XVIII International Conference on Water Resources
CMWR 2010
J. Carrera (Ed)
CIMNE, Barcelona 2010

METHOD OF LEAST SQUARES FOR COMPUTING NORMALS AND CURVATURES FROM 2D VOLUME FRACTIONS

Poorya A. Ferdowsi[*] and Markus Bussmann[†]

Department of Mechanical and Industrial Engineering, University of Toronto

[*] e-mail: p.ferdowsi@utoronto.ca, web page: http://individual.utoronto.ca/pferdowsi/
[†] e-mail: bussmann@mie.utoronto.ca, web page: http://www.mie.utoronto.ca/faculty/bussmann

Keywords: volume of fluid, interface reconstruction, normals, curvatures, height functions

Summary: A new method for calculating normals and curvatures from 2D volume fractions is presented. The method fits a curve (line or arc) to discrete values of fluid heights, by minimizing a discretized least squares error. By fixing the fitting curve, the normal to and curvature of an interface segment can be determined.

1 INTRODUCTION

Applications of the volume-of-fluid (VoF) method for numerical simulation of interfacial flows are extensive. In the VoF method, the discrete volume fractions of one phase, f, defined as the ratio of the phase fluid volume to the cell volume, are used to reconstruct the interface[1,2]. Those cells with $0 < f < 1$ are considered to contain a portion of the interface, that is commonly reconstructed by piecewise linear segments (PLIC). The local curvature of the interface must also be calculated from the volume fractions, that in turn is used to calculate the surface tension force that can be incorporated into the Navier-Stokes equation via the CSF method[3,4]. In this paper, a new method for calculating normals and curvatures from 2D volume fractions on a Cartesian mesh is presented.

In order to reconstruct a piecewise linear segment of an interface, a normal and an intercept are required. Given a normal, the intercept can be calculated analytically or iteratively and is unique[1]; it is the approach to calculating the normal to an interface that differentiates various VoF reconstruction algorithms. The height function method (HF) is a popular gradient-based[3,5,6] technique for computation of normals and curvatures. The HF method first decides the primary orientation of the interface, horizontal or vertical, by a simple centered difference. Then a stencil is oriented more perpendicular to the interface, as shown in Figure 1(a), and the volume of fluid in each of the three columns of the stencil is calculated and converted to a height; in 2D this corresponds to:

$$H_\beta = \frac{\sum_{\alpha=-3}^{3} f_{\beta,j+\alpha}\delta y_{j+\alpha}}{\delta x_\beta}, \quad \text{for } \beta = i-1,\ i,\ i+1 \tag{1}$$

The normal components and curvature of the interface can then be calculated as:

$$n_x = x_{i+1} - x_{i-1} \tag{2}$$

$$n_y = H_{i+1} - H_{i-1}$$

$$\kappa = -\frac{\ddot{H}}{\left[1 + \left({n_y}/{n_x} \right)^2 \right]^{\frac{3}{2}}}$$

where $\ddot{H}$ is the discretized second-order derivative calculated by central differencing. The HF method yields first order accurate normals and second order accurate curvatures. To obtain second order accurate normals, Ferdowsi et al.[5] presented a sub-grid method for calculating normals in cells where the interface intersects two adjacent sides of a cell.

An alternative to gradient-based methods are fitting techniques, usually used to minimize an error function that represents the difference between actual volume fractions (areas in 2D) and reconstructed volume fractions associated with a fitted interface. LVIRA[7], for example, minimizes the difference between actual volume fractions $f_{i,j}$ and those obtained by extending a piecewise linear segment into the surrounding 3×3 stencil, $\tilde{f}_{i,j}$, by iteratively varying the normal:

$$\tilde{E}_{i,j}^2 = \sum_{\alpha=-1}^{1} \sum_{\beta=-1}^{1} \left(f_{i+\alpha,j+\beta} - \tilde{f}_{i+\alpha,j+\beta} \right)^2 \tag{3}$$

Brent's algorithm[8] can be used to minimize equation (3), for which the derivatives of the target function are unavailable. As a result, however, the algorithm converges slowly and is computationally expensive. The ELVIRA method[7] is a related technique, but the normal is chosen from only six candidates, based on forward, backward and central differencing in both the horizontal and vertical directions. This dramatically reduces the costs, but the method is still more expensive than the HF technique, yet yields the same first order accuracy[5]. As for computing curvatures via fitting methods, Chorin[9] estimated interface curvature by approximating an interface by an osculating circle, that was determined by reproducing exact values of f in the neighborhood of an interface cell. Poo et al.[10] fit a second-order polynomial to volume fractions in a 3×3 stencil, then used an equation similar to equation (2) to calculate the curvature. However, none of these methods can exactly predict the known curvature of a circle.

Here we introduce a new mixed least squares HF (LHF) method, in which an error function similar to equation (3) is minimized subject to a constraint. Although any fitting curve could be used, we restrict ourselves here to fitting a line (the discretized tangent to a curve) and a circle (a discretized osculating circle). Fitting a circle, LHF captures the exact curvature of a circle, and produces second-order accurate normals and first-order accurate curvatures for general interfaces.

2 THEORY

Analogous to equation (3), the least squares height function (LHF) method minimizes the difference between two actual heights H and those obtained from a fitted curve, $\tilde{H}$:

$$\tilde{E}_{i,j}^2 = \left(H_{i-1} - \tilde{H}_{i-1}\right)^2 + \left(H_{i+1} - \tilde{H}_{i+1}\right)^2 \tag{4}$$

subject to the constraint that the fitted curve preserve the central height of the stencil H_i:

$$H_i = \tilde{H}_i \tag{5}$$

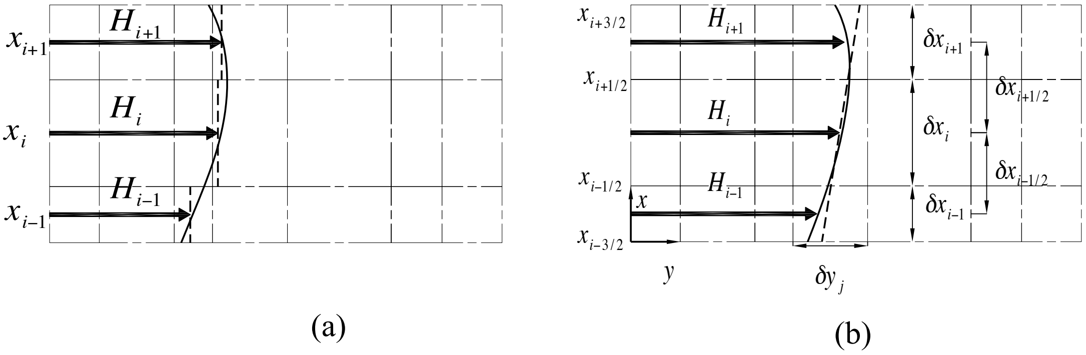

Figure 1: (a) Definition of heights in a horizontal stencil, (b) parameters used in the LHF method. The actual interface in indicated by the solid line while the dashed line is a linear fit.

Any fitting curve (e.g. line, parabola, circle) with no more than three degrees of freedom could be used for such an optimization problem. Here we restrict ourselves to presenting the method for a line and a circular arc. Note, however, that the second derivative of a line is zero, and so one cannot calculate interface curvature with such a fit. Nevertheless, we present the linear fit because one obtains an explicit expression for the normal that is equavalnt to the HF method on a uniform mesh, but different than the usual expression in a non-uniform grid. The grid parameters used in these analyses are shown in figure 1(b).

2.1 Linear Fit

To begin, we compute heights in the HF stencil subject to a monotonicity condition[11]. Using the linear fit $y = mx + c$ to minimize equation (4) subject to the constraint (5) yields the following expressions on a general Cartesian grid:

$$m = \frac{\delta x_{i+1}^2 \delta x_{i+1/2}^2 \Delta H + \delta x_{i-1}^2 \delta x_{i-1/2}^2 \nabla H}{\delta x_{i+1}^2 \delta x_{i+1/2}^2 + \delta x_{i-1}^2 \delta x_{i-1/2}^2} \tag{6}$$

$$c = H_i - m x_i$$

where Δ and ∇ represent forward and backward differencing, respectively, defined as:

$$\Delta H = \frac{H_{i+1} - H_i}{\delta x_{i+1/2}} \tag{7}$$

$$\nabla H = \frac{H_i - H_{i-1}}{\delta x_{i-1/2}}$$

On a uniform grid, equation (6) reduces to that of the original HF, but on a non-uniform grid the expressions are different from the HF technique.
To assess the linear fit, we use equation (6) to calculate $\tilde{H}$ in equation (4) and obtain:

$$\tilde{E}_{i,j} = \frac{\delta x_{i+1} \delta x_{i-1} \delta x_{i+1/2} \delta x_{i-1/2}}{\sqrt{\delta x_{i+1}^2 \delta x_{i+1/2}^2 + \delta x_{i-1}^2 \delta x_{i-1/2}^2}} \left| \Delta H - \nabla H \right| \tag{8}$$

On a uniform this difference reduces to:

$$\tilde{E}_{i,j} = \frac{\delta x^3}{\sqrt{2}} \left| h_{i,j}^{(4)} \right| \tag{9}$$

where $h_{i,j}^{(4)}$ is the fourth-order derivative of the actual interface function. If any slope other than that computed by equation (6) is used, $\tilde{E}_{i,j} = O\left(\delta x^2\right)$.

2.2 Circular Arc Fit

The arc that we use is a part of a circle defined by:

$$\left(x - x_c\right)^2 + \left(y - y_c\right)^2 = \rho^2 \tag{10}$$

where $\left(x_c, y_c\right)$ is the coordinate of the circle centre and ρ is the circle radius. With this fit the minimization of equation (4) subject to constraint (5) reduces to solving the following system of equations for any interface function:

$$\begin{cases} H_{i-1} = \tilde{H}_{i-1} \\ H_i = \tilde{H}_i \\ H_{i+1} = \tilde{H}_{i+1} \end{cases} \tag{11}$$

This is a nonlinear system of equations, and so a Newton-Raphson[8] (NR) technique was implemented, that yields the following iterative linear system of equations:

$$\nabla \tilde{H}^{(k-1)} \delta = \tilde{H}^{(k-1)} - H \tag{12}$$

where:

$$\delta = \begin{Bmatrix} \delta x_c \\ \delta y_c \\ \delta \rho \end{Bmatrix}, \; \tilde{H}^{(k-1)} = \begin{Bmatrix} \tilde{H}_{i-1} \\ \tilde{H}_i \\ \tilde{H}_{i+1} \end{Bmatrix}^{(k-1)}, \; H = \begin{Bmatrix} H_{i-1} \\ H_i \\ H_{i+1} \end{Bmatrix} \tag{13}$$

and $\nabla \tilde{H}^{(k-1)}$ is the Jacobian computed at iteration $k-1$. Convergence requires a reasonable initial guess, that we introduce at the end of this section. We derived exact expressions for $\tilde{H}^{(k-1)}$, the heights associated with the fitted circle, based on the following assumptions: that the interface normal is in the first quadrant, the second derivative of the interface is positive, the interface intersects only opposite sides of the stencil, and that the interface is more vertical than horizontal. Any other configurations can be transformed into this configuration, or treated similarly.

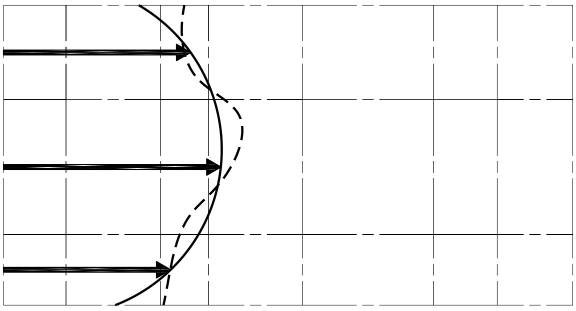

Figure 2: The initial guess is the circle that passes through the end points of the three arrows that indicates the heights. The dashed line is the actual interface.

In what follows we assume that the bottom of the stencil, at $y = y_0$, has been shifted to the origin (see figure 1(b)). The fluid heights confined to the arc and the sides of central column are then calculated as:

$$\tilde{H}_i = \frac{A(x_{i+1/2}) - A(x_{i-1/2})}{\delta x_i} \tag{14}$$

where:

$$A(x) = \frac{x - x_c}{2}\sqrt{\rho^2 - (x - x_c)^2} + \frac{\rho^2}{2}\tan^{-1}\frac{x - x_c}{\sqrt{\rho^2 - (x - x_c)^2}} + x y_c \tag{15}$$

with analogous expressions for the left and right columns. Although it may be possible to derive analytical expressions for the Jacobian, we chose to calculate it numerically via the forward differencing technique with an increment of 10^{-5}. When the arc has been determined, the reciprocal of its radius gives the interface curvature, and the normal at the mid-point of the arc, is the interface normal.

Returning to the crucial issue of obtaining a good initial guess in order to assure convergence of the NR iterations, we determined that passing a circle through the three end-points of the heights illustrated in figure 2 yielded good results as long as the interface was well-enough resolved (roughly, $\kappa\Delta x < 1/8$). We have developed an approach for less well-resolved interfaces, that we present elsewhere[12].

3 RESULTS

Here we assess the accuracy of the LHF method by computing normals and curvatures for a circle and for an ellipse:

$$x^2 + y^2 = 1 \tag{16}$$

$$x^2 + 4y^2 = 1$$

The norm introduced by Ferdowsi et al.[5] was used to evaluate the accuracy of the method; calculated unit normals, $\hat{n}_{num}$, are compared to the average normal $\hat{n}_{avg}$ of an exact interface:

$$L_\infty^{(n)} = \bigcup_{i=1}^{N} \max\left(\cos^{-1}\left(\hat{n}_{num} \cdot \hat{n}_{avg}\right)_i\right) \tag{17}$$

where N is the number of interface cells. For curvature, similarly one can write:

$$L_\infty^{(\kappa)} = \bigcup_{i=1}^{N} \max\left(\left|\kappa_{num} - \kappa_{avg}\right|_i\right) \tag{18}$$

Analogous to $\hat{n}_{avg}$, κ_{avg} is the average curvature of the interface segment within a cell. For the ellipse, the average values were integrated numerically; for the circle, the curvature is a constant and $\hat{n}_{avg}$ is the interface normal at the mid-point of the interface segment.

Tests were run on a unit square. The exact volume fractions were computed analytically for both the circle and ellipse. The convergence tolerance of the NR method was set to 10^{-15}. The system of equations (11) was normalized by the initial radius of the circle; otherwise, due to machine round-off error, the method did not converge to machine accuracy. As mentioned previously, on a uniform grid linear fit yields the same results as the HF method and it is not necessary to analyze it here. Hereafter, we only compare HF with LHF.

It suffices to say that the LHF method captures the curvature and normals for a circle to machine precision if an exact Jacobian is known. But as we calculated the Jacobian numerically, the curvature and normals were calculated to within 10^{-8}, where the error varies with the differencing step. The method captured the circle curvature after no more than five iterations, which was deemed computationally reasonable.

ρ	LHF		HF		IG	
Δ	L_∞	Rate	L_∞	Rate	L_∞	Rate
8	$\sim 10^{-8}$	–	7.85×10^{-2}	–	3.16×10^{-3}	–
16	$\sim 10^{-8}$	–	2.86×10^{-2}	1.46	6.54×10^{-4}	2.27
32	$\sim 10^{-8}$	–	1.86×10^{-2}	0.62	1.74×10^{-4}	1.91
64	$\sim 10^{-8}$	–	9.27×10^{-3}	1.00	4.16×10^{-5}	2.06

Table 1: Error norms for normals to a circle of unit radius on a uniform square grid.

ρ	LHF		HF		IG	
Δ	L_∞	Rate	L_∞	Rate	L_∞	Rate
8	$\sim 10^{-8}$	–	6.18×10^{-3}	–	5.99×10^{-3}	–
16	$\sim 10^{-8}$	–	1.56×10^{-3}	1.98	1.65×10^{-3}	1.86
32	$\sim 10^{-8}$	–	3.64×10^{-4}	2.10	3.60×10^{-4}	2.2
64	$\sim 10^{-8}$	–	9.27×10^{-5}	1.97	9.41×10^{-5}	1.94

Table 2: Curvature error norms, for a circle of unit radius on a uniform square grid.

Table 1 and 2 compare the accuracy of the LHF method with the standard HF. The last column in both tables, IG, present the results corresponding to the initial guess without iteration. Interestingly, the accuracy of HF and IG are similar for curvatures, and not only does IG yield more accurate normals, but with a higher order of accuracy. We see that HF is first order accurate for normals, while IG is second order. When computing curvature for a circle, both methods are second order accurate.

ρ_y	LHF		HF		IG	
Δ	L_∞	Rate	L_∞	Rate	L_∞	Rate
16	2.88×10^{-3}	–	5.25×10^{-2}	–	2.80×10^{-3}	–
32	7.48×10^{-4}	1.94	2.54×10^{-2}	1.05	7.44×10^{-4}	1.91
64	1.85×10^{-4}	2.02	1.48×10^{-2}	0.78	1.85×10^{-4}	2.01
128	4.62×10^{-5}	2.00	7.29×10^{-3}	1.02	4.62×10^{-5}	2.00

Table 3: Error norms for normals to an ellipse with $\rho_x/\rho_y=2$ on a uniform square grid.

ρ_y	LHF		HF		IG	
Δ	L_∞	Rate	L_∞	Rate	L_∞	Rate
16	1.70×10^{-2}	–	1.79×10^{-2}	–	1.82×10^{-2}	–
32	8.46×10^{-3}	1.02	9.22×10^{-3}	0.96	9.30×10^{-3}	0.97
64	4.93×10^{-3}	0.78	4.94×10^{-3}	0.90	4.93×10^{-3}	0.92
128	2.55×10^{-3}	0.95	2.56×10^{-3}	0.95	2.55×10^{-3}	0.95

Table 4: Curvature error norms for an ellipse with $\rho_x/\rho_y=2$ on a uniform square grid.

The second test case is for an ellipse with $\rho_x/\rho_y=2$. Errors are presented in Table 3 and 4. Both the LHF method and the IG yield normals that are second order accurate compared to the first order accurate HF values, and with increasing refinement the LHF normals are much more accurate. For curvatures, the improvement is negligible as all three methods are first-order accurate, as occurs for any interface of non-constant curvature.

4 CONCLUSIONS

A new method has been developed for computing interface normals and curvature from 2D volume fractions, that yields exact values for a circle, and yields second order accurate normals and first order accurate curvatures for general interfaces. With mesh refinement, the Newton-Raphson iteration used to solve the least squares problem did not improve the accuracy of the method, perhaps due to the numerical error associated with calculating the Jacobian via a forward differencing method (we would expect that an analytical Jaconbian would improve the accuracy of the method). Wherever the radius of the curvature is high, improvement of the initial guess is inevitable; otherwise, NR does not converge. Such a case needs further investigations.

REFERENCES

[1] W.J. Rider, D.B. Kothe, “Reconstructing volume tracking”, *J. Comp. Phys.*, **141**, 112-152 (1998).

[2] R. Scardovelli, S. Zaleski, “Direct numerical simulation of free-surface and interfacial flow”, *Annu. Rev. Fluid Mech.*, **31**, 567-603 (1999).

[3] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, M.W. Williams, “A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework”, *J. Comp. Phys.*, **213**, 141-173 (2006).

[4] J.U. Brackbill, D.B. Kothe, C. Zemach, “A Continuum Method for Modeling Surface Tension”, *J. Comp. Phys.*, **100**, 335-354 (1992).

[5] P.A. Ferdowsi and M. Bussmann, “Second-order accurate normals from height functions”, *J. Comp. Phys.*, **227**, 9293-9302 (2008).

[6] S.J. Cummins, M.M. Francois, D.B. Kothe, “Estimating curvature from volume fractions”, *Comput. Struct.*, **83**, 425–434 (2005).

[7] J.E. Pilliod Jr. And E.G. Puckett, “Second-order accurate volume-of-fluid algorithms for tracking material interfaces”, *J. Comp. Phys.*, **227**, 9293-9302 (2008).

[8] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical recipes in C : The art of scientific computing*, Cambridge University Press, New York, 1992.

[9] A.J. Chorin, “Curvature and solidification”, *J. Comp. Phys.*, **58**, 472-490 (1985).

[10] J.Y. Poo and N. Ashgriz, “A computational method for determining curvatures”, *J. Comp. Phys.*, **84**, 483-491 (1989).

[11]M. Malik, E.S.-C. Fan, M. Bussmann, “Adaptive VOF with curvature-based refinement”, *Int. J. Numer. Meth. Fluids*, **55,** 693-712 (2007).

[12] P.A. Ferdowsi, M. Bussmann, “Computing curvature from volume fractions in under-resolved regions”, In *Proceedings of the 18th Annual Conference of the CFD Society of Canada*, London, Canada, 2010