Lecture notes
Introduction to numerical methods for interfacial flows

by Stephane Popinet

June 8, 2011

Table of contents

1 Basic equations

Given a control volume V of boundary S, the variable-density incompressible Euler equations can be written in integral form.

Mass conservation:

$$\frac{d}{dt}\int_V \rho\, dv + \oint_S \rho\, \boldsymbol{u}\cdot\boldsymbol{n}\, ds = 0 \tag{1}$$

Momentum conservation:

$$\frac{d}{dt}\int_V \rho\, \boldsymbol{u}\, dv + \oint_S \left(-p\,\boldsymbol{n} + \rho\, \boldsymbol{u}\,(\boldsymbol{u}\cdot\boldsymbol{n})\right) ds = \int_V \boldsymbol{f}\, dv \tag{2}$$

Incompressibility:

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0$$

with ρ the density, $\boldsymbol{u}$ the velocity, p the pressure, $\boldsymbol{n}$ the unit vector normal to S and $\boldsymbol{f}$ a force per unit volume. Of course, incompressibility does not imply (spatially-)constant density, it only means that material particles (advected by the flow) have a constant density (as reflected in the mass conservation equation).

From a numerical point of view, this set of equations can be decomposed in three subproblems:

1. advection of quantities: ρ and $\rho\,\boldsymbol{u}$,
2. computation of the surface tension volumetric force $\boldsymbol{f}_\sigma$,
3. imposition of the incompressibility condition.

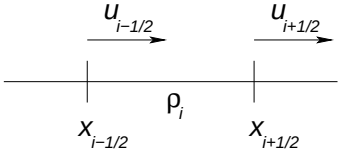
In the following we will introduce the concepts required to solve 1 and 2 in the context of interfacial flows. The imposition of incompressibility 3 is not specific to interfaces and we refer the reader to the existing litterature on this topic [3].

2 Advection scheme

As a starting point, we can assume that the scalar field ρ and vector field $\rho\,\boldsymbol{u}$ are generic and

share common properties such as continuity, differentiability, etc... A generic scheme can thus be derived which could be suitable for advection of either ρ or $\rho\,\boldsymbol{u}$. For simplicity we will consider below only advection of a scalar quantity ρ (which is not necessarily density).

Let us first consider a one-dimensional discrete version of equation (1). If space is discretised like this



where $u_{i+1/2}$ is the "face-averaged" velocity

$$u_{i+1/2} = \oint_{S_{i+1/2}} u\,ds$$

and ρ_i is the "cell-averaged" density,

$$\rho_i = \frac{1}{\Delta}\int_{V_i} \rho\,dv = \rho_i = \frac{1}{\Delta}\int_{x_{i-1/2}}^{x_{i+1/2}} \rho\,dx$$

with

$$\Delta = x_{i+1/2} - x_{i-1/2}$$

a simple discrete version of (1) can then be written

$$\Delta\,\frac{d}{dt}\rho_i = \rho_{i-1/2}\,u_{i-1/2} - \rho_{i+1/2}\,u_{i+1/2}.$$

Note that the dimension of the one-dimensional "cell volume" is Δ while the dimension of the zero-dimensional "face area" is unity. This evolution equation is only valid for the cell average ρ_i, so the first question is how to compute the "face-average" $\rho_{i+1/2}$ from the cell average ρ_i? A natural choice is

$$\rho_{i+1/2} = \frac{\rho_i + \rho_{i+1}}{2}$$

The resulting scheme is known as a "centered advection scheme". The interpolated value $\rho_{i+1/2}$ is second-order accurate i.e. it will be exact for ρ varying linearly. In more general cases, the interpolation error will scale to leading order as $O(\Delta^2)$.

For practical use, we also need to discretise time. This can be done for example using a first-order time-discretisation such as

$$\Delta\,\frac{d}{dt}\rho_i \simeq \Delta\,\frac{\rho_i^{n+1} - \rho_i}{\Delta t} = \rho_{i-1/2}\,u_{i-1/2} - \rho_{i+1/2}\,u_{i+1/2}$$

where n is the discrete timestep. Combining it all together gives the explicit update algorithm

$$\rho_i^{n+1} = \rho_i^n + \frac{\Delta t}{2\,\Delta}\left[(\rho_i^n + \rho_{i-1}^n)\,u_{i-1/2} - (\rho_i^n + \rho_{i+1}^n)\,u_{i+1/2}\right] \tag{3}$$

This algorithm is formally second-order accurate in space and first-order accurate in time. It is also *discretely conservative* by construction since the fluxes at the boundaries between cells cancel out exactly. We will also assume that this scheme is stable provided a Courant-Friedrich-Levy (CFL) condition is verified i.e.

$$\Delta t \leqslant \frac{\Delta}{U} \tag{4}$$

Can we use this scheme in the case of flows with interfaces? In one dimension, an interface is entirely defined by its position x_I and the density can be expressed as

$$\rho(x) = \begin{cases} \rho_1 \text{ if } x < x_I \\ \rho_2 \text{ if } x \geqslant x_I \end{cases}$$

where ρ_1 and ρ_2 are the densities of the phases on the left and right sides of the interface respectively. Note also that in one dimension the incompressibility condition

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0,$$

has for only trivial solution $u = \text{constant} = U$. Figure 1 illustrates what happens if we use scheme

(3) to advect this density field. Only the first three timesteps are shown. It is clear that the scheme creates oscillations of increasing amplitude. The CFL condition (4) is not sufficient for stability.

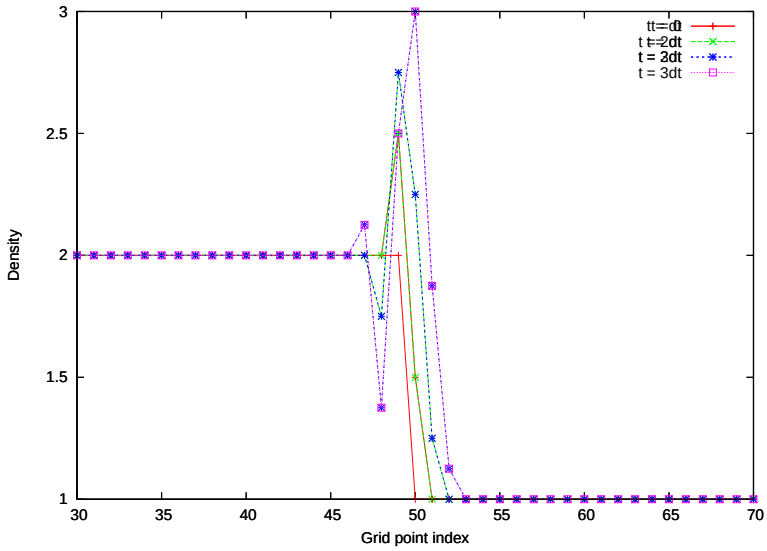


Figure 1. Explicit centered advection scheme applied to a discontinuous density field. $\rho_1 = 2$, $\rho_2 = 1$.

2.1 First-order upwind scheme

This instability is well-known for naive centered schemes. A simple way to construct a stable scheme is to use "upwinding" i.e. to construct *asymmetric* fluxes which take into account the direction of propagation. For example if we assume that U is positive, scheme (3) can be modified as

$$\rho_i^{n+1} = \rho_i^n + \frac{\Delta t}{\Delta}\left[\rho_{i-1}^n\, u_{i-1/2} - \rho_i^n\, u_{i+1/2}\right] \tag{5}$$

where we see that the face values of ρ needed to compute the fluxes are now approximated as

$$\rho_{i+1/2} = \rho_i$$

This "extrapolation" of the centered values is of course only first-order accurate, that is, it will be exact only for $\rho =$ constant. Does this work better for advection of a discontinuous density field? Figure 2 illustrates the result.

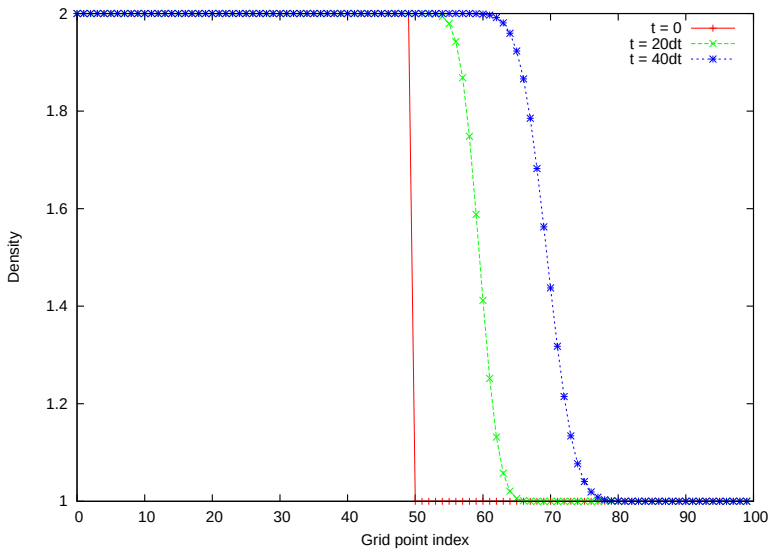


Figure 2. First-order upwind advection scheme applied to a discontinuous density field. $\rho_1 = 2$, $\rho_2 = 1$.

This looks much better than the centered scheme. In particular the density stays bounded between 1 and 2, however the front does not remain sharp. Actually the thickness of the front (i.e. the interface) increases with time. This is a manifestation of *numerical diffusion*. Can we do better?

2.2 Second-order upwind scheme

We see that we used first-order extrapolation to derive the "upwind" face-centered value of ρ. A simple idea would be to use an higher-order upwind extrapolation. For example, the second-order extrapolation

$$\rho_{i+1/2} = \rho_i + (\rho_i - \rho_{i-1})/2$$

This gives the second-order upwind scheme

$$\rho_i^{n+1} = \rho_i^n + \frac{\Delta t}{2\,\Delta}\left[(3\,\rho_{i-1}^n - \rho_{i-2}^n)\,u_{i-1/2} - (3\,\rho_i^n - \rho_{i-1}^n)\,u_{i+1/2}\right] \tag{6}$$

Figure 3 illustrates the results for this scheme.

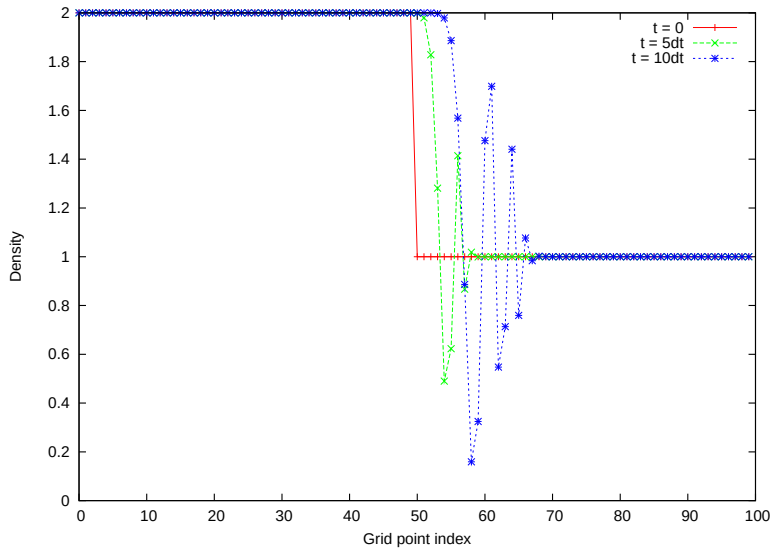


Figure 3. Second-order upwind advection scheme applied to a discontinuous density field. $\rho_1 = 2$, $\rho_2 = 1$.

While this looks better than the second-order centered scheme, there are large oscillations "downstream" of the interface i.e. the scheme looks less diffusive than the first-order upwind scheme but is not stable enough. One way to increase stability is to use "slope-limiters". We can generalise the extrapolation operator as

$$\rho_{i+1/2} = \rho_i + \phi(r)\,(\rho_i - \rho_{i-1})/2$$

with

$$r = \frac{\rho_i - \rho_{i-1}}{\rho_{i+1} - \rho_i},$$

and ϕ a "slope limiting" function. The idea is simple. If ρ is non-oscillatory then r (the "curvature" of field ρ) is of order one and the second-order scheme (6) can be used (i.e. $\phi(r) = 1$), otherwise the stable first-order scheme (5) should be used (i.e. $\phi(r) = 0$). The detail of the depen-

dence of ϕ on r is largely determined from empirical considerations and is problem-dependent. A simple choice is the "minmod" limiter

$$\phi(r) = \begin{cases} r \text{ if } r < 1 \\ 1 \text{ otherwise} \end{cases}$$

which is the most dissipative of the Total-Variation-Diminishing (TVD) scheme. Another choice is the "van Leer" limiter

$$\phi(r) = \frac{r + |r|}{1 + |r|}$$

which is generally a good compromise between dissipation and stability.

For the interface advection problem this gives the results of Figure 4 and 5 respectively.

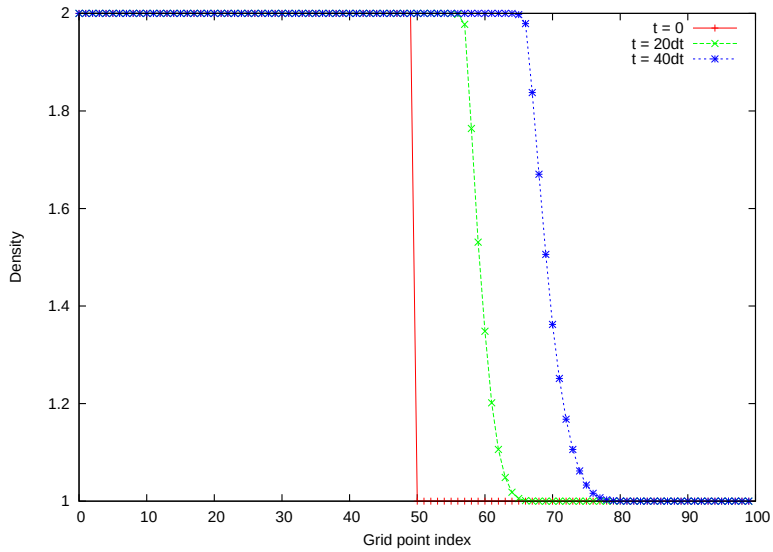


Figure 4. Minmod-limited upwind advection scheme applied to a discontinuous density field. $\rho_1 = 2$, $\rho_2 = 1$.

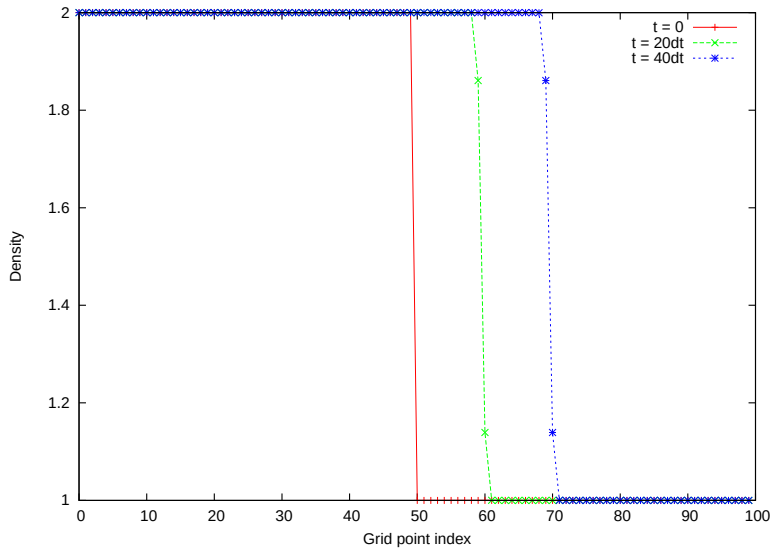


Figure 5. Van Leer-limited upwind advection scheme applied to a discontinuous density field. $\rho_1 = 2$, $\rho_2 = 1$.

It is clear that the van Leer limiter in particular considerably reduces the numerical diffusion compared to the first-order upwind scheme (Figure 2) and without the oscillations of the second-order upwind scheme (Figure 3). Note however that the interface is not sharp anymore but seems to stabilise to a thickness of about four grid points.

This slope-limited scheme can be extended to more than one dimension and to a second-order time discretisation. This gives the Bell-Collela-Glaz (1989) scheme implemented in Gerris to advect *diffusive* tracers i.e. an advection–diffusion equation of the form

$$\frac{d}{dt}\int_V \rho\, dv + \oint_S (\rho\, \boldsymbol{u} + \epsilon\, \boldsymbol{\nabla}\rho)\cdot \boldsymbol{n}\, ds = 0 \tag{7}$$

where ϵ is a diffusion coefficient. The emphasis on *diffusive* is important. If ρ is diffusive and the diffusion coefficient is large enough compared to the numerical diffusion then the solution of (7) can be approximated correctly. Numerical diffusion depends both on the advection scheme used and on spatial resolution. For example, if a high-enough resolution was used, the first-order upwind scheme of Figure 2 could give results close to the van Leer-limited results of Figure 5.

Can we tolerate this numerical diffusion in the case of "real" interfaces? In a finite volume sense the jump in physical properties caused by a real interface should be entirely contained within a single finite volume. This means that strictly speaking, even the limited thickening of the interface caused by the van Leer limiter is not acceptable. We will see however, that this requirement of interface sharpness is often loosened in practice. A more problematic issue is that even for "good" schemes such as the van Leer-limited scheme above, the remaining numerical diffusion will cause the interface to thicken "forever". Depending on the physical problem studied, this thickening can be fast compared to the timescales of interest and can quickly degrade the quality of the solution, and this with little hope of ever returning back to a sharp interface (in contrast to shocks for example).

2.3 Interface tracking

Can we do better? Yes, if we make use of the fact that for interfacial flows, all the information is contained within the interface itself. For example for our one-dimensional problem above, the density field ρ is entirely determined by $x_I(t)$: the position of the interface. If we discretise ρ on equidistant finite-volumes $V_i = [x_{i-1/2}: x_{i+1/2}]$ then we have

$$\rho_i = \begin{cases} \rho_1 & \text{if } x_{i+1/2} \leqslant x_I \\ \rho_2 & \text{if } x_{i-1/2} \geqslant x_I \\ \rho_2 + (\rho_1 - \rho_2)\,\frac{x_I - x_{i-1/2}}{x_{i+1/2} - x_{i-1/2}} & \text{otherwise} \end{cases}$$

This can be rewritten

$$\rho_i = c_i\,\rho_1 + (1 - c_i)\,\rho_2 \tag{8}$$

with

$$c_i = \begin{cases} 1 & \text{if } x_{i+1/2} \leqslant x_I \\ 0 & \text{if } x_{i-1/2} \geqslant x_I \\ \frac{x_I - x_{i-1/2}}{x_{i+1/2} - x_{i-1/2}} & \text{otherwise} \end{cases} \tag{9}$$

the *volume fraction* of the first phase (of density ρ_1).

Although these particular relations are specific to our one-dimensional example, they can be generalised to any interface in any dimension such that

$$\rho_{i,j,\ldots,n} = c_{i,j,\ldots,n}\,\rho_1 + (1 - c_{i,j,\ldots,n})\,\rho_2$$

and

$$c_{i,j,\ldots,n} = f(\boldsymbol{x}_I)$$

where $\boldsymbol{x}_I$ is a description of the position of the interface in n-dimensional space.

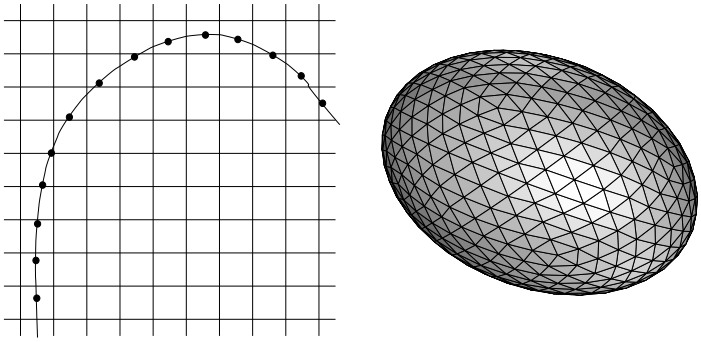
What is $\boldsymbol{x}_I$ in practice? How is it updated as it evolves in time? Several methods are possible. For our one dimensional example, x_I is the position of the interface which can be seen as the position of a material particule transported by the flow i.e. it verifies the evolution equation

$$\frac{dx_I}{dt} = U$$

If we now consider the case of an interface in two or three dimensions, we could describe the shape of the interface by distributing a finite number of material particules on the interface. The positions of these particules (and thus the shape of the interface) could then be updated using

$$\frac{d\,\boldsymbol{x}_I^k}{dt} = \boldsymbol{u}$$

This equation can be discretised and solved with high accuracy. This class of methods are often called *front-tracking* or *marker* methods. Figure 6 illustrate typical discretisations of interfaces in two or three dimensions.



two dimensions three dimensions

Figure 6. Discretisations of an interface for *marker* or *front-tracking* methods.

Once the updated position $\boldsymbol{x}_I$ of the interface is known, the volume fraction field (and thus the density field) can be obtained using relations similar to (9).

Marker-based methods are intuitively easy to understand but they also have significant drawbacks (which they share with other *Lagrangian* methods). In particular, nothing guarantees that the interface shape remains properly discretised when the positions of individual marker particule evolve in time. One can easily imagine for example that particules will accumulate at *stagnation points* in the flow. To ensure an appropriate description of the interface, marker particules need to be redistributed periodically along the interface, which complicates practical implementations. Changing interface topology (i.e. coalescence or breakup) can also be difficult to deal with, as explicit "surgery" needs to be performed on the marker representation (Figure 6).

One way to work around these difficulties is to define the interface position *implicitly* i.e. through an arbitrary function $\mathcal{F}$ such that

$$\mathcal{F}(\boldsymbol{x}_I) = 0 \tag{10}$$

For example a circular interface centered on the origin and of radius unity could be defined as

$$\boldsymbol{x}_I = \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{such that} \quad \mathcal{F}(\boldsymbol{x}_I) = x^2 + y^2 - 1 = 0$$

Rather than discretise the interface itself, one can then choose to discretise the implicit function $\mathcal{F}$, for example on the same grid used to discretise the other quantities (such as c, ρ and $\boldsymbol{u}$). One can then solve (10) locally to recover $\boldsymbol{x}_I$ and from there use relations similar to (9) to derive c. Of course $\mathcal{F}$ must also evolve in time to reflect the evolution of the interface. What is the evolution equation for $\mathcal{F}$? A simple choice is to write

$$\frac{d}{dt}\int_V \mathcal{F}\, dv + \oint_S \mathcal{F}\boldsymbol{u}\cdot\boldsymbol{n}\, ds = 0 \tag{11}$$

that is $\mathcal{F}$ is simply advected by the flow as an ordinary tracer. It is then trivial to show that $\boldsymbol{x}_I$ (as defined by (10)) also verifies

$$\frac{d\boldsymbol{x}_I}{dt} = \boldsymbol{u}$$

This is the principle of the *levelset* method. Compared to an explicit marker discretisation of the interface, it has several advantages. In particular $\mathcal{F}$ can be discretised using the standard techniques and data structures used for other fields and topology changes can be handled without complications.

At first sight, it is not obvious that we have gained anything by replacing the original advection equation for density (1) with the advection equation for the implicit function (11). The important difference between (1) and (11) though, is that we are free to choose the form of $\mathcal{F}$ as long as it verifies (10). In particular, we can choose $\mathcal{F}$ so that it is a smooth (i.e. differentiable) function of space, in contrast to the discontinuous function ρ (in the case of an interface). This means that the general schemes we derived above to solve (1) and (11) will be much more accurate for $\mathcal{F}$ than for ρ. Furthermore, the remaining numerical diffusion for $\mathcal{F}$ will not impact on the sharpness of the interface, since c will be derived from (sharp) relations analogous to (9).

Another important (but detrimental) difference between (1) and (11) is that although (11) is written in conservation form, conservation of $\mathcal{F}$ does not have any physical interpretation. Indeed, in the general case, the only physical interpretation of $\mathcal{F}$ is indirect, through the definition of the interface position (10). Although it is often interesting to define $\mathcal{F}$ as the signed distance to the interface (which gives it a physical meaning), solving (11) does not guarantee that it remains so as the interface evolves. These differences have important practical consequences. Although numerical diffusion of $\mathcal{F}$ does not degrade the sharpness of the interface, it does degrade the accuracy with which the position of the interface $\boldsymbol{x}_I$ is estimated (because the gradient of $\mathcal{F}$ near the interface becomes smaller). This in turn can cause large errors in mass conservation (because conservation of $\mathcal{F}$ is not linked to conservation of c). In order to avoid losing mass, it is thus necessary to periodically re-initialise $\mathcal{F}$ (for example to the exact signed distance function to the interface), which complicates the method.

2.4 Volume-Of-Fluid

We have seen that interface-tracking methods circumvent the problem of excessive numerical diffusion when solving (1) for interfacial fronts by using auxilliary fields (marker points or implicit functions) to track and update a description of the position of the interface. We know however that from a finite-volume point of view, the interface is uniquely defined by the discontinuity of the density field ρ (or equivalently of the volume fraction field c). Rather than involve auxilliary quantities is it possible to derive a scheme which would update c (and ρ) directly according to (1) while avoiding numerical diffusion entirely?

Let us start with the one-dimensional example studied previously. We can formulate a generic first-order in time advection scheme as

$$c_i^{n+1} = c_i^n + \frac{\Delta t}{\Delta}\left[F_{i-1/2} - F_{i+1/2}\right] \tag{12}$$

where $F_{i+1/2}$ is the *flux* through the right-hand-side boundary of a finite volume. We derived previously the following approximations for F

$$F_{i+1/2} = \begin{cases} \frac{c_i + c_{i+1}}{2}\, u_{i+1/2} & \text{centered scheme} \\ c_i\, u_{i+1/2} & \text{first order upwind scheme} \\ \frac{(3\, c_i - c_{i-1})}{2}\, u_{i+1/2} & \text{second order upwind} \\ \ldots & \end{cases}$$

If we now assume that c is discontinuous and bounded between zero and one, can we derive a better estimate for the fluxes? Based on Figure 7 the answer is yes.

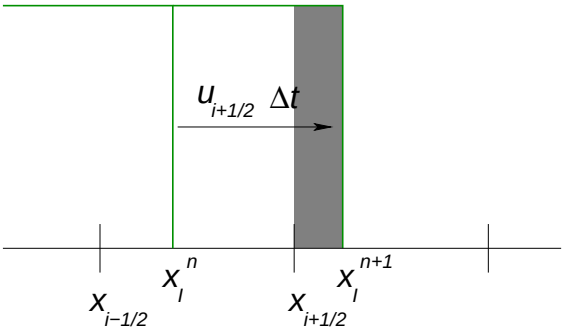


Figure 7. Advection of a sharp interface and corresponding fluxes. The gray area is the flux $F_{i+1/2}$.

We first note that in the one-dimensional case the position of the interface is given by

$$x_I = \begin{cases} x_{i-1/2} + c_i\,\Delta & \text{for } 0 < c_i < 1 \\ \text{undefined} & \text{otherwise} \end{cases} \tag{13}$$

That is, the information on the interface position can be *reconstructed* directly from the volume fraction field c. The fluxes can then be computed *geometrically* as (see Figure 7)

$$F_{i+1/2} = \begin{cases} 0 & \text{if } x_I + u_{i+1/2}\,\Delta t \leqslant x_{i+1/2} \\ u_{i+1/2} & \text{if } x_I \geqslant x_{i+1/2} \\ \frac{1}{\Delta t}\left(x_I + u_{i+1/2}\,\Delta t - x_{i+1/2}\right) & \text{otherwise} \end{cases}$$

Using the relation between x_I and c and simplifying then gives

$$F_{i+1/2} = \begin{cases} 0 & \text{if } \Delta t \leqslant (1 - c_i)\,\Delta / u_{i+1/2} \\ u_{i+1/2} + \frac{\Delta}{\Delta t}\,(c_i - 1) & \text{otherwise} \end{cases} \tag{14}$$

Note that given that u = constant for an incompressible one-dimensional flow, this scheme is exact in the one-dimensional case. For reference, Figure 8 illustrates the solution obtained using this scheme (combined with formula (8) to recover the density). As expected the interface thickness is at most one cell. The overall scheme is conservative and non-diffusive and does not involve fields other than the "primitive field" c. As illustrated here, it is also very simple (at least in one dimension).

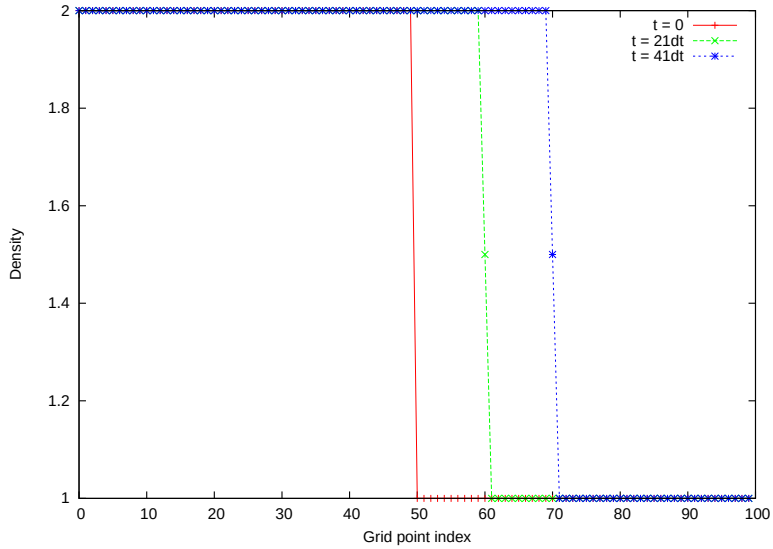


Figure 8. VOF advection scheme applied to a discontinuous density field. $\rho_1 = 2$, $\rho_2 = 1$.

Can we generalise this scheme to more than one dimension? The simplest way to do so is to use *split-direction* advection i.e. use the one-dimensional scheme above alternatively in each direction of propagation. This is the basic idea for the original VOF method. Using the one dimensional scheme means that the interface is always *reconstructed* perpendicularly to the direction of propagation using formula (13). An example of the corresponding interface reconstruction is given in Figure 9.

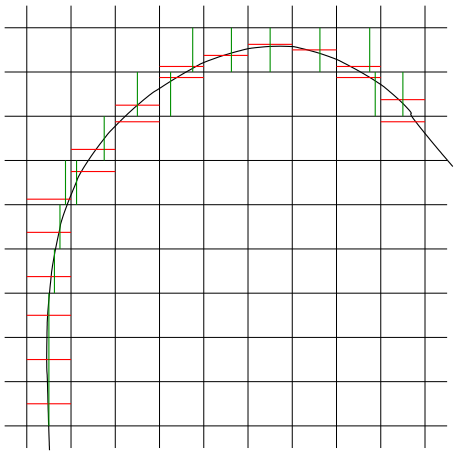


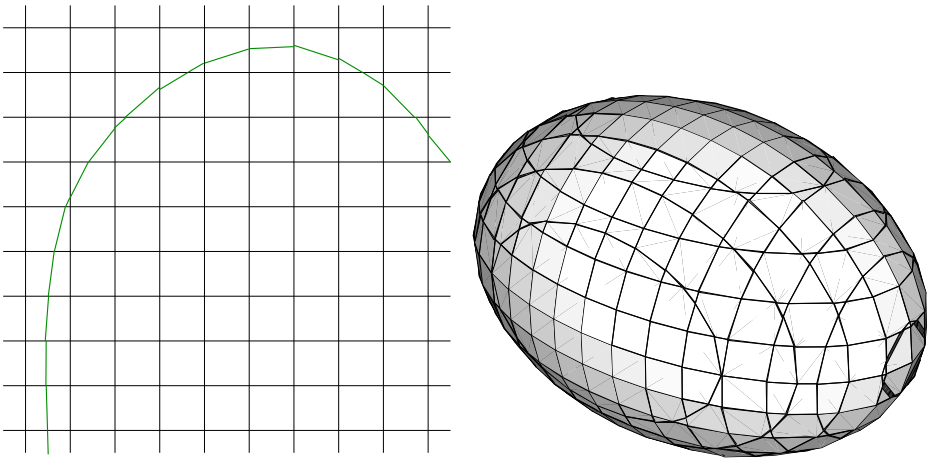
Figure 9. Simple Line Interface Calculation (SLIC) interface reconstruction. The interface is reconstrcuted alternatively as the green lines (for horizontal advection) and red lines (for vertical advection).

The SLIC scheme is non-diffusive and conservative but does not work very well. Small pieces of interfaces can easily split off the main body etc... Note also that since this scheme can not reconstruct straight interfaces at arbitrary angles to the mesh, it is only formally first-order accurate in space.

To improve the numerical scheme, the first step is to generalise formula (13) i.e. find a relation between the volume fraction field c and a description of the interface geometry $\boldsymbol{x}_I$ valid also in more than one dimension. A simple idea is to approximate the interface locally as a straight line inclined at an arbitrary angle (rather than the vertical or horizontal segments of the SLIC scheme). For example one could describe the interface locally as

$$\boldsymbol{m}_i \cdot \boldsymbol{x} = \alpha_i$$

where $\boldsymbol{m}_i$ is the local (discrete) vector normal to the interface and α_i is linked to the position of the planar interface. Taking the notation above, the interface is now represented by a discrete set $\boldsymbol{x}_I^i = (\boldsymbol{m}_i, \alpha_i)$ of interfacial segments (see Figure 10). Note that the resulting interface representation is piecewise-linear i.e. the segments are usually not exactly connected at their endpoints. Also, the method should be able to represent a straight interface at an arbitrary angle and is thus second-order in space (at least in principle).



two dimensions three dimensions

Figure 10. Piecewise Linear Interface Calculation (PLIC) interface reconstruction.

What is the relation between ($\boldsymbol{m}_i$, α_i) and c_i i.e. the equivalent of formula (13) for PLIC? An obvious relation is that the volume fraction c_i multiplied by the volume of the control volume V_i must equal the volume of the intersection of V_i with the half-plane defined by ($\boldsymbol{m}_i$, α_i) (Figure 10). This can be summarised as

$$c_i = \text{volume}(V_i \cap (\boldsymbol{m}_i, \alpha_i))/\text{volume}(V_i) = v(V_i, \boldsymbol{m}_i, \alpha_i) \tag{15}$$

The exact form of v depends on the details of the implementation (such as the shape of the control volume V_i) but only involves elementary geometric considerations. For Cartesian cells (i.e. squares in 2D and cubes in 3D) it can be relatively simply implemented in a generic function. Function v has one important property: for a given $\boldsymbol{m}_i$ and c_i, there is a unique α_i verifying (15). This can be summarised as

$$\alpha_i = v^{-1}(V_i, \boldsymbol{m}_i, c_i) \tag{16}$$

As before for Cartesian cells the inverse function v^{-1} can easily be implemented as a generic function. To close the problem, we thus need to find a way to derive $\boldsymbol{m}_i$ from c_i. In the continuum limit it is easy to show that

$$\boldsymbol{m} = -\frac{\boldsymbol{\nabla} c}{\|\boldsymbol{\nabla} c\|}$$

A first simple approach is to discretise this relation as

$$\boldsymbol{m}_i = -\frac{\boldsymbol{\nabla}_h c_i}{\|\boldsymbol{\nabla}_h c_i\|} \tag{17}$$

where $\boldsymbol{\nabla}_h$ is a discrete gradient operator, for example, in two dimensions one could choose

$$\boldsymbol{\nabla}_h c_{i,j} = \frac{1}{2\,\Delta}\begin{pmatrix} c_{i+1,j} - c_{i-1,j} \\ c_{i,j+1} - c_{i,j-1} \end{pmatrix} \tag{18}$$

We will see later that this is not a really good choice but that will do for now. Combining (17) and (16) gives the general *interface reconstruction* scheme which allows to derive the geometric interface description $(\boldsymbol{m}_i, \alpha_i)$ from the volume fraction field c_i.

Following the one-dimensional procedure above, the second step is to derive the geometrical advection flux from the reconstructed interface (i.e. the equivalent of formula (14) above). To illustrate the principle of the method, we will only consider uniform advection in one direction

(i.e. one-dimensional advection of a 2D interface). Figure 7 gives a geometrical representation of a simple flux calculation. Although an explicit formula is slightly more difficult to derive than for (14) it is clear than this can be done again using simple geometrical calculations (i.e. a combination of areas of triangles).

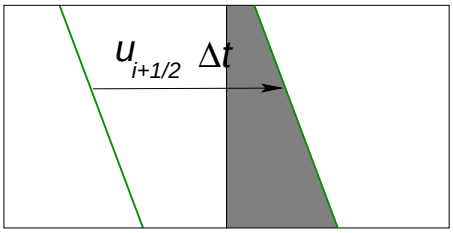


Figure 11. Simple one-dimensional advection combined with PLIC interface reconstruction. The gray area is the flux $F_{i+1/2,j}$.

How can this simple one-dimensional scheme be extended to two dimensions? A simple solution is to extend scheme (12) as

$$c_{i,j}^{n+1} = c_{i,j}^{n} + \frac{\Delta t}{\Delta}\left[F_{i-1/2,j} - F_{i+1/2,j} + F_{i,j-1/2} - F_{i,j+1/2}\right] \tag{19}$$

where each of the fluxes $F_{i\pm1/2,j\pm1/2}$ is computed using the geometrical flux estimate illustrated in Figure (11) (in the respective directions). This scheme is conservative by construction, however it will not guarantee that

$$0 \leqslant c_{i,j}^{n+1} \leqslant 1 \tag{20}$$

That is because *transverse* or *diagonal* fluxes are not computed correctly. In practice what is often done is to arbitrarily enforce (20) by chopping off any excess volume fraction. The resulting scheme is thus not strictly mass conserving anymore. In practice however the mass conservation properties of the resulting scheme are still very acceptable.

Nonetheless, given that one of the main theoretical advantage of the VOF method is mass conservation, it is important to derive schemes which are strictly mass conserving. This is possible but not as simple as the approach described here and is beyong the scope of this introduction. The recent book by Tryggvason, Scardovelli and Zaleski is a good reference for such advanced VOF schemes [3].

2.4.1 Deriving geometric quantities from the volume fraction field

We have seen above that an important step in the VOF scheme is *interface reconstruction* i.e. deriving geometrical information (position, normal direction etc...) about the interface directly from the volume fraction field c. We will see later that interface *curvature* is also an important quantity which needs to be derived when computing surface tension. What are the relations between these geometrical quantities and the volume fraction field c? To answer this question, it is useful to first consider an interface defined implicitly by a smooth function $\mathcal{F}$ such that

$$\mathcal{F}(\boldsymbol{x}_I) = 0$$

(see equation (10)) above. In that case the interface unit normal is given by

$$\boldsymbol{n} = -\frac{\boldsymbol{\nabla}\mathcal{F}}{|\boldsymbol{\nabla}\mathcal{F}|}, \tag{21}$$

and the interface curvature by

$$\kappa = -\boldsymbol{\nabla}\cdot\boldsymbol{n} = \boldsymbol{\nabla}\cdot\left(\frac{\boldsymbol{\nabla}\mathcal{F}}{|\boldsymbol{\nabla}\mathcal{F}|}\right) \tag{22}$$

For smooth functions the relations (21) and (22) can be discretised using standard finite difference operators such as (18) above. Using standard numerical analysis it is also relatively simple to derive the formal order of accuracy of these discrete operators. If care is taken, reasonably accurate schemes can thus be derived to compute $\boldsymbol{n}$ and κ from $\mathcal{F}$. This is the essence of the *levelset* method, and this apparent simplicity largely justifies the choice of this method for interface representation and advection (despite the difficulties related to mass conservation).

What happens to relations (21) and (22) if $\mathcal{F}$ is not a smooth function i.e. if we replace $\mathcal{F}$ by c? These relations still hold, however they need to be interpreted in the context of *distributions*. For example (21) becomes

$$\boldsymbol{\nabla} c = -\,\delta_S\,\boldsymbol{n}, \tag{23}$$

where δ_S is the equivalent of the Dirac delta function for a surface (rather than a point). The convergence with spatial resolution of the standard finite difference operators (such as (18)) rely on the differentiability of $\mathcal{F}$, so that these convergence results cannot be extended to the discontinuous field c. This is why in practice using (18) to derive $\boldsymbol{n}$ from c does not give good results.

A first simple idea is to replace c with a "smoother" approximation $\tilde{c}$ and apply the discrete versions of (21) and (22) on this field instead of c. The smooth field $\tilde{c}$ can be constructed in various ways, for example by filtering c spatially. This was historically how surface tension was first implemented in VOF methods (see Brackbill 1992). For simple filtering techniques this does not work very well either however (for example the curvature estimate often fails to converge with grid resolution). Rather than use these somewhat naive filtering techniques, one could also reconstruct a smooth levelset function $\mathcal{F}$ from c (i.e. $\tilde{c} = \mathcal{F}$). This is the principle of the coupled VOF-Levelset method (CLSVOF) method which works well in practice.

Introducing an intermediate field is not very satisfactory however, for reasons related to an increased complexity and also a loss of accuracy and consistency when switching between multiple interface representations. We will see now how the *Height Function* (HF) method allows to derive accurate estimates of geometrical quantities directly from the volume fraction c. In two dimensions one could choose to define an interface explicitly as

$$y = h(x)$$

Of course only univalued interfaces can be described with such a (graph) function. We will see later how to circumvent this problem. With this definition, the normal $\boldsymbol{n}$ and curvature κ can be computed using their standard definitions

$$\boldsymbol{n} = \frac{1}{\sqrt{1+h'^2}} \begin{pmatrix} h' \\ -1 \end{pmatrix} \tag{24}$$

$$\kappa = \frac{h''}{\left(1+h'^2\right)^{3/2}} \tag{25}$$

Using a discrete description of h on a regular grid one can then derive simple finite difference schemes to compute $\boldsymbol{n}$ and κ. For example using

$$h'_i = \frac{h_{i+1} - h_{i-1}}{2\,\Delta} + O(\Delta^2) \tag{26}$$

$$h''_i = \frac{h_{i+1} - 2\,h_i + h_{i-1}}{\Delta^2} + O(\Delta^2) \tag{27}$$

where the order of the approximation is deduced from classical analysis applied to smooth functions. What is the relation between h and c? If a cell is cut by the interface we have

$$c_i = \frac{1}{\Delta^2} \int_{x_{i-1/2}}^{x_{i+1/2}} \left(h(x) - y_{j-1/2}\right) dx$$

which can also be written

$$c_i = \frac{h_i - y_{j-1/2}}{\Delta}$$

with

$$h_i = \frac{1}{\Delta} \int_{x_{i-1/2}}^{x_{i+1/2}} h(x)\, dx$$

The reciprocal relation is simply (Figure 12)

$$h_i = \Delta \sum_j c_{i,j}$$

This relation is exact and combined with (24), (25), (26) and (27) gives a simple scheme which allows to derive $\boldsymbol{n}$ and κ directly from c with formal second-order spatial accuracy (it can also be extended to higher order using wider stencils in (26) and (27)).

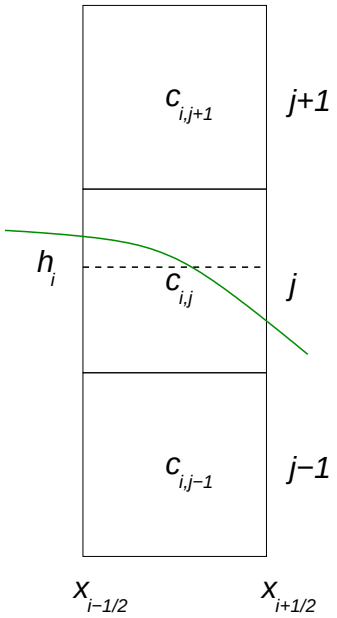


Figure 12. Relation between the height-function h and the volume fraction c.

We have seen however that the height function description is consistent only for univalued interfaces. A simple way to lift this constraint is to define the height-function locally. Depending on the overall (rough) orientation of the interface, one can choose to define the height function either along the x- or y-coordinate (Figure 13). If the interface is resolved with a fine enough mesh, a consistent discretisation of the height function can be constructed. In the few limit cases where the interface is not well-resolved, other consistent height-function approximations can still be constructed (see Popinet, 2009 for examples). Although we presented only the two-dimensional case, the height-function method is trivially extended to three dimensions.

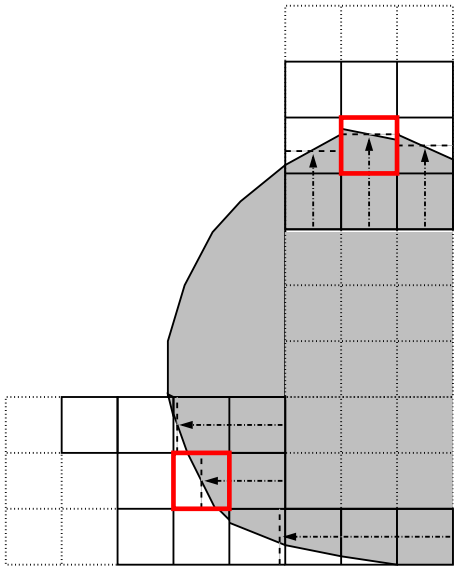


Figure 13. Choice of the direction of the local height-function reconstruction.

3 Surface tension

The surface tension term is usually added as a force per unit volume in the momentum equation (2). In the case where the surface tension coefficient is constant along the interface, this can be written

$$\boldsymbol{f}_\sigma = \kappa\, \boldsymbol{n}\, \delta_S$$

i.e. the surface tension force is proportional to the surface tension coefficient σ, is normal to the interface and is non-zero only on the interface. For a finite-volume method, we are more interested in

$$\int_V \boldsymbol{f}_\sigma\, dv = \sigma \int_V \kappa\, \boldsymbol{n}\, \delta_S\, dv$$

For the two-dimensional case, using the first Serret-Frenet relation

$$\oint_S \kappa\, \boldsymbol{n}\, ds = \oint_S d\boldsymbol{t}$$

where $\boldsymbol{t}$ is the unit tangent vector to the interface, we get (see Figure 14)

$$\sigma \oint_A^B \kappa\, \boldsymbol{n}\, ds = \sigma \oint_A^B d\boldsymbol{t} = \sigma\, (\boldsymbol{t}_B - \boldsymbol{t}_A) \tag{28}$$

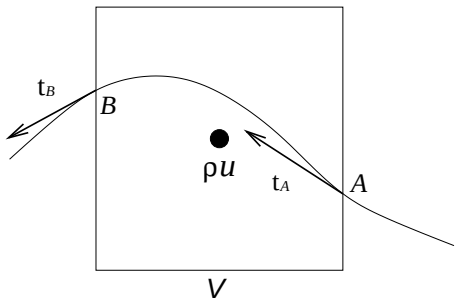


Figure 14. Surface tension term integrated over a control volume.

With this formulation we recover the intuitive definition of the surface tension force. Surface tension itself is of course a force per unit area *tangential* to the surface, but its volume-integrated resultant is a force per unit volume *normal* to the interface. Aside from this intuitive interpretation, this formulation has the advantage of being momentum-conserving. Since by construction, the surface tension forces $\sigma\ \boldsymbol{t}$ at the boundary between two control volumes are of equal magnitude but opposite sign, the (discrete) integral of the tension surface forces along the interface vanish, so that no net momentum is imparted to the interface.

An important difficulty with this method in practice however, is that it requires accurate and consistent interface tangent estimates at the boundaries between control volumes. This is a difficulty when using a VOF method because we have seen previously that the VOF interface representation is only piecewise-linear, so that interface segments (and normals) are not continuous at the boundaries between control volumes. On the other hand, the method can be applied relatively easily when using a marker or levelset representation of the interface.

What are the other options for VOF methods? One can choose to go back to the volume-intetgrated definition of the surface tension term and use relation (23) to write

$$\int_V \boldsymbol{f}_\sigma\, dv = \sigma \int_V \kappa\, \boldsymbol{n}\, \delta_S\, dv = -\sigma \int_V \kappa\, \boldsymbol{\nabla} c\, dv$$

which suggests the following discrete approximation

$$\sigma \int_{V_i} \boldsymbol{f}_\sigma \, dv \approx -\sigma \, \Delta^2 \, \kappa_i \boldsymbol{\nabla}_i \, c \tag{29}$$

where κ_i is a volume-averaged curvature which can be computed from c_i using the the height-function method above. This formulation is the basis for the Continuum-Surface-Force (CSF) method of Brackbill (1992). Note that one of the drawbacks of this method is that it is not strictly momentum-conserving anymore. We can also guess that it will be less accurate than (28) since we approximate the gradient of a discontinuous function $\boldsymbol{\nabla} c$ with a continuous discrete version $\boldsymbol{\nabla}_i \, c$.

3.1 Laplace's relation for a circular drop: spurious currents

A useful case to consider when assessing the relative merits of different schemes for the representation of surface tension is the simple case of a droplet in equilibrium. In that case the volume-integrated equations reduce to

$$\oint_S -p \, \boldsymbol{n} \, ds = \int_V \boldsymbol{f}_\sigma \, dv = \sigma \oint_S \kappa \, \boldsymbol{n} \, ds$$

which has a solution only if κ is a constant and

$$[p]_S = -\sigma \, \kappa$$

where $[p]_S$ is the jump of the pressure across the interface. The equilibrium solution is thus a circular interface where the surface tension force is exactly balanced by the pressure jump across the interface. This is known as "Laplace's law" for circular droplets.

What happens when trying to solve this problem numerically? One could for example choose to discretise the pressure term for the horizontal velocity component as

$$\oint_S -p \, \boldsymbol{n} \, ds \approx -\Delta^2 \boldsymbol{\nabla}_i \, p$$

while discretising the surface tension term using the exact scheme (28), which would give the discrete equilibrium condition

$$-\Delta^2 \, \boldsymbol{\nabla}_i \, p = \sigma \, (\boldsymbol{t}_B - \boldsymbol{t}_A)$$

This condition is usually not trivially satisfied, actually it is easy to show that with this particular choice of discretisation of the pressure gradient, it cannot be satisfied for non-trivial interface configurations (such as a circular interface). Where does this inconsistency come from? In the general case, the terms on either side of the discrete equilibrium condition are discretised using entirely different schemes. The discretisation errors associated with either terms will thus behave differently (even if both schemes are formally second-order accurate for example). If care is not taken, there is thus no reason for the discretisation errors on either side of the discrete equilibrium relation to cancel out exactly.

Why does this matter in practice? Figure 15 illustrate a typical solution obtained using a numerical scheme where the discrete balance is not verified. The velocity field observed is purely numerical and is often known as "spurious currents". These current are continuously fed by the numerical imbalance and are only bounded by viscous dissipation. If viscous dissipation is low and surface tension is high (such as in water–air applications or liquid metals) they can be strong enough to even "atomise" the interface. Spurious currents are thus an important problem for most methods using a naive implementation of surface tension terms.

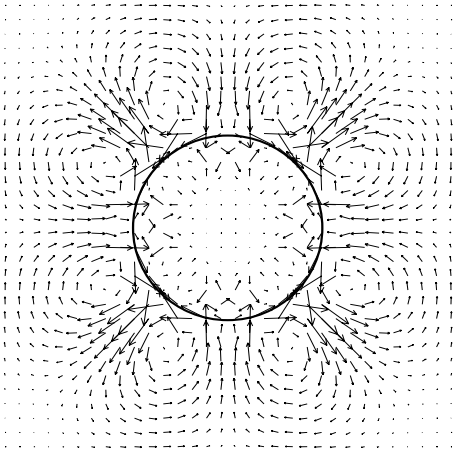


Figure 15. Spurious currents around a circular droplet using the original CSF method for surface tension terms.

To solve this problem one must look in detail at the properties of the numerical schemes in the case of the circular interface. This is where the CSF method reveals a very useful property. For the CSF scheme the discrete equilibrium condition for the circular interface can be written

$$\tilde{\boldsymbol{\nabla}}_i p = \sigma \kappa_i \boldsymbol{\nabla}_i c \tag{30}$$

where a different notation has been deliberately chosen for the discrete gradient operator $\tilde{\boldsymbol{\nabla}}_i$ applied to the pressure and that applied to the volume fraction $\boldsymbol{\nabla}_i$. Indeed, for naive implementations of the CSF formulation (such as the original CSF method), the two operators are usually different (because in particular of the constraints imposed by the projection method and the choice of spatial discretisation). If care is taken however, the terms on either side of the discrete equilibrium condition can be discretised using the same discrete operator $\boldsymbol{\nabla}_i$: this combination is often known as the *balanced CSF* method. *If the curvature κ_i is constant everywhere*, then (30) admits a trivial discrete solution which is

$$p_i = \sigma \kappa c_i$$

Note that we have only demonstrated the existence of an exact discrete equilibrium solution for a particular combination of numerical schemes. Whether this solution is reached in practice also depends on other aspects of the code i.e. time integration, projection methods etc... The other important requirement is that the discrete curvature estimate κ_i must be constant. The combination of schemes used within Gerris: balanced CSF and Height–Function curvature estimation, has been demonstrated to always converge toward such a discrete equilibrium solution [2].

Bibliography

[1] J.U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of computational physics*, 100(2):335–354, 1992.

[2] S. Popinet. An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics*, 228(16):5838–5866, 2009.

[3] G. Tryggvason, R. Scardovelli, and S. Zaleski. *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge University Press, 2011.