This document was prepared for the ETI by third parties under contract to the ETI. The ETI is making these documents and data available to the public to inform the debate on low carbon energy innovation and deployment.

Programme Area: Marine

Project: PerAWAT

Title: Numerical Modelling of Tidal Turbine Arrays Involving Interactions within an Array: Development of the Level-Set Free Surface Model

Abstract:

This report forms part of work package WG3WP2, the objective of which is to: a) model the interaction of three dimensional unsteady flow and turbine wakes within an array; b) verify available numerical models; and c) implement an appropriate free surface model within Code_Saturne. This report provides the reader with an introduction, methodology and guide to implementation of the work conducted to provide a suitable means of tidal stream modelling. The tidal stream modelling is seen as an important component to the whole project since it provides the necessary upstream boundary condition to a small array of marine current turbines at the meso-scale using EDF's Computational Fluid Dynamic (CFD) solver 'Code_Saturne'.

Context:

The Performance Assessment of Wave and Tidal Array Systems (PerAWaT) project, launched in October 2009 with £8m of ETI investment. The project delivered validated, commercial software tools capable of significantly reducing the levels of uncertainty associated with predicting the energy yield of major wave and tidal stream energy arrays. It also produced information that will help reduce commercial risk of future large scale wave and tidal array developments.

PerAWaT (MA 1003) Report – (WG3 WP2 D1) Numerical modelling of Tidal Turbine arrays involving interactions within an array: Development of the level-set free surface model

Participant lead on the deliverable:- **D.M. Ingram**
Other participant:- **D.A. Olivieri**
Institute Energy Systems, School of Engineering, University of Edinburgh, The King's Buildings, Mayfield Road, Edinburgh EH9 3JL UK
Version v3.0
Date of submission:- **09/02/2011**

Nomenclature

$\vec{A}$, $\vec{B}$	vector lines formed between the centres of a cell iel and its neighbour cell with index number
C	Constant that preserves the consevation of φ globally
con(iel,i)	Connectivity function giving the index number of the cell face i associated with cell index number iel
C_I	Set of nodes not including the I_h[th] node but is a subset of the combined set of
$\mathrm{H}(s)$	Heaviside function such that $\mathrm{H}(s)$ = 1 if s > 0, otherwise $\mathrm{H}(s)$ = 0 for s $\leq$ 0
iface(i,ifac)	Code Saturne's connectivity function giving the index number of the i[th] neighbour cell about cell face index number ifac
I_h	I_h[th] node in the halo region of the narrow band region
κ	Set of simplices in a region of the narrow band such that it surrounds an isosurface
nbcell(iel,i)	Cell connectivity function giving the index number of a neighbour cell associated with a central cell with an index number of iel
η_K	Simplexwise correction function associated with the K[th] local grid mesh simplex volume function
N_I	Number of simplexes that contain the node *I* which the K^{th} simplex is a part of
nfac	Total number of cell faces from all field cells(code Saturne parameter)
P	Set of nodes with a narrow band region with a set of κ simplices.
$\Re$	Set of halo nodes of the narow band région in which I_h[th] node is a part of $\Re$
S_h	Zero Level Set surface
Simplex	Any triangle or tetrahedron formed from the cell centres of existing mesh cells which arround an isosurface region for the purpose of geometric based re-distancing of that region of the isosurface.
t	Model time in seconds
$\vec{u}$	Interface velocity of the isocontour, wave.
u_x	*x* component of $\vec{u}$
u_y	*y* component of $\vec{u}$
u_z	z component of $\vec{u}$
$\vec{x}$	Position vector in the flow field
$\vec{y}$	Position vector on isocontour S_h
φ	Level Set scalar
φ^*	Signed distance function from a flow field point $\vec{x}$ the closest point on the iscontour S_h
ψ	Correction funtion to correct or redistance φ
ΔV_K	Local grid mesh simplex volume function for simplex K in the narrow band
ΔV	Global volume function for a set κ of simplices
ξ_{I}	Nodewise correction function at node I for a number of simplices that contain node I
Ω	Flow field
θ	The angle between two vectors $\vec{A}$ and $\vec{B}$
$\vec{X}_J$	Position vector of a the J[th] node such that J belongs to the C_I set of nodes in the narrow band région
$\vec{X}_{I_h}$	Position vector of a the I_h^{th} node such that I_h belongs to the $\Re$ set of nodes in the narrow band région

Table of contents

1) Executive summary

This report forms part of the deliverable D1 of work package WG3WP2. The WG3WP2 objectives are:

- To model the interaction of three dimensional unsteady flow and turbine wakes within an array.
- To verify available numerical models
- To implement an appropriate free surface model within Code_Saturne.

The work associated with this deliverable can now allow Level Set modelling to be conducted without the problem of Code Saturne having no cell connectivity. This formed the most central piece of work of the deliverable. Therefore WG3 WP2 D1 has set the groundwork for implementing later work on the free-surface or upstream boundary conditions.

What is known now that was not known before is the need for cell to cell connectivity in Code Saturne module development for Level set modelling. Also the initial literature survey has indicated that the best approach for the re-distancing in the Level set scheme should be a geometric mass-preserving method, which can work with general polyhedral meshing. These problems were solved by the author's own coding strategy involving an 'Object-Oriented' approach to building the cell-to cell connectivity and the narrow band model which has the job of re-distancing the Level Set scalar. The implications of these problems involve the use of external modules without the need to create new specific fortran 90 modules. Hence re-coding the existing specific modules known as USCLIM.f90, USINIV.f90 and USPROJ.f90 have only been changed as a result.

To achieve these objectives, the PerAWaT project will make use of the existing parallel, high accuracy open source, CFD code Code_Saturne (developed by EDF) and will extend the model to provide a mechanism for modelling the performance of a small array of marine current turbines at the meso-scale.

The content of the deliverable can be best summarized by the following points:

- Since the onset of this work progress has been made in extending Code Saturne's modelling capabilities to Level Set free surface modelling. This involves a geometric mass-preserving re-distancing scheme, which is acknowledged currently as the most effective re-distancing method, which can work with general polyhedral meshing.
- Experience has be gained in Code Saturne's Fortran 90 programming structure to allow more flexible modelling involving the use of Fortran modules. This has allowed the author to overcome the major problem with Code Saturne lack of cell to neighbour connectivity, and is covered in detail in sections 4.1 and 4.3.
- The implications of Code Saturne's lack of cell to neighbour connectivity has involved significant code development delays which was initially unforeseen. Fortunately, with the help of an object oriented Fortran 90 coding approach these issues will not make major inroads into the final code's performance.

This report provides the reader with an introduction, methodology and implementation of the work conducted to provide a suitable means of tidal stream[1] modelling. The tidal stream modelling is seen as an important component to the whole project since it provides the necessary upstream boundary condition to a small array of marine current turbines at the

[1] The term tidal stream is refers to the currents associated with the tides, generally near a coastline or harbor.

meso-scale using EDF's CFD solver Code_Saturne. Ultimately this work will follow a process of development with the existing flow solver. Namely, (1) The implementation of a free surface model which satisfies the zero-tangential shear boundary condition, (2) The implementation of an unsteady upstream boundary condition to introduce large scale, synthetic, turbulent eddy structures into the domain, and (3) The development of a parameterised actuator disk model of a horizontal-axis marine current turbine.

The introduction covers the description of the Level Set method and why it was chosen as the best approach to free surface modelling, with reference to the most recent improvements in the numerical scheme, namely improved mass conservation features. The methodology section covers the more detailed description of the actual Level Set method used, with particular reference to the author's implementation of an advanced mass preserving re-initialisation method and how it can be applied to either structured or unstructured grid systems. The implementation covers an in depth description of the modules (open source code) for Code_Saturne implementing a free-surface boundary condition and an unsteady upstream boundary condition.

The deliverable meets the acceptance criteria of (a) modules implementing the level-set method which can be compiled into and executed within Code_Saturne for output from the flow code, (b) a report describing the approach and assumptions used in the modules (c) graphical output which confirms the functionality of the code in terms of level set routine preserving the sharpness the isocontour over a given number of time steps and mesh resolution. (See section 3.2 for more details.

2) Introduction

The application of Code_Saturne will require three major developments with the existing flow solver:

1. The implementation of a free surface model which satisfies the zero-tangential shear boundary condition;
2. The implementation of an unsteady upstream boundary condition to introduce large scale, synthetic, turbulent eddy structures into the domain; and,
3. The development of a parameterised actuator disk model of a horizontal-axis marine current turbine.

This report addresses the point (1), the implementation of a free surface model, which satisfies the zero-tangential shear boundary condition. The inclusion of such a free surface is thought to be important to the accurate modelling of the flow field in and around and array of turbines, as energy extraction may lead to deformation of the free surface and subsequent changes in the pressure and velocity distributions around the turbines. Furthermore, such a boundary condition is critical if the impact of surface waves on the turbines is to be investigated.

Presently, a variety of numerical methods have been proposed for modelling the free surface, or more explicitly modelling immiscible fluids with interfaces. These involve methods such as Surface Fitting [1], Density Function [2], Front-Tracking [3], Smoothed Particle Hydrodynamics [4], Volume of Fluid (VOF) [5], Free Surface Capturing [6] and Level Set Method [7,8].

In surface-fitting methods [1] the flow equations are solved in the liquid region only with the free surface treated as a moving upper boundary to the computational domain. This method is very efficient for simple free surface problems, but its applicability is limited by the

skew ness[2] of the resulting computational grid. Another obvious limitation of the method is that the effects of water–air interactions such as the trapping of an air bubble in the water ambient[3] cannot be predicted, as in this case no explicit boundary conditions can be specified at the interface [11]. Surface-tracking methods, however, simulate both fluid regions on a fixed grid system, and the free surface is identified by a marker function such as the volume fraction in the widely used VOF method [5, 9,12-14]. Unfortunately, VOF has drawbacks due to the crudity of the method leading to the need for large number of cells and occurrence of inaccuracies in calculating intrinsic properties of the wave front such as curvature etc. Also VOF evolution will involve complex speed function which is difficult to model. Smoothed Particle Hydrodynamics also suffer the problem of poor surface definition without resorting to computationally inefficient large particle usage. The Level Set approach however, allows numerically efficient computations of curves and surfaces on a Cartesian grid without having to parameterize these objects.

From the outset of the project it was proposed that a level-set approach be adopted and that the boundary condition described by Watanabe, Saruwatari and Ingram [15] be implemented, as this approach satisfies the zero-tangential-shear condition on the free surface and has been successfully applied to the propagation of ocean surface waves – without the dissipation associated with Volume of Fluid (VOF) methods.

2.1 The Level Set Method

The principle of the Level set is to represent the interface S_h as a zero-level set function φ such that its evolution can be described by

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot (\vec{u}\varphi) = 0, \qquad (1)$$

Where $\vec{u} = (u_x, u_y, u_z)$ is the interface velocity in the three dimensional flow field Ω. In this case, since the interface is simply advected with flow, the velocity, $\vec{u}$, is that obtained from Code Saturne's solution of the Navier-Stokes equations. Both the level set function and the velocity field are functions of $(\vec{x}, t), \vec{x} \in \Omega, t > 0$. Unfortunately, Equation (1) does not arise from conservation law, since it does not imply that the zero-level set of φ will propagate at the correct speed, implying that a mass conservation error will occur at the moving interface. Recent efforts towards improving this have seen methods as elaborate as level-set- based adaptive Characteristics-Based Matching method [16], which involves the use Adaptive Mesh Refinement applied in the main flow transport solver and the Level Set Method combined. In view of the fact that the main flow transport solver involved with Code Saturne works only with fixed unstructured Cartesian grids, the authors have therefore selected the approach of Ausas et al [17] as an up to date Level Set Method.

The Partial Differential Equation (PDE) numerical solution of the hyperbolic equation given Equation (1) will involve an upwind finite difference technique, which is automatically provided by Code Saturne's flow field solver for the selected scalar φ. However, the Level set method will not guarantee the conservation of the φ. Unfortunately the scalar φ will experience a mass loss error as it evolves with time. The usual approach to minimise errors is to set φ as an assigned distanced function to the interface surface S_h and periodically re-initialize or "re-distance" φ, so that Level Set distortion and numerical errors are kept under control.

[2] The term skew ness refers to angle between the vector normal of the cell face between two neighbouring cells and the line vector connecting their centres [10].

[3] Water-air interactions are of significance in wave modelling particularly with overtopping [11].

In the case of uniform and rotational flow the shape of the level set may get severely distorted and the level set may vanish over several time steps. The need for re-distancing of the Level set is therefore required. The usual approach to perform re-distancing would involve high-order finite difference schemes, such as high-order essentially non-oscillatory (ENO) schemes like Hamiltonian-Jacobi Partial Differential equations [7,8,18]. Unfortunately, this approach has not been applied to unstructured polyhedral grid reliably, making the geometric mass-preserving re-distancing scheme [17] the best candidate over PDE-based schemes. The approach of Ausas et al [17] involves a geometric mass-preserving re-distancing scheme for φ, as opposed to usual PDE-based re-distancing schemes and this is achieved by means of a localized mass correction without the use of adjustable parameters. To implement the approach of Ausas et al [17] will require use of Code Saturne's flow field solver and the development of the re-distancing routine which is used periodically a to readjust the scalar φ.

The Level set scalar φ would perfectly represent a signed distance function φ^* if conservation was automatically maintained. Therefore only

$$\varphi = \varphi^* + \psi \qquad (2)$$

is possible with a suitable correction scalar function ψ[4]. In Figure 1 the signed distanced function φ^* is defined, which is the closest signed distance to the isocontour S_h, in which $\vec{y}$ belongs to the isocontour for a given point in space $\vec{x}$

$$\varphi(\vec{x})^* = \text{sign}[\varphi(\vec{x})] \min_{\vec{y} \in S_h} \|\vec{x} - \vec{y}\| \qquad (3)$$

3) Methodology

To implement the approach of Ausas et al [17] will require use of Code Saturne's flow field solver and the development of the re-distancing routine which is used periodically a to readjust the scalar φ. The following two subsections provide a detailed description of the developments required to Code Saturne order to implement the Geometric Mass Preserving Re-distancing Method and the theory behind the Geometric Mass-Preserving Re-distancing Method.

3.1 Background to the re-distancing process.

Unfortunately the scalar φ will experience a mass loss error as it evolves with time and therefore will require periodically a re-distancing routine to readjust the scalar φ. There are two approaches to re-distancing as mentioned earlier known as Geometric mass-preserving re-distancing [17] and Partial Differential Equation-based re-distancing [7,8,18]. The author chose the first approach mainly because of its ease of use in unstructured grids. Now readjustments to φ are conducted locally at each mesh node, rather than globally as with Partial Differential Equation-based schemes, as a result Geometric mass-preserving re-distancing will offer better accuracy overall. How this is done is covered in detail in section 3.2

3.2 Geometric mass-preserving re-distancing

[4] The explanation as to why re-distancing is needed, which is indicated in Equation (2), is the most generally adopted description of the re-distancing process see references [7,8,15,17,18]

The geometric mass-preserving re-distancing approach [17] offers improved mass conservation compared to schemes based on the solution of Hamiltonian-Jacobi Partial Differential equations [7,8,18], which have been widely used until now. The adjustments to φ are confined to a narrow band of primary neighbour cells about the computed zero isocontour as shown in Figure 4 as red cells, which amounts to one cell either side of the isocontour. Also there are a set of secondary or halo cells, shown in pink that surround the primary cells. These provide a means of primary cell to secondary cell exchange as the isocontour changes position at each time step. In order for Code Saturne to achieve this degree of control over each cell in the flow field the author had to develop a means of cell to locally surrounding neighbour cell connectivity, which did not exist with Code Saturne initially. As a result of achieving this connectivity a set of primary and secondary cells could be linked up to form a narrow band object as seen in Figure 3. The light and dark blue cells form the set of primary cells with the isocontour at the interface between light and dark cells. The light green being the halo cells. This forms the basis of each indexed narrow band object that finally forms a spine of narrow band objects as seen in Figure 4. As the model time advances each narrow band object will inhabit different neighbouring cells as the scalar values given at the cell centres change. This will alter the position of the narrow band in a way as shown in Figure7 as a band of white cells. As stated previously the change in φ at each cell centre will involve both that from advection of $\vec{u}$ and the secondary influence from re-distancing of φ. The process by which φ is corrected can be outlined briefly if we consider any cluster of narrow band object nodes such that part of the isocontour S_h involved will be a subset of the set of κ simplexes involved where their φ values change sign.

1. The centres of the Finite Volume cells used in the transportation algorithm to calculate φ form the vertices of the triangulation necessary for the re-distancing algorithm. In Figure 5 the reconstruction of isocontour S_h from calculated values φ will provide the means to calculate the signed distance function φ^* at each node in the narrow band. The disagreement between these values form the basis to calculate the mass correction function ψ

   $$\varphi = \varphi^* + \psi \,. \qquad (4)$$

2. To obtain the mass correction a piecewise-constant function η_K is found for each simplex formed out of first neighbour cell centres of the narrow band objects, such that the difference in volumes defined by φ and φ^* would be corrected at each simplex K such that

   $$\Delta V_K(\varphi,\ \varphi^* + \eta_K) = 0 \,. \qquad (5)$$

   Where ΔV_K is defined as follows

   $$\Delta V_\kappa\left(\varphi(\vec{x}),\varphi^*(\vec{x})+\eta_\kappa\right) = \int_\kappa \left[\mathrm{H}(\varphi(\vec{x})) - \mathrm{H}(\varphi^*(\vec{x})+\eta_\kappa)\right]\cdot d\vec{x} \qquad (6)$$

   Where $\mathrm{H}(s)$ is the Heaviside function such that the function is unity if $s > 0$ and zero otherwise. The value of η_K is then determined by a false position method to obtain η_K for that particular simplex. Ultimately, considering ΔV for the set of κ simplices in this region would suggest for a set of simplex wide solutions of η_K

   $$\Delta V\left(\varphi,\varphi^*\right) = \sum_{K\in\mathrm{K}} \Delta V_K\left(\varphi,\varphi^*\right) \qquad (7)$$

   In view of η_K discontinuous nature a node wide solution ξ_{I} is sort at each node I

3. The simplex wise contributions η_K at each node of the first neighbour cells, which are a part of the simplexes in the narrow band are then averaged at each node to provide a node wise correction such that

$$\xi_I = \frac{1}{N_I}\sum \eta_K \ , \qquad (8)$$

$$\psi_I = C\xi_I \qquad (9)$$

where N_I the number of simplexes that contain the node I which the K^{th} simplex is a part of and C the constant that preserves the mass conservation of φ globally such that

$$\Delta V\left(\varphi, \varphi^* + C\xi\right) = 0 \qquad (10)$$

The solution of equation (10) will to find C again involves a false position method.

4. The mass corrected values of φ amongst all nodes of the first neighbour cells will now provide a boundary condition for the re-initialisation of φ on the rest of the mesh points in the halo cells. If we consider just the halo cells with φ positive, for say a $\Re$ set of halo nodes in which $I_h \in \Re$ we use a distance along an edge approximation such for I_h such that

$$\varphi(X_{I_h}) = \min_{J \in C_I} [\varphi(\vec{X}_J) + |\vec{X}_{I_h} + \vec{X}_J|] \qquad (11)$$

Where C_I is a set of nodes that are connected I_h but I_h^{th} node is not included in the C_I set of nodes but $C_I \subset (\mathrm{P} \cup \Re)$, P being the set of nodes associated with κ simplices. Equation (11) provides an edge distance approximation. For each simplex, and for each node J of the simplex at position $\vec{X}_J$, φ is interpolated linearly on the opposite face FJ, using the current values at the nodes generated by the earlier stage involving the first neighbour cells. Then, a tentative new value η_J at node J is calculated such that

$$\eta_J = \min_{\vec{x} \in F_J} [\varphi(\vec{x}) + |\vec{X}_{I_h} + \vec{x}|] \qquad (12)$$

$$\varphi\left(\vec{X}_J\right) = \eta_J \text{ if } \varphi\left(\vec{X}_J\right) > \eta_J \qquad (13)$$

5. As a result of the re-distancing routine given above, some of the first and secondary neighbour cells of the narrow band have altered. New cells will have to be inhabited and old cells evacuated as a result of the change in position of the isocontour, as seen by the band of white coloured cells shown in Figure 7.

4. Implementation

This section describes the implementation of the level-set method and its associated reconstruction algorithm in Code_Saturne. (In order for the reader to understand the Code Saturne run directory structure, the reader is advised to refer to EDF's practical user's guide [19]).

In order to demonstrate the implementation of the level-set advection and re-distancing algorithms a simple test case based on flow passed a 2D cylinder at a Reynolds number of 5000 has been used. The zero-contour of φ has been set 1.5 cylinder diameters above the centre of the cylinder. The purpose of the test is to show that the algorithms have

been correctly implemented on an unstructured grid and to show that the routines are able to use a scalar quantity with the flow and adjust the scalar quantity to ensure that it's gradient is close to unity in the neighbourhood of the zero contour. These routines are at the heart of the level-set approach to free surface modelling, which is an important building block for the free-surface flow solver.

Because the boundary conditions have not been implemented (this will be done for D2 as specified in the technology contract) the cylinder cannot truly be said to be submerged, nor is the zero level-set of phi a free-surface at the moment. The normal test for the advection and re-distancing of the level-set function would be Zelesak's disk [20], however this cannot be used because Code_Saturne is solving the Navier-Stokes equations rather than the linear-advection equation. Further tests such as the submerged hydrofoil [21,22], surface piercing plate [23], low amplitude sloshing [24] etc need the zero-tangential shear boundary condition to be implemented together with the ghost-fluid method and this will be done for D2. D1 should therefore be seen as an interim deliverable in this respect.

4.1 Connectivity

The internal representation of the grid within Code Saturne has presented some difficulties. The solver represents the unstructured grid using a list of faces with pointers to the left and right control volumes, rather than providing a list of control volumes together with their connectivity information as required by the re-distancing algorithm. The first stage of implementation has therefore required the development of a set of library routines to compute the connectivity information for the halo cells surrounding the interface from the face connectivity list. These routines have been implemented within the CONNECTIVITY. Fortran-90 module coded into the preamble of the user routines. Face connectivity information provided by the NBCELL data structure is processed to find adjacent control volumes and from these a set of simplexes is constructed. Since the grid used is fully unstructured the list of control volumes computed has no implicit order and so must be sorted before re-distancing can be performed.

Below is a sample of code taken from the file USINIV.f90, which constructs the connectivity between a given cell of index `iel` and in this case 1 to six neighbour cells associated with each 3D hexahedra grid,

```
!===============================================================================
!Building connectivity between particular indexed cell in the flow field and
!its local neighbour cells. For a hexahedral cell this involves 6 neighbours
!===============================================================================
     do ifac = 1, nfac
        iel1 = ifacel(1, ifac)
        iel2 = ifacel(2,ifac)
        switch1 = .true.
        switch2 = .true.
        do i = 1, 6
           if ((switch1).and.(con(iel1,i).eq.(-ihuge) )) then
              con(iel1,i) = ifac
              switch1 = .false.
           endif
           if ((switch2).and.(con(iel2,i).eq.(-ihuge) )) then
              con(iel2,i) = ifac
              switch2 = .false.
           endif
        enddo
     enddo
```

The connectivity function `con(iel,i)` developed by the author provides the necessary process of indexed cell → local face connectivity for a given cell index `iel` and a randomly selected

cell face `i` . To complete the process of indexed cell → local face → local neighbour cell connectivity an extra piece of coding was needed in the form of another connectivity function developed by the author known as `nbcell(iel,i)`. As the reader can see in this initial coding exercise it has been set up specifically to deal with only three dimensional hexahedral meshing by having the number of cell faces restricted to six. In the next stage of development the code will be upgraded to meet any general unstructured polyhedral mesh.

From the code given above access to the i^{th} neighbour cell of the iel^{th} field cell is allowed through the function `nbcell(iel,i)`. Unfortunately, this will not provide an orderly access to these neighbour cells about the iel^{th} field cell, since we are dealing with general unstructured polyhedral Cartesian field cells. Therefore, there is a need to have an orderly access to cell faces `i` order for the flow field to be decomposed in the triangular meshing indicated in Figures 1 and 5. Therefore, to complete the development of the connectivity function `nbcell(iel,i)` the function needed to be sorted in a manner as shown in Figure 2, whereby the angle θ between two vectors $\vec{A}$ and $\vec{B}$ formed at one neighbour cell and another in relation to a central chosen cell of index `iel`, would provide a sorting attribute to sort the neighbour cells in an orderly way. From the development of the connectivity function `nbcell(iel,i)` a structured cellar arrangement can be made of the narrow band cells.

```
!================================================================
! Create 'nbcell(iel,Number_Of_Faces)' which provides connectivity
! between a cell centre of cell 'iel' and 1 to 'Number_Of_Faces'
! neighbour cells over 1 to ncel mesh cells in the flow field for
! any general polyhedral mesh (note in this case this is hexahedral)
!================================================================
    do iel = 1, ncel
     do ifac = 1,6
        if (con(iel,ifac).gt.(-ihuge)) then
           iel2 = ifacel(2,con(iel,ifac))
           iel1 = ifacel(1,con(iel,ifac))
           if (iel1.ne.iel) then
              nbcell(iel,ifac) = iel1
           endif
           if (iel2.ne.iel) then
              nbcell(iel,ifac) = iel2
           endif
        endif
     enddo
```

4.2 Re-distancing

The re-distancing algorithm has been implemented in several of the user routines, which provide an interface to the main Code_Saturne solver. The routine `USCLIM.f90`, which sets up the boundary conditions for the flow has been modified to allow properly distanced values of φ to be introduced through an inflow boundary. This routine does not need to make use of connectivity information since it is applied only to boundary faces. The routine `USINV.f90` sets up the initial values of the level-set across the entire fluid domain and contains the `CONECTIVITY` module described above. Furthermore, `USINV.f90`, also sets up the initial data structure containing the narrow band cells surrounding the zero-contour of the scalar variable. The initial configuration of the narrow band is set up about the specified isocontour position for time zero by using a set of narrow band node elements as seen in Figure 3. These elements finally form a spine of narrow band elements as shown in Figure 4. Each narrow band elements consist of two primary cells, which are positioned either side of the particular part of the isocontour. In addition there are also halo cells either side of the primary. As model time advances, the value of Level Set φ will change in accordance with the transport model and the

re-distancing algorithm as discussed earlier in section 3.2. At the end of each time step `USPROJ.f90` is called. In this subroutine, the re-distancing algorithm is given and implements the geometric mass preserving algorithm [1]. This algorithm monitors and updates the narrow band data structure as the free surface moves across the background mesh.
The `USCLIM`, `USINV` and `USPROJ` routines, containing self-documented source code, are contained in the attached files.

4.3 Experience gained from implementation

The lessons that have been learned so far from the implementation of the project can be best summarised from the following points

- Before writing the technology contract, the internal structure of Code_Saturne should have been better understood in more detail, particularly the connectivity issues of Code_Saturne. For example, the available functions that were given in Code_Saturne such as `ifacel(2,nfac) and ifabor(nfabor)` only provided cell face to neighbour cell connectivity, making direct cell to neighbour cell connectivity impossible without the need to develop what the author has created in terms of connectivity function `nbcell(iel,ifac)`. (See section 4.1 for more explanations on these functions). This would have then been taken into account in the timings of D1 for the development of connectivity function nbcell(iel,ifac).
- The initial training provided by EDF's R&D centre in Paris gave a useful introduction to Code_Saturne but should have made us aware of Code_Saturne's connectivity issues. A longer training period at Paris backed up by further training after three months into the contract at Paris again would cover coding practices to allow more effective Fortran 90 library module development, rather than rely on just online user group assistance which involves large time delays.
- Code_Saturne online support has provided a limited service leaving us to spend inordinate amounts of time trying to solve issues beyond that expected in the time period of D1. Better online support would be appreciated particularly manuals written in English and not partly written in French and English.

For the future, for projects of this nature, the authors' recommendations would include a more involved training programme for a user of Code_Saturne. This would be conducted at EDF's training centre in Paris and would involve an initial introductory session with Code_Saturne spread over a week. A month later, another week of training would be conducted involving advanced programming practice. This would help with Fortran90 library module development specific to the needs of the contract. (Note in our case this would involve connectivity module library development). Extra studies of running and debugging code and the use of graphical methods in data post-processing with Code_Saturne would prove very useful. The venue of the training sessions could be given either at Paris at EDF's R&D centre or as an online training conferencing programme.

The implications of such changes in initial training would result in more effective time management later on. This means that rather than have the researcher find out about the limitations of Code_Saturne's interface haphazardly in the D1 period, the researcher would be better prepared and subsequently more productive in delivery.

A review of how this problem escalated in the D1 period starts at the beginning of the contract in April. The author had a short introduction to Code_Saturne through a two day course given in Paris at EDF's R&D centre involving other members of the PeraWat team. After this point, progress was made in using Code_Saturne with the help of EDF's Code_Saturne support team. By the time the author had got to the point of having generated a customized mesh in CGN format, the author was ready to start coding on the development of

the re-distancing scheme. This brought progress up to June where the author discovered that there is a connectivity issue with Code_Saturne. The author attempted to contact Code_Saturne's support team only to discover there was only one person available to help on these issues. The person involved was a Mr Fournier who later attempted to resolve the issue by suggesting a re-verse engineering approach. Subsequently, Mr Fournier became unable to resolve the problem and the author left to solve the problem alone. A period of a month was spent with the author finally solving this problem. During this period EDF and UoE were involved but EDF's team did not escalate this issue to the project manager. As a result, by early October the project manager finally became aware of these problems as a result of the submission of WG3 WP2 D1.

Following the submission of WG3 WP2 D1 to the project manager, Garrad-Hassan refused to submit the deliverable to the ETI claiming that the report failed to meet the acceptability criteria as it did not contain a detailed verification with comparisons to either experimental results or results from other models. In addition GH required that the source code was added in text form to the report as an Annex. Following the refusal to submit D1 there followed a lengthy process of arbitration between GH and UoE regarding the content and presentation of the report. This involved the formatting and style of the report, and the information required and the interpretation of the deliverable descriptions in the technology contract. Much of this would have been unnecessary had the following steps been taken:

1. A clear process for the delivery of reports had been known and allowed for the in work timings when the technology contract was being written. We strongly suggest that future contracts allow an internal deadline for submission of reports 1 month before the deadline for receipt by the ETI.

2. The project manager had provided all partners with a template for reports, including information on the sections required specifically by the ETI (executive summary, experience gained from implementation, etc). This would be particularly beneficial to partners who are used to producing scientific papers and detailed technical reports on their work and whose expectations of the structure and content of a deliverable may differ from those of the project manager.

3. It should also be clear to all parties, including the project manager and the ETI, which deliverables comprise, for example, software or executable code accompanied by a commentary and which a detailed technical report.

3. The existence of a clear process for the signing off of reports at both the partner institution, the sub-project manager and the project manager before submission to the ETI. This process should allow for comments to be returned to the authors in a timely fashion and for them to be addressed and signed off without being bounced forward and backward by e-mail requiring further corrections beyond those points originally raised. Furthermore, where the project manager lacks the necessary technical background to review a report, a suitable alternative reviewer should be identified in advance.

4. When it became apparent that there were inconsistencies in the work package and deliverable descriptions in the technology contract (particularly between the text and summary tables), this issue should have been raised with the ETI and the situation clarified.

5. Results

To address the connectivity issues highlighted in Section 3&4 a set of new library routines has been implemented in the CONNECTIVITY module, which is held within the existing Code_Saturne routine usiniv.f90. The Level set method using the geometric mass preserving re-distancing algorithm has been implemented via developments to the existing Code_Saturne routines of USCLIM.f90 and USPHYS.f90. The source code containing the USCLIM, USINV and USPROJ routines are included on the accompanying CD with the modifications made as part of D1 highlighted.

To demonstrate the required functionality of the modifications standard free-surface flow test cases are used. Flow over a submersed 2D cylinder has been considered because it is the standard and thus provides a means of verifying the implementation of the level-set method in Code_Saturne. The specifics of the 2D flow test involved a von Karman Vortex Street modelling test case at a Reynolds Number of 5000. The isocontour was set at 1.5 cylinder diameters above the centre of gravity of the cylinder as shown in Figures 4 and 7. The scalar φ is easily set up with Code_Saturne and advected in a manner dictated by Code_Saturne flow solver for $\vec{u}$, using a laminar flow assumption that is with no eddy viscosity model involved.

The necessary data files for running the above case test are held within the CASE and STUDY directories provided on the accompanying CD.

6. Conclusions

The work package objective of WG3 WP2 D1 was to implement a Level-Set free surface model with Code_Saturne. Despite the major unforeseen problem associated with Code Saturne's lack of cell to neighbour cell connectivity, a basic implementation of the free surface level-set boundary condition has been achieved. Figures 3, 4 and 7 provide evidence to this in the form of a flow field test, which is represented in a discretized form as an unstructured mesh, modelling the flow over a submerged 2D cylinder. At this stage of the work-package (D1), Figure 7 provides a demonstration of the functioning code and is therefore not to be considered a validation. It serves to demonstrate that the code implementing the method has successfully been compiled into Code Saturne and executed within the solver. After the next deliverable (WG3 WP2 D2) a boundary free-surface condition will have been included and at this stage it will be possible to validate the code by comparison to the experimental data. This will involve a 2D test case with water flow over a submerged hydro-foil for which experimental data is available, and will be cross-validated by comparison with other CFD codes such as flow3D, and CF/X.

The deliverable meets the acceptance criteria of (a) modules implementing the level-set method which can be compiled into and executed within Code_Saturne for output from the flow code, (b) a report describing the approach and assumptions used in the modules (c) graphical output which confirms the functionality of the code in terms of the level set routine preserving the sharpness the isocontour over a given number of time steps and mesh resolution. (See section 3.2 for more details. Also, see the accompanying CD which lists code modules with new generated code highlighted).

7. References

[1]Farm, J., Martinelli, L., Jameson, A. Fast multigrid method for solving incompressible hydrodynamic problems with free surfaces. AIAA Journal 1994. 32, 1175–1182.
[2] Park, J., Kim, M., Miyata, H.. *Full non-linear free-surface simulations by a 3D viscous numerical wave tank.* International Journal for Numerical Methods in Fluids.1999. 29 (6), 658–703
[3]Unverdi, S., Tryggvason, G.. *A front-tracking method for viscous, incompressible multifluid flow*. Journal of Computational Physics 1992. 100, 25–37.
[4] Monaghan, J. *Simulating free surface flows with SPH*. Journal of Computational Physics.1994. 110, 399–406.
[5] Hirt, C., Nichols, B. *Volume of fluid (VoF) methods for dynamics of free boundaries.* Journal of Computational Physics.1981. 39, 201–225.
[6] Kelecy, F., Pletcher, R. *The development of a free surface capturing approach for multidimensional free surface flows in closed containers.* Journal of Computational Physics.1997. 138, 939–980.
[7] Osher S, Fedkiw R. *Level Set Methods and Dynamic Implicit Surfaces*. Publisher: Springer, Applied Mathematical Sciences. 2003: ISBN 978-0-387-95482-0.
[8] Sussman M, Fatemi E. *An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow*. SIAM J. Sci.Comput. 1999; 20:1165–1191.
[9] Youngs, D. *Time dependent multi-material flow with large fluid distortion.* In: Morton, K., Baines, M. (Eds.), Numerical Methods for Fluid Dynamics. Academic Press, London. 1982. 237–285.
[10]Online details of *"Meshing_CompressibleModels.pdf"* http://www.ara.bme.hu/oktatas/tantargy/NEPTUN/BMEGEATAM05/2009-2010II/ea_lecture/Meshing_CompressibleModels.pdf
[11] Ingram, D.M., Gao, F., Causon, D.M., Mingham, C.G., Troch, P. *Numerical investigations of wave overtopping at coastal structures*. Coastal Engineering. 2009. 56, 190-202
[12] Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S., Zanetti, G. *Modelling merging and fragmentation in multiphase flows with SURFER.* Journal of Computational Physics. 1994.113, 134–147.
[13] Ubbink, O., Issa, R. *A method for capturing sharp fluid interfaces on arbitrary meshes.* Journal of Computational Physics. 1999. 153, 26–50.
[14] Troch, P., Li, T., De Rouke, J., Ingram, D. *Wave interactionwith a sea dike using a VOF finite-volume method*. In: Chung, J., Prevosto, M.,Mizutani, N., Kim, G., Grilli, S. (Eds.), Proceedings of the13th International Offshore and Polar EngineeringConference, vol.3. International Society of Offshore and Polar Engineering. 2003. 325–332.
[15] Watanabe,Y., Saruwatari, A., Ingram, D.M. *Free-surface flows under impacting droplets* .Journal of Computational Physics.2008. 227:2344-2365
[16] Norgaliev, R.R., Dinh,T.N., Theofanous, T.G. *Sharp Treatment of Surface Tension and Viscous Stresses in Multifluid Dynamics*. Proceedings of the 17th American Institute of Aeronautics and Astronautics Computational Fluid Dynamics Conference. 2005.(Paper 2005-5349).1-20
[17] Roberto F. Ausas, Enzo A. Dari and Gustavo C. Buscaglia. *A geometric mass preserving redistancing scheme for level set function*. International Journal for Numerical Methods in Fluids. Published online : 9 FEB 2010, DOI: 10.1002/fld.2227
[18] Sethian J.A,. *Level Set Methods and Fast Marching Methods.* Publisher: Cambridge University Press. 2000: ISBN 0 521 64557 3.
[19] Online details of *"Code Saturne version 2.0.0-rc1 practical user's guide"* EDF 2010; 10-12 :http://www.code-saturne.org/

[20] Zalesak, S. (1979) *Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids*, J. Comp. Phys. 31, 335–362.
[21] Duncan, J. H. (1983) *The breaking and non-breaking wave resistance of a two-dimensional hydrofoil*. J. Fluid Mech. 126, 507–520
[22] Hino, T., Martinelli, L. & Jameson, A. (1993) *A finite volume method with unstructured grid for free-surface flow*. Proc. 6th International Conference on Num. Ship Hydrodynamics, Iowa City, IA, 173–194.
[23] Klakaa, K, Penrosea, JD, Horsleya, RR and Renilsonb, MR (2005) *Hydrodynamic tests on a fixed plate in uniform flow*, Experimental Thermal and Fluid Science 30(2) 131-139
[24] Dutta, S and Laha, K (2000) *Analysis of the small amplitude sloshing of a liquid in a rigid container of arbitrary shape using a low-order boundary element method*, International Journal for Numerical Methods in Engineering 47(9), 1633–1648.

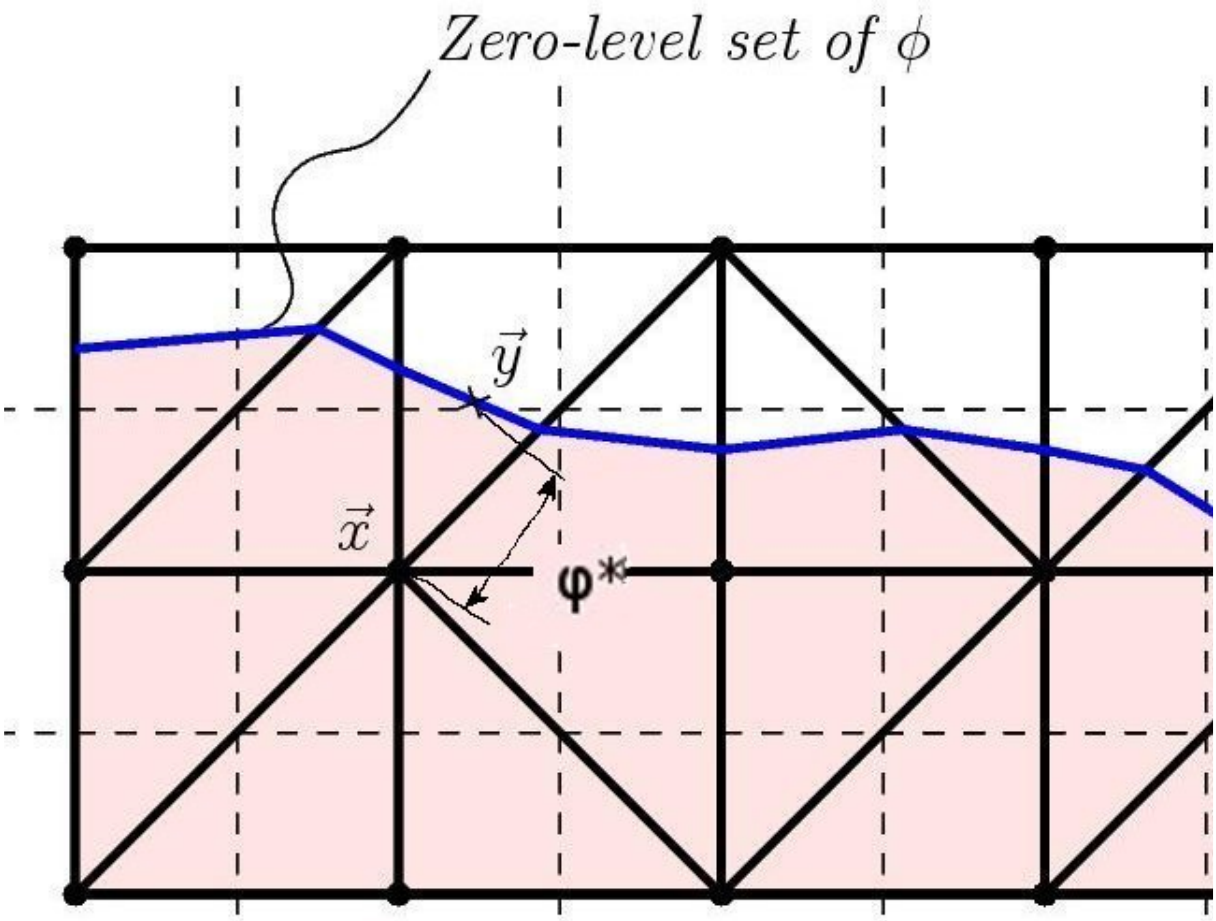

Figure 1: The definition of the signed distance function φ^* (Image based on reference [15])

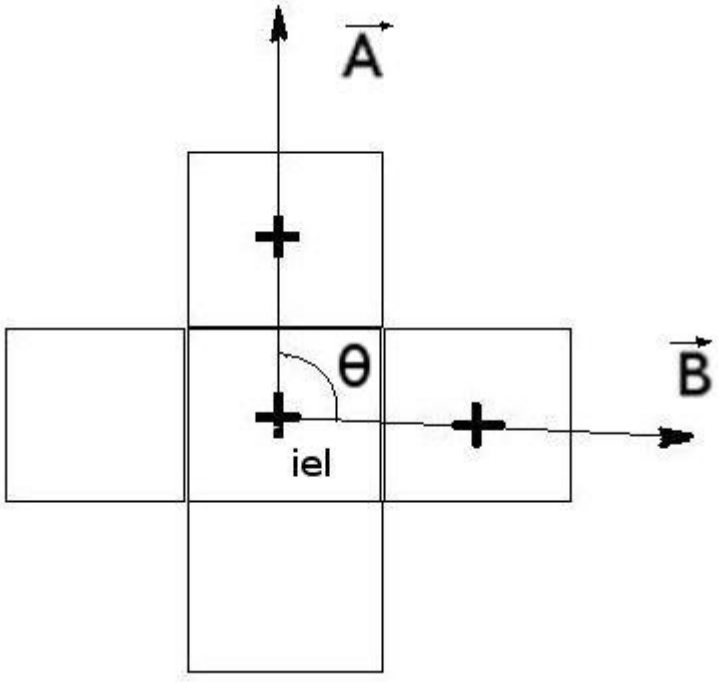

Figure 2: Sorting neighbour cells surrounding centre cell "iel" with attribute θ angle between vectors $\vec{A}$ and $\vec{B}$ which pass though the centre of these neighbour cells.

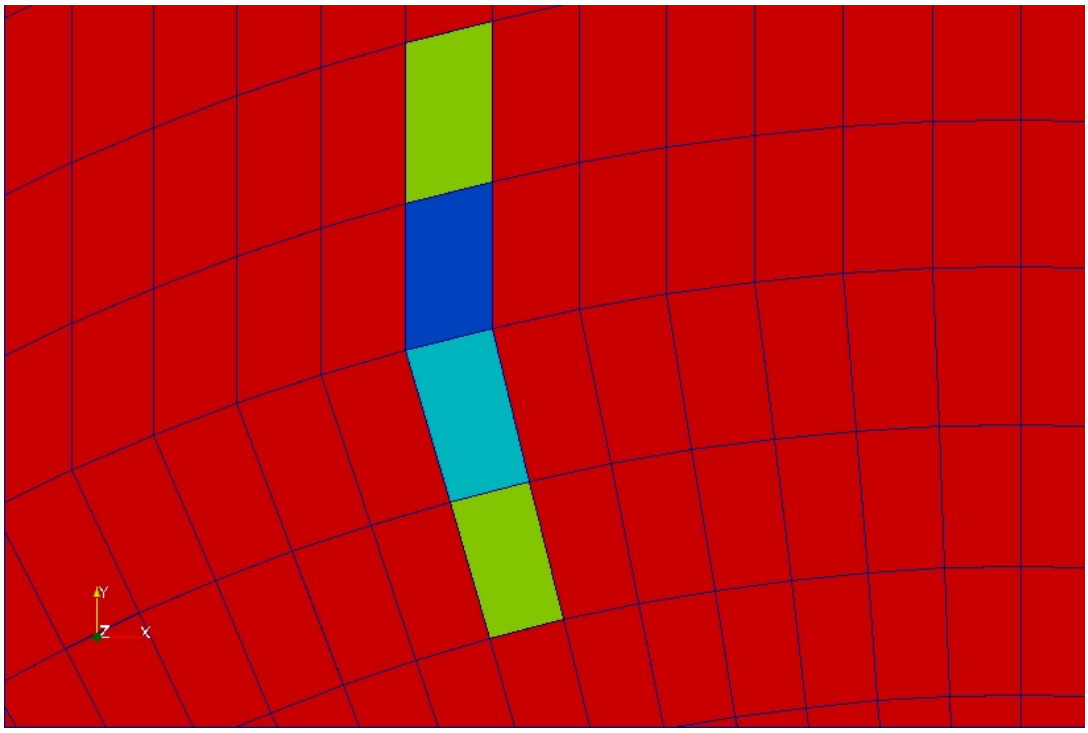

Figure 3: A narrow band node element: The light and dark blue cells form the set of primary cells, with the isocontour at the interface between light and dark cells. The light green being the halo cells.

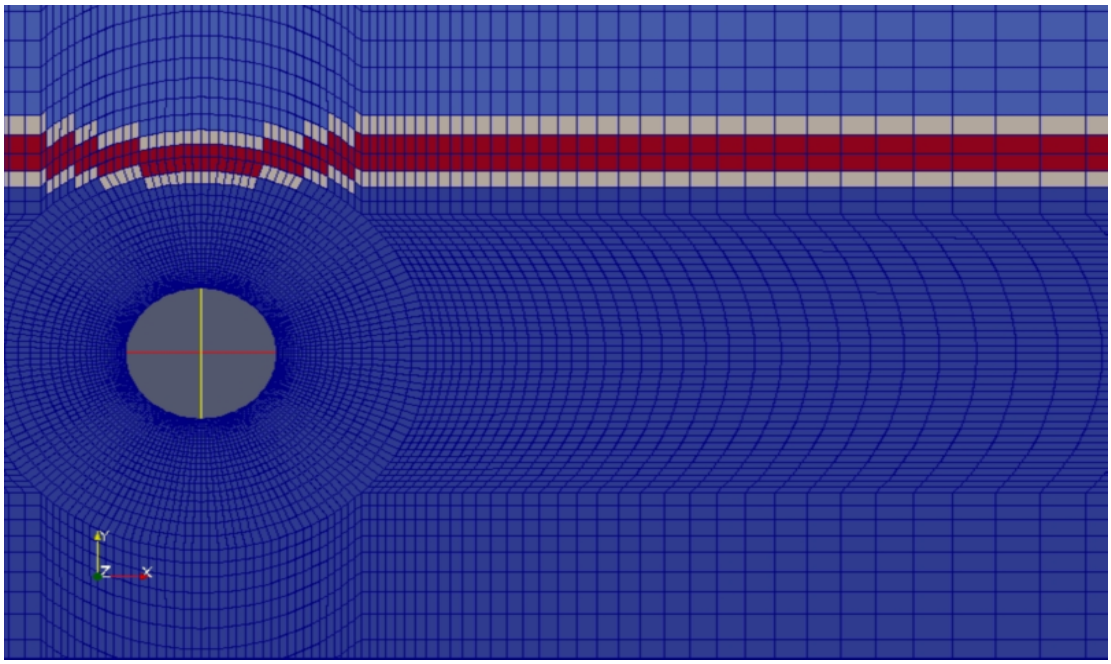

Figure 4: A narrow band: The dark red cells form the set of primary cells, with the isocontour between the two horizontal bands of dark red cells. The light red being the halo cells.

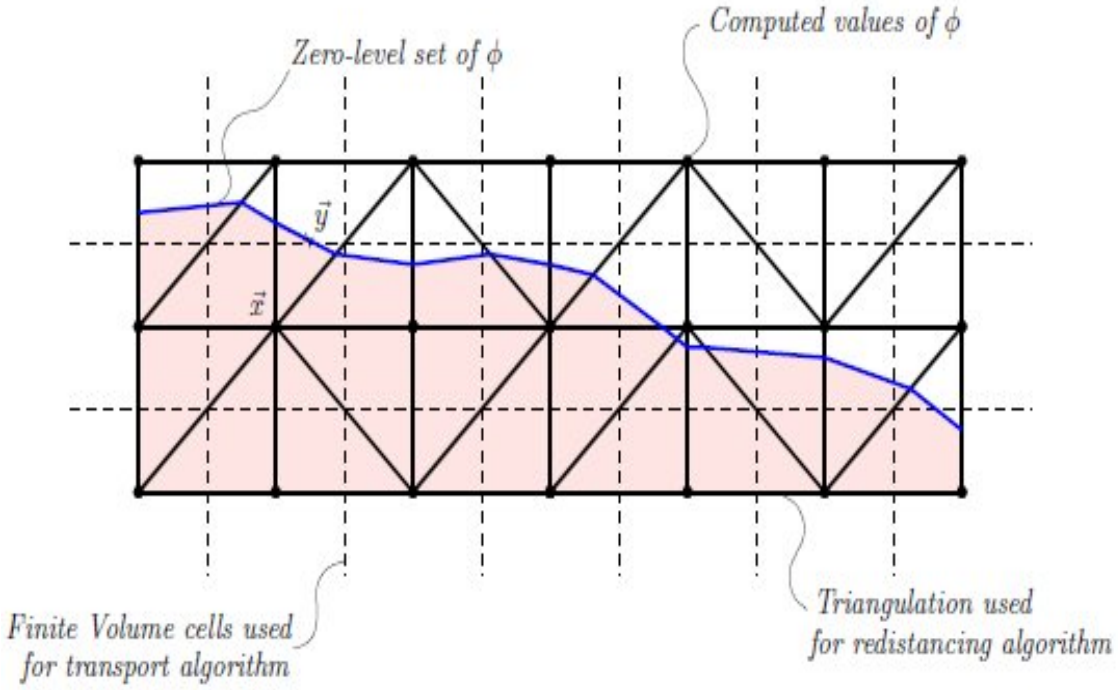

Figure 5: Schematic drawing of the finite volume discretization cells, (shown in dotted lines), and the associated triangulation, (shown in black lines lines), for the re-distancing algorithm [15]

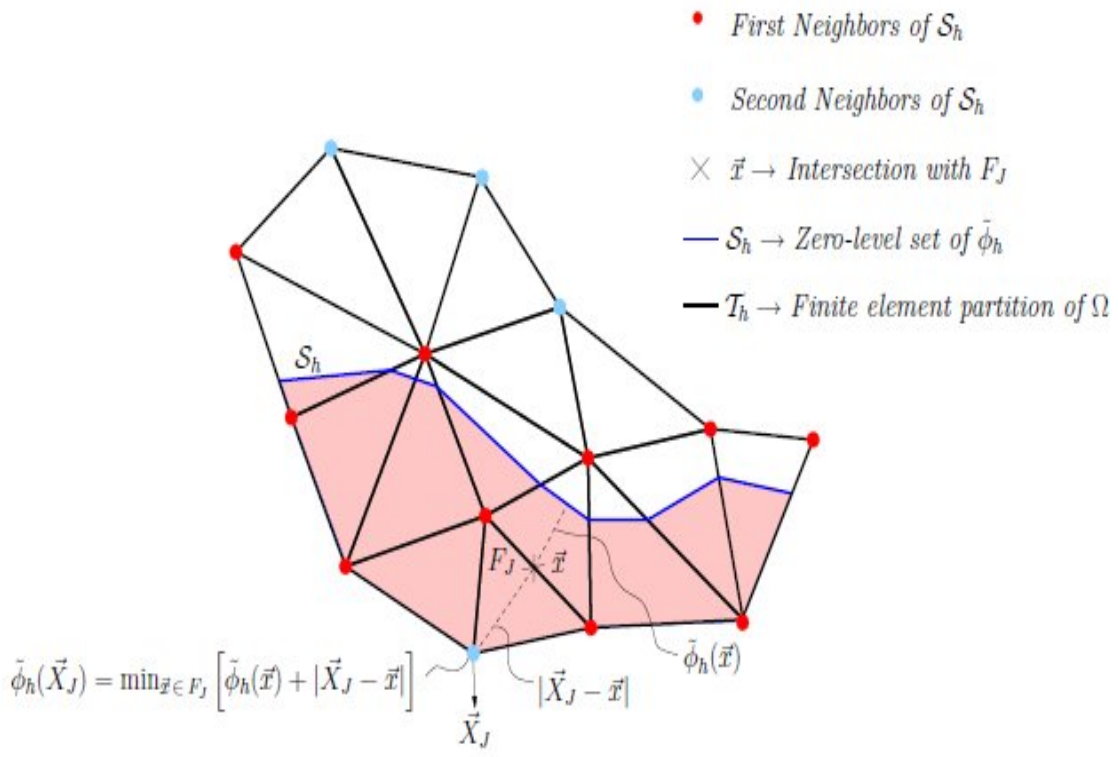

Figure 6: Schematic drawing of step 4 involving update of scalar $\varphi(\vec{X}_J)$ using φ values from first neighbour cells

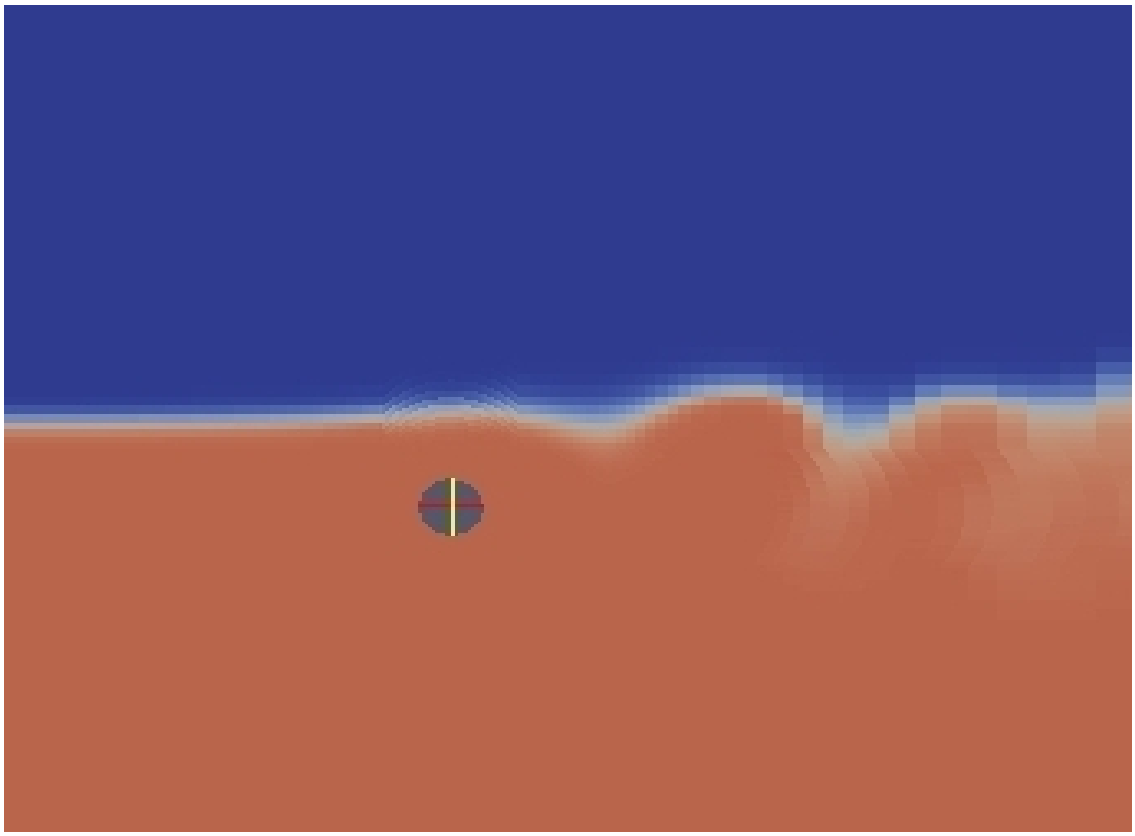

Figure 7: Changing distribution of the level-set function, ϕ, transport at a velocity, for flow passed cylinder.