral of $c(\boldsymbol{x})$ does exist.

For Equation (2.25), Looking from (2.13) and (2.26), it can be derived that

$$\lim_{h\to 0}\int_{\Delta V}\boldsymbol{F}_{sv}(\boldsymbol{x})\mathrm{d}^3x = \lim_{h\to 0}\int_{V}\sigma\kappa(\boldsymbol{x})\hat{n}(\boldsymbol{x})\delta(\hat{n}(\boldsymbol{x}_s)\cdot(\boldsymbol{x}-\boldsymbol{x}_s))\mathrm{d}^3x,$$

and thus

$$\boldsymbol{F}_{sv} = \sigma\kappa(\boldsymbol{x})\hat{n}(\boldsymbol{x})\delta(\hat{n}(\boldsymbol{x}_s)\cdot(\boldsymbol{x}-\boldsymbol{x}_s)).$$

This expression for the volume force is not ideal since there is still a δ term. So from Equation (2.24) the following can be expressed:

$$\lim_{h\to 0}\int \hat{n}(\boldsymbol{x}_{s0})\cdot\nabla\tilde{c}(\boldsymbol{x})\mathrm{d}^3x = [c]$$

$$\Leftrightarrow$$

$$\lim_{h\to 0}\int \hat{n}(\boldsymbol{x}_{s0})\cdot\nabla\tilde{c}(\boldsymbol{x})\mathrm{d}^3x = \int [c]\delta(\hat{n}(\boldsymbol{x}_{s0})\cdot(\boldsymbol{x}-\boldsymbol{x}_{s0}))\mathrm{d}^3x$$

$$\Leftrightarrow$$

$$\lim_{h\to 0}\int \nabla\tilde{c}(\boldsymbol{x})\mathrm{d}^3x = \int n(\hat{\boldsymbol{x}}_{s0})[c]\delta(\hat{n}\cdot(\boldsymbol{x}-\boldsymbol{x}_{s0}))\mathrm{d}^3x = \int \nabla c(\boldsymbol{x})\mathrm{d}^3x \tag{3.4}$$

so in a weak sense the following holds for $\boldsymbol{v}\in\mathbb{R}^n$ a test vector:

$$\lim_{h\to 0}\nabla\tilde{c}(\boldsymbol{v}) = \nabla c(\boldsymbol{v}),$$

but this is definitely not true in the strong sense.

Taking a look at Equation (2.29)

$$\begin{aligned}\int_{p_1}^{p_2}\boldsymbol{F}_{sv}(\boldsymbol{x})d(\hat{n}\cdot\boldsymbol{x}) &\\ &= \int_{c_1}^{c_2}\left(\sigma\kappa(\boldsymbol{x})\frac{\nabla\tilde{c}(\boldsymbol{x})}{[c]}\right)\hat{n}\mathrm{d}n\\ &= \int_{c_1}^{c_2}\left(\sigma\kappa(\boldsymbol{x})(\boldsymbol{x})\frac{\nabla\tilde{c}(\boldsymbol{x})}{[c]}\right)\hat{n}\mathrm{d}n\\ &= \int_{c_1}^{c_2}\sigma\kappa(\boldsymbol{x})\hat{n}(\boldsymbol{x})\frac{\frac{\partial\tilde{c}(\boldsymbol{x})}{\partial\hat{n}}}{[c]}\mathrm{d}n.\end{aligned} \tag{3.5}$$

Integrating this will result in $\sigma\kappa(\boldsymbol{x})\hat{n}(\boldsymbol{x})$. The problem that arises is $\tilde{c}(\boldsymbol{x})$ is still dependent on h, this could create errors if $h\neq 0$.

3.2 Problems with current implementation of the Continuous Surface Force model

Two major problems with the implementation of the CSF model are:

1. The density weighting.
2. The use of interface curvature, instead of the local curvature.

Many implementations of the original CSF model don't use the density weighting. The density weighting can be applied as a scaling to the volume force according to Brackbill et al. [1]:

$$\boldsymbol{F}_{sv}(\boldsymbol{x}) = \sigma\kappa(\boldsymbol{x})\delta(\hat{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_s))g(\boldsymbol{x}),$$

with $g(\boldsymbol{x})$ being the density weight function defined as $c(\boldsymbol{x})/ < c >$. A few examples of a different implementation can be seen in Baltussen et al. [3, p. 68] and Denner Fabian and van Wachem Berend G. M. [4, p. 222]. According to Baltussen et al. [3, p. 68] the density weighted volume force is:

$$F_\sigma = 2F\sigma\kappa\boldsymbol{n},$$

where F is the phase fraction.
According to Denner Fabian and van Wachem Berend G. M. [4, p. 222] the density weighted volume force is:

$$f_{s,i} = \sigma\kappa\frac{\partial\gamma}{\partial x_i},$$

where γ is the color function.
From these examples it is clear that the implementation of the density weighting is not what Brackbill proposed. In the next chapter the density weighting will be investigated and the possible reason why many interpretations of the original CSF model do not use the density weighting proposed by Brackbill.

The MCLS algorithm applies the local curvature which is justified by Brackill and the correlation with respect to ascent speed is worse than with methods that use the interface curvature. This will be investigated in chapter 5.

Chapter 4

Density weighting

In section III of Brackbill's paper the density is proposed as mollified indicator function. The reason Brackbill proposes the density weight function is to keep neighbouring contours in the transition region a constant distance from the interface if the surface force is applied. If the density weighting is not applied and the surface force is taken to be constant, lighter particles will have a larger acceleration than heavier particles. This could make the transition region increase in width which is not desirable, or the transition region could decrease in width which is desirable. The density weighting will make sure that the interface will keep the same thickness. From the examples mentioned before not many implementations use the density weight function. The Scientific Computing group of DIAM had inconsistent results when the density weighting was applied and when it was not applied. So applying the density weight function has some consequence which will be investigated.
The density weight function is defined according to Brackbill et al. [1] as:

$$g(\boldsymbol{x}) = \rho(\boldsymbol{x}) / < \rho > , \tag{4.1}$$

where $< \rho >= \frac{1}{2}(\rho_1 + \rho_2)$ with ρ_1 and ρ_2 the density in the fluids. This weight function equals 1 when $\lim_{h \to 0}$. The weight function can then be applied as a scaling to the volume force to get according to Brackbill et al. [1]:

$$\boldsymbol{F}_{sv}(\boldsymbol{x}) = \sigma \kappa(\boldsymbol{x}) \frac{\nabla \rho(\boldsymbol{x})}{[\rho]} \frac{\rho(\boldsymbol{x})}{< \rho >}, \tag{4.2}$$

and then if this is substituted in (2.15) the acceleration due to the volume force will only depend on density gradients and not the density itself. In practice due to limitations of numerical modelling h will always have some nonzero value and as a result this volume force will be dependent on the density, this will result in differences when using the density weight function

and when it is not used. From Equation (2.1) the pressure difference can be expressed as a function of the surface tension:

$$p_1 - p_2 = \sigma\kappa. \tag{4.3}$$

From Equation (4.3) it can be seen that the pressure difference is independent of the density. Expressing the pressure difference as a function of the volume force without the density weighting:

$$p_1 - p_2 = \int \sigma\kappa(\boldsymbol{x}) \frac{\nabla\rho(\boldsymbol{x})}{[\rho]} \mathrm{d}^3x. \tag{4.4}$$

This equation is also not dependent on the density. To see what the consequences are when using the density weighting, two cases will be looked at these can be seen in Figure 4.1 and Figure 4.2.

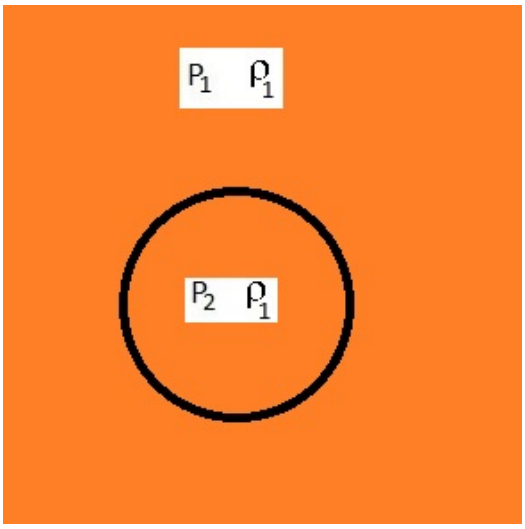

Figure 4.1: A bubble with mean curvature κ, P_2 and ρ_1, in a space with P_1 and ρ_1.

In this case assuming nonzero h, the densities ρ are the same inside and outside the bubble, so $g(\boldsymbol{x}) = \rho_1 / < \rho >= \rho_1/\rho_1 = 1$. In this case the weight function equals 1 for every h, so the volume force does not depend on the densities inside and outside the bubble. As stated before in practice h will always be a nonzero value, this confirms Brackbill's statement of using a weight function.

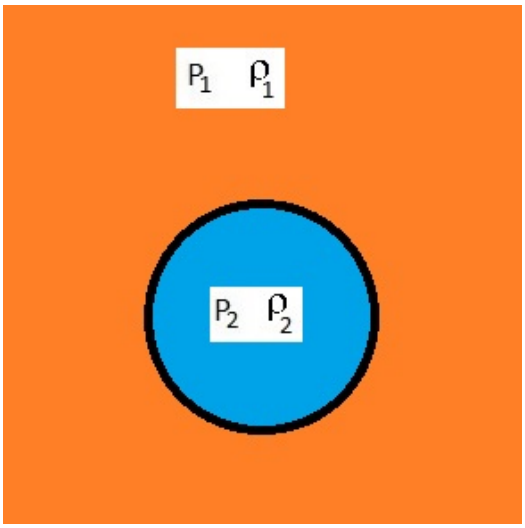

Figure 4.2: A bubble with mean curvature κ, P_2 and ρ_2, in a space with P_1 and ρ_1.

In this case assuming nonzero h, the densities ρ are not the same inside and outside the bubble, so $g(\boldsymbol{x}) = \rho_1 / < \rho >= \frac{\rho_1}{\frac{1}{2}(\rho_1+\rho_2)} \neq 1$ on one side of the interface and $g(\boldsymbol{x}) = \rho_2 / < \rho >= \frac{\rho_2}{\frac{1}{2}(\rho_1+\rho_2)} \neq 1$ on the other side of the interface. In this case the weight function equals 1 only for $\lim_{h\to 0}$. As stated before in practice h will always be a nonzero value, so the volume force does depend on the densities inside and outside the bubble. When using a finite h and the densities inside and outside the bubble are not the same, the weight function can not be used.

An example is given for this consequence in 2 dimensions, for a circular bubble with diameter of 1 cm, $\sigma = 0.0070$ N/m for air and water, the density of water is approximately 1000 kg/m^3 and the density for air is approximately 1.2 kg/m^3. This example is for the surface tension for water-water and water-air.
Surface force for water-water without density weighting is $\sigma\kappa = 1.4$ N/m^2.
Surface force for water-water with density weighting is $\sigma\kappa \cdot g(\boldsymbol{x}) = 1.4 \cdot$

$\frac{1000}{\frac{1}{2}(1000+1000)} = 1.4$ N/m^2. So the surface forces are the same if the density inside and outside the bubble are the same.
Surface force for water-air without density weighting is $\sigma\kappa = 1.4$ N/m^2.
Surface force for air-water with density weighting is $\sigma\kappa \cdot g(\boldsymbol{x}) = 1.4 \cdot \frac{1.2}{\frac{1}{2}(1000+1.2)} \approx 3.4 \cdot 10^{-3}$ N/m^2.
Surface force for water-air with density weighting is $\sigma\kappa \cdot g(\boldsymbol{x}) = 1.4 \cdot \frac{1000}{\frac{1}{2}(1000+1.2)} \approx 2.8$ N/m^2. So the surface forces are affected by the density if the density inside and outside the bubble are not the same.

Chapter 5

Calculating the curvature of an arbitrary interface

As mentioned earlier the use of the interface curvature could be the cause that leads to better results than if the local curvature is used. In this chapter it will be shown that the calculating the interface curvature is very important. This will be done in two steps. The first step is to determine the curvature of cells which are near the interface and the second step is to distribute the determined curvature to the rest of the cells. The reason for calculating the curvature near the interface is, because the interface is unknown, so the location of the interface can only be approximated.

To get an understanding first of all a simple but nontrivial interface will be taken which is an ellipse and to create a more real scenario a star shaped interface will be used which is made up of ellipses. In practice the interface is not always convex for instance a skirted bubble. The skirted bubble is one of the hardest cases due to the high variation in curvature of the interface, but this report will not include the skirted bubble. The difference between the ellipse and the star is that the ellipse is convex and star is not.
The curvature for an ellipse can be calculated analytically, because the analytical expression is known for an ellipse and thus also the analytical expression for its curvature. Since the star is made of ellipses, the curvature can be calculated for every individual ellipse. The star will be made C^1 continuous and this means that the curvature for the whole star cannot be calculated analytically for every point. In this report the points that have an undefined interface curvature will be left out and in practice these points do not exist if surface tension is present.
In practice the interface is defined implicitly by the level set function in a discrete equidistant lattice. The level set function is the distance from for

every lattice point to the interface. The ellipse and the star will function as reference to compute the level set function and explicit interface curvature. The level set function will be used to calculate the interface curvature when the interface is only known implicitly and the explicit interface curvature will be used to verify these approaches.

Second of all the curvature will be calculated in cells which are near the interface. The level set function can be used to find the local curvature for the lattice and the location of the interface. With the location of the interface and the local curvature of the lattice the curvature near the interface can be calculated. Two methods will be used to calculate the curvature near the interface these are 'interpolation of the local curvature' and 'the method of concentric circles' by Meland et al. [2]

Finally the curvature near the interface will be distributed to the rest of the cells using a linear transport equation.

5.1 Analytical determination of the interface curvature for an elliptical interface

First the interface curvature will be calculated analytically, then the distance between a cell and an elliptical interface needs to be calculated and finally the curvature will be assigned to the corresponding cells.

5.1.1 Curvature of an elliptical interface

Curvature is the amount a certain curve deviates from a straight line. The curvature can be found with:

$$\kappa(t) = \left\| \frac{\mathrm{d}\boldsymbol{T}(t)}{\mathrm{d}s} \right\|, \tag{5.1}$$

where κ is the curvature and $\boldsymbol{T}$ is the unit tangent vector.
Rewriting $\frac{\mathrm{d}\boldsymbol{T}(t)}{\mathrm{d}s}$ so any vector function $\boldsymbol{r}(t)$ can be used

$$\frac{\mathrm{d}\boldsymbol{T}(t)}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{T}(t)}{\mathrm{d}s} \cdot \frac{\mathrm{d}s}{\mathrm{d}t} \Rightarrow \frac{\mathrm{d}\boldsymbol{T}(t)}{\mathrm{d}s} = \frac{\frac{\mathrm{d}\boldsymbol{T}(t)}{\mathrm{d}t}}{\|\boldsymbol{r}'(t)\|}. \tag{5.2}$$

An expression for an ellipse with half axes a and b is:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1. \tag{5.3}$$

To find the curvature for an ellipse it has to be parametrized. Using
$x = a\,\cos(t)$ and $y = b\,\sin(t)$ as parametrization it can be expressed as a vector function:

$$\boldsymbol{r}(t) =< a\,\cos(t), b\,\sin(t) > . \tag{5.4}$$

An expression for the unit tangent function is:

$$\boldsymbol{T}(t) = \frac{\boldsymbol{r}'(t)}{\|\boldsymbol{r}'(t)\|}. \tag{5.5}$$

So when (5.4) and (5.5) are combined:

$$\boldsymbol{T}(t) = \left\langle \frac{-a\,\sin(t)}{\sqrt{a^2\,\sin^2(t) + b^2\,\cos^2(t)}}, \frac{b\,\cos(t)}{\sqrt{a^2\,\sin^2(t) + b^2\,\cos^2(t)}} \right\rangle, \tag{5.6}$$

and now if (5.1), (5.2), (5.4) and (5.6) are combined:

$$\kappa(t) = \left\| \frac{\frac{\mathrm{d}\boldsymbol{T}(t)}{\mathrm{d}t}}{\|\boldsymbol{r}'(t)\|} \right\| = \frac{ab}{\left(\sqrt{a^2\,\sin^2(t) + b^2\,\cos^2(t)}\right)^3}. \tag{5.7}$$

This is an expression for the curvature of an ellipse which satisfies (5.3).

5.1.2 Distance from a point to the elliptical interface

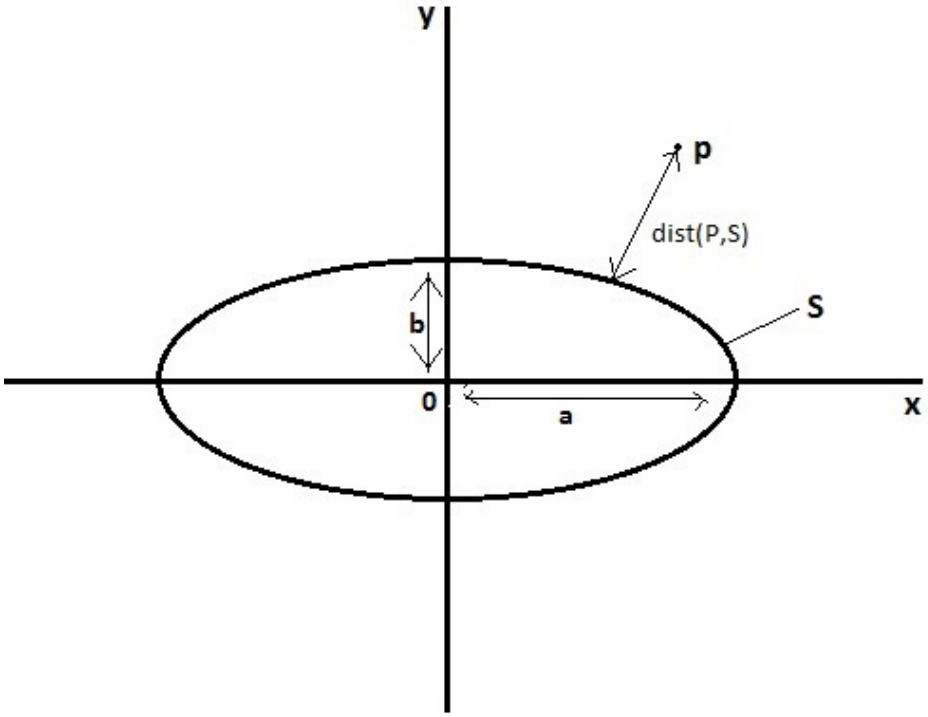

Figure 5.1: An ellipse S with center in the cross section of the axis and a random point P and the distance between point P and ellipse S. a and b are the semi major and semi minor axis

The distance from a point P and the ellipse S which can be seen in Figure 5.1 will be defined as $\text{dist}(\text{P}, \text{S}) \equiv \min_{\boldsymbol{x} \in S} \text{dist}(\text{P}, \boldsymbol{x})$.
There are a few cases where finding the distance between the ellipse and a point P can be made easier, all these cases will made with reference to Figure 5.1.

If $x_p = 0$ and $y_p = 0$.
The distance is just b.

If $x_p = 0$ and $y_p \neq 0$.
The distance is either b$-y_p$ or $-$b$-y_p$.

If $x_p \neq 0$ and $y_p = 0$ and $x_p > \sqrt{a^2 - b^2}$.
The distance is either a$-x_p$ or $-$a$-x_p$. If $x_p \leq \sqrt{a^2 - b^2}$ then it will be between the focal points and the distance will be different

If $x_p \neq 0$ and $y_p \neq 0$.
The distance from a point P and the ellipse will be the distance between a point on the ellipse whose normal is in line with point P. The ellipse in Figure 5.1 satisfies (5.3), so take a random point E in the ellipse with coordinates (x, y) and point P with coordinates (x_p, y_p) the slope between these two point can be expressed as:

$$\text{slope}(\text{E}, \text{P}) = \frac{y - y_p}{x - x_p}. \tag{5.8}$$

To find the normal it is needed that the slope(E, P) to be exactly the slope of the normal which goes through point P.
slope(normal) $\cdot$ slope(tangent) $= -1$ can be used to determine the point on the ellipse which is closest to point P, so if slope(E, P) $\cdot$ slope(tangent) $= -1$ then the point on the ellipse which is closest to point P has been found.

If the derivative is taken from (5.3) with respect to x, the slope(tangent) can be found:

$$\frac{\mathrm{d}(\frac{x^2}{a^2} + \frac{y^2}{b^2})}{\mathrm{d}x} = 0 \Leftrightarrow \frac{2x}{a^2} + \frac{2yy'}{b^2} = 0 \Leftrightarrow y' = -\frac{b^2 x}{a^2 y}, \tag{5.9}$$

this is the slope(tangent).
Combining (5.8) and (5.9):

$$\text{slope}(\text{E}, \text{P}) \cdot \text{slope}(\text{tangent}) \rightarrow \frac{y - y_p}{x - x_p} \cdot -\frac{b^2 x}{a^2 y} = -1,$$

this can be solved for y which gives us:

$$y = \frac{b^2 y_p x}{b^2 x - a^2 (x - x_p)}. \tag{5.10}$$

Substituting (5.10) in (5.3) gives:

$$\frac{x^2}{a^2} + \frac{(\frac{b^2 y_p x}{b^2 x - a^2 (x - x_p)})^2}{b^2} = 1,$$

solving this for x results in a quartic equation:

$$C_0 x^4 + C_1 x^3 + C_2 x^2 + C_3 x + C_4 = 0, \tag{5.11}$$

with

$$
\begin{aligned}
C_0 &= b^6 + b^2a^4 - 2a^2b^4 \\
C_1 &= 2a^2b^4x_p - 2a^4b^2x_p \\
C_2 &= b^2a^4x_p^2 + a^2b^4y_p^2 - a^2b^6 - a^6b^2 + 2a^4b^4 \\
C_3 &= -2a^4b^4x_p + 2a^6b^2x_p \\
C_4 &= -b^2a^6x_p^2.
\end{aligned}
$$

Finding the roots for (5.11) will give four x-coordinates for the point E on the ellipse that is the closest to point P. Using (5.10) will then give four y-coordinates for the point E on the ellipse that is the closest to point P. From these four E points it can then be determined which is closest to point P. So the distance is $\min\left(\sqrt{(x-x_p)^2 + (y-y_p)^2}\right)$

If $x_p \neq 0$ and $y_p = 0$ and $x_p \leq \sqrt{a^2 - b^2}$.
A problem arises when certain points P have equal distances to different points E, these problems occur when the points P are on the x-axis or y-axis. If the ellipse in Figure 5.1 is used it can be seen that $a > b$ so this means that all the points P on the y-axis have the point E on the y-axis as closest point. The points P that are on the x-axis and outside of the ellipse have the point E on the x-axis as closest point.

For the points within the ellipse and on the x-axis the following has to be solved:

$$d^2 = (x_p - x_E)^2 + (y_p - y_E)^2, \tag{5.12}$$

where d is the distance between the point P on the x-axis and an arbitrary point E on the ellipse. It is known that $y_p = 0$ because the point P is on the y-axis and it is also known from (5.3) that $y_E^2 = b^2 - \left(\frac{bx_E}{a}\right)^2$. Filling these in (5.13) gives:

$$d^2 = x_p^2 + x_E^2 - 2x_px_E + b^2 - \frac{b^2x_E^2}{a^2}.$$

The equation has to be differentiated with respect to x_E to find the distance. So differentiating and solving for x_E gives:

$$x_E = \frac{x_p}{1 - \frac{b^2}{a^2}}. \tag{5.13}$$

So the distance is $\min\left(\sqrt{(x_p - x_E)^2 + (y_p - y_E)^2}\right)$

5.1.3 Contour lines of an elliptical interface

For each point P it holds that the curvature at this point corresponds to the curvature of the interface at the point on the interface which is closest to the point P, this will result in a contour of curvature of corresponding points. To verify if the algorithm used is correct, it can be applied to a circle. This circle has a radius of 3 and the corresponding curvature for a circle is $\frac{1}{r}$ with r being the radius, so in this case the curvature should be $\frac{1}{3}$. The lattice used for these figures is a 10 by 10 lattice with step size of 1. Every quadrant of the circle has to be the same, so only the first quadrant is going to be shown. So first a lattice is needed and finding the corresponding point on the circle which is the closest, this can be seen in Figure 5.2. It should hold that the curvature stays constant on the lattice. Figure 5.3 shows the difference between the calculated curvature of the lattice points and the analytical curvature. It can be seen that the relative difference fluctuates, this is probably due to numerical errors when solving the nonlinear equations in MATLAB™. These numerical errors are typically in the range of 10^{-15} to 10^{-16}. Since the curvature should stay constant it can be expectated that all the points are connected in the contour line. From Figure 5.4 it can be seen that the contour line connects most of the points together, but due to the numerical errors it does not connect all the points. So from these results it can be said that the algorithm full fills our expectations.

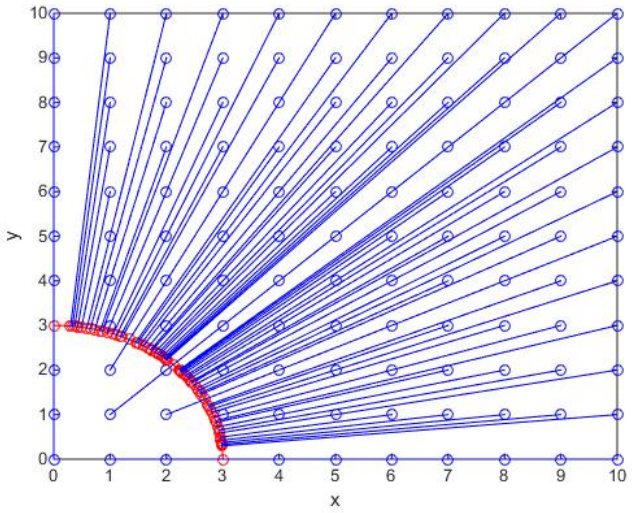

Figure 5.2: The first quadrant of an equidistant lattice with a circle as contour zero. the blue circles are points P from the lattice and the small red circles are the corresponding points E from the ellipse.

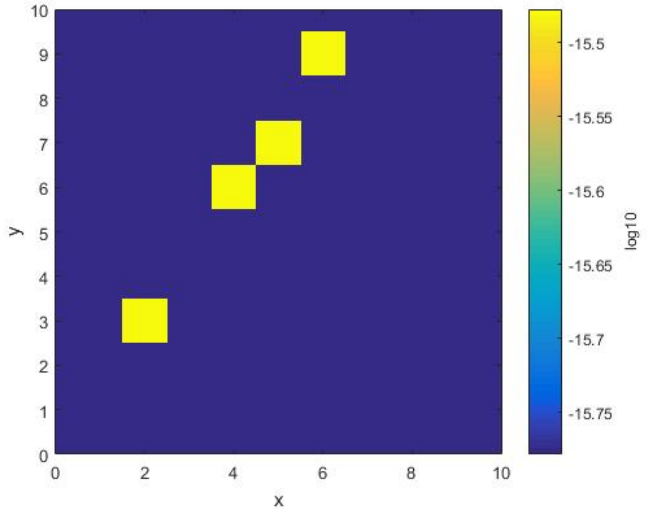

Figure 5.3: The first quadrant of difference between the calculated curvature of points P and the analytical curvature from a circle, the errors are from MATLAB™.

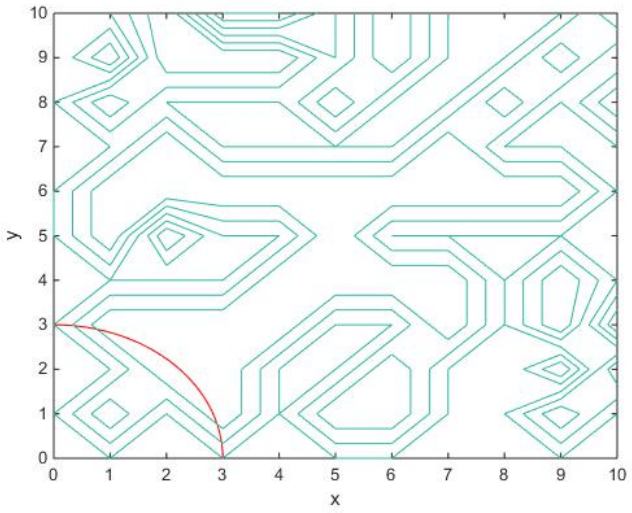

Figure 5.4: The first quadrant of the contour of curvature of corresponding points for the circle, the contour should cover all the points but this is not the case due to numerical errors.

Now take a lattice but this time use an ellipse, Figure 5.5 shows the configuration used. The ellipse has a semi major axis of 4 and a semi minor axis of 1 and the lattice used for these figures is a 10 by 10 lattice with step size of 1. It is expected that the curvature for an ellipse is largest at the edge of the semi major axis and the curvature to be smallest at the edge of the semi minor axis, Figure 5.6 verifies this expectation. The contour of curvature of corresponding points for an ellipse should have multiple contours around the area where the curvature changes the most, this can be seen in Figure 5.7. This figure however has jagged lines and is probably due to MATLAB™ not being able to interpolate accurately when there are not enough points and the curvature changes drastically. A higher density lattice solves the problem of the jagged lines. If the distance is taken from the lattice points the level sets from this ellipse can be achieved, this can be seen in Figure 5.8. From this figure it can be seen that the level set are not ellipses, because of how the distance is defined. The jagged lines in this figure is probably due to the same problem as in the last figure. The local curvature of the lattice points can be calculated with the level set.

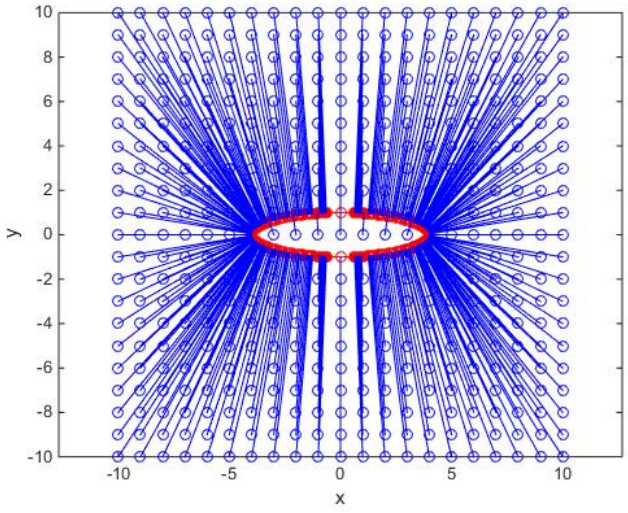

Figure 5.5: An equidistant lattice with an ellipse as contour zero. the blue circles are points P and the small red circles are the corresponding points E.

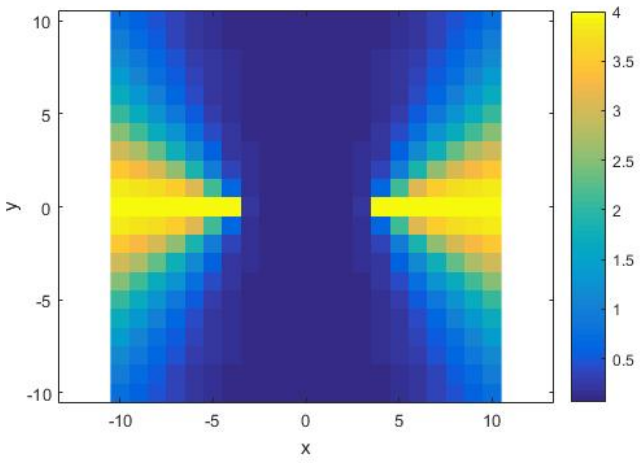

Figure 5.6: The curvature of points P from an ellipse, the curvature is the largest at the edge of the semi major axis and the curvature is the smallest at the edge of the semi minor axis.

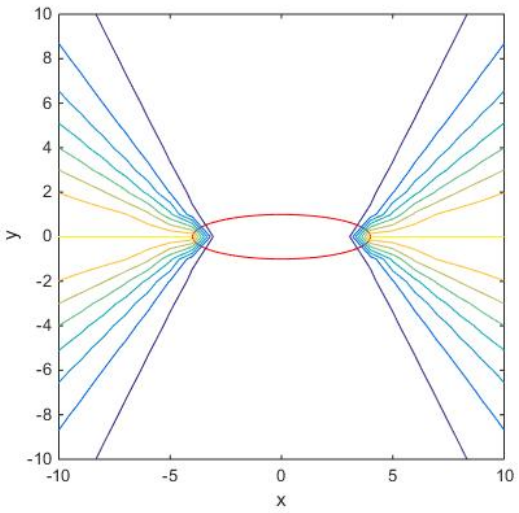

Figure 5.7: Contour of curvature of corresponding points for the ellipse, the contour shows where the change in curvature is the largest.

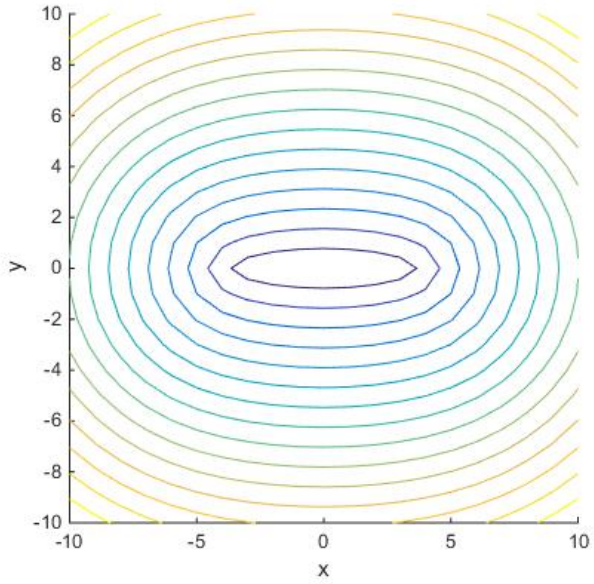

Figure 5.8: The level sets of an ellipse, here the level set is the signed distance function. It can be seen that the level sets are not ellipses.

5.2 Analytical determination of the interface curvature for a star shaped interface

The difference between the star and the ellipse is that the star is not convex and the ellipse is. For the star five half ellipses are going to be used and five pieces of ellipses are going to be used, which will connect the half ellipses of the star. Important is this star will be C^1 continuous, so this means that the curvature does not exist on the connecting points of the half ellipse and the ellipse piece. From here on this report will neglect the contribution of these points, since they have a nonexistent curvature.

5.2.1 constructing the star shaped interface

A configuration is shown in Figure 5.9

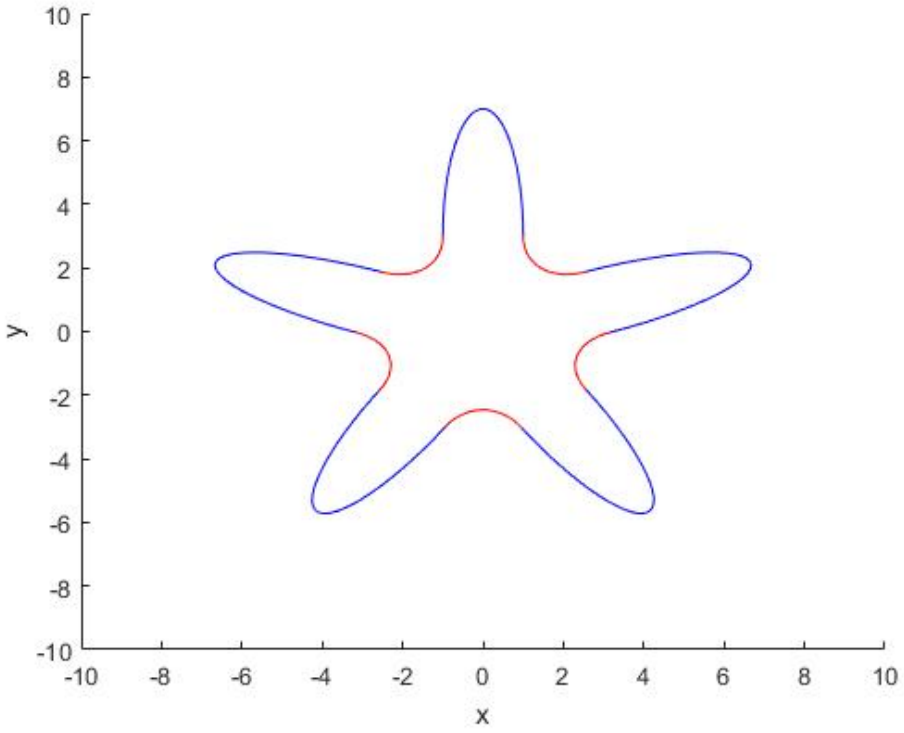

Figure 5.9: A star made of ellipses. The five half ellipses are shown in blue which serve as the points of the star and ellipse piece is shown in red which connects the points of the star.

For the half ellipses a normal ellipse can be used and using Equation (5.4)

to parametrize the ellipse in polar coordinates. First the half ellipse gets rotated using the rotation matrix and then it gets translated until it is in the desired position. For the connecting piece, rotate the half ellipses as shown in the following figure:

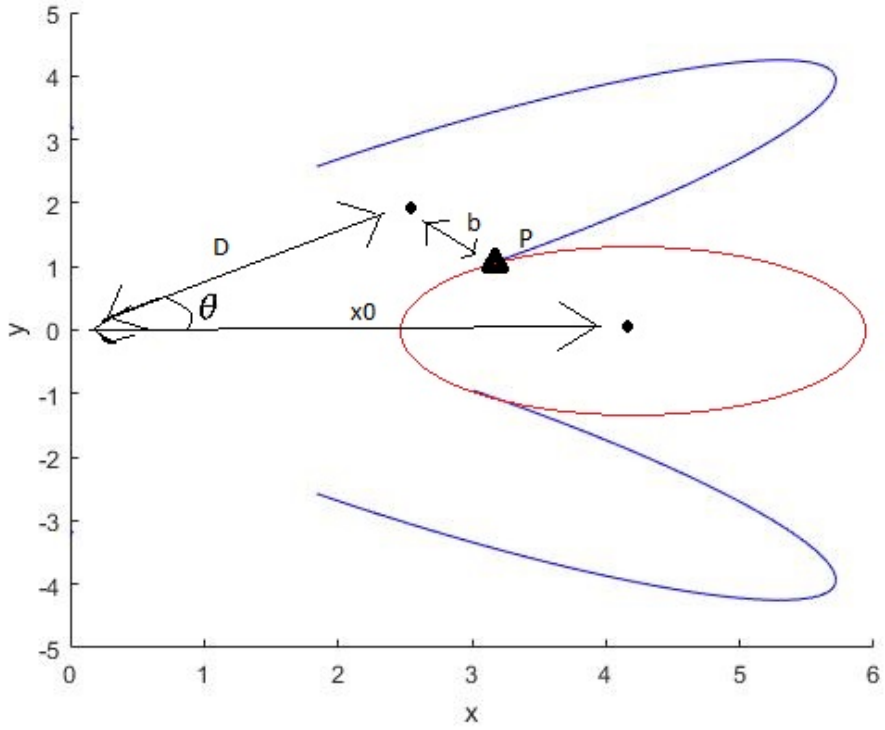

Figure 5.10: A close up of Figure 5.9, here blue are the half ellipses and red is the connecting piece of an ellipse which connects the half ellipses. The dots are the middle points of the ellipses, the triangle is the point P which is the connecting point between the blue and red ellipse.

In this figure $\theta = \frac{\pi}{5}$, D = 3, a = 4 and b = 1. The star is going to be C^1 continuous so the value and the tangent vector of the blue ellipse and red ellipse need to be aligned in point P see Figure 5.10. For the red ellipse the following parametrization
$\boldsymbol{r}(t) =< a_c \cos(t), b_c \sin(t) >$ can be used and $\boldsymbol{T} = \left\langle \frac{-a_c \sin(t)}{\sqrt{a_c^2 \sin^2(t) + b_c^2 \cos^2(t)}}, \frac{b_c \cos(t)}{\sqrt{a_c^2 \sin^2(t) + b_c^2 \cos^2(t)}} \right\rangle$.
This will leave us with 4 equations and 4 unknowns a_c, b_c, $x0$, t. The 4 equations are:

1. $D\cos(\frac{\pi}{5}) + b\sin(\frac{pi}{5}) - x0 = a_c\cos(t)$
2. $D\cos(\frac{\pi}{5}) - b\sin(\frac{pi}{5}) = b_c\cos(t)$
3. $-\cos(\frac{\pi}{5}) = \frac{-a_c\ \sin(t)}{\sqrt{a_c^2\ \sin^2(t)+b_c^2\ \cos^2(t)}}$
4. $-\sin(\frac{\pi}{5}) = \frac{b_c\ \cos(t)}{\sqrt{a_c^2\ \sin^2(t)+b_c^2\ \cos^2(t)}}$

Solving these 4 equations will give the parametrization of the red ellipse which will make the star C^1 continuous.

5.2.2 Curvature of a star shaped interface

The curvature can not be calculated for every point, because the star is only C^1 continuous. These points have undefined curvature and will be left out in this report. A solution for this problem could be to use a spline between each ellipse, but due to time constraints this was not done. The curvature for the star that are defined can be calculated by calculating the curvature for each of the 10 ellipse pieces. This will create 10 matrices with curvature in MATLAB™. To get the correct curvature on the lattice, it needs to be mapped to the 10 distance matrices of the star.

5.2.3 Distance from a point to the star shaped interface

The distance from a point to the star shaped interface is the minimum of all the distances from that point to the 10 ellipse pieces of the star.

5.2.4 Contour lines of a star shaped interface

The level set signed distance function of this star can be seen in the following figure:

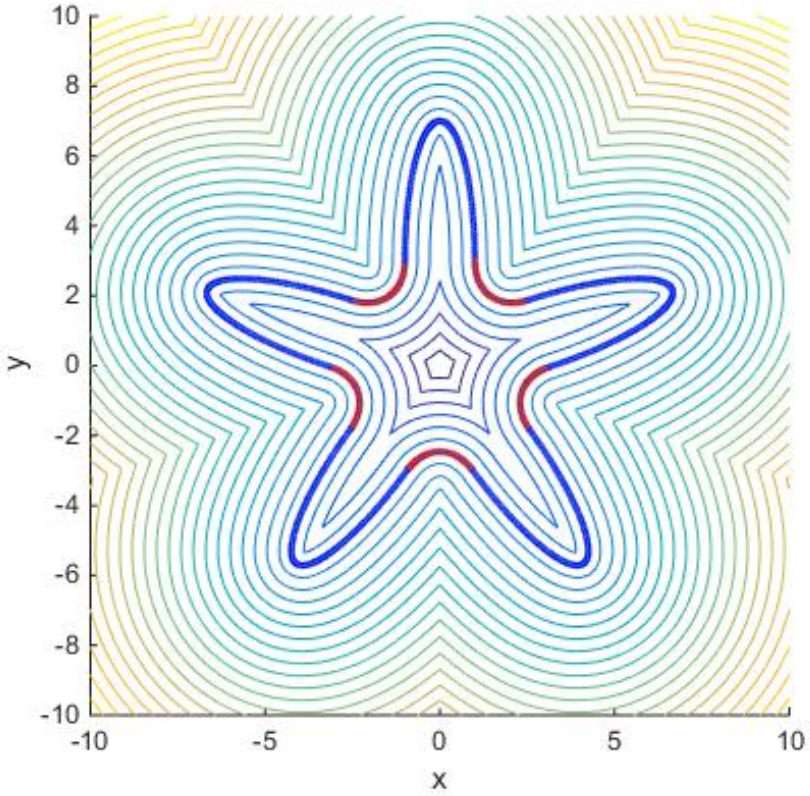

Figure 5.11: This is the level set signed distance function, the thick dotted line in the colors blue and red is the level set zero.

It can be seen from Figure 5.11 that the level set signed distance function is the same for all five half ellipses and it is the same for all five ellipse pieces.

The exact curvature for this star can be seen in the following figure:

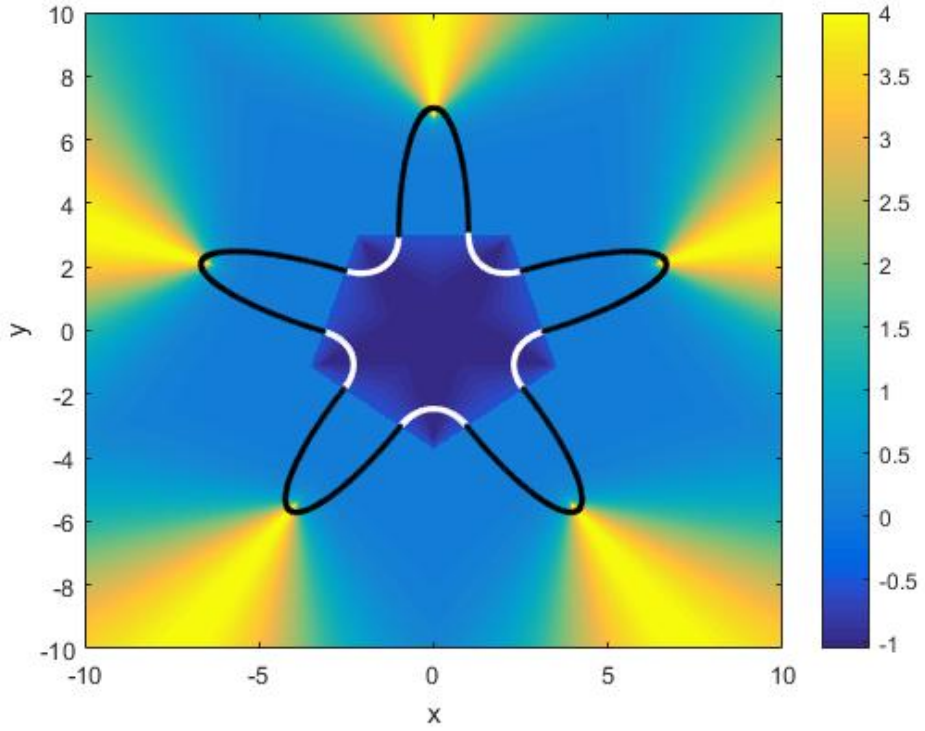

Figure 5.12: The curvature of the star shape is visible, the curvature is higher where the star bends. The colorbar represents the value of the curvature at a certain point. The original star can be seen in black and white.

In Figure 5.12 it can be seen that there are negative curvatures, this is due to the white ellipse pieces having a negative curvature.

5.3 Determining curvature near an implicitly defined interface

The first step to find the curvature without knowing the exact interface is to find the location of the interface using the level set function, the second step is to calculate the local curvature for the lattice points and the last step is to use the local curvature from the lattice points to find the curvature on the location of the interface. The last step will be done with two different methods, the first method will make use of interpolation and the second method will be a Norwegian method of concentric circles by Meland et al. [2].

5.3.1 Finding the interface using the level set function

Without knowing where the interface is, it can still be found using a level set function. Lets say that the gradient from point P goes through the level set function in point A then assuming that the gradient stays constant and thus that the gradient from point A points in about the same direction as point P, this can be seen in Figure 5.13.

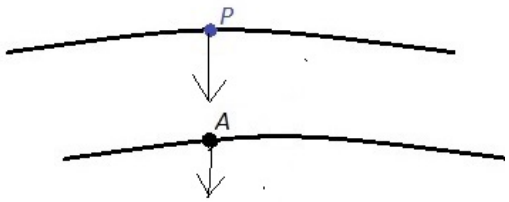

Figure 5.13: A point P which is on the unknown interface and a point A on a known level set, with respective normal gradient.

So the gradient of A can be used to find the direction in which point P lies and then the distance is needed from point P to A to find the location of point P. The coordinates of point P can be expressed as a function

$$\boldsymbol{X_P}(t) = \boldsymbol{X}_A + t\frac{\nabla\phi_A}{|\nabla\phi_A|}, \tag{5.14}$$

where $\boldsymbol{X_P}(t)$ is the coordinate vector for point P, $\boldsymbol{X}_A$ are the coordinates from point A, t is the distance from point A to point P and $\frac{\nabla\phi_A}{|\nabla\phi_A|}$ is the

normal gradient from point A.
If the signed distance function is used as level set function then it is quite easy, the distance is then given. The coordinates of point P can be expressed as a function of the distances from the points on the level set

$$\boldsymbol{X_P}(\phi_A) = \boldsymbol{X}_A + \phi_A \frac{\nabla \phi_A}{|\nabla \phi_A|}. \tag{5.15}$$

The problem with the assumption that the gradient stays the same is that it only holds for interfaces and level sets which have a relatively small curvature. If the curvature is relatively large then the difference in gradient between two relatively close points will be large, this can be seen in Figure 5.14. A second problem is that if point A is to far from point P the gradient could differ strongly from the gradient of point P and therefor point P would not be found, but a different point.

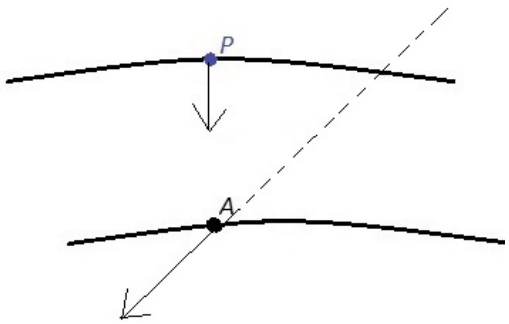

Figure 5.14: A point P which is on the unknown interface and a point A on a known level set, but now the gradient of point A differs strongly from the gradient of point P.

5.3.2 Computation of the local curvature

Finding the curvature for the level set function can be done by using the signed distance function. The local curvature can be calculated numerically for each point.

The curvature can be defined by:

$$\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}. \tag{5.16}$$

This can be rewritten according to Brackbill et al. [1] as:

$$\kappa = \frac{\nabla \cdot \nabla \phi}{|\nabla \phi|} - \frac{\nabla \phi \cdot [(\nabla \phi \cdot \nabla) \nabla \phi]}{|\nabla \phi|^3}. \tag{5.17}$$

The signed distance function can be used in (5.17), but to make it easier to use with the signed distance function it can be expressed according to Brackbill et al. [1] as:

$$\kappa = \frac{\nabla \cdot \nabla \phi}{|\nabla \phi|} - \frac{1}{|\nabla \phi|^3} \sum_i \sum_j \frac{\partial \phi}{\partial x_i} \frac{\partial \phi}{\partial x_j} \frac{\partial^2 \phi}{\partial x_i \partial x_j}. \tag{5.18}$$

The local curvature in 2 dimensions can be found by using central differences and the signed distance function.

8 neighbouring points are needed to find the local curvature of a point for the central derivatives as can be seen in Figure 5.15.
The central difference approximation for the first derivative is:

$$\frac{\partial \phi}{\partial x_i} = \frac{1}{2h}(Q_{i+1,j} - Q_{i-1,j}) + O\left(h^2\right), \tag{5.19}$$

where h is the step size. The central difference approximation for the second derivative is:

$$\frac{\partial^2 \phi}{\partial x_i^2} = \frac{1}{h^2}(Q_{i+1,j} - 2Q_{i,j} + Q_{i-1,j}) + O\left(h^2\right), \tag{5.20}$$

and the central difference approximation for the mixed derivative is:

$$\frac{\partial^2 \phi}{\partial x_i \partial x_j} = \frac{1}{4h^2}(Q_{i+1,j+1} - Q_{i+1,j-1} - Q_{i-1,j+1} + Q_{i-1,j-1}) + O\left(h^2\right). \tag{5.21}$$

The length of the gradient can be expressed by:

$$|\nabla \phi| = \sqrt{\sum_i \frac{\partial \phi}{\partial x_i}}. \tag{5.22}$$

All these finite differences have second order accuracy

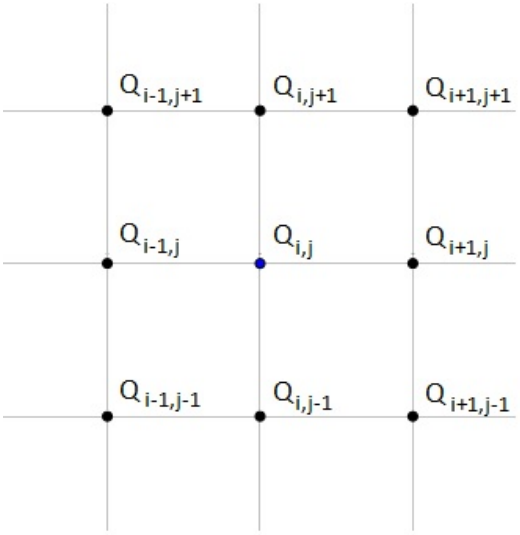

Figure 5.15: A lattice of 9 points with point $Q_{i,j}$ in the middle of the lattice, the local curvature of $Q_{i,j}$ will be calculated using the 8 neighbouring points.

5.3.3 Interpolation of the local curvature to the interface location

Calculation of the curvature for an arbitrary point can be done by interpolation, given the local curvature for the vertices of computational cells. An arbitrary point has 4 vertices which are closest to it. These vertices will be named neighbouring points. The curvature for an arbitrary point can be calculated by interpolating the local curvature of its neighbouring points.
The interpolation method that is used has to be a second order approximation and it should be easy to implement.
The easiest method would be linear interpolation, but this only works in the 1 dimensional case, since this report will cover a 2 dimensional plane it is important to do linear interpolation in two directions. The method that can be used for interpolating in two directions is called bilinear interpolation. It uses four neighbouring points and linearly interpolates over the two directions to calculate the curvature of point P, the process is illustrated in Figure 5.16. First linear interpolation will be applied along x-direction. This gives:

$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(B) + \frac{x - x_1}{x_2 - x_1} f(D)$$

$$f(x, y_2) \approx \frac{x_2 - x}{x_2 - x_1} f(A) + \frac{x - x_1}{x_2 - x_1} f(C),$$

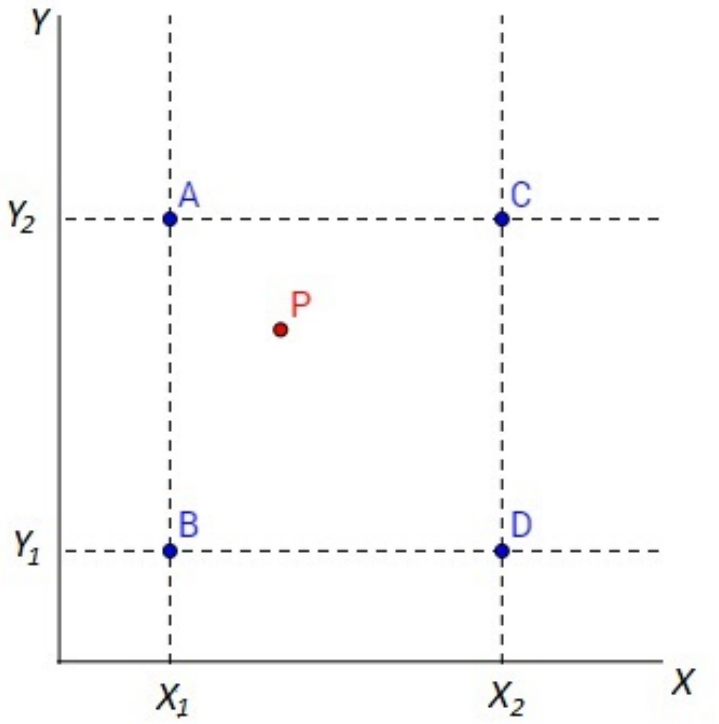

Figure 5.16: Point P of which the curvature will be calculated and points A,B,C and D are the four neighbouring points from the lattice of which the local curvature are known.

where $A = (x_1, y_2), B = (x_1, y_1), C = (x_2, y_2), D = (x_2, y_1)$.
Then linear interpolation will be applied along the y direction :

$$f(x,y) \approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2),$$

and this will result in:

$$\begin{aligned} f(x,y) \approx & \frac{(y_2 - y)(x_2 - x)}{(y_2 - y_1)(x_2 - x_1)} f(B) + \frac{(y_2 - y)(x - x_1)}{(y_2 - y_1)(x_2 - x_1)} f(D) \\ & + \frac{(y - y_1)(x_2 - x)}{(y_2 - y_1)(x_2 - x_1)} f(A) + \frac{(y - y_1)(x - x_1)}{(y_2 - y_1)(x_2 - x_1)} f(C). \end{aligned} \tag{5.23}$$

In these test cases a lattice 10 by 10 was used with step size of 1/64. The ellipse had a semi major axis of 4 and a semi minor axis of 1 and for the star see section 5.2.1. The relative difference can be taken between the interpolation method and the analytical solution of the ellipse to see how accurate the interpolation method is, this can be seen in Figure 5.17

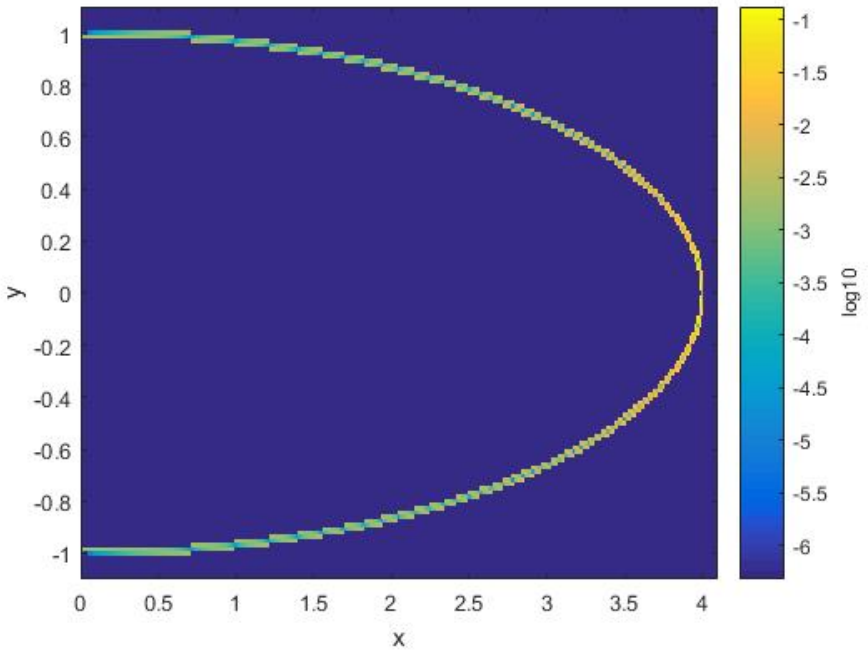

Figure 5.17: The relative difference between the interpolation method and the analytical solution of the curvature for the ellipse. Only part of the ellipse is shown due to symmetry in the y-axis. In this figure the log was taken of the relative difference to make the figure clearer.

The relative difference for the star shape can be seen in Figure 5.18.
From Figure 5.18 it can be seen what happens if the algorithm is used on the points which have undefined curvature. Figure 5.19 shows the errors if those points are taken out.

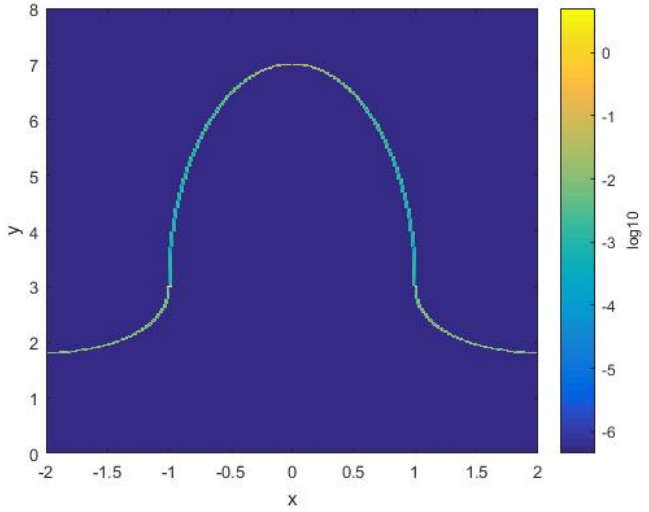

Figure 5.18: The relative difference between the interpolation method and the analytical solution of the curvature for the star. In this figure the log was taken of the relative difference to make the figure clearer.

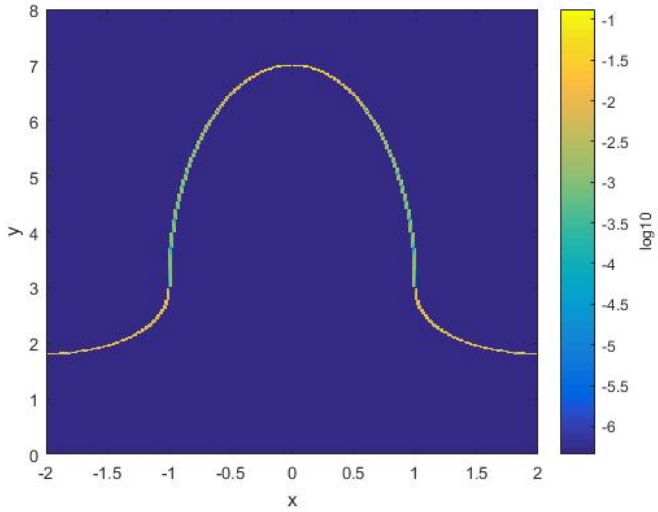

Figure 5.19: The relative difference between the interpolation method and the analytical solution of the curvature for the star. Only the top half ellipse of the star is shown, because it is the same for all the other half ellipses. In this figure the log was taken of the relative difference to make the figure clearer.

Richardson extrapolation can be used to verify that the interpolation leads

to a second order accuracy. If $N(h)$ which depends on the step size h is the approximation of M the difference gives us the error, this error can be written as a polynomial dependent on h:

$$M - N(h) = K_0 h^{\alpha_0} + K_1 h^{\alpha_1} + K_2 h^{\alpha_2} + \cdots, \tag{5.24}$$

where K_i are unknown constants and α_i are constants such that $h^{\alpha_i} > h^{\alpha_{i+1}}$, α_0 dictates the order of the approximation. Equation (5.24) can be approximated, because h is very small:

$$M - N(h) \approx K_0 h^{\alpha_0} + \mathcal{O}(h^{\alpha_1}) \approx K h^{\alpha}. \tag{5.25}$$

If Equation (5.25) is used for smaller step sizes it can be written as an expression for α which represents the order:

$$\begin{aligned} M - N(h) &= K h^{\alpha} \\ M - N(\frac{h}{2}) &= K \left(\frac{h}{2}\right)^{\alpha} \\ M - N(\frac{h}{4}) &= K \left(\frac{h}{4}\right)^{\alpha}, \end{aligned} \tag{5.26}$$

subtract one the equations by the other to get:

$$\begin{aligned} N(\frac{h}{2}) - N(h) &= K h^{\alpha} \left(1 - \frac{1}{2}^{\alpha}\right) \\ N(\frac{h}{4}) - N(\frac{h}{2}) &= K \left(\frac{h}{2}\right)^{\alpha} \left(1 - \frac{1}{2}^{\alpha}\right). \end{aligned} \tag{5.27}$$

An expression for α can be found:

$$\alpha = \frac{\log\left(\frac{N(\frac{h}{2}) - N(h)}{N(\frac{h}{4}) - N(\frac{h}{2})}\right)}{\log(2)}. \tag{5.28}$$

The expectation is that the bilinear interpolation method has a second order accuracy and when using (5.27) an order of 1.6 is achieved, which means that this has an order of 2. This is not exactly 2 due to rounding errors from MATLAB™ when calculating the local curvature and calculating the interpolation of the local curvature.

5.3.4 The method of concentric circles

There is another way the curvature for an arbitrary point can be calculated using neighbouring points. If it is assumed that the isolines of the level set form concentric circles which is introduced by Meland et al. [2] the local curvature of neighbouring points can be used to find the interface curvature. The curvature is dependent on the radius of curvature, so in order to use concentric circles the radius of curvature has to be very large compared to the distance from an arbitrary point to the interface. In Figure 5.20 the concept is illustrated.

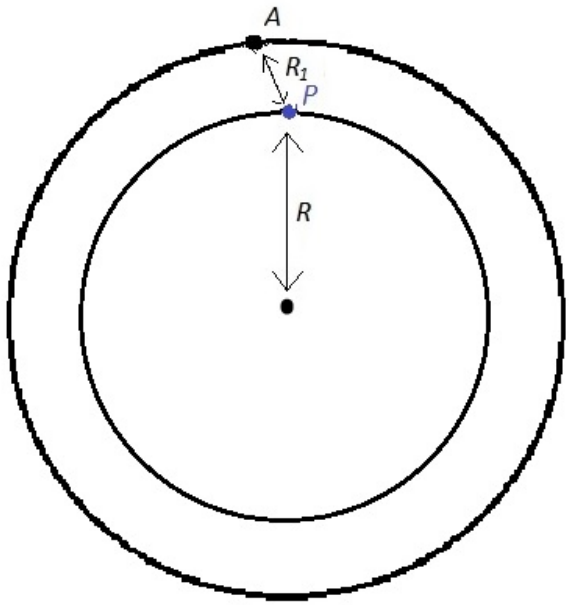

Figure 5.20: The curvature of point P will be calculated and Point A of which the local curvature is known which are located on concentric circles, where R is the radius of curvature for point P and R_1 is the level set for point A.

So assuming that R is very large and $R >> R_1$ the local curvature can be used from A to calculate the curvature for P. An expression for the curvature in point P:

$$\kappa = \frac{1}{R}. \tag{5.29}$$

Since R is very large the radius of curvature can be approximated for point A as $R + R1$, then a similar expression can be given for point A:

$$\kappa^* = \frac{1}{R + R1}. \tag{5.30}$$

Using (5.29) and (5.30) an expression for the curvature of point P can be achieved using the curvature from point A

$$\kappa = \frac{\kappa^*}{1 - \kappa^* R1}, \tag{5.31}$$

where R is the radius of curvature and R_1 is the level set for a point A.

In these test cases a lattice 10 by 10 was used with step size of 1/64. The ellipse had a semi major axis of 4 and a semi minor axis of 1 and for the star see section 5.2.1. The relative error with respect to the analytical curvature for this method can be seen in Figure 5.21

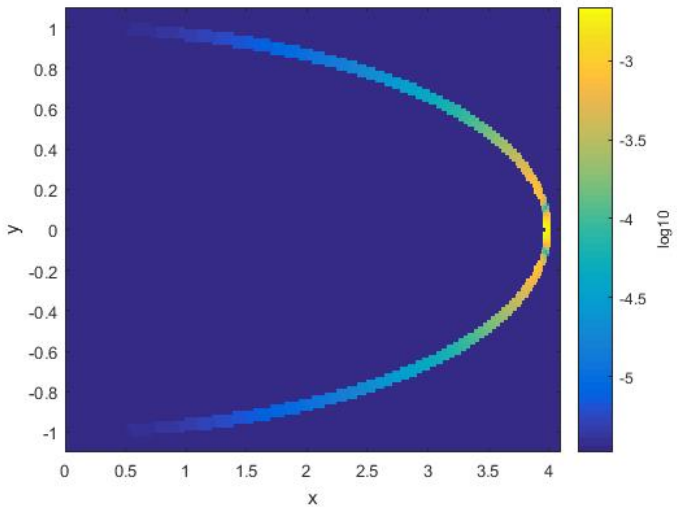

Figure 5.21: The relative difference between the method of concentric circles and the analytical solution of the curvature for the ellipse. Only part of the ellipse is shown due to symmetry in the y axis. In this figure the log was taken of the relative difference to make the figure clearer.

The relative difference for the star shape can be seen in Figure 5.22. From Figure 5.22 it can be seen what happens if the algorithm is used on the points which have undefined curvature. Figure 5.23 shows the errors if those points are taken out.

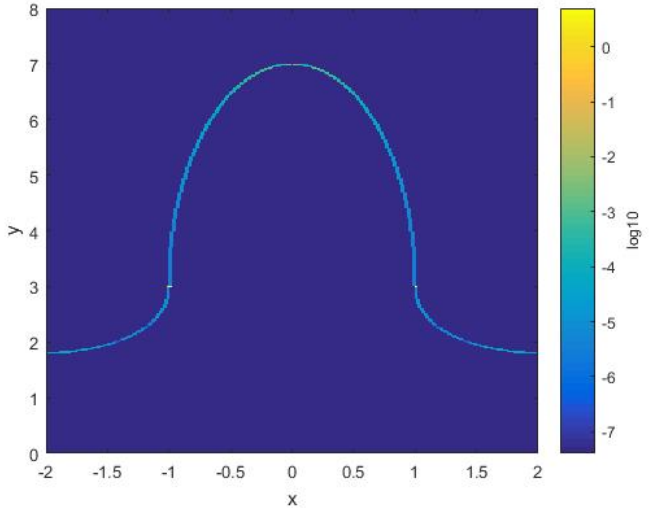

Figure 5.22: The relative difference between the method of concentric circles and the analytical solution of the curvature for the star. In this figure the log was taken of the relative difference to make the figure clearer.

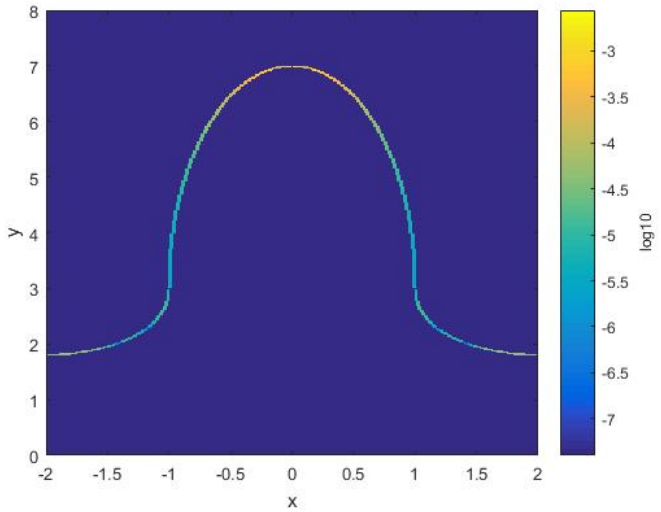

Figure 5.23: The relative difference between the method of concentric circles and the analytical solution for the star. Only the top half ellipse of the star is shown, because it is the same for all the other half ellipses. In this figure the log was taken of the relative difference to make the figure clearer.

5.3.5 Comparing the interpolation method and the concentric circle method

The two methods of calculating the curvature near the interface will be compared in terms of accuracy and work time. These are the method of interpolation and the method of concentric circles. Figures 5.17 and 5.21 show the relative difference between the respective method and the analytical solution for the elliptical interface. Figures 5.19 and 5.23 show the relative difference between the respective method and the analytical solution for the star shaped interface.

Looking at the accuracy. From Figure 5.21 can be seen that the method of concentric circles for the elliptical interface has a maximum of $10^{-2.5} \approx 0.003$ relative difference and from Figure 5.17 it can be seen that the method of interpolation for the elliptical interface has a maximum of $10^{-1} = 0.1$ relative difference. So the concentric circles method is much more accurate in comparison to the method of interpolation.

From Figure 5.23 it can be seen that the concentric circles method for the star shaped interface has a maximum of $10^{-2.5} \approx 0.003$ relative difference. From Figure 5.19 it can be seen that the interpolation method for the star shaped interface has a maximum of $10^{-1} = 0.1$. So the concentric circles method is much more accurate in comparison to the method of interpolation.

The work time is defined as the time that MATLAB™ takes to finish executing the code, MATLAB's 'tic' 'toc' was used. The mean work time of the method of concentric circles is 2.62 seconds and interpolation function it is 2.95 seconds. It must be noted that MATLAB's 'tic' 'toc' is a bit arbitrary, so since the time difference is not that large there is not a clear faster method.

The method of concentric circles has a higher accuracy and takes about the same time as the interpolation method does, so the best choice is to use the method of concentric circles.

5.4 Distribution of the curvature to other cells

The concentric circles method can calculate the curvature near the interface accurately. The best method to get the curvature to the rest of the cells is distributing the curvature by extrapolating. Extrapolating will distribute the curvature in the desired direction and will be much faster than interpolation. The distribution of the curvature from the interface to the lattice points can be seen in Figure 5.24:

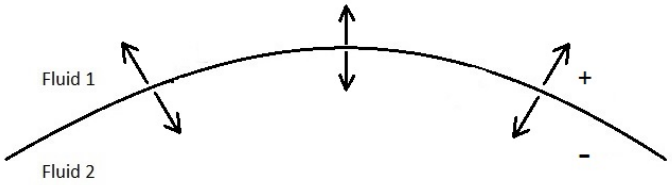

Figure 5.24: The interface between two fluids with fluid 1 being '+' and fluid being '-'. The curvature needs to be distributed along the normal and negative normal to the lattice points of the interface.

This will be done using a linear transport equation which can be expressed as:

$$\frac{\partial \kappa}{\partial t} = -\boldsymbol{u}\nabla \cdot \kappa. \tag{5.32}$$

The velocity is in this case is the direction in which the curvature will be distributed this velocity can be expressed as:

$$\boldsymbol{u} = \frac{\nabla \phi}{|\nabla \phi|} \cdot \text{sign}(\phi), \tag{5.33}$$

the sign(ϕ) will be negative if ϕ is negative and positive if ϕ is positive.
Equation (5.32) will be solved numerically using an upwind discretization to get the solution. The upwind discretization makes use of adaptive finite differences to numerically simulate the propagation direction of the curvature. Note that the upwind discretization is first order accurate even though everything up till now has had second order accuracy. This will be done in 2 dimensions:

$$\frac{\partial \kappa_{ij}}{\partial t} = -\frac{u_x \partial(\kappa_{ij})}{\partial x} - \frac{u_y \partial(\kappa_{ij})}{\partial y}$$

$$\Leftrightarrow$$

$$\kappa_{ij}|_{n+1} = \kappa_{ij}|_n - \delta t \left(\frac{u_x(\kappa_{ij}|_n - \kappa_{i-1j}|_n)}{h} + \frac{u_y(\kappa_{ij}|_n - \kappa_{ij-1}|_n)}{h} \right), \tag{5.34}$$

where δt is the time step size.

It is important to know when implementing this equation that the curvature near the interface must be kept the same. When distributing the curvature to the other points it is also important that points on the '+' side only get information from the interface or other points on the '+' side and not get information from the '-' side. The sign of the velocity can be used to discern the '+' and '-' side in 2 dimensions:

$$\begin{aligned}
&\kappa_{ij}|_{n+1} = \\
&\kappa_{ij}|_n - \delta t \frac{1}{2}(1 + \text{sign}(u_x)) \cdot \left(\frac{u_x(\kappa_{ij}|_n - \kappa_{i-1j}|_n)}{h} \right) \\
&- \delta t \frac{1}{2}(1 + \text{sign}(u_x)) \left(\frac{u_y(\kappa_{ij}|_n - \kappa_{ij-1}|_n)}{h} \right) \\
&- \delta t \frac{1}{2}(1 - \text{sign}(u_x)) \cdot \left(\frac{u_x(\kappa_{i+1j}|_n - \kappa_{ij}|_n)}{h} \right) \\
&- \delta t \frac{1}{2}(1 - \text{sign}(u_y)) \left(\frac{u_y(\kappa_{ij+1}|_n - \kappa_{ij}|_n)}{h} \right),
\end{aligned} \tag{5.35}$$

In section 5.3.5 two methods were compared to determine the curvature near an unknown interface, it was concluded that the method of concentric circles is the best method to use for our case. First the method of concentric circles is used to determine the curvature near an unknown interface and then the determined curvature will be distributed to the other points.

5.4.1 Distribution of the curvature of the ellipse to the other cells

If Equation (5.34) is implemented, the curvature distribution can be seen over time and what the relative difference is between each iteration and the reference model. It does not matter what the start value is of the lattice when the distributing is done, because the equilibrium solution of the lattice will be the same it only affects how many iteration are needed to achieve the same accuracy. In these figures the distribution was used on a lattice with only zero values as start value instead of the lattice with the local curvature, this was done to make the figures clearer. The relative difference after 100 iterations can be seen in Figure 5.25. The relative difference after 200 iterations can be seen in Figure 5.26. The relative difference after 300 iterations can be seen in Figure 5.27. From these three figures it can be seen

that the relative difference over the whole lattice is decreasing, since the band with low relative difference is increasing in size with more iterations. In my implementation only between a rectangle that surrounds the ellipse and a rectangle that is surrounded by the ellipse was used to observe the behavior of the distribution. The ellipse that was used had a semi major axis of 4 and a semi minor axis of 1, a lattice 10 by 10 was used with step size of 1/64 and a time step of 1/1280

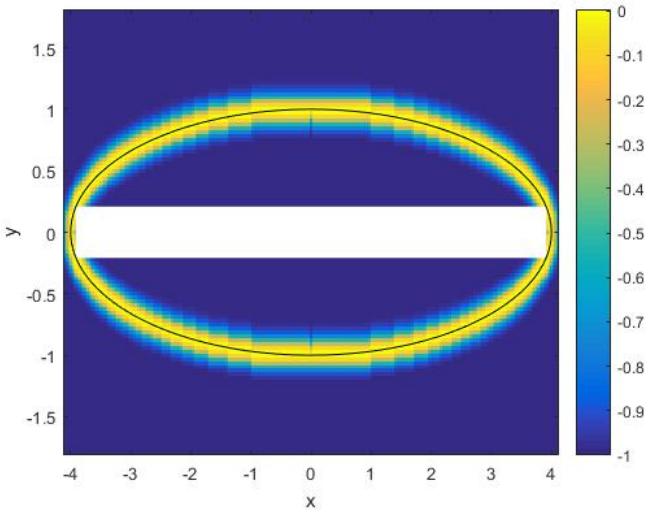

Figure 5.25: The relative difference of the distributed curvature and the analytical solution after 100 time steps. The white rectangle is the rectangle that was taken out and the black line is the interface.

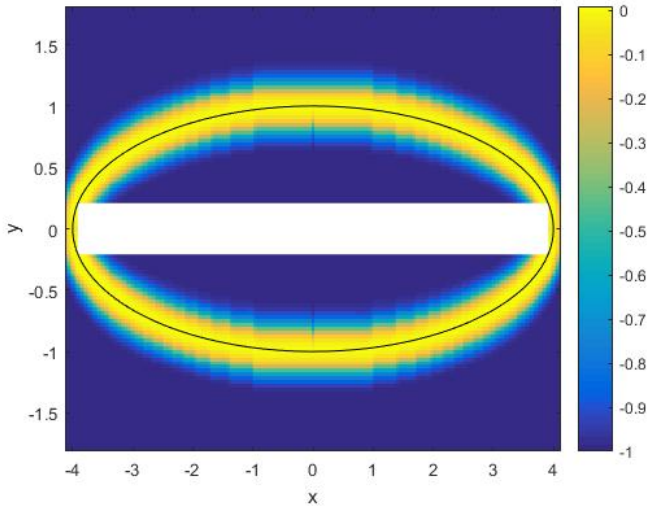

Figure 5.26: The relative difference of the distributed curvature and the analytical solution after 200 time steps. The white rectangle is the rectangle that was taken out and the black line is the interface.

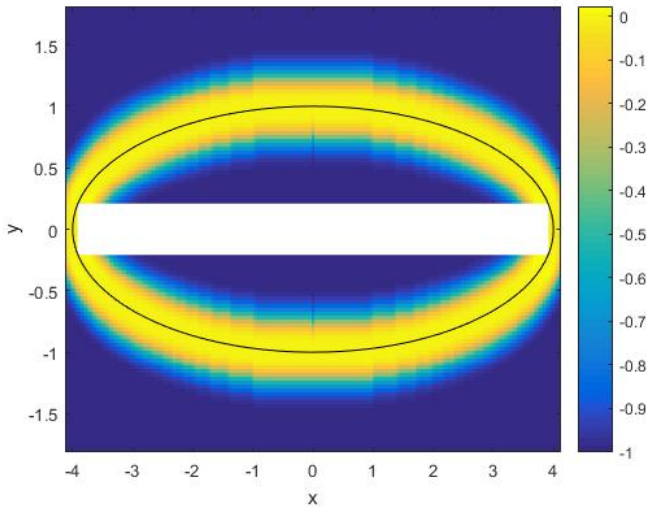

Figure 5.27: The relative difference of the distributed curvature and the analytical solution after 300 time steps. The white rectangle is the rectangle that was taken out and the black line is the interface.

It is not enough to have the overall relative difference of the lattice decrease, it is also important that the relative difference decreases with each iteration for an arbitrary point.

So for a point with $x = 4.125$ and $y = 0$ after 100 time steps it has a relative difference of -0.8330, after 200 time steps -0.1850 and after 300 time steps -0.0148. So the relative difference for an arbitrary point decreases as more time steps have passed.

5.4.2 Distributing the curvature of the star to the other cells

If Equation (5.34) is implemented, the curvature distribution can be seen over time and what the relative difference is between each iteration and the reference model. It does not matter what the start value is of the lattice when the distributing is done, because the equilibrium solution of the lattice will be the same it only affects how many iteration are needed to achieve the same accuracy. In these figures the distribution was used on a lattice with only zero values instead of the lattice with the local curvature, this was done to make the figures clearer. The relative difference after 100 iterations can be seen in Figure 5.28. The relative difference after 200 iterations can be seen in Figure 5.29. The relative difference after 300 iterations can be seen in Figure 5.30. From these three figures it can be seen that the relative difference over the whole lattice is decreasing, since the band with low relative difference is increasing in size with more iterations. Two lines with a high relative difference can be seen in the figures, this is due to the curvature of the analytical solution not existing there. In my implementation only a square surrounding one of the connection pieces was used to observe the behavior of the distribution. Details on the star can be found in section 5.2.1, a lattice 10 by 10 was used with step size of 1/64 and a time step of 1/1280

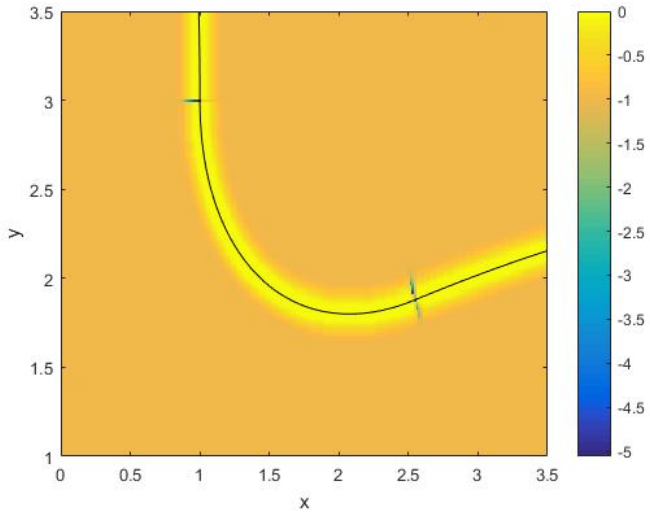

Figure 5.28: The relative difference of the distributed curvature of a connecting piece of the star and the analytical solution after 100 time steps. The black line is the interface.

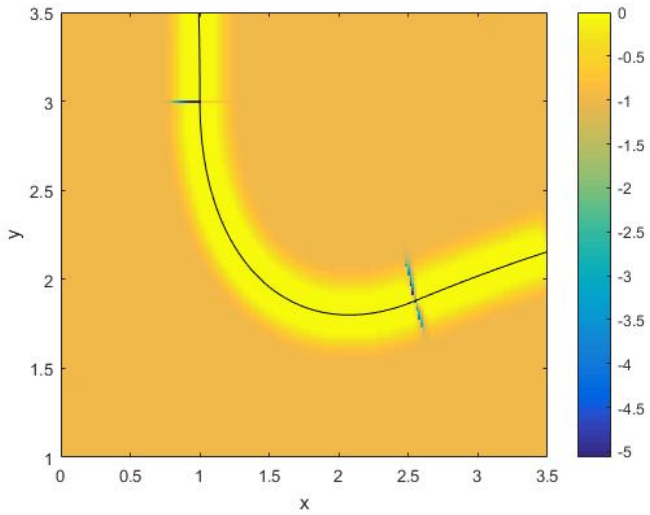

Figure 5.29: The relative difference of the distributed curvature of a connecting piece of the star and the analytical solution after 200 time steps. The black line is the interface.

It is also important that the relative difference decreases with each iteration for an arbitrary point.
So for a point with $x = 1.25$ and $y = 1.875$ after 100 time steps it has a

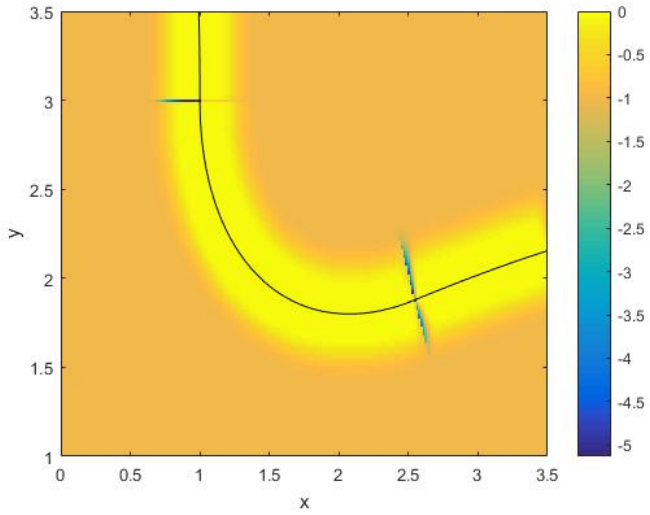

Figure 5.30: The relative difference of the distributed curvature of a connecting piece of the star and the analytical solution after 300 time steps. The black line is the interface.

relative difference of -0.9999, after 200 time steps -0.8549 and after 300 time steps -0.2448. So the relative difference for an arbitrary point decreases as more time steps have passed.

Chapter 6

The influence of using the calculated curvature as opposed to the local curvature

One of the differences that other research groups make compared to the Scientific computing group of DIAM when implementing the CSF model is that other research groups use the interface curvature, instead of the local curvature. In the next section it will be clarified that using the local curvature is a cause for having a worse correlation with respect to ascent speed than if the interface curvature is used. In practice the usual thickness of the transition region is $\frac{3h}{2}$. In these test cases a lattice 10 by 10 was used with step size of 1/64 and a time step of 1/1280. The ellipse had a semi major axis of 4 and a semi minor axis of 1 and for the star see section 5.2.1

In the previous chapter a way to calculate the curvature on a lattice was discussed by using the curvature of the interface. The relative difference between the exact model and the local curvature of the ellipse can be seen in Figure 6.1 and the relative difference between the exact model and the calculated curvature of the ellipse can be seen in Figure 6.2. In the Figures only part of the lattice was calculated $\frac{3h}{2}$ from the interface, since the curvature is needed in a small band along the interface in real world practice.

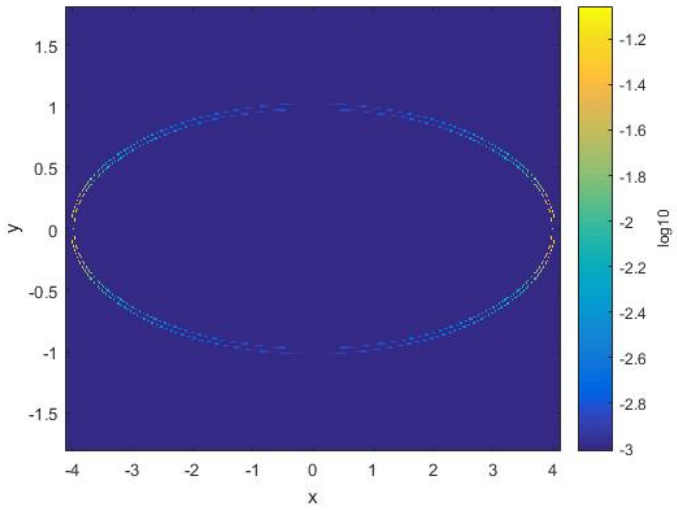

Figure 6.1: The relative difference of the local curvature and the analytical solution of the ellipse in a range of $\frac{3h}{2}$ from the interface. In this figure the log was taken of the relative difference to make the figure clearer.

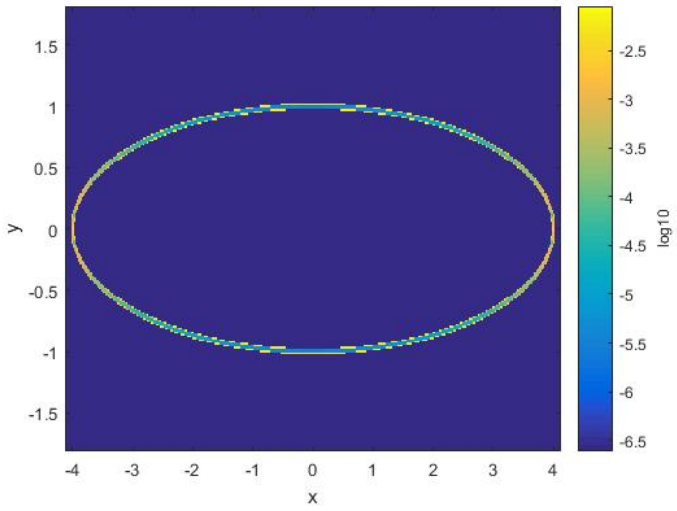

Figure 6.2: The relative difference of the calculated curvature and the analytical solution of the ellipse in a range of $\frac{3h}{2}$ from the interface. In this figure the log was taken of the relative difference to make the figure clearer.

From Figure 6.1 and Figure 6.2 it can be seen that for the ellipse the calculated curvature error is magnitude of 10 smaller than the local curvature

error.

The same can be done for a lattice with 4 times as less points this can be seen in Figure 6.3 and Figure 6.4

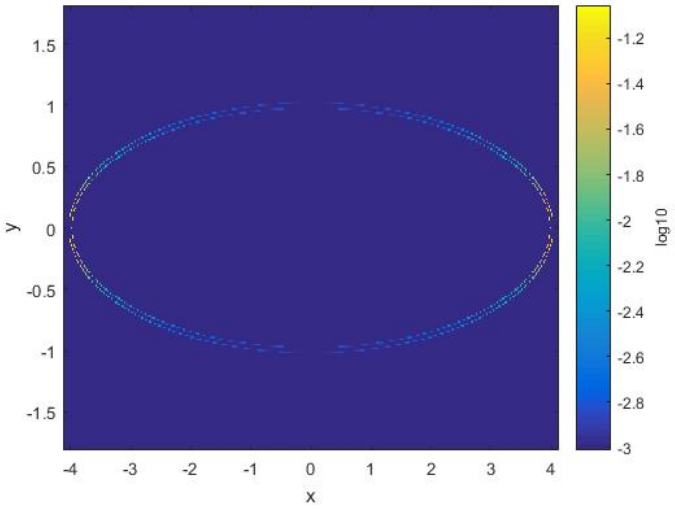

Figure 6.3: The relative difference of the local curvature and the analytical solution of a 4 times less dense ellipse in a range of $\frac{3h}{2}$ from the interface. In this figure the log was taken of the relative difference to make the figure clearer.

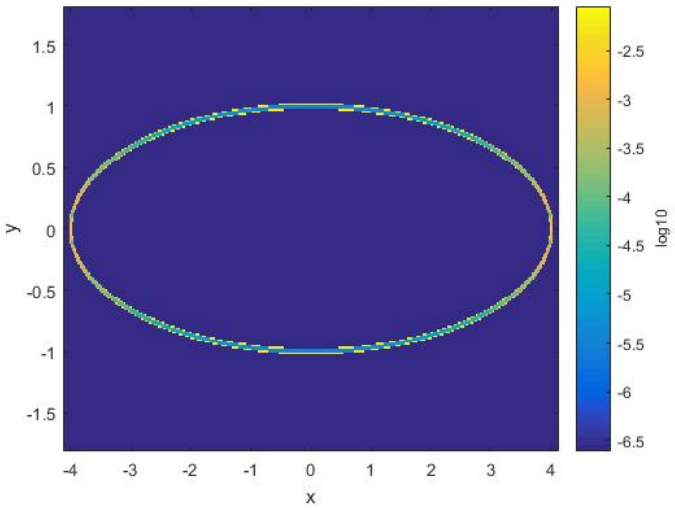

Figure 6.4: The relative difference of the calculated curvature and the analytical solution of a 4 times less dense ellipse in a range of $\frac{3h}{2}$ from the interface. In this figure the log was taken of the relative difference to make the figure clearer.

From Figure 6.3 and Figure 6.4 it can be seen that for the ellipse, with a 4 times less dense lattice, the calculated curvature error is magnitude of 1.5 smaller than the local curvature error.

For the star the relative difference between the exact model and the local curvature can be seen in Figure 6.5 and the relative difference between the exact model and the calculated curvature can be seen in Figure 6.6. In the Figures only part of the lattice was calculated $\frac{3h}{2}$ from the interface, since only a small part of the curvature is needed in the real world practice.

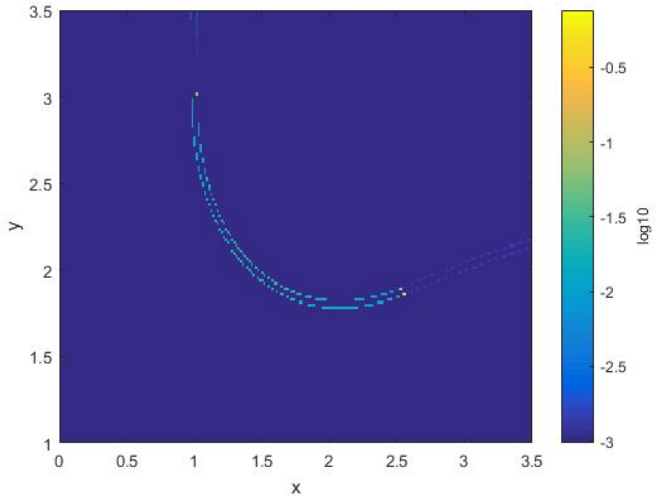

Figure 6.5: The relative difference of the local curvature and the analytical solution of a connecting piece of the star in a range of $\frac{3h}{2}$ from the interface. The points where the curvature does not exist were taken out. In this figure the log was taken of the relative difference to make the figure clearer.

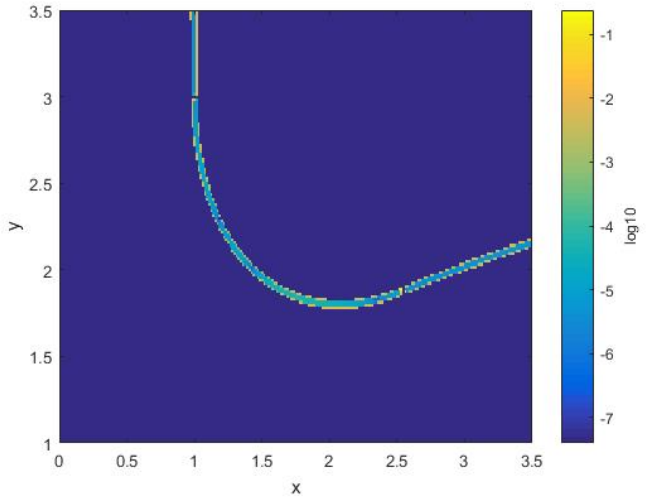

Figure 6.6: The relative difference of the calculated curvature and the analytical solution of a connecting piece of the star in a range of $\frac{3h}{2}$ from the interface. The points where the curvature does not exist were taken out. In this figure the log was taken of the relative difference to make the figure clearer.

From Figure 6.5 and Figure 6.6 it can be seen that for the star the calculated curvature error is magnitude of 2 smaller than the local curvature error. This result is only possible, if the points with nonexistent curvature are not included.

Chapter 7

Conclusions and recommendations

From the analysis of the Brackbill paper, only one equation was mathematically incorrect this was Equation (2.25). In the weak sense Equation (2.25) is correct, but in the strong sense it is not a solution. A solution would be to keep everything as Equation (3.4).

The consequence of using the density weight function is that the pressure difference for a finite interface will be dependent on the density difference, which it should not be. There will be a difference for the implementation of the CSF model if the density weight function is used or not. For instance the example with water-air the surface force with density weighting was 100 times smaller than without the density weighting. So the density weighting should not be used for finite h.

Two methods were tested for determining the curvature near the interface it turns out that the concentric circles method is more accurate than the interpolation method. This is true if $R >> R_1$ see Figure 5.20 as reference, this was the case in this report. Calculating the curvature near the interface and distributing it, can result in a magnitude of 10 times more accurate than using the local curvature over a thickness of $\frac{3h}{2}$.

The method of calculating the curvature is worth the effort for the accuracy gain in the curvature even in the not convex test case it achieved a accuracy gain of 2.

The whole method of calculating the curvature was done with a second order accuracy, the distribution was done with a first order accuracy. Even with a

first order accuracy the results are clear that calculating the curvature and distributing it is more accurate than using the local curvature, this would mean that if the distribution was done with a second order accuracy the curvature could have been even more accurate on the cells.

Future research on this subject using this report could be using a spline in between the ellipses for the star. In this report the star had a discontinuous curvature and the points that had undefined curvature had to be omitted.

If the method of calculating and distributing the curvature is used it could be useful to use a second order accuracy for the distribution.

Bibliography

[1] Brackbill J.U., Kothe D.B. and Zemach, C. (1992)
A Continuum Method for Modeling Surface Tension.*
JOURNAL OF COMPUTATIONAL PHYSICS

[2] Meland R., Gran I.R., R. Olsen and Munkejord, S.T. (2007)
Reduction of parasitic currents in level-set calculations with a consistent discretization of the surface-tension force for the CSF model.
16th Australasian Fluid Mechanics Conference Crown Plaza, Gold Coast, Australia

[3] Baltussen M.W., Kuipers J.A.M. and Deen, N.G. (2014)
A critical comparison of surface tension models for the volume of fluid method.
Chemical Engineering Science

[4] Denner Fabian and van Wachem Berend G. M. (2014)
FULLY-COUPLED BALANCED-FORCE VOF FRAMEWORK FOR ARBITRARY MESHES WITH LEAST-SQUARES CURVATURE EVALUATION FROM VOLUME FRACTIONS.
Numerical Heat Transfer, Part B: Fundamentals