27

Fluid Dynamics Without Fluids

Marco Marcon
Politecnico di Milano, Dipartimento di Elettronica e Informazione, Milano
Italy

1. Introduction

This chapter will discuss some interesting real applications where Fluid Dynamics equations found fruitful applications without dealing with "strictly speaking" fluids. In particular, thanks to the large set of analyses performed over different kinds of fluids in different operating and boundary conditions, a wide range of Computational Fluid Dynamics algorithms flourished tackling different aspects, from convergence rate, to stability according to the discretization, to multigrid and linearization problems. This robust and thorough background, both on theoretical and on practical aspects, made Computational Fluid Dynamics (CFD) appealing also to other sciences and applications where Fluid Dynamics equations, or similar equations very close to them, can be useful in describing complex phenomena not related to fluids. Some applications that will be discussed concern, e.g., Geometry of liquid snowflakes whose contour is growing steered by curvature, staring from a circle. Furthermore Image Restoration and Segmentation can also benefit from CFD since a set of evolutionary algorithms, based on level-set curvature flow equations, plays a fundamental role in steering active contours or snakes through the noise present in the image till the complete warping of the desired framed object. Also in this case advanced techniques like Ghost Fluids Method for two competing fluids dynamics can be used to separate different objects in images. Other interesting applications that will be described concern applicability of CFD to surface extraction from cloud of points. This is a common problem when complex clouds of points, representing 3D objects or scenes are obtained by laser scanners or multi-camera vision systems. These points represent unambiguous features from corners or sharp edges and the final 3D closed surface must fit on these points smoothly interpolating empty space between them. Also in this case CFD can provide useful tools to define the evolution of a 3D surface representing the border between two competing fluids, one representing the "inside" and the other the "outside" of the object itself. The two fluids evolution will stop when surface sticks on all the 3D points: the viscosity of the two fluids will control the smoothness of this surface that will wrap the cloud and turbulence is used to model injection into grooves or narrow holes. This chapter will also discuss another interesting application of CFD to robotic navigation in complex environments where we are looking for the best path, both in terms of length and distance from objects, through a set of obstacles, different terrains traversability or path slope. Also in this case an imaginary fluid with a predefined viscosity floods from the robot position through the whole environment, its front

evolution speed, accordingly to CFD, will be slower in narrow passages and, once it reaches the target, it will define the easiest way.

2. Snowflakes and phase interfaces

In many physical problems outlines of a considered fluid have different speeds at different points, and in many cases local speed is directly connected to the curvature. In the following we will show how a geometrical contour, represented by a 2D line or a 3D surface of the evolving shape changes from point to point accordingly to the local geometrical properties of the contour itself. The presented approach can be applied to a generic N-dimensional case and we can refer to the moving contour as a general hypersurface, anyway in the following, we will limit without loss of generalization our considerations to a 3D space where fluid boundary is represented by a closed surface.
In particular Sethian (1989) showed how algorithms based on direct parametrization of the surface evolution could present ambiguous and error-prone solutions due to the local error propagation while global approaches based on implicit representations can result in a much more robust solution. This kind of approaches usually relies on higher dimensional functions for which the considered surface is just represented by a level-set. The contour motion can then be described applying the proper Hamilton-Jacobi equation related to a hyperbolic conservation law to the implicit function and tracking the particular level-set representing the contour itself.
Implicit function formulation is much more robust to numerical techniques with respect to direct methods and can be implemented starting from an original closed and non-intersecting surface: two famous physical examples are the combustion model of a flame and the grow of a snowflake: Markstein (1951) assumed that thin flame fingers close to their ends are cooler than their inner part and move slower that hot nonconvex regions: the proposed evolution equation assumes that flame contour speed is inversely proportional to its curvature κ, (where κ is positive for convex regions and negative for nonconvex ones). This assumption come from the fact that in highly convex regions, flame particles collide with a high number of slower and cooler surrounding particles that slow down their motion. The symmetrical effect is present in spikes growing in snowflakes. In particular these two last examples, flames and snowflakes, represent two very good examples where simple equations can explain very well complex phenomena. For a comprehensive description of ice crystal growth we suggest Libbrecht (2005), while in the following we describe how solid-liquid interface dynamics are steered by curvature in liquid snowflakes (or Tyndall figures). We refer to some recent studies (Hennessy (2010), Hobbs (2010)) on cylindrical discs of liquid in superheated crystals of ice. In this case the snowflake is made of water inside ice and its geometrical shape evolution is determined by boundary condition between solid and liquid phase. In Fig. 1 there is a typical snapshot of a Liquid snowflake.

2.1 Liquid snowflakes surface evolution

The typical modelization of snowflakes surface growing is the Gibbs-Thomson equation where the interface temperature T_I, between the solid and the liquid part is ruled by the equation:

$$T_I(\kappa) = T_M\left(1 - \kappa\frac{\gamma}{\rho L}\right) \tag{1}$$

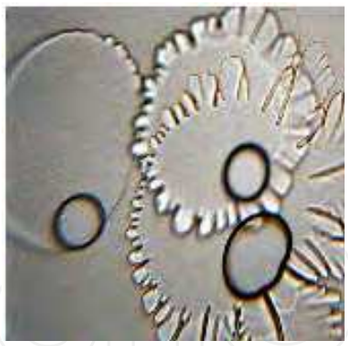


Fig. 1. A typical Tyndall figure (Liquid snowflake)

Where T_M is the equilibrium temperature of a planar solid-liquid interface (273.15K for water), κ is the curvature, γ is the surface energy per unit area for the solid-liquid interface, ρ is the density of a specific phase but in this case we are assuming equal density both for ice and water and L is the latent heat of fusion. The steering interface force is then due to the pressure variation across the interface and depends from curvature:

$$p_s - p_l = \rho \left(s_l - s_s\right)\left(T_I - T_M\right) = \rho L \left(\frac{T_I\left(\kappa\right)}{T_M} - 1\right) \tag{2}$$

Where the subscript l and s refer to liquid and solid state respectively and s indicates the entropy per unit mass, for further details refer to (Hills & Roberts (1993)).
A further improvement in the Gibbs-Thomson model considers surface energy anisotropy due to crystal ice structure (six fold symmetry), this will add a dependence of the surface energy γ from the relative orientation of the surface with respect to the underlying crystal. Accordingly to (Davis (2001)) a six fold symmetrical crystal will have a:

$$\gamma_6 = \gamma_0 \left[1 + \frac{\sigma_n}{n^2 - 1}\cos\left(n\theta\right)\right]\Bigg|_{n=6} \tag{3}$$

Where σ_n represents the anisotropy degree and the complete interface temperature will be, accordingly to (Pimpinelli & Villain (1999));

$$T_I\left(\kappa\right) = T_M\left(1 - \kappa\frac{\gamma_6 + \frac{d^2\gamma_6}{d\theta^2}}{\rho L}\right) \tag{4}$$

In particular, accordingly to equations 3 and 4, whenever $|\sigma_n| < 1$ we are in a week anisotropy condition, i.e. the surface energy γ is always positive, and, at equilibrium, the surface will resemble a smooth line, while, for anisotropy values greater that one, as for the snowflakes water example, the thermodynamic equilibrium implies a closed polygonal made of straight line segments (Dobrushin et al. (1992)).
Once the Temperature of the interface is determined the shape of the boundary between the liquid and solid part and its time evolution can be recovered from the energy conservation, in particular, in order for the ice-water interface to advance inside the solid part, two energy contributions must be provided: one for solid melting and one for interface stretching. The melting energy that must be provided per unit of time for a surface A that is moving inside

the solid at a normal speed v_n is $l_{mel} = \rho L A v_n$ while the energy contribution per unit of time due to surface stretching is $l_{sur} = \gamma \frac{dA}{dt}$; which, accordingly to the differential geometry can be written as:

$$l_{sur} = \gamma \frac{dA}{dt} = A\gamma\kappa v_n \tag{5}$$

The surface advancement is steered by two opposite heat diffusive fluxes: one inside the liquid part and one inside the solid part; respectively: $-k_l \nabla T_l$ and $k_s \nabla T_s$ where k indicates the thermal conductivity of the considered phase.
The equilibrium of energy equations per area and time unit can be written as:

$$\frac{l_{mel}}{A} + \frac{l_{sur}}{A} = (-k_l \nabla T_l + k_s \nabla T_s) \cdot \mathbf{n} \tag{6}$$

i.e.:

$$(\rho L + \gamma\kappa)\, v_n = (-k_l \nabla T_l + k_s \nabla T_s) \cdot \mathbf{n} \tag{7}$$

Where $\mathbf{n}$ is surface normal oriented from the liquid part towards the solid one; this equation is called the Stefan condition. The numerical solution for the liquid-solid interface evolution, accordingly to the Stefan condition, requires some particular precaution due to accuracy that must be placed in surface tracking. Since the analyzed problem is far from thermodynamic equilibrium due to the superheated temperature of the solid, some typical approaches, e.g. the Enthalpy model (Shastri & Allen (1998)), are not accurate. In fact, Enthalpy approach, even if requires only the evaluation of a single parabolic PDE at each time step, assumes that the interface is within a mushy solid-liquid region and the evolving interface is just approximated. A possible extension of Enthalpy approach to account for superheated solid state could be phase-field models; anyway an accurate interface tracking requires a very small grid spacing resulting in a high computational cost (Biben et al. (2005)). One of the method that in recent years is getting growing attention is Level-set: in the following of this chapter we will show it versatility and accuracy in many CFD applications and in liquid snowflakes shape definition.

3. Level-set methods

For most of real CFD systems symbolic formulation that easily yield to symbolic mathematical solution is unfeasible, and the goal becomes to find the system modelization with the best trade-off between accuracy and computational complexity accounting for local, global and independent properties of the moving surface. In particular we will refer to "local" as properties associated to local surface features (e.g. curvature, local density or temperature), "global" as properties associated to overall surface shape or extension, while independent properties are not linked to the surface itself, like a flow below the surface. The Level-set method aims to track a thin propagating interface over time, it outperforms many other approaches handling topological complexities such as corners and cusps, and in handling complexities in the evolving interface such as entropy conditions and weak solutions. A detailed description of level-set can be found in (Sethian (1999)).
The general idea is to define a function $\phi(p,t)$ over the whole space-time domain, where p indicates a point in the space and t represents the time variable. The initial surface is the zero level-set of an implicit function for $t = 0$: i.e. the starting surface is the set of points p so that $\phi(p,t) = 0$ the value of the ϕ function for non-interface points is, the signed distance from the surface where 'signed' means that different signs indicate internal or external position with

respect to the closed surface. The discrete formulation requires that the value of ϕ is initially evaluated over all the mesh points and the steering equations would require the update of each point at each time step. This computational complexity of order $O(n^3)$ at each time step could be unacceptable in many practical cases so many faster update solutions have been proposed; the most common are:

- Narrow band: the updated mesh points are only those in a small stripe surrounding the interface, this will reduce computation to $O(n^2)$ but as soon as the interface is approaching the stripe boundary all values must be recomputed (more details in (Adalsteinsson & Sethain (1995))
- Octree: relevant mesh points and high detailed surface regions can be divided in sub-grids, e.g. a cube can be split in eight sub-cubes (the method name come from here) and the procedure can be iterated for higher precision. More details can be found in (Losasso et al. (2004))
- Sparse block grid: The whole domain is subdivided in blocks representing clusters of the original grid points (they can be overlapped or not) then the update is evaluated only on blocks containing the surface narrow band. (Further details in (Bridson (2003))

3.1 Level-set evolution

Accordingly to the previous description the set of points belonging to the surface will satisfy the zero level-set condition:$\phi\left(\mathbf{p}(t),t\right)=0$ where $\mathbf{p}(t)$ represents the path of a surface point. The first temporal derivative will become:

$$\frac{d\phi\left(\mathbf{p}(t),t\right)}{dt}=\frac{\partial\phi}{\partial t}+\nabla\phi\cdot\frac{d\mathbf{p}(t)}{dt}=0 \tag{8}$$

Considering the normal unitary vector to a generic point of ϕ for a generic level-set (contour), $\mathbf{n}=\frac{\nabla\phi}{|\nabla\phi|}$ the speed function v along this direction is then:

$$v\left(t\right)=\frac{d\mathbf{p}\left(t\right)}{dt}\cdot\mathbf{n} \tag{9}$$

obtaining:

$$\frac{d\phi\left(\mathbf{p}(t),t\right)}{dt}=\frac{\partial\phi}{\partial t}+v\left(t\right)|\nabla\phi|=0 \tag{10}$$

In the cases considered in the following and in many physical cases the interface evolution is ruled by a Hamilton-Jacobi equation and the local velocity can be assumed proportional to the local curvature of the surface itself:

$$v=v_0-\epsilon\kappa \tag{11}$$

where ϵ is a constant and

$$\kappa=\frac{\phi_{xx}\phi_y^2-2\phi_y\phi_x\phi_{xy}+\phi_{yy}\phi_x^2}{\left(\phi_x^2+\phi_y^2\right)^{3/2}} \tag{12}$$

.

We can rewrite ϕ as follows:

$$\frac{\partial\phi}{\partial t}=-v_0\left|\nabla\phi\right|+\epsilon\kappa\left|\nabla\phi\right| \tag{13}$$

3.2 Liquid snowflakes surface evolution

Assuming that the ice is radiated with an external electromagnetic radiation normal to its surface (a focused beam with a gaussian shape) the heat will follow an exponential decay penetration, accordingly to the following heat propagation equation:

$$I(r,z) = I_0 e^{-(r/r_b)^2 - \alpha z} \tag{14}$$

where I_0 is the central beam intensity, r_b is its standard deviation while r is the in-plane coordinate for the ice surface. Furthermore z is the coordinate orthogonal to the ice surface and α is a penetration coefficient. The formation of the liquid snowflake is then in the in-plane direction. A level-set function can the be defined where ϕ is the normalized temperature that follows heat PDE:

$$\begin{cases} \frac{\partial \phi}{\partial t} = \nabla^2 \phi + e^{-(r/\beta)^2} & \textit{for the liquid} \\ c_{pr} \frac{\partial \phi}{\partial t} = k_r \nabla^2 \phi + \alpha_r e^{-(r/\beta)^2} & \textit{for the solid} \end{cases} \tag{15}$$

where $c_{pr} = \frac{c_{ps}}{c_{pl}}$ is the relative measure of heat capacities, $k_r = \frac{k_s}{k_l}$ is the relative thermal conductivity and $\alpha_r = \frac{\alpha_s}{\alpha_l}$ is the relative absorbtion coefficient.
The liquid/solid interface is then associated to the zero level-set in a sort of rescaled Celsius scale. Experimental observations show that liquid snow flakes often begin as circular discs which then grow outwards in a radially symmetric manner. However, after a certain amount of time, the interface becomes unstable and small, sinusoidal perturbations with a well defined wave-number appear. In Fig. 2 it is possible to see the growing of a liquid snowflake starting from a circle and then evolving with equally spaced rippling.

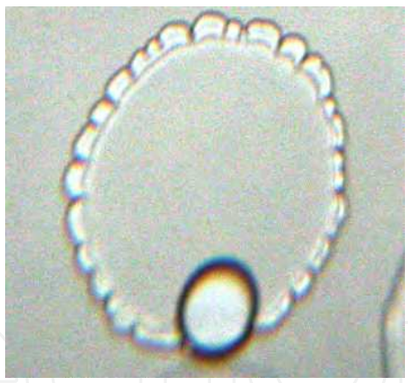


Fig. 2. Liquid snow flakes formation starting from a circle and the evolving, accordingly to linear instabilities with sinusoidal perturbations

The eq. 15, accordingly to the problem symmetry can be written in polar coordinates for the quasi-steadily condition:

$$\begin{cases} \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \phi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 \phi}{\partial \theta^2} + e^{-(r/\beta)^2} = 0 & \textit{for the liquid} \\ \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \phi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 \phi}{\partial \theta^2} + \frac{\alpha_r}{k_r} e^{-(r/\beta)^2} = 0 & \textit{for the solid} \end{cases} \tag{16}$$

Even small linear perturbations must satisfy the equations above and an integer number of oscillations (first order approximation) will constitute the zero level-set. Assuming a starting

contour equal to a circle of radius R the curvature will be equal to $\kappa = -\frac{1}{R}$ The time evolution of the level-set accordingly to the aforemention equations is represented in Fig. 3 where a very good agreement of the zero level-set with real data is reached. For further details about analytical integration of eq. 16 refer to (Hennessy (2010)).

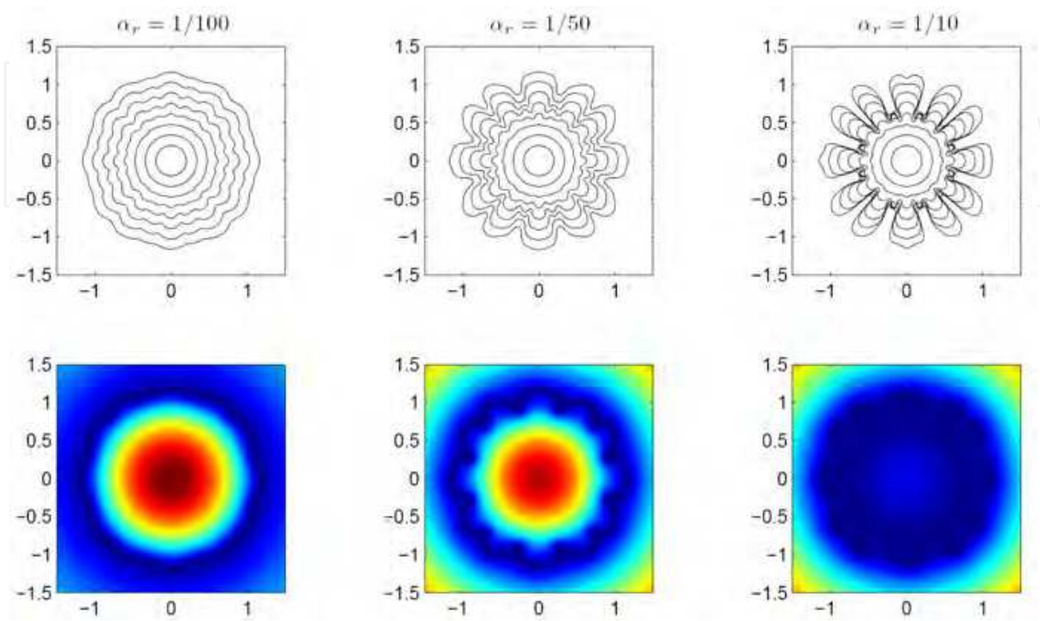


Fig. 3. on the top raw: different snapshots of the zero level set evolution varying the relative absorbition α_r; instead on the bottom line different level set values (different temperatures) are presented for the larger curve of the raw above

4. Motion planning for traveling robots in complex environments

Fluid dynamics and Level-set formulation can also be fruitfully adopted in Motion planning Algorithms. In particular the flexibility of the presented technique for time optimal motion planning with moving obstacles can be easily integrated with further constraints like different terrains traversability, path narrowing or fuel economy. Many classical techniques are present in literature for shortest path search with stationary obstacles, a good survey can be found in Latombe (1991). The space-time representation of obstacles moving in a 2D environment, assuming that one axis will be the time variable will be a solid structure. In Fig. 4 there is the representation of a circular obstacle moving in the$x$-y plane along x direction with an abrupt slow down.

As shown in Kimmel et al. (1998) the search for the time-optimal path within the aforementioned 3D time-space can be reduced to the search in a reduced set of 2D regions forming a Multivalued Distance Map (MDM). A Distance Map from a point **p** is a function that assigns to each point **q** in the considered domain a value corresponding to the "minimal length" (geodesic). The simplest approach uses a contour with a constant speed propagating from a small circle around **p** and the distance assigned to each crossed point **q** is proportional to the propagation time. If obstacles are moving the optimal path will be the minimal geodesic only when it does not cross any moving obstacles in the scene. For all the other cases a

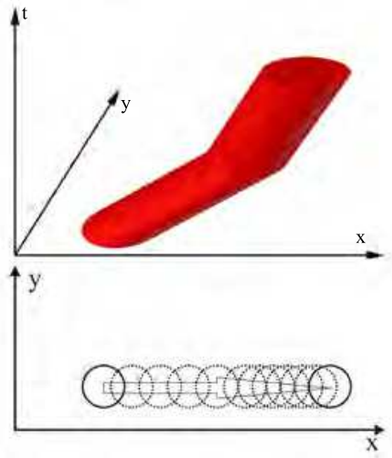


Fig. 4. a space-time representation for a circular obstacle moving in the *x-y* plane along the x direction that slow down its speed

more sophisticated approach is required: MDM will keep track of multiple crossing of the propagating contour when moving obstacles are present: in particular $MDM_p(\mathbf{q},\mathbf{1})$ will indicate the first time at which the point $\mathbf{q}$ is reached from $\mathbf{p}$ while $MDM_p(\mathbf{q},\mathbf{2})$ will indicate the second time at which, accordingly to the moving objects the propagating contour can reach $\mathbf{q}$, and so on. We will call $\mathcal{N}(\mathbf{q})$ the number of times that the propagating front will cross that point. Once the MDM is built for all points in the domain, the best path is obtained using a back-track procedure.

Consider now, given the source point $\mathbf{p}$, a continuous surface $\mathcal{S}$ in the space-time domain, centered in $\mathbf{p}$ indicating the wavefront contour as a function of time: $\mathcal{S} = \{(x,y,t(x,y))\}$, where a contour at a specific geodesic distance $\bar{d}$, $\alpha(\bar{d})$ is given by:

$$\alpha(\bar{d}) = \{\mathbf{q} | d_g(\mathbf{p},\mathbf{q}) = \bar{d}\} \tag{17}$$

where $d_g(\mathbf{p},\mathbf{q})$ is the minimal geodesic distance along the surface $\mathcal{S}$. Accordingly to Kimmel et al. (1995), due to the continuity and smoothness of $\mathcal{S}$ we can define a curvilinear abscissa u for $\alpha(u,\bar{d})$.

If we assume that the motion velocity is constant, V without loss of generality we can also assume that $V = 1$ then the geodesic distance $d = d_g(\mathbf{p},\mathbf{q})$ coincides with time t and $\alpha(u,d) = \alpha(u,t)$ the equal distance contour evolution rule is:

$$\frac{\partial \alpha(u,t)}{\partial t} = \mathbf{N} \times \mathbf{T} \tag{18}$$

where $\mathbf{T}$ is the unit vector tangent to α while $\mathbf{N}$ is the unit vector normal to the surface $\mathcal{S}$.

To avoid point discontinuities the starting contour $\alpha(u,0)$ is a small circle around the point $\mathbf{p}$ and the projection of equation 18 on the *x-y* plane will be the velocity vector orthogonal to the actual front, V_N:

$$V_N = \mathcal{P}(\mathbf{N} \times \mathbf{T}) \cdot \mathbf{n} \tag{19}$$

where $\mathcal{P}()$ is the projector on the x-y plane and $\mathbf{n} = (n_x, n_y)$ is the unit normal to the planar curve in the x-y plane.

The evolution of the planar contour will then be:

$$\frac{\partial \mathcal{P}\left(\alpha\left(t\right)\right)}{\partial t} = -V_N \mathbf{n} \tag{20}$$

In Fig. 5 there is a graphical representation of the aforementioned quantities.

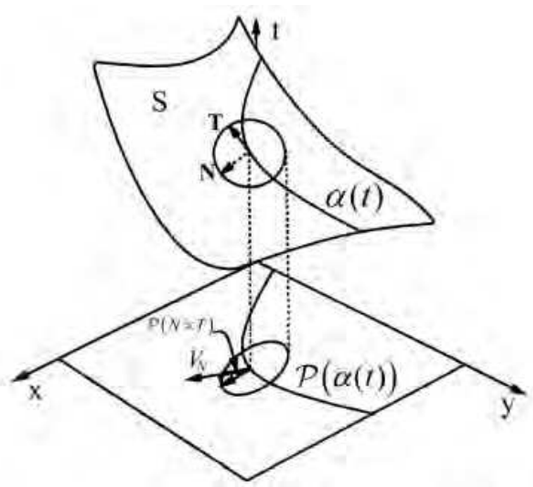


Fig. 5. The time evolution of variable α, and the definition of the moving front velocity V_N

Calling $p = \frac{\partial t(x,y)}{\partial x}$ and $q = \frac{\partial t(x,y)}{\partial y}$, accordingly to (Kimmel et al. (1995)) we can write:

$$\mathbf{N} = \frac{(-p,-q,1)}{\sqrt{1+p^2+q^2}} \tag{21}$$

and:

$$V_N = \sqrt{\frac{\left(1+q^2\right)n_x^2 + \left(1+p^2\right)n_y^2 - 2pqn_xn_y}{1+p^2+q^2}} \tag{22}$$

Since the minimal geodesics are perpendicular to the equal distance contours on the surface, we can track the optimal trajectory starting from the destination point and, considering the equal distance contour α crossing it we have to move backward thanks to the orthogonality between the optimal path and the equal distance contour:

$$\eta = -\left(\mathbf{N} \times \mathbf{T}\right) \tag{23}$$

Using an arc length parametrization for the curve α we can write $r = \frac{\partial MDM_p}{\partial x}$, $l = \frac{\partial MDM_p}{\partial y}$ and using a normalization function

$$\psi\left(p,q,r,l\right) = \frac{p^2+q^2+1}{\sqrt{l^2\left(1+p^2\right)+r^2\left(1+q^2\right)-2pqrl}} \tag{24}$$

and the backtracking rule will then consist in tracking, from destination point $\mathbf{q}$ over the surface $\mathcal{S}$, the best path β so that:

$$\frac{\partial \beta}{\partial t} = \eta = \psi\left(p,q,r,l\right)\left[r\left(1+q^2\right) - pql, l\left(1+p^2\right) - pqr, pr+ql\right] \tag{25}$$

If there are moving obstacles along the path the MDM_p becomes multivalued and it could be possible that some points change their index along the optimal path, to account for this all distance values must be evaluated at a candidate point.

4.1 The Level-set approach

Moving obstacles introduce new boundary constraints and dealing with them could be very complex following the previously describer approach, a simpler way could be using the level-set approach where the projection of α on the x-y plane is the zero level set $\phi = 0$ and ϕ will be negative in the interior and positive in the exterior of the zero level set.

$$\phi\left(\mathcal{P}\left(\alpha\left(t\right)\right), t\right) = 0 \tag{26}$$

A graphical representation of this process can be seen in Fig. 6.
The next step requires the definition of the evolution law for ϕ, that, accordingly to the chain rule can be written as:

$$\nabla\phi\left(\mathcal{P}\left(\alpha\left(t\right)\right), t\right)\frac{\partial\mathcal{P}\left(\alpha\left(t\right)\right)}{\partial t} + \frac{\partial\phi\left(\mathcal{P}\left(\alpha\left(t\right)\right), t\right)}{\partial t} = 0 \tag{27}$$

Furthermore, accordingly to this level-set formulation, the in-plane normal vector can be written as $\mathbf{n} = \frac{\nabla\phi}{\|\nabla\phi\|}$. This relation together with the Lagrangian evolution equation:

$$\frac{\partial\phi\left(\mathcal{P}\left(\alpha\left(t\right)\right), t\right)}{\partial t} = -V_N\left\|\nabla\phi\right\| = \sqrt{\frac{\left(1+q^2\right)\phi_x^2 + \left(1+p^2\right)\phi_y^2 - 2pq\phi_x\phi_y}{1+p^2+q^2}} \tag{28}$$

This is the Eulerial formulation for curve evolution.

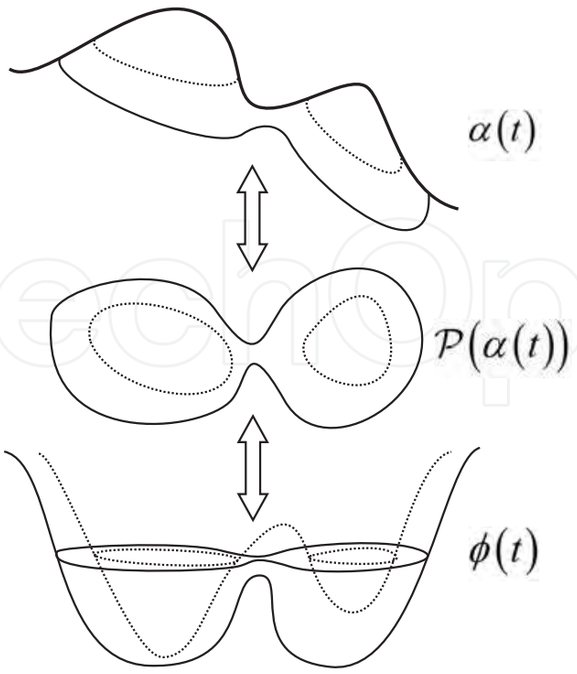


Fig. 6. Implicit representation of the projection of the α function through the level-set.

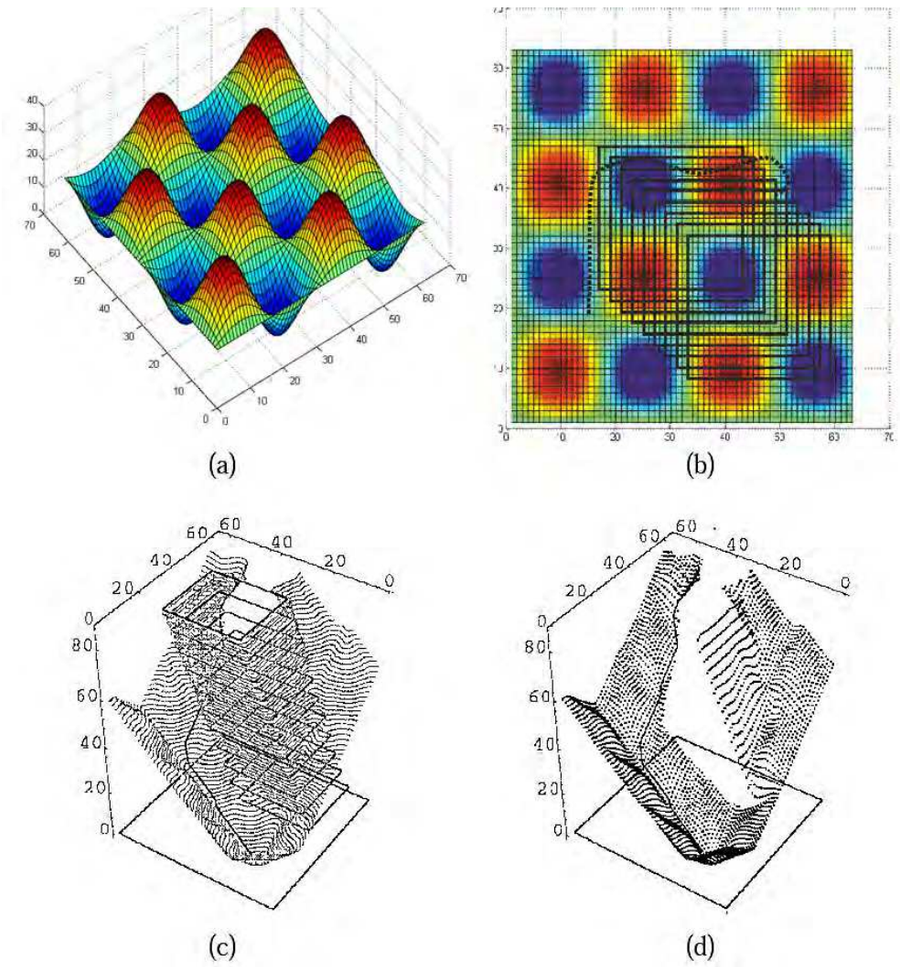


Fig. 7. (a) the terrain definition, (b) a squared object moves across the scene. (c) $\alpha(u,t)$ for different values of t representation (d)Back tracking from the final point toward the starting one along the estimated geodesic path

The steps for the best path definition can then be summarized as:

1. Evolve the PDE (eq.28) till the destination point is reached.
2. back-track along the optimal path till the origin.

In Fig. 7 there is the representation of level-set evolution, accordingly to moving objects, in an "egg-box" terrain where a square is moving across the scene.
Terrain traversability can be easily integrated in the presented approach since it is possible to define a map $F(x,y)$ describing the terrain traversability according to continuous values from 1 (optimal traversability) to 0 (impassable) and the optimal path can be obtained just changing eq. 18 into

$$\frac{\partial \mathcal{P}(\alpha(t))}{\partial t} = -F(x,y)\, V_N \mathbf{n} \tag{29}$$

Further velocity dependencies, e.g. from the vehicle weight reduction due to fuel combustion, can be accounted introducing a time dependance in V_N. More complex conditions can be

easily added just acting on V_N, e.g. if the Robot has a specific shape the velocity can reflect the ability to walk through a narrow path or to turn at a specific point. In some cases the $F(x, y)$ correcting function is called viscosity for it similarity to fluid property that can prevent or slow down motion in particular circumstances.

5. Surface reconstruction from cloud of points

The temporal evolution of a volumetric implicit function can also be fruitfully applied to reconstruct 3D surfaces form large set of unorganized sample points. The evolving front can be thought as the surface that separates two different fluids obeying specific fluid dynamics laws. One remarkable feature of this approach is its ability to model complex topologies using a set of intuitive tools derived from fluid physics: Global and local surface descriptors are used allowing the parallelization of the algorithm on different object parts working with different resolutions and accuracies. The problem of building surfaces from unorganized sets of 3D points has recently gained a great deal of attention. In fact, in addition to being an interesting problem of topology extraction from geometric information, its applications are becoming more and more numerous. For example, the acquisition of large sets of 3D points is becoming easier and more affordable using, for example, 3D-scanners (*Registration and fusion of intensity and range data for 3d modelling of real world scenes* (2003)). The presented approach is particularly important in applications where objects are better described by their external surface rather than by simple unorganized data (clouds of points, data slices, etc.). For example, in medical applications based on Tomography scans or NMRs it is often necessary to visualize some specific tissues such as the external surface of an organ starting from the acquired 3D points. This can be achieved by selecting the points that belong to a specific class (organ boundary, tissue, etc.) and then generating the surface from their interpolation. In most cases the definition of this surface is an ill-posed problem as there is no unique way to connect points of a dataset into a surface, therefore it is often necessary to introduce constraints for globally or locally controlling the surface behavior. As a matter of fact, the resulting surface often turns out to exhibit a complex topology due to noise in the acquired data or ambiguities in the case of non-convex objects Hoppe (1994). In order to overcome such problems, surface wrapping algorithms need to incorporate specific constraints on the quality of the data fitting (surface closeness to the acquired points), on the maximum surface curvature and roughness, on the number of resulting triangles, etc. The existing surface reconstruction methods can be classified into two broad categories: the former describes the surface as an implicit function while the latter describes it in an explicit form. Explicit (boundary) representations describe the surface in terms of point connections, and traditional approaches are based on Delaunay triangulation, Voronoi diagrams (Amenta et al. (1998)) or NURBS (Piegel & Tiller (1996)).
Here we will show how Level-set technique together with Volume of Fluid end Fluid Dynamics equations can be fruitfully applied to obtain a wrapping surface with intuitive parameters tuning. To evaluate quality of the final reconstruction an important parameter that can also be used as stopping condition for the surface evolution is an energy functional defined as:

$$E(\Gamma) = \left[\int_{\Gamma} d^p(\mathbf{x})\, ds \right]^{1/p} \quad , 1 \leqslant p \leqslant \infty \tag{30}$$

where Γ represents the surface, $\mathbf{x}$ is the set of 3D points, d is the euclidean distance between the point $\mathbf{x}$ and the closest point on the surface Γ and $1 \leq p \leq \infty$ is the order of the functional distance: for $p = 2$ it gives a root mean square error while for $p = 1$ we just focus on the point with the maximum distance from Γ. looking for the surface that minimizes the functional (30) is similar to enveloping the data set with a membrane with certain elastic properties and have this membrane to evolve in time until it comes to rest. This time evolution paradigm can be described as:

$$\frac{\partial \Gamma}{\partial t} = -\left[\int_{\Gamma} d^{p}\left(\mathbf{x}\right) ds\right]^{\frac{1}{p}-1} d^{p-1}\left(\mathbf{x}\right)\left[\nabla d\left(\mathbf{x}\right)\cdot\mathbf{n} + \frac{1}{p} d\left(\mathbf{x}\right)\kappa\right]\mathbf{n} \tag{31}$$

Euler-Lagrange equation can the be used to find its minimum:

$$d^{p-1}\left(\mathbf{x}\right)\left[\nabla d\left(\mathbf{x}\right)\cdot\mathbf{n} + \frac{1}{p} d\left(\mathbf{x}\right)\kappa\right] \tag{32}$$

where $\mathbf{n}$ points accordingly to the evolving point direction, κ indicates the mean curvature, $\nabla d\left(\mathbf{x}\right)\cdot\mathbf{n}$ represents the surface attraction term and $d\left(\mathbf{x}\right)\kappa$ is the surface tension. An important consideration is that function $d\left(\mathbf{x}\right)$ will automatically impose that the final surface is more flexible where the 3D points dataset is denser and more rigid in low density regions: this will allow smoother regions where we have less samples and accurate fitting in highly sampled regions. Fluids, with their properties like surface tension and viscosity represent an optimal modelization for the warping surface providing us with a flexible tool for this goal. Also in this case we present a Level-set approach, in particular we will adopt an implicit function ϕ over the spatial 3D domain so that $\phi\left(\mathbf{x}, t\right) = 0$ represent the warping surface $\Gamma\left(t\right)$ at time t, for the initialization we use $\phi\left(\mathbf{x}, t\right)|t = 0 = D$ where D is the signed distance of $\mathbf{x}$ from $\Gamma\left(t\right)$ (positive values indicate external points while negative ones represent internal points). One possible approach to make surface Γ to converge towards the cloud of points is, accordingly to Zhao et al. (2001), to use the fluids convection defining a velocity field $\mathbf{v}\left(\mathbf{x}\right)$ accordingly to:

$$\frac{\partial \Gamma\left(t\right)}{\partial t} = \mathbf{v}\left(\Gamma\left(t\right)\right) \tag{33}$$

In particular we adopted, as a steering potential field, the distance function $d\left(\mathbf{x}\right)$ so that the velocity field $\mathbf{v}$ is oriented towards the minima of the potential field: $\mathbf{v} = -\nabla d\left(\mathbf{x}\right)$. So, accordingly to eq. 8, the level-set evolution in this case can be defined as:

$$\frac{\partial \phi}{\partial t} = \nabla d\left(\mathbf{x}\right)\cdot\nabla\phi \tag{34}$$

A major computational drawback comes from the fact that the ϕ function at each time iteration becomes more and more compressed presenting a growing slope across the zero level-set which could introduce numerical imprecisions. To solve this problem some solutions are present in literature, for example a level-set re-initialization in a region around the zero level: this is particularly useful for Fast Marching methods which, as described in section 3 for the Narrow Band approach Sethian (1999). Other approaches, instead, suggest a different implementation for the Hamilton Jacobi PDE like Gomes & Faugeras (1999). In particular in

Marcon et al. (2008) a quite different interpretation for the level-set function is presented that aims to conciliate Level-set upgrade, narrow band approach, discretization issues and Volume of Fluid (VoF) method: in particular, the VoF method (see Noh & Woodward (1976)) has been formulated in a variety of forms and re-introduced under different names such as the "cell method" and the "partial fractions method". The key idea behind this technique is to define a fixed computational grid and assign to each grid cell a value that describes the relative proportions of two materials contained in that cell. Our particular modeling metaphor is based on two immiscible fluids of opposite densities $\rho = 1$ and $\rho = -1$, which identify the surfaceŠs outside and inside, respectively. More specifically, cells completely filled with outer fluid take on the value +1 while cells that are only filled with inner fluid will take on the value -1. When filled with a mix of both fluids, the cell is assigned an intermediate value in the $]-1,1[$ range, which measures the relative density of the mix (i.e. a value of zero indicates the same amount, half and half of the two fluids inside the cell). As a first step we define a bounding box that completely encloses the point cloud. We assume that this box is completely filled with inner fluid and that the whole space outside of the box is filled with outer fluid. We define the rules of evolution so that both fluids are set in motion by the presence of data points, which act as attractors. This fluid migration takes place in compliance with laws of conservation. According to such rules, the front starts propagating inward (towards the cloud of points) and stops only when the inner fluid is confined "inside" the cloud of points and the interface between the two fluids wraps the point-cloud. At each iteration step we will replace the value of points on the domain boundary $\mathcal{B}$ with 1 (this is equivalent to place outer fluid sources on the domain boundary), the cells with values greater than 1 will be clipped with value '1' while cells with values lower that -1 will be clipped to -1: this will be equivalent to remove excess fluids from cells without any need to introduce further fluid sources or drains. Two main contribution to fluid motion are considered: convective and diffusive: the first one is important for steering the zero level-set toward the cloud of points while the second has an important role in smoothing evolving front. Accordingly to the Gauss theorem when no sources or drains are present the fluid density will follow eq. (35):

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (\mathbf{v}\rho) + \nabla \cdot (\xi \nabla \rho) \tag{35}$$

where ρ indicates the fluid density, $\mathbf{v}\rho$ is the convective term indicating the amount of fluid per volume unit that is transported by the velocity field while $\xi \nabla \rho$ represents the diffusive part where ξ is the diffusive term (further details in (Gueyffier et al. (1999))). In this approach a point will be assigned a velocity that is proportional to the vector $\mathbf{v}$ that joins that point to the closest data point. This way data points act as attractors for both internal and external fluids and, therefore, for the evolving front: each sample point represents a central vector field. Even if this attraction field has no physical equivalent, it is physically consistent thanks to its irrotational characteristic (see Marcon et al. (2008) for further details). This fact guarantees algorithmic convergence. In Figures 8 and 9 it is possible to see the direction and intensity of the steering field toward the cloud of points and different level-set curves; in particular in the magnification of zones A and B is possible to see how the steering field automatically stops external liquid from entering the object (all the internal arrows point outside) while, in the convex part, the steering field force the outer fluid to enter the convex region.
In Fig. 10 we represent the final state for the wrapping of a two dimensional cloud of points.

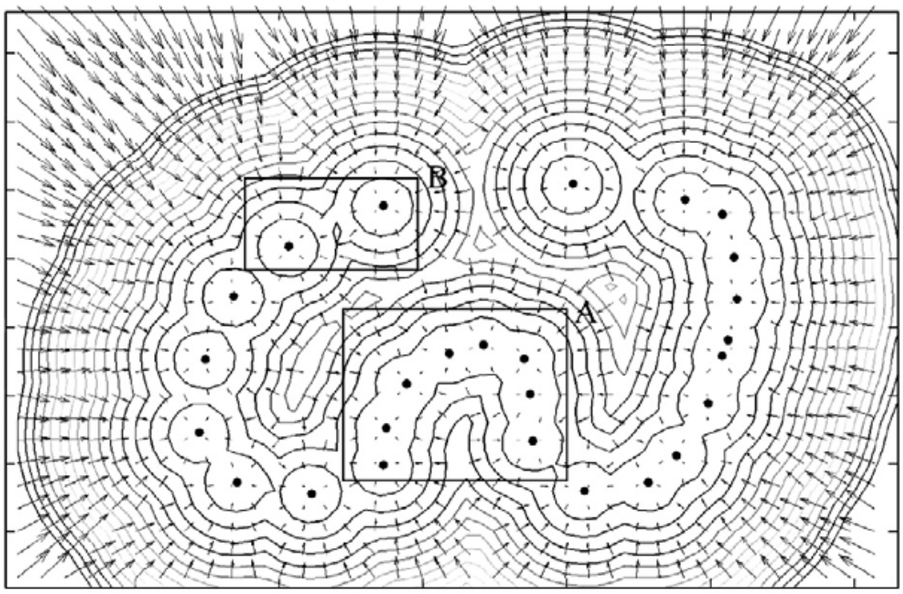


Fig. 8. a 2D non-convex example, arrows indicate the steering velocity field toward the cloud of points and different level-sets are also represented. Zones A and B are zoomed in Fig.9

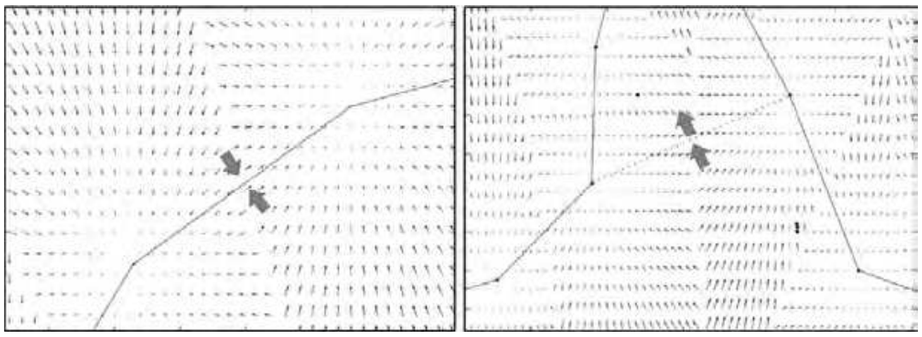


Fig. 9. zoom of zone A and B of Fig. 8 in particular different behavior of the steering field prevents external fluid from entering the convex part in B while favor its penetration in A (nonconvex region)

Further aspects from fluid dynamics can be fruitfully applied, e.g. the diffusive parameter ξ in eq. 35 is a delicate parameter as it is strictly connected to fluid viscosity. High values of ξ result in a smooth but often inaccurate surface while low values result in a surface that strictly honors the data but is sensitive to acquisition noise. As a matter of fact for laser scanner acquisitions the portions of the object with high samples density (usually due to multiple acquisitions from different viewpoints) requires high accuracy and/or present complex topologies, while smoother or less critical areas usually exhibit a lower sample density. In order to account for that, we can establish an inverse dependency between the diffusive constant ξ and the local point density:

$$\xi = \frac{\alpha}{1 + \rho_p} \tag{36}$$

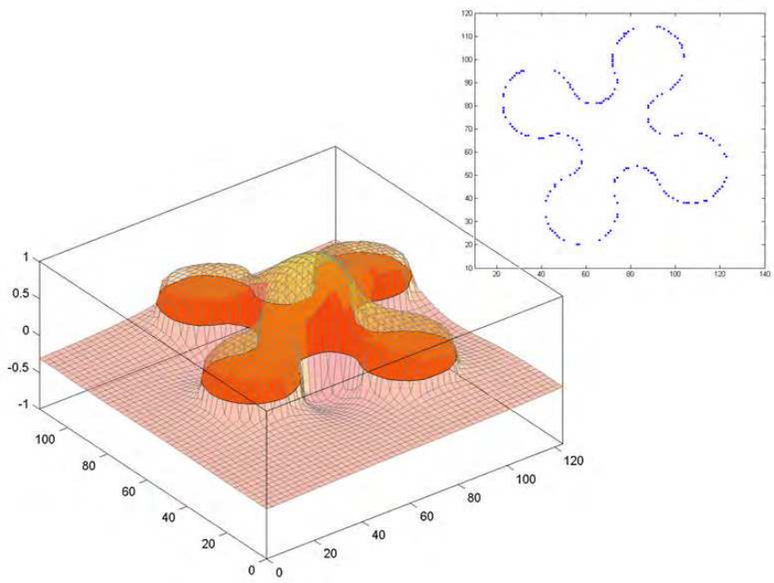


Fig. 10. the wrapping of a set of 2D points (top right), the zero level-set at the end of the evolution represent the joining line, inner points have positive ψ while external ones have negative values

where α is a parameter that indicates the estimated noise in the point location (the higher the value the smoother the surface). The density ρ_p at a given point is defined as the number of samples that fall into a cube of volume V/N around that point, where V is the volume of the point cloud (approximated by the volume of the smallest parallelepiped that contains the point cloud) and N is the total number of samples. This means that if all the samples would be uniformly distributed in the considered volume (like in a cubic crystal) in each cubic cell of volume V/N there would be just one sample. In our case, since we have to define the ρ_p variable for each voxel, we define it as the sum of all the points falling inside a cube of volume V/N centered at the voxel center. In surface regions with a low point density the diffusive constant ξ corresponds to a smooth surface, while in high-density zones ξ becomes very low, to guarantee that the surface will strictly honor the sample points.
The level-set evolution equation is then the following:

$$\begin{cases} \frac{\partial \phi(\mathbf{x},t)}{\partial t} = \nabla \cdot (\mathbf{v}\phi(\mathbf{x},t)) + \xi \nabla^2 \phi(\mathbf{x},t), & t \geqslant 0 \\ \phi(\mathbf{x},t) = +1, & t \geqslant 0, \mathbf{x} \in \mathcal{B} \\ \phi(\mathbf{x},0) = -1 & \mathbf{x} \notin \mathcal{B} \end{cases} \tag{37}$$

where the second raw indicates that the boundary $\mathcal{B}$ of the considered domain is always kept equal to +1 assuming that those points are sources of external fluid. The third raw indicates that at the starting point all the points inside the domain are filled with the internal fluid. In Fig. 11 the level-set evolves toward the cloud of points which is a laser scanner acquisition of a plaster wolf.

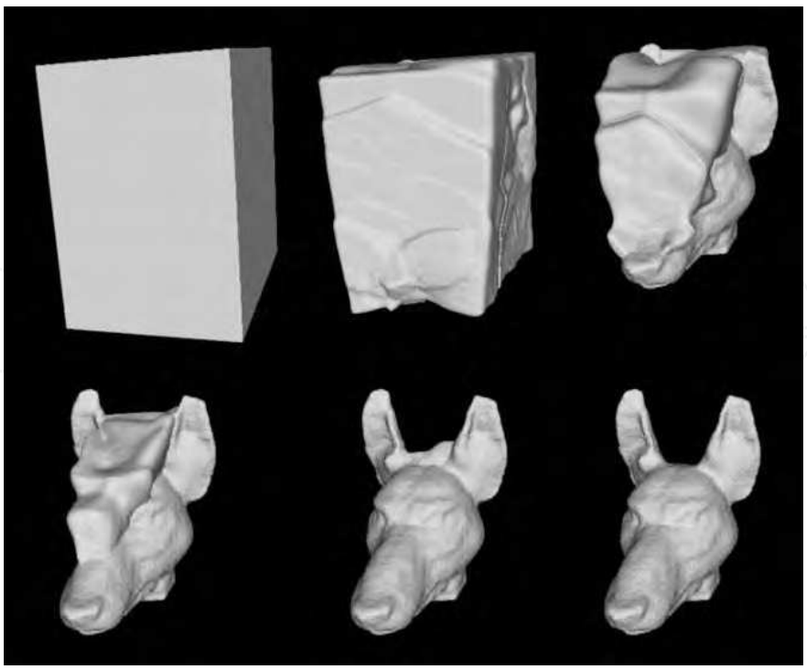


Fig. 11. Evolution of the zero level-set toward a 3D cloud of points acquired by a laser scanner. The level-set starting from the boundary of the considered volume (top left) evolves towards the final cloud of points.

5.1 Turbolence and convex regions

In some particular cases, where high curvature is present in nonconvex regions with a small entrance, external fluid could be inhibited from entering (e.g. consider the cloud of points of Fig. 12).

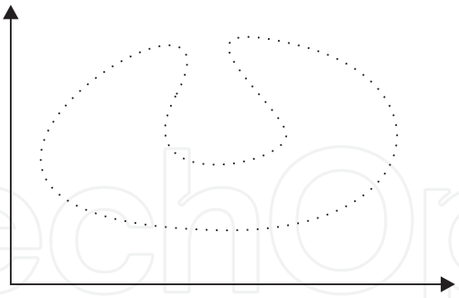


Fig. 12. a set of points presenting a high curvature region within a concavity

For cases like these a further help could come from vorticity. In particular, since our steering field is based on the distance from the closest point it will be a conservative irrotational vector field and when the outer fluid will approach the concavity mouth it will be pushed to its borders without the opportunity to get inside (we underline that this condition is different from the one in Fig. 8 where the concavity of the region A does not present high curvature and external fluid is correctly steered inside the region). A possible solution consists in introducing a rotational component in the steering field, this could be obtained adding periodically a small random displacement for each point around its original position: it will result in a vorticity

contribution: $\zeta = \nabla \times \mathbf{v}(\mathbf{x})$ the resulting effect is that inner fluid in convex regions is pushed inside and outside with the same probability while in nonconvex regions inner fluid that went inside stays there and progressively fills the region. Particular care must be placed in introducing the amount of random points displacement since convergence of the algorithm is no more guaranteed by the conservative vector field.

6. Image processing and viscous noise removal

Fluid dynamics applied together with level-set method can also provide a powerful tool for noise removal in complex images (i.e. when noise is not simply removable using intensity thresholds of histogram analysis), a Total Variation denoising approach is presented in (Osher & Fedkiw (2002)) where the regularization algorithm is based on this PDE:

$$Lu = \lambda Ru, \tag{38}$$

where u is the image represented as a 2D function ($u(x,y)$) that indicates intensity value for each point; R is the regularization operator and L is the time-space operator. A very common energy functional $E(u)$ whose minimization provide the denoised image is the Mumford-Shah multiscale segmentation model:

$$\min_u E(u) = \frac{1}{2}\int_\Omega (u-u_0)^2 dx + \mu \int_{\Omega\backslash\Gamma} |\nabla u|^2 dx + \nu \mathcal{H}^2(\Gamma) \tag{39}$$

where u_0 is the given noisy image, Ω is the image function domain and μ is a weighting coefficient for the average smoothness of regions divided by the contour Γ. ν is the weight coefficient for the total contour length expressed by the Haussdorf measure $\mathcal{H}$.
The minimization of $E(u)$ will provide an piecewise-smooth approximation of the initial u_0 image, Γ has the role of approximating the edges in the image u_0 and u will be smooth only in $\Omega\backslash\Gamma$; Accordingly to the level-set formulation the implicit function will be ϕ and in particular $\phi(\mathbf{x}) = 0$ represents the contour Γ separating region of positive values of ϕ from the region of negative values. In particular $u^+(\mathbf{x})$ will be the intensity function representing values of the region where $\phi(\mathbf{x}) > 0$ while $u^-(\mathbf{x})$ will represent the intensity of points in the $\phi(\mathbf{x}) < 0$ region. Eq. 39 then become:

$$\begin{aligned}\min_u E(u^+, u^-, \phi) = &\int_\Omega (u^+ - u_0)^2 H(\phi)\, d\mathbf{x} + \int_\Omega (u^- - u_0)^2 (1 - H(\phi)) d\mathbf{x} + \\ &+\mu \int_\Omega |\nabla u^+|^2 H(\phi)\, d\mathbf{x} + \mu \int_\Omega |\nabla u^-|^2 (1 - H(\phi))\, d\mathbf{x} + \nu \int_\Omega |\nabla H(\phi)|\end{aligned} \tag{40}$$

Where $H()$ represents the Heaviside step function and the last term, $\int_\Omega |\nabla H(\phi)|$ is the integral in the sense of geometric measure, i.e. for a 2D case, is the length of contour Γ.
The minimization can be done iteratively on u^+ , u^- and on ϕ, in particular, minimizing with respect to u^+ and u^-, i.e. deriving eq. 40, will give:

$$u^+ - u_0 = \mu \nabla^2 u^+ \tag{41}$$

and

$$u^- - u_0 = \mu \nabla^2 u^- \tag{42}$$

which correspond to the diffusive part inside the borders defined by Γ, since $\frac{\partial u^{+,-}}{\partial \mathbf{n}} = 0$ (i.e. no diffusion will take place across the boundary keeping edges sharp). While, deriving with respect to ϕ will give:

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left[-\left|u^+ - u_0\right|^2 - \mu \left|\nabla u^+\right|^2 + \left|u^- - u_0\right|^2 + \mu \left|\nabla u^-\right|^2 + \nu \nabla \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right] \tag{43}$$

Accordingly to a steepest descendent approach. The δ_ϵ indicates that the delta of Dirac, in a discrete environment, must be calculated in a narrow band and, to prevent discontinuities u^+ has to be extended, in the narrow band, for $\phi < 0$ values and also u^- has to be extended in the narrow band for $\phi > 0$. A possible approach to these extension could be the "Ghost fluid Method" (Fedkiw et al. (1999)) where the interface is treated as a moving boundary. Solving the governing equations (Terashima & Tryggvason (2009)) with the extension of the discontinuous variables across the fluids interface, (for real CFD applications this is usually entropy), will reduce smearing of discontinuities providing smooth convergence to the final image without oscillations. In Fig. 13 is represented how the contour between the two image parts is steered accordingly to the Ghost Fluid Method in the narrow band while in Fig. 14 it is possible to see the evolution of the contour across the noisy image: diffusive terms smooth uniform regions reducing noise while the contour evolution smoothly warp the objects.

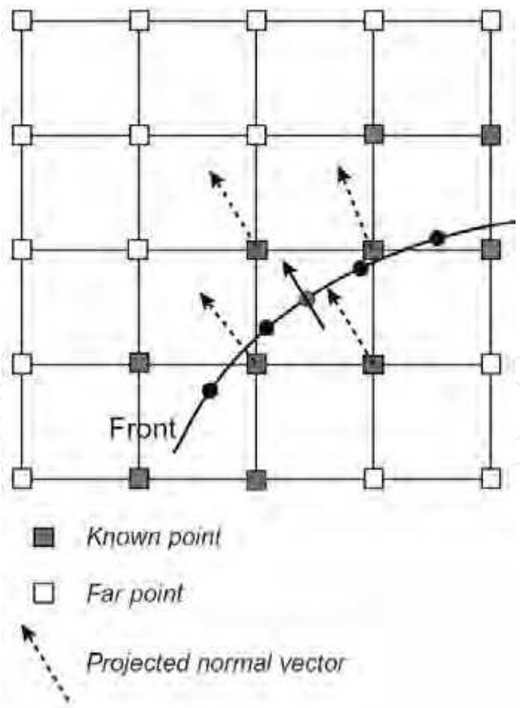


Fig. 13. update procedure in the narrow band points

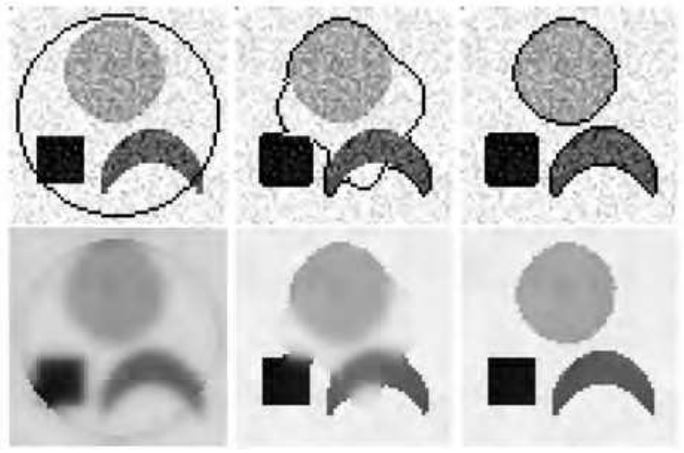


Fig. 14. zero level set evolution accordingly to the Mumford-Shah energy minimization for object segmentation and image denoising

7. Conclusion

In this chapter we have shown some significant implementations of Computational Fluid Dynamics equations in completely different contexts ranging from image denoising to optimal path research for moving robots. In recent literature it is also possible to find different applications to many other research fields: The main reason for this large diffusion and reuse of CFD equations can be mainly addressed to:

- the actual deep insight and knowledge both into the theoretical and computational aspect of fluids.
- The accuracy and stability of modelling equations that can grant convergence in a wide variety of applications.
- Versatility and adaptability of different models from multiphase, to viscous or compressible fluids that provide the user with many intuitive and easy-to-tune parameters allowing an easy adaptation to different contexts.
- The close relation to physical quantities that in many cases can automatically provide the minimum energy solution.

All these aspects concurred in making CFD a science that goes well beyond fluids analysis providing a self consistent, well known, spread of tools to find fruitful and unlimited applications in engineering, computer science and other scientific research fields.

8. References

Adalsteinsson, D. & Sethain, J. A. (1995). A fast level set method for propagating interfaces, *Journal of Computational Physics* 118(2): 269–277.

Amenta, N., Bern, M. & Eppstein, D. (1998). The crust and the β-skeleton: combinatorial curve reconstruction, *Graph. Models Image Process* 60(2): 125–135.

Biben, T., Klaus, K. & Chaouqi, M. (2005). Phase-field approach to three-dimensional vesicle dynamics, *Phys. Rev. E* 72(4).

Bridson, R. (2003). *Computational aspects of dynamic surfaces*, PhD thesis, University of British Columbia.

Davis, S. H. (2001). *Theory of solidification*, Cambridge University Press.

Dobrushin, R. L., Koteckỳ, R. & Shlosman, S. (1992). *Wulff Construction: A Global Shape from Local Interaction*, American Mathematical Society.

Fedkiw, R. P., Aslam, T., Merriman, B. & Osher., S. (1999). A nonoscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method), *Journal of Computational Physics* 152(2): 457–492.

Gomes, J. & Faugeras, O. (1999). Reconciling distance functions and level sets, tech. rep. 3666, *Technical report*, INRIA Sophia Antipolis.

Gueyffier, D., Li, J., Nadim, A., Scardovelli, R. & Zaleski, S. (1999). Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, *J. Comput. Phys.* 152: 423–456.

Hennessy, M. G. (2010). *Liquid Snowflake Formation in Superheated Ice*, PhD thesis, University of Oxford.

Hills, R. N. & Roberts, P. H. (1993). A note on the kinetic conditions at a supercooled interface, *Int. Comm. Heat Mass Transfer* 20: 407–416.

Hobbs, P. V. (2010). *Ice Physics*, Oxford University Press.

Hoppe, H. (1994). *Surface reconstruction from unorganized points*, PhD thesis, University of Washington.

Kimmel, R., Amir, A. & Bruckstein, A. M. (1995). Finding shortest paths on surfaces using level sets propagation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17(6): 635–640.

Kimmel, R., Kyriati, N. & Bruckstein, A. M. (1998). Multivauled distance maps fro motion planning on surfaces with moving obstacles, *IEEE Trans. on Robotics and automation* 14(3): 427–436.

Latombe, J. C. (1991). *Robot Motion Planning*, Kluwer.

Libbrecht, K. G. (2005). The physics of snow crystals, *Rep. Prog. Phys. 68*, Norman Bridge Laboratory of Physics, California Institute of Technology.

Losasso, F., F. Gibou, F. & Fedkiw, R. (2004). Simulating water and smoke with an octree data structure, *ACM Transactions on Graphics* 23(3): 457–462.

Marcon, M., Piccarreta, L., Sarti, A. & Tubaro, S. (2008). Fast pde approach to surface reconstruction from large cloud of points, *Computer Vision and Image Understanding* 112: 274–285.

Markstein, G. H. (1951). Experimental and theoretical studies of flame front stability, *J. Aero. Sci.* 18: 199.

Noh, W. & Woodward, P. (1976). A simple line interface calculation, *in* Springer-Verlag (ed.), *Proceedings of 5th International Conference on Fluid Dynamics*.

Osher, S. & Fedkiw, R. (2002). *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag.

Piegel, L. & Tiller, W. (1996). *The NURBS Book*, second ed. edn, Springer-Verlag, Berlin.

Pimpinelli, A. & Villain, J. (1999). *Physics of Crystal Growth*, Cambridge University Press.

Registration and fusion of intensity and range data for 3d modelling of real world scenes (2003). Vol. Fourth International Conference on 3-D Digital Imaging and Modeling, 3DIM Proceedings.

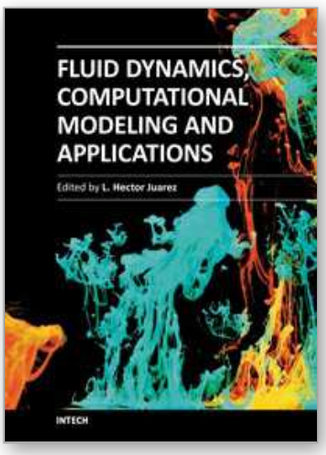
Sethian, J. A. (1989). A review of recent numerical algorithms for hypersurfaces moving with curvature-dependent speed, *Journal Differential Geometry* 31: 131–161.

Sethian, J. A. (1999). *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press.

Shastri, S. S. & Allen, R. M. (1998). Method of lines and enthalpy method for solving moving boundary problems, *International Communications in Heat and Mass Transfer* 5(4): 531–540.

Terashima, H. & Tryggvason, G. (2009). A front-tracking/ghost-fluid method for fluid interfaces in compressible flows, *Journal of Computational Physics* 228: 4012–4037.

Zhao, H., Osher, S. & Fedkiw, R. (2001). Fast surface reconstruction using the level set method, *Proceedings of IEEE Workshop on Variational and Level Set Methods in Computer Vision (VLSM 2001)*.



Fluid Dynamics, Computational Modeling and Applications

Edited by Dr. L. Hector Juarez

ISBN 978-953-51-0052-2
Hard cover, 660 pages
Publisher InTech
Published online 24, February, 2012
Published in print edition February, 2012

The content of this book covers several up-to-date topics in fluid dynamics, computational modeling and its applications, and it is intended to serve as a general reference for scientists, engineers, and graduate students. The book is comprised of 30 chapters divided into 5 parts, which include: winds, building and risk prevention; multiphase flow, structures and gases; heat transfer, combustion and energy; medical and biomechanical applications; and other important themes. This book also provides a comprehensive overview of computational fluid dynamics and applications, without excluding experimental and theoretical aspects.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Marco Marcon (2012). Fluid Dynamics Without Fluids, Fluid Dynamics, Computational Modeling and Applications, Dr. L. Hector Juarez (Ed.), ISBN: 978-953-51-0052-2, InTech, Available from: http://www.intechopen.com/books/fluid-dynamics-computational-modeling-and-applications/fluid-dynamics-without-fluids

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen