V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

SEGMENT PATCHING OF THE HIGHER ORDER INTERFACE RECONSTRUCTION FOR THE VOLUME OF FLUID METHOD

Joris C.G. Verschaeve*

*Department of Energy and Process Engineering,
Norwegian University of Science and Technology,
N-7491 Trondheim, Norway
joris.verschaeve@ntnu.no

Key words: B-Splines, Two-phase flow

Abstract. *We derive a more rigorous mathematical formulation for the interface reconstruction problem in the framework of the volume of fluid method. This leads to an interpolatation problem which can be solved systematically by means of B-spline interpolation allowing a higher order description of the interface segments. These segments need to be patched. Three different methods to patch these segments are derived. Numerical verification shows that a higher order description of the interface can be obtained by the present method.*

1 INTRODUCTION

Two-phase flow is a common issue in industry. This flow consists of two immiscible fluids separated by an interface. Different numerical methods have been devised to compute the evolution of this interface. In general these methods can be grouped into two classes, the interface tracking methods and the interface capturing methods [13]. The interface tracking methods such as particle methods [4] and the segment projection method [16] follow the interface in a Lagrangian manner, whereas interface capturing methods, such as the level set method [15] and the volume of fluid method (VOF) [13, 1] obtain the evolution of the interface indirectly by extracting it from a field defined in the entire space. For the level set method the evolution of the level set function is solved whereas the volume of fluid method computes the evolution of the volume fraction field.
The volume fraction field denotes the amount of one phase in each cell of the computational domain. The evolution of this volume fraction field is the central task of the VOF method. It can be subdivided into to two steps: the interface reconstruction step and the advection step. The interface reconstruction step computes the interface position at time t using the volume fraction field at time t. The advection step advects the volume fraction field from time t to time $t + \Delta t$. The VOF method has its origins in the works of [3] and [7]. Substantial improvement of the interface reconstruction has been achieved with the piecewise linear interface computation (PLIC) method by Youngs [17] in 1982. However

the resulting interface is approximated by discontinuous linear segments which makes it necessarily to estimate the curvature by additional approximation schemes. The PLIC method is second order accurate with respect to the grid spacing. A third order parabolic interface reconstruction scheme based on a very similar approach has been derived by Price *et al.* in 1998 [10]. However due to the fact that a numerical minimization has to be performed in each cell to find all the coefficients of the interface parabola, the method enjoys less popularity. More recently in 2004 Lopez *et al.* [5] used a parametric cubic spline interpolation through the midpoints of the PLIC interface segments and obtain a smoother description of the interface. However, although cubic splines are known to interpolate a function with fourth order accuracy, their method inherits the second order accuracy of the PLIC method since it is based on the same approach.
In the present discussion we describe in more detail a very recent development of higher order interface reconstruction. Higher order interface reconstruction approximates the interface not as a piecewise function but as a continuous line. Opposed to the other methods it is not based on the computation of an interface normal using the volume fraction field. Instead a mathematical reformulation of the interface reconstruction problem in terms of segments' integrals is found which allows to transform the problem of finding the interface into a problem of interpolating a function through a given set of points. Similar to the segment projection method (SPM) [16], the interface is divided into segments for which a set of *interpolation* points can be found from the volume fraction field. These interpolation points allow a better-defined basis to approximate the interface. An important issue in the high order interface reconstruction is the patching of these interface segments. The present discussion focuses on this aspect of the high order interface reconstruction.
The present discussion is organized as follows: In the next section basic principles of the VOF method necessary for the present discussion are summarized, followed by a short section on the segment projection method. Thereafter we present the mathematical reformulation of the interface reconstruction problem, section 4, and explain the basic principles of the method, followed by a section on the interpolation. A more detailed discussion of the segment patching is presented in section 6. A section on numerical verifications follows before concluding the present discussion in section 8.

2 THE VOLUME OF FLUID METHOD [13, 1]

For a two-phase flow problem of a red and blue fluid, the two-dimensional domain of definition Ω of our system can for each point in time t be divided into a region Ω_{red} occupied by the red fluid and a region Ω_{blue} occupied by the blue fluid, cf. figure 1.

These regions are separated by an interface which is the common boundary of both regions. A normal can be defined at each point on the interface. Predicting the evolution of the interface is the central problem of two-phase flow computations. Both fluids, the red and the blue, are supposed to be incompressible. Therefore the continuity equation needs to be satisfied by both fluids. We can define a characteristic function χ which is

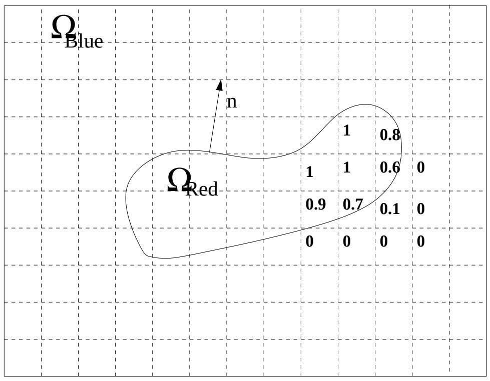

Figure 1: The two dimensional domain Ω with the red fluid occupying the region Ω_{red} and the blue fluid occupying the region Ω_{blue}. The boundary of Ω_{red} and Ω_{blue} separates both fluids and is called the interface. At each point of the interface a normal $\vec{n}$ can be defined. The domain Ω is divided into small control volumes, the cells. Each cell contains a number between zero and unity, the volume fraction, indicating the amount of red fluid in the cell.

unity in the regions of the red fluid and vanishes in the regions of the blue fluid:

$$\chi(x, y, t) = \begin{cases} 0 \text{ if } (x, y) \in \Omega_{blue}(t) \\ 1 \text{ if } (x, y) \in \Omega_{red}(t) \end{cases} . \tag{1}$$

We assume the surface of the cell normalized to unity. This characteristic function χ is discontinuous across the interface. Therefore a geometric approach to the evolution of χ is used in the volume of fluid method. The computational domain is cut into small control volumes, *cells*, cf. figure 1. For the present method it is important that the domain is cut into rectangular cells. This seems more restrictive than it actually is, since the coupling of the interface to the Navier-Stokes solver is treated as a moving boundary which is not aligned with the grid axes. Therefore also the treatment of a stationary boundary not aligned with the grid axes should be feasible. For each cell the volume fraction $C_{i,j}$ is computed by

$$C_{i,j}(t) = \int\limits_{i,j} \chi \, dxdy. \tag{2}$$

The volume fraction $C_{i,j}$ can be seen as the surface occupied by the red fluid in each cell. Blue cells have a volume fraction of zero whereas red cells have a volume fraction of unity. Mixed cells have a volume fraction in between zero and unity. The volume of fluid method proposes a scheme to compute the evolution of $C_{i,j}$. This is done in two steps, the interface reconstruction step and the advection step. The interface reconstruction step, cf. figure 2, computes the interface position starting from the volume fractions in each cell. In the present discussion we focus on this step of the volume of fluid method. This interface is then used as an input to the Navier-Stokes solver which with its help computes the velocity field $\vec{u}$ at time $t + \Delta t$. Knowing the velocity field $\vec{u}$ at times t and $t + \Delta t$, the volume fractions $C_{i,j}$ at time t and hence the interface at t, the advection step tells us

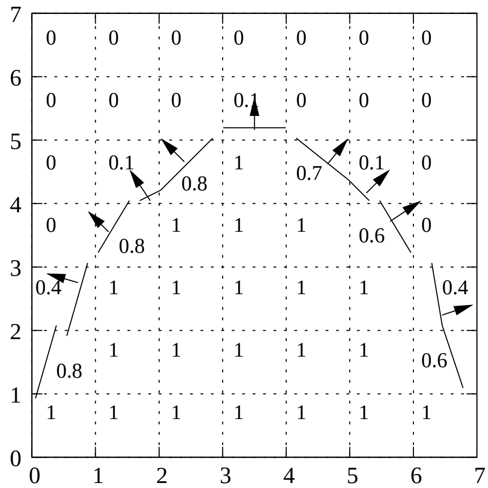

Figure 2: Interface reconstruction by Youngs' method [17]. In each mixed cell the interface normal is computed. This normal defines a perpendicular line which is unequivocally defined by the fact that the volume under the line should equal the volume fraction in the cell.

how to compute the volume fractions $C_{i,j}$ at time $t + \Delta t$. This is done by tracing back the cells from their position at $t + \Delta t$ to their position at t, cf. figure 3. Having the interface at time t, the volume fraction in each back traced cell is computed by evaluating the volume under the interface. For more details on the advection step, we refer to the work of Cervone *et al.* [2] whose advection scheme is used throughout the present discussion. For the interface reconstruction method this gives two constraints: not only needs the interface to be reconstructed as accurately as possible from the volume fractions, but in addition the interface reconstruction method needs to allow an exact evaluation of the surface under the interface in order to assure surface conservation.

The most widely used method to reconstruct the interface is the piecewise linear interface computation (PLIC) by Youngs [17]. This method computes first an approximation to the interface normal $\vec{n}$ in a each mixed cell by computing the gradient of the volume fraction field:

$$\vec{n} \approx \frac{\nabla C}{||\nabla C||}. \tag{3}$$

This normal defines a unique perpendicular line cutting the mixed cell in such a way that the surface under the line matches exactly the volume fraction in this cell, cf. figure 2. A drawback of the PLIC method is that it approximates the interface as a piecewise line. Therefore additional approximation schemes are necessary to estimate the curvature of the interface.

In this discussion we present an interface reconstruction scheme based not on an approximation of the interface normal, but on a more rigorous mathematical formulation in terms of the primitive of a segment. Dividing the interface into segments and describing the evolution of each segment is the central object of the segment projection method (SPM) [16], briefly outlined in the next section.

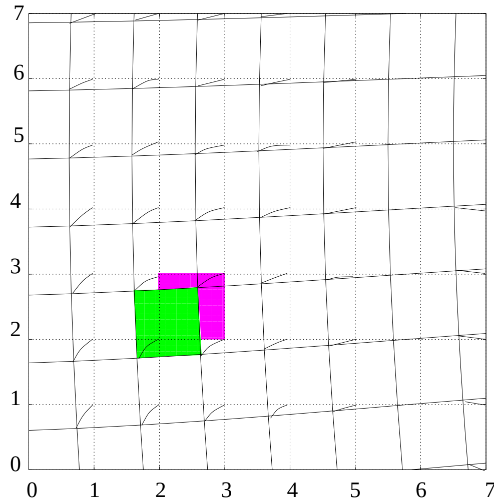

Figure 3: Using the velocity field $\vec{u}$ at time t ant $t + \Delta t$. The advection step computes the position at time t of the points included in the cells at time $t + \Delta t$. This is depicted by the green deformed square whose points are advected into to pink cell at time $t + \Delta t$.

3 THE SEGMENT PROJECTION METHOD [16]

The segment projection method was introduced by Tornberg and Engquist in 2003 [16]. The strategy of the method is to decompose the interface into x or y-monotone segments, cf. figure 4, and to advect each segment individually.

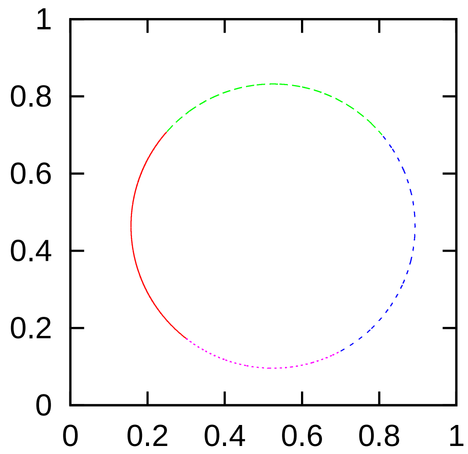

Figure 4: Decomposition of the interface of a circle into four segments.

The segments are described by a mapping taking the coordinate of the projected segment as an argument and returning the interface position. The circle in question, cf. figure 4, can be decomposed into four segments which each are described by a mapping defined on an interval:

$$
\begin{aligned}
y &\rightarrow f_1(y) \text{ on } [a_1, b_1]\\
x &\rightarrow f_2(x) \text{ on } [a_2, b_2]\\
y &\rightarrow f_3(y) \text{ on } [a_3, b_3]\\
x &\rightarrow f_4(x) \text{ on } [a_4, b_4].
\end{aligned}
$$

The evolution of these mappings is then solved in a Lagrangian manner. Therefore the segment projection method belongs to the interface tracking methods. The present discussion uses the concept of decomposing the interface into segments but relates the segment's interval to the volume fraction field leading to a set of interpolation points for the segment's *primitive*. These concepts are elucidated in the next section.

4 MATHEMATICAL REFORMULATION OF THE INTERFACE RECONSTRUCTION PROBLEM

The volume fraction $C_{i,j}$ can be interpreted as the surface of red fluid occupying the cell i, j.

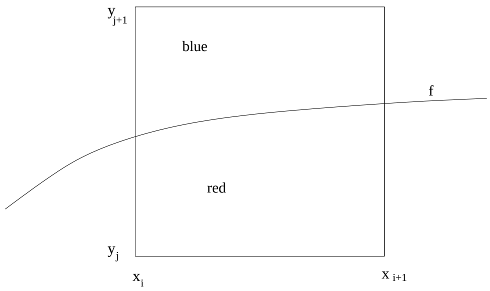

Figure 5: A cell intersected by the interface.

For a cell intersected by the interface as in figure 5, the surface integral of χ can be rewritten as a one-dimensional integral of the interface segment f:

$$
C_{i,j} = \int\int_{i,j} \chi \, dx dy = \int_{x_i}^{x_{i+1}} \int_{y_j}^{f(x)} dy dx = \int_{x_i}^{x_{i+1}} f(x)\, dx - (x_{i+1} - x_i) y_j. \tag{4}
$$

Thus along a segment we only need to sum up the volume fractions which we can relate to the integral of the interface segment f, cf. figure 6:

$$
A_i = \sum_j C_{i,j} = \int_{x_i}^{x_{i+1}} f(x)\, dx. \tag{5}
$$

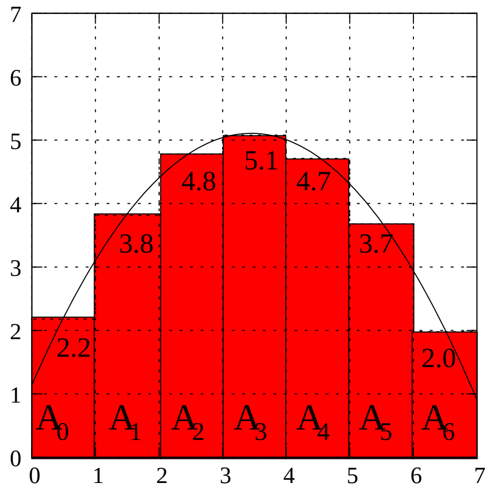

Figure 6: Summing up the volume fractions along y for a segment along x, gives us the area A_i under the interface between x_i and x_{i+1}.

By the fundamental theorem of integral and differential calculus, the integral of the interface segment is equal to the difference of the segment's primitive evaluated at the integration boundaries:

$$A_i = \int_{x_i}^{x_{i+1}} f(x)\,dx = F(x_{i+1}) - F(x_i). \tag{6}$$

This leads to a set of equations for each A_i. We can express $F_i = F(x_i)$ in terms of the A_i by solving (6) recursively:

$$F_i = \sum_{l=0}^{i-1} A_l + F_0. \tag{7}$$

This leads to a set of pairs for each segment:

$$(x_i, F_i) \quad i = 0, \ldots, n, \tag{8}$$

where n is the number of pairs. In the following we will call this set of pairs, equation (8), the *interpolation* points. The interface reconstruction problem can thus be divided into three subproblems:

1. Divide the interface into segments.

2. Compute the interpolation points for each segments.

3. Find an interpolant interpolating the interpolation points for each segment.

The first two points are of a rather technical nature and will not be treated in the present discussion. It needs to be underscored that so far we did not do any approximation but only found a different mathematical formulation for the interface reconstruction problem in terms of an interpolation problem. Once we have found an interpolant F, we obtain the segment f by differentiating F once:

$$f = \frac{dF}{dx}. \tag{9}$$

The computation of the normal $\vec{n}$ involves the second derivative of F and the computation of the curvature the second and the third derivative of F. The reason why we choose to find first the segment's primitive F instead of finding the segment f directly, for example in terms of finite differences of the interpolation points, is the following. Most advection schemes approximate the back traced cell by polygons [2]. Knowing the segment's primitive allows us to compute the surface under the segment exactly by evaluating the segment's primitive at the intersections between the segment and the polygon, cf. figure 7.

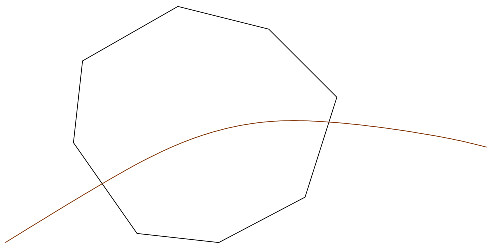

Figure 7: Back traced cell approximated by a polygon. Computing the surface under the interface is done by evaluating the surface primitive at the intersections between the polygon and the interface.

This ensures surface conservation of the algorithm. Since the abscissae of the interpolation points, equation (8), cannot be chosen arbitrarily but are the result of the division of the interface into segments, the method of interpolation used to find an approximation of F must be able to deal with arbitrary point distributions. In the following section we show how an interpolation by B-splines [6, 8, 14] can solve the problem.

5 INTERPOLATION BY B-SPLINES [6, 8, 14]

Knowing the interpolation points (x_i, F_i), $i = 0, \ldots, n$ for a segment, we use an interpolation s_P by B-splines of order P to find an approximation of F:

$$F(x) \approx s_P(x) = \sum_{i=-P}^{n-1} c_i B_i^P(x), \tag{10}$$

where c_i are the unknown coefficients which should be chosen in such a way, that $s_P(x_i) = F_i$, $i = 0, \ldots, n$; B_i^P is the B-Spline of order P associated with the knot x_i [8]. We assume

P being an odd integer. B-Splines have finite support [6, 8, 14] and therefore by imposing that the interpolant s_P should take the values F_i at x_i we obtain n conditions for the $n+P$ unknown coefficients c_i:

$$c_{i-P}B^P_{i-P}(x_i) + c_{i-P+1}B^P_{i-P+1}(x_i) + \ldots + c_{i-1}B^P_{i-1}(x_i) = F_i. \tag{11}$$

The missing $P-1$ conditions are obtained by imposing that the first $(P-1)/2$-derivatives of s_P should equal the first $(P-1)/2$- derivatives of the segment's primitive at each ending, x_0 and x_n. Since these $(P-1)/2$ derivatives are unknown a priori, opposed to the F_0 and F_n which are given by the volume fraction field, we need to approximate them by some method. This is the subject of the following section where we will derive different strategies for obtaining the derivatives of F at x_0 and x_n.
Having now $n+P$ conditions for $n+P$ unknowns c_i, we can solve the band-diagonal system (11). Band-diagonal systems can be solved very efficiently in order n number of steps [9]. Therefore the additional cost of B-Spline interpolation is subdominant to the overall cost.

6 SEGMENT PATCHING

Opposed to the segment projection method, it is not possible to let overlap the segments, since this would not ensure exact mass conservation in the back traced cell containing the overlap. Therefore we need a point at which two consecutive segments match in order to obtain a continuous interface. This point is called the patching point. We devised three methods to obtain the $(P-1)/2$ derivatives at the endings of the segments in order to patch the segments.

1. Finite Differences
   Using the interpolation points, equation (8), we can by means of finite differences compute the $(P-1)/2$ derivatives at the endings of each segment.

2. Continuity
   Using Faa di Bruno's formula [12] to relate the higher order derivatives of a function to its inverse, $P-1$ constraints can be obtained for the $P-1$ unknowns at each ending of two consecutive splines ($(P-1)/2$ from each spline ending). Without loss of generality we write $x \mapsto F_1(x)$ and $y \mapsto F_2(y)$, then $y = f_1(x) = \frac{dF_1(x)}{dx}$ is the first segment and $x = f_2(y) = \frac{dF_2(y)}{dy}$ is the second segment. The first few constraints at the patching point (x_0, y_0) can be written as follows:

$$y_0 = f_1(x_0) \text{ or } x_0 = f_2(y_0) \tag{12}$$

$$\frac{df_1(x_0)}{dx}\frac{df_2(y_0)}{dy} - 1 = 0 \tag{13}$$

$$\frac{d^2 f_1(x_0)}{dx^2}\left(\frac{df_2(y_0)}{dy}\right)^3 + \frac{d^2 f_2(y_0)}{dy^2} = 0 \tag{14}$$

For a spline of order $P = 3$, the coordinates of the patching point (x_0, y_0) are the unknowns of the system. They can be adjusted in such a way that constraints (12-13) are satisfied in order to obtain continuity in the first derivative of the segments. Practically this is done by choosing an initial value for x_0 and y_0 and iteration by means of a Newton-Raphson method until the constraints are satisfied.

3. Parametric Periodic Spline
Having an initial guess for the patching points and for the higher derivatives at the endings of the segments, we can construct a spline through the interpolation points for each segment. By differentiation we can compute the interface position for each segments. Sampling a set of interface points for each segment gives us an ordered set of points:

$$\{(x_i, y_i)|i = 1, \dots m\}, \tag{15}$$

where m is the number of points. Since the interface is a periodic line, there exists a periodic parametrization $(x(t), y(t))$ such that:

$$\begin{aligned} x(t_i) &= x_i \\ y(t_i) &= y_i \\ x(t_{i+m}) &= x_i \\ y(t_{i+m}) &= y_i \end{aligned}$$

In order to find the periodic sequence t_i, $i = 1, \dots m$ we propose a simple "fix-point" scheme. An initial guess t_i^0 is computed by evaluating the Euclidean distance between the point (x_i, y_i) and (x_{i-1}, y_{i-1}) and adding this to t_{i-1}^0:

$$t_i^0 = t_{i-1}^0 + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \text{ and } t_1^0 = 0. \tag{16}$$

This periodic sequence is used to construct a periodic parametric spline s^0 interpolating the points (x_i, y_i), $i = 1, \dots, m$. At a certain iteration n of our fix-point scheme, using the periodic parametric spline s^n we obtain t_i^{n+1} by computing the arc length d^n between (x_i, y_i) and (x_{i-1}, y_{i-1}):

$$d^n = \int_{t_{i-1}^n}^{t_i^n} \sqrt{\left(\frac{ds_x^n(t)}{dt}\right)^2 + \left(\frac{ds_y^n(t)}{dt}\right)^2} dt, \tag{17}$$

and adding it to t_{i-1}^{n+1}:

$$t_i^{n+1} = t_{i-1}^{n+1} + d^n. \tag{18}$$

Interpolating the points (x_i, y_i) using the t_i^{n+1} gives us the periodic parametric spline s^{n+1} of iteration $n + 1$.

This simple fix-point scheme will gives us a periodic parametrization of the interface. Once we have this parametrization we can evaluate the derivatives of our periodic parametric spline at the patching points between the segments and using Faa di Bruno's formula [12] find a new guess for the derivatives at the endings of the segments. This new guess for the derivatives allows us to construct more accurate interpolations for the segment primitives which in turn allows us to obtain more accurate interface points (x_i, y_i). This can then again be used to construct a new periodic parametric spline interpolating these points using the above method. This gives rise to a second (outer) fix point scheme allowing us to reach very accurate interpolants. The order of the interpolating splines used for the fix-point schemes should increase with increasing accuracy in order to keep the error contribution due to the interpolation subdominant compared to the error of the derivatives at the ending of the segments.

7 NUMERICAL VERIFICATION

As a numerical verification of our higher order interface reconstruction scheme and the segment patching schemes, we chose the interface reconstruction test of Rider and Kothe [11] and Lopez *et al.* [5]. The test of Rider and Kothe consists in reconstructing an interface from the exact volume fraction field of a circle with radius $r = 0.368$ centered at $(0.525, 0.464)$ in a two-dimensional box of sidelenght 1, cf. figure 8.

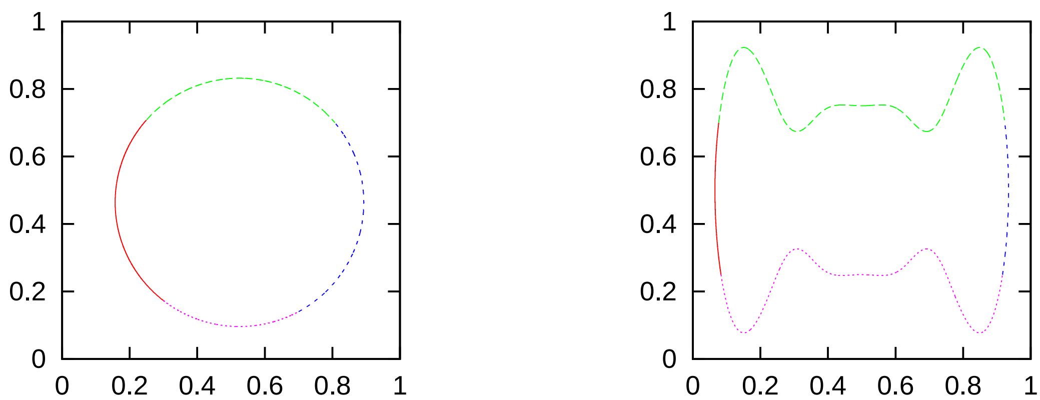

Figure 8: Examples of interfaces used as a numerical verification of the present interface reconstruction scheme. Left: Circle of Rider and Kothe [11]. Right: Bat-shaped interface of Lopez *et al.* [5]. The figures display the interface segments (in different colors) reconstructed by the present interface reconstruction scheme.

The test of Lopez *et al.* requires the reconstruction of a bat shaped interface given by the following equation:

$$\cos(6\pi x)(4x-2)^2 + (4y-2)^2 = 1$$

in a two-dimensional box of side length 1, cf. figure 8. In a first step we combined the segment patching by finite differences and continuity in such a way, that we first use the value of the derivatives of the segment's primitive obtained by finite differentiating the interpolation points. If the continuity of one of the first $P-1$ derivatives across the patching point is not satisfied, we use a Newton-Raphson scheme to improve the derivatives at the patching point. For better resolved interfaces the finite differentiation is most often accurate enough. An example of the patched segments for the circle of Rider and Kothe and the bat-shaped interface of Lopez *et al.* can be seen in figure 8. Performing

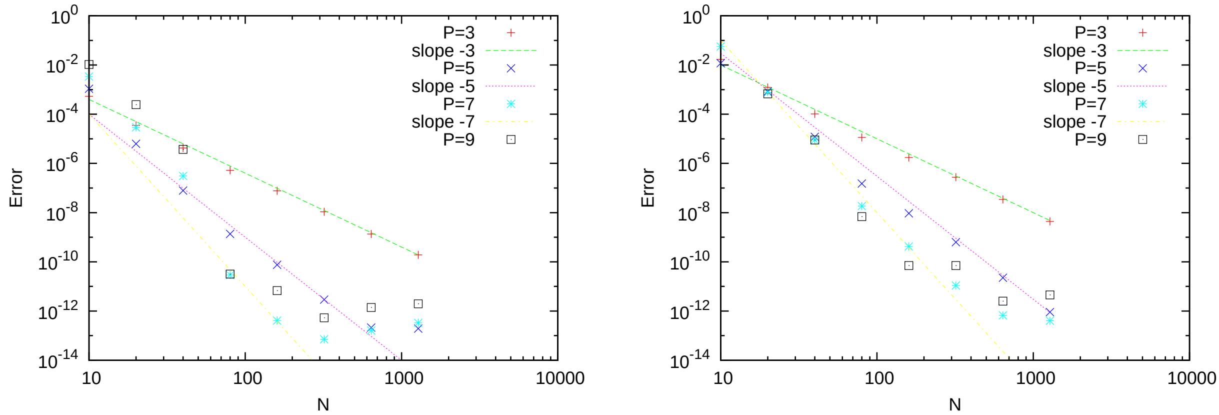

Figure 9: Error decrease of the present interface reconstruction scheme for different orders P of B-spline interpolation and different resolutions N (the number of cells along the side lengths of the boxes. Left: Circle of Rider and Kothe [11]. Right: Bat-shaped interface of Lopez *et al.* [5].

these tests for different resolutions, cf. figure 9, we see that the error decreases with third order accuracy for a third order spline and with fifth order for a fifth order spline. This is what we might expect given the error bound for B-spline interpolation in [6] on page 52. The convergence for seventh order splines seems to be seventh order, however for ninth order it is difficult to say whether the convergence is ninth order or not. This is due to the inaccurate estimation of the derivatives at the segment endings. In order to obtain more accurate estimations of the derivatives we use the fix-point scheme above. The resulting error decrease for the periodic parametric spline is shown in figure 10, revealing an error decrease as expected by the error bound in [6] on page 52. Fixing the resolution N and increasing the order P of the B-splines leads to a spectral, meaning faster than algebraic, convergence of the interpolation for the circle of Kothe, cf. figure 11. This is however not always the case as can be seen for the error decrease of the periodic parametric splines for the bat-shaped interface of Lopez *et al.*, cf. figure 12, since increasing the order of the B-splines can be done without increasing the spatial resolution of the interpolation. This is opposed to classical spectral methods, such as Chebyshev polynomials or Legendre polynomials where increasing the order of the interpolation increases the number of Gauß' points at which the function is interpolated. Therefore an aliasing error of interpolation can persist if the function is not well resolved when using B-splines. In order to test

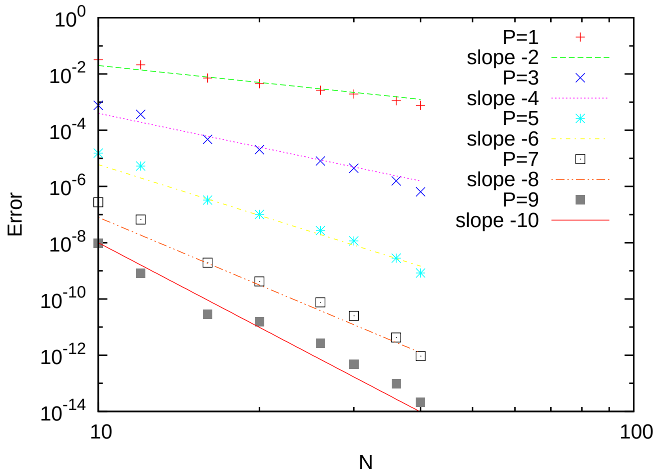

Figure 10: Error decrease of a parametric periodic spline for different orders P of interpolation and different resolutions N for the circle of Rider and Kothe [11].

the practical value of the current interface reconstruction method, we coupled it to the advection scheme of Cervone *et al.* [2] and perform the Vortex test of Rider and Kothe [11] using B-splines of order 3 with a resolution of $N = 32$ cells along each axis, a CFL-number of 1/2 and a period of $T = 2$. The position of the segments for $T = 1$ and $T = 2$ can be seen in figure 13.

Although it is also difficult for the present method to predict the position of the tail of the drop correctly, the resulting error is smaller than for a PLIC interface reconstruction.

8 CONCLUSIONS

The more rigorous mathematical formulation of the interface reconstruction problem in terms of an interpolation problem for the segment's primitive, allowed us to derive a higher order interface reconstruction scheme using B-splines. A main difficulty is the patching of these segments. We derived three strategies to achieve an accurate patching: estimation of the derivatives at the endings of the segments by finite differences, solving a set of constraints to guarantee continuity of first derivatives of the segments and patching by constructing a periodic parametric spline using the periodicity of the interface. Numerical tests showed that for an order ≤ 7, finite differencing and the continuity constraints are enough to patch the segments. Better accuracy can be obtained by using periodic parametric splines, if the drop shape is well resolved.

REFERENCES

[1] D. J. Benson. Volume of fluid interface reconstruction methods for multi-material problems. *Applied Mechanical Review*, 55(2):151–165, 2002.

[2] A. Cervone, S. Manservisi, R. Scardovelli, and S. Zaleski. A geometrical predictor-

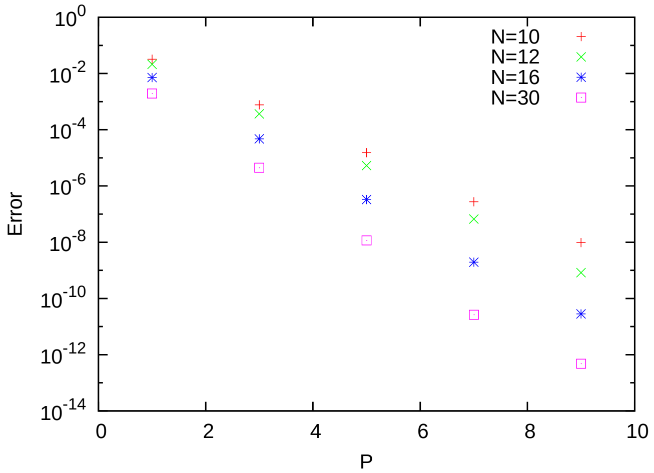

Figure 11: Error decrease of a parametric periodic spline in function of the order P of interpolation for different resolutions N for the circle of Rider and Kothe [11]. The scaling of the P axis is linear.

corrector advection scheme and its application to the volume fraction function. *Journal of Computational Physics*, 228:406–419, 2009.

[3] R. DeBar. Fundamentals of the kraken code. Technical Report UCIR-760, Lawrence Livermore Nat. Lab., 1974.

[4] J. Glimm, E. Isaacson, and . Marchesin, D. nad McBryan. Front tracking for hyperbolic systems. *Advances of Applied Mathematics*, 2:91, 1981.

[5] J. López, J. Hernández, P. Gómez, and F. Faura. A volume of fluid method based on multidimensional advection and spline interface reconstruction. *Journal of Computational Physics*, 195:718–742, 2004.

[6] G. Micula and S. Micula. *Handbook of Splines*. Kluwer Academic Publishers, 1999.

[7] W. Noh and P. Woodward. Slic (simple line interface calculation). In *Proceedings of the 5th International Conference on Fluid Dynamics*, volume 59, pages 330–340, 1976.

[8] G. M. Phillips. *Interpolation and Approximation by Polynomials*. Springer Verlag, New York, 2003.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recepies in C++*. Cambridge University Press, second edition, 2002.

[10] G. Price, G. Reader, R. Rowe, and J. Bugg. A piecewise parabolic interface calculation for volume tracking. In *Proceedings of 6th Annual Conference of the Computational Fluid Dynamics Society of Canada*, Victoria, British Columbia. University of Victoria.

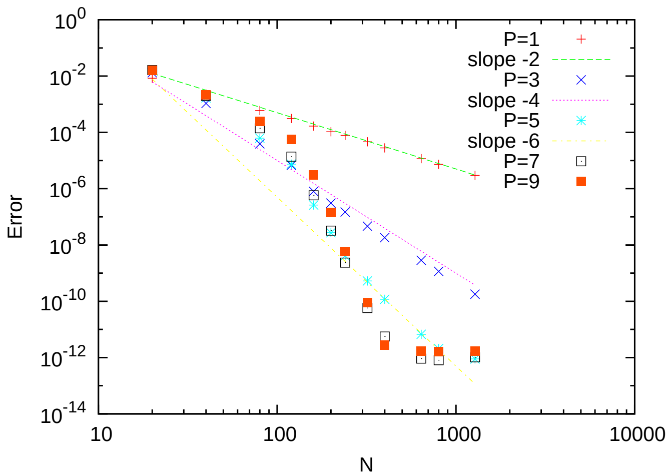

Figure 12: Error decrease of a parametric periodic spline for different orders P of interpolation and different resolutions N for the bat-shaped interface of Lopez *et al.* [5]. Higher order interpolations only start to converge with a higher spatial resolution of the interface shape.

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

[12] S. Roman. The formula of faa di bruno. *The American Mathematical Monthly*, 87(10):805–809, 1980.

[13] R. Scardovelli and S. Zaleski. Direct numerical simulation of free surface and interface flow. *Annual Review of Fluid Mechanics*, 31:567, 1999.

[14] L. L. Schumaker. *Spline Functions: Basic Theory*. John Wiley and Sons, 1981.

[15] M. Sussman, P. Smereka, and O. S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

[16] A.-K. Tornberg and B. Engquist. The segment projection method for interface tracking. *Communications on Pure and Applied Mathematics*, LVI:47–79, 2003.

[17] D. L. Youngs. Time dependent multi-material flow with large fluid distortion. *Numerical Methods for Fluid Dynamics*, pages 273–285, 1982.

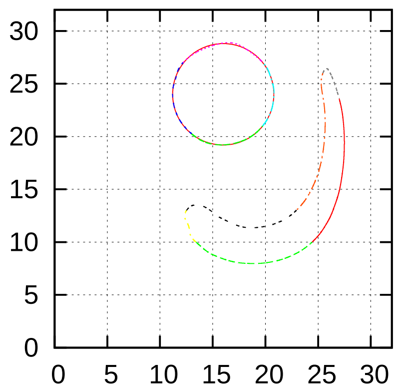

Figure 13: Vortex test of Rider and Kothe [11] using the present interface reconstruction method. The resolution is 32 cells in each spatial direction, a cfl number of 1/2 and a period of 2.