Natural Element Simulation of Free Surface Flows

Author: Andrés S. Galavís Borden

Directors: David González and Elías Cueto

Doctoral Degree in
Computational Mechanics

January, 2011

Universidad Zaragoza
1542

Group of Structural Mechanics and Material Modeling

Natural Element Simulation of Free Surface Flows

Author: Andrés S. Galavís Borden

Directors: David González and Elías Cueto

Doctoral Degree in
Computational Mechanics

January, 2011

Universidad Zaragoza
1542

Group of Structural Mechanics and Material Modeling

Ni todas las tesis escritas y por escribir podrían igualar el valor de tu amor, compañía, el balance que traes a mi vida y la fuerza que me das. Aún así, esta tesis es para ti, Su.

Acknowledgements

Quiero ofrecer un sincero agradecimiento a todos quienes por medio del cumplimento de su trabajo han hecho posible el desarrollo de esta tesis, así como mi estancia en la universidad e incluso en este país. A todos quienes fueron mas allá de sus obligaciones, les debo una gratitud mucho mas profunda, pues estas buenas intenciones y acciones son el origen del buen sabor de boca que me queda de esta vivencia.

Quiero darle un agradecimiento especial a mis tutores, los Doctores Elías Cueto y David González, por su guía, ayuda, apoyo y paciencia durante estos años. Con ustedes me encuentro en deuda de gratitud tanto a nivel académico como personal y espero que en un futuro me brinden la posibilidad de retribuirles al menos parte. También deseo agradecer al Doctor Manuel Doblaré, por abrirme las puertas del grupo y proporcionarme la posibilidad de emprender esta etapa. Agradezco también los buenos recuerdos que mantengo de una gran cantidad de compañeros, tanto del grupo como de convivencia, algunos presentes, algunos ya en otros lugares.

A medida que avanzo por esta página se hace mas difícil encontrar epítetos apropiados, capaces de reflejar el aprecio que siento por las personas que han pasado de ser meros compañeros a grandes amigos. Y aunque me disculpo primero antes quienes deje fuera de esta lista (tanto por espacio como porque es de madrugada y mi mente divaga) y segundo, por la aparente frialdad al mencionarlos en orden alfabético; deseo nombrar a Angélica, Berto, los Heidenreich-Prozzi, Jose Félix, Olfa, Oresti, Raúl, Siamak, Teng, Victor, Vincenzo... Y aunque no los nombre, a los amigos de afuera de la universidad también les agradezco.

Finalmente deseo expresar un agradecimiento máximo (y hago notar que utilizo el superlativo) a mi familia grande, con todos sus apellidos Galavís Borden Valdivia Schebesta Delgado... aunque no llame tanto como debería, ahí están siempre.

Gracias!

Natural Element simulation of free surface flows

Abstract

Numerical simulation of free surface flows remains to be a problem of utmost difficulty in the field of Computational Mechanics. The origin of these problems is twofold. On one side, the most typical description of the fluid kinematics is the Eulerian description. But this description is by no means the most adequate for describing the evolution of the free surface, which would be most easily described in a Lagrangian approach.

On the other side, if a Lagrangian description of the movement is preferred, then the most extended numerical techniques for solving the resulting Navier-Stokes equations, such as finite elements, finite differences or finite volumes, become extremely intricate.

In this thesis a different approach has been pursued. Meshless methods have been chosen to approximate the Navier-Stokes equations, and particularly, the natural element method has been chosen due to its particular characteristics. Among these, one can cite the exact imposition of essential boundary conditions, the ability for high-order approximations, and its strong link with the geometrical structure of the description of the free surface that has been chosen.

In this thesis a second order in time natural neighbour Lagrange-Galerkin method has been developed. This method has demonstrated excellent results in problems where previous approaches failed. In addition, a shape constructor method has been proposed for the automatic extraction of the geometry of the domain as it evolves in time. It is based on the concept of α-shapes, but two additional α-filtrations are performed on top of the traditional algorithm that make it much more powerful and less sensitive on the choice of the parameters.

Finally, the developed method has been applied to a particularly challenging problem, which is that of the Worthington jet and, in general, free surface, non-Newtonian fluid mechanics.

Simulación de flujos con superficie libre por el método de los elementos naturales

Resumen

La simulación numérica de flujos con superficie libre continúa siendo hoy día un problema de extrema dificultad en el campo de la Mecánica Computacional. El origen de estas dificultades tiene dos vertientes. Por un lado, el hecho de que la cinemática de los fluidos se describe habitualmente y de manera natural en un marco euleriano. Pero este tipo de descripción no es, en modo alguno, la más apropiada para describir el movimiento de la superficie libre, que aceptaría de un modo mucho más natural una descripción lagrangiana.

Por otro lado, si se escoge una descripción lagrangiana, las técnicas numéricas más extendidas (elementos finitos, diferencias finitas o volúmenes finitos, entre otros), se vuelven extremadamente complejos en su aplicación a las ecuaciones resultantes de Navier-Stokes.

En esta tesis de plantea una aproximación al problema de Navier-Stokes con superficie libre completamente diferente. Se han escogido los métodos sin malla, y más concretamente, el método de los elementos naturales, para realizar la simulación. Éste último se ha escogido por sus peculiares características, entre las cuales cabe citar la imposición exacta de condiciones de contorno esenciales, su capacidad de desarrollar aproximaciones de alto orden y su estrecho lazo con la estructura geométrica que se ha escogido para la descripción del movimiento de la superficie libre.

En esta tesis se ha desarrollado un método de Lagrange-Galerkin de vecindad natural de segundo orden en el tiempo, que ha mostrado excelentes resultados en problemas en los que intentos previos han fallado. Además se ha desarrollado un constructor de formas para la extracción automática de la geometría del dominio conforme ésta evoluciona en el tiempo. El método desarrollado se basa en la técnica de formas α, pero se han añadido dos filtrados adicionales que hacen que método sea mucho más robusto y menos sensible a la elección de los parámetros.

Finalmente, el método recién desarrollado se ha aplicado a un problema especialmente complicado como es el del jet de Worthington y, en general, a los flujos de fluidos no newtonianos con superficie libre.

Contents

Contents

Document **23**

1 Introduction **25**

1.1 State-of-the-art for meshless methods 27

1.2 State-of-the-art in the simulation of free-surface flows 29

1.3 Structure of the thesis 30

2 The Natural Element Method **33**

2.1 Natural neighbors 34

2.1.1 Voronoi Tesellation 34

2.1.2 Delaunay Triangulation 35

2.2 Natural Neighbour Interpolation 36

2.2.1 Sibson Interpolant 36

2.2.2 Non-Sibson Interpolant 38

2.3 Natural Element Shape Function Properties 39

2.3.1 Meshless Character 39

2.3.2 Strictly Interpolant Character 40

2.3.3 Approximation 40

2.3.4 Partition of Unity and Positivity 42

2.3.5 Linear Consistency 42

2.3.6 Smoothness 43

2.4 Shape Function Construction 43

2.4.1 Useful formulas 43

2.4.2 Bowyer-Watson Algorithm 44

2.4.3 Laserre's Algorithm 51
2.5 Imposition of Essential Boundary Conditions 52
2.6 α-shape Based Natural Elements 52
2.7 High-Order NEM 53
2.7.1 Revisiting the de Boor's algorithm for B-splines 54
2.7.2 B-spline surfaces constructed over natural neighbor interpolation 56

3 An Updated Lagrangian Approach to Fluid Dynamics 61
3.1 Kinematic Description of Fluid Flows 61
3.1.1 Eulerian Description 62
3.1.2 Lagrangian Description 63
3.1.3 Arbitrary Lagrangian-Eulerian (ALE) 64
3.1.4 Material and Spatial Time Derivatives 65
3.2 Governing Equations 66
3.2.1 Conservation Equations 67
3.2.2 Constitutive equations for a Newtonian fluid 68
3.2.3 Navier-Stokes Equations 68
3.3 The Method of Characteristics 69
3.3.1 Characteristic Lines 70
3.3.2 Solution Strategies Based on Characteristic Lines 70
3.4 Lagrange-Galerkin Methods Based on Characteristic Lines 72
3.4.1 Weak Formulation 72
3.4.2 Time Discretization 73
3.4.3 Algorithmic Issues 74
3.5 Numerical Examples 75
3.5.1 Sloshing 76
3.6 Conclusions 81

4 Free Surface Treatment 83
4.1 Theory of α-shapes . . . 86
4.1.1 Definition of the family of α-shapes . . . 87
4.1.2 How to choose the α-value . . . 87
4.1.3 Problems with the α-shape technique . . . 90
4.2 Proposed algorithm . . . 91
4.2.1 Choosing a and b. . . . 93
4.3 Examples . . . 95
4.3.1 Benchmarking . . . 95
4.3.2 2D wave breaking . . . 99
4.3.3 An extension to 3D problems. Aluminium extrusion . . . 101
4.4 Conclusions . . . 103

5 Applications to Non-Newtonian fluid flows 109
5.1 Governing Equations: The Oldroyd-B Fluid Model . . . 110
5.1.1 Model Derivation . . . 111
5.1.2 Model Implementation . . . 115
5.1.3 Model Validation: Fully developed flow Inside a Pipe . . . 116
5.2 Other, more complex, non-Newtonian Examples . . . 120
5.2.1 Die Swelling . . . 120
5.2.2 Drop Splashing. Worthington Jet . . . 124
5.3 Conclusions . . . 126

6 Ending Remarks 131
6.1 Original developments in this thesis . . . 133
6.2 Future work . . . 134

Bibliography 137

Appendix 147

A Publications in peer-reviewed journals 149
A.1 Computational Mechanics (2008) 42:467–479 . . . 150
A.2 European Journal of Computational Mechanics (2006) 16:323–336 . . . 163

List of Figures

2.1 Voronoi Tessellation of a set S of points. 35
2.2 Empty circumcircle. Delaunay triangulation. Degenerated triangulation 36
2.3 Modified Voronoi diagram by the inclusion of point $\boldsymbol{x}$ 37
2.4 Sibson shape function. (Courtesy N. Sukumar) 38
2.5 Laplace shape function construction. 39
2.6 Laplace shape function (Courtesy N.Sukumar) 40
2.7 Natural neighbor coordinates in 1D. 41
2.8 Starting point for the Bowyer-Watson algorithm. 46
2.9 Natural neighbor search. 47
2.10 Triangles t_i and their circumcenters $\boldsymbol{c}_i(\boldsymbol{x})$. 48
2.11 Subset of triangles($t(\boldsymbol{c}_i(\boldsymbol{x}), \boldsymbol{c}_j(\boldsymbol{x}), \boldsymbol{v})$) and their areas α_{it}. 49
2.12 The sum of α_{it} over all t_is yields the area of the second order Voronoi cells. 50
2.13 Schematic representation of the de Boor's algorithm. 55
2.14 Schematic representation of the de Boor's algorithm employing natural neighbours. Between parentheses, the domain of each function. 57
2.15 Schematic representation of the proposed algorithm 58
2.16 Shape of a typical function N_{IJ} for a set of irregularly distributed sites. . 59

3.1 Eulerian approach. ◯ Numerical nodes. △ Material particles. 63
3.2 Lagrangian approach. ◯ Numerical nodes. △ Material particles 64
3.3 ALE approach. ◯ Numerical nodes. △ Material particles 65
3.4 Initial configuration for a 2-D sloshing problem. 76
3.5 Evolution in time of the vertical displacement and velocities at both edges of the free surface. The numerical solution is shown in red, in purple the analytic solution. 78

3.6 Vertical displacement and velocities at both edges of the free surface. The second-order solution is shown in blue, the first-order solution in red and the analytic solution in a purple line. 79
3.7 Velocity field at time $t = 0.005$. 79
3.8 Velocity field at time $t = 0.455$. 80
3.9 Velocity field at time $t = 1.205$. 80

4.1 Evolution of the family of α-shapes of a cloud of points representing a wave breaking on a beach. 88
4.2 Medial axis of a two-dimensional curve. 89
4.3 Computation of the LFS at a point p. 89
4.4 Evolution of the LFS at the neighbourhood of two surfaces getting into contact. 91
4.5 Spurious detection of contact at the crest of the wave. 91
4.6 Relationship between the modifying factors. Variation in any of the deformation factors (a) and in alpha value (b). 94
4.7 Method validation. Cases studied of a 2D ball drop over a flat surface. (a), $0 \leq \boldsymbol{v}_2 < \boldsymbol{v}_1$ and (b), $\boldsymbol{v}_2 = \boldsymbol{v}_1,\ 0 \leq \beta \leq \pi/2$ 96
4.8 Drop approaching a surface. Both families of cases studied were tested on the same set of nodes (a).The resultant geometry of the domain provided the standard α-shape (b) and the modified method (c) are depicted. 97
4.9 Drop approaching a still plate (detail of the contact zone). In light grey the triangles filtrated by the proposed techniques are shown. 98
4.10 Detail of the velocity field at the wave crest 99
4.11 Wave before breaking. Velocity vectors (a), α-shape without additional filtration (b) and shape reconstructed with the new approach (c). In this cases the parameters used where $\alpha = 9, a = 10, b = 1$. 100
4.12 Volume conservation for the standard and the proposed technique. . . . 100
4.13 Schematic geometry of the die for the extrusion of a hollow profile. . . . 104
4.14 Sequence of aluminium flow at the early stages of the extrusion of a hollow cylindrical profile. 105
4.15 Extrusion process. Instant before contact of the two metal flows. 106

5.1 A sample dumbbell. 111

5.2 Flow schematics 117
5.3 Extra-stress field for a fully developed pipe flow: τ_{yy} 118
5.4 Extra-stress field for a fully developed pipe flow: τ_{xy} 118
5.5 Extra-stress field for a fully developed pipe flow: τ_{yy} at $y = 1$ 119
5.6 Extra-stress field for a fully developed pipe flow: τ_{xy} at $y = 1$ 119
5.7 Velocity profile for a fully developed pipe flow: $\boldsymbol{V}_y$ at $y = 1$ 120
5.8 Swelling comparison between fluids with different Weissenberg numbers 122
5.9 Swelling ratio against We. Different numerical models compared against an approximated theoretical solution by Tanner (2005). 123
5.10 Drop falling, example of a Worthington Jet formation. (Taken from Cheny and Walters (1999)) 124
5.11 Initial configuration for the falling drop (left) and detail in the vicinity of the drop (right). 126
5.12 Maximum height reached by the Newtonian fluid and Oldroyd fluids of $We_{ef} = 0.1$ and $We_{ef} = 0.5$ 127
5.13 Comparision between Newtonian fluid (left) and Oldroyd-B fluid (right) with $We = 0.5$. $t = 0$. 127
5.14 t=10 ms. The crown is visible at this point. 128
5.15 t=75 ms. The jet is already formed. 128
5.16 t=105 ms. Oldroyd fluid reaches its maximum height. 128
5.17 t=110 ms. Newtonian fluid reaches its maximum height. 129
5.18 t=150 ms. Both jets are already decreasing. 129

Document

Chapter 1

Introduction

The contemporary significance of numerical methods in a wide range of fields is a fact already accepted. This is evidenced by the large number of such tools routinely applied in the industrial sector. Additionally, the increased performance of computers, which took place in the last twenty or thirty years, has allowed the numerical simulation of a large number and variety of phenomena studied by various disciplines. Whether in the fields of mathematics, engineering, physics or medicine, computer simulations can provide quantitative results in highly complex processes. Problems such as the mechanical response of solids, fluids, and even living tissues, have been successfully addressed through the development of mathematical formulations of the laws that govern those behaviors. However, we are far from providing a mathematical description for every problem and even more from solving them —if that would be even possible—with the techniques currently available.

This thesis originated in the framework of the project ``Meshless simulation of fluid-structure interaction'', funded by the Spanish Ministry of Science and Innovation. One of the most difficult problems we faced during the development of such project was the design of an efficient computational technique which could face effectively the burden associated with free-surface fluid-solid interaction. The complexity associated with this phenomenon was mainly due to the complex nature of the free surface problem, which, added to the inherent fluid-structure interaction difficulties, led to numerous difficulties.

In the first half of the decade of 1990 a new family of numerical methods arose that were coined as *meshless methods*. These methods share one common characteristic, despite the wide range of names and different techniques that they encompass. This characteristic is that, either based upon Galerkin or collocation techniques, meshless methods do not greatly suffer of mesh distortion, and hence their name. They thus

appeared as a natural choice for the problem at hand. Meshless methods allow for an updated Lagrangian description of the fluid flow, thus avoiding remeshing nor complex descriptions such as Arbitrary Lagrangian Eulerian methods (ALE).

This thesis proposes using the natural element method to study certain flow features in the presence of a free surface for both Newtonian and Viscoelastic fluids. To this end it is necessary to improve the robustness of this technique, to implement the fluid model focus of our study and to solve some algorithmic aspects which are troublesome for this method, as will be deeply described later on.

In this thesis we will move away from traditional models in at least two ways. On the one hand, the fluid mechanics is usually treated from an Eulerian approach, which assumes that there are fixed observation points from which sampled particles allow to describe the state of flux at a given time instant. This way of studying flows is very useful for stationary situations or for internal flows, where the shape of the volume occupied by the liquid does not change, or if it does, at least is possible to predict where to place the observation points. However, the Eulerian point of view is not equally suitable when we are in the presence of largely changing free surfaces. An arbitrary Lagrangian Eulerian (ALE) approach has been developed and it is well established by now, although it does not come free of problems. The approach we adopt in this thesis is the use of an updated Lagrangian scheme, in which each observation point moves associated to a material particle. We consider that this strategy will be the most appropriate to track domains evolving in time, typical of the problems that concern us. This scheme has been successfully used in González (2004) for the case of Newtonian fluids. This method for fluid flow simulation will be fully explained on Chapter 3, where some examples showing the technique capabilities will be presented as well.

The second relevant aspect regarding the way on which this work parts from the norm is on the numerical solution of the fluid (Navier-Stokes) equations. The Finite Element Method is the most widely used and developed numerical technique for the approximate solution of partial differential equations. Applying this method to problems with heterogeneous materials, anisotropic or nonlinear behavior has yielded high-quality results. It seems logical, therefore, that the first choice for the approximate solution of solid mechanics equations, would be precisely the FEM. This choice is not so clear in the field of fluid mechanics. Its election presents not a few drawbacks to contend with. First, in order of construction, and perhaps also in order of difficulty, would be the making of a mesh that accurately represents the domain under study. The resolution by the finite element method involves a discretization of the domain. This is necessary both for the required numerical integration of Galerkin method and for the very construction

of FEM shape functions.

While great strides on improving these procedures have been made and some functionality has been achieved, computational mesh generation is still a very active research field. It has a large number of groups dedicated to it, being far from a solved problem or sufficiently automated (as an amused activity, the interested reader can be bothered to perform a quick search for papers published on 2010 on this topic). On this topic, three-dimensional domains represent additional and particular difficulties. In fact, mesh generation is still one of the most time consuming parts in the process creating a model by the finite element method, and to some extent could be perceived almost as an art.

It is on the numerical method of solution, where the most important selection in the direction of this thesis was made. On this matter a road less well-traveled has been chosen, so the Finite Volume Methods (very popular in commercial simulations of fluids) and Finite Differences (more typical of academia) that are usually used with Eulerian schemes where set aside. Also ignored has been the method of Finite Elements, which as already mentioned, despite being the most popular approaches used in Lagrangian (both industry and academia), is not capable of withstanding large deformations domains—at least not without modifications to the method. The employed technique in this work has been the Natural Element Method, which belongs to the family of aforementioned meshless methods. The characteristic of these is that they do not require fixed information about the connectivity between nodes, which is why there is no problem in following material particles even if they change their neighborhood at each time step.

1.1 State-of-the-art for meshless methods

Several names have been proposed for these methods in an attempt to characterize them: methods of particles, finite point, element-free methods, diffuse elements, etc. each time putting emphasis on a characteristic of each method. Still, the commonality between them is that the nodal connectivity is obtained through a search algorithm in a process transparent to the user, releasing the burden of generating a suitable mesh for the domain at each stage of the simulation.

Most meshless methods continue to progress, even though the last two decades produced great advances in their understanding. Just to cite an example, works like (Babuška and Melenk, 1996) have provided some of the theoretical basis needed to

identify the approximation spaces from which meshless shape functions can be constructed.

Many methods have found their niche application, and have proven to be adequate in problems of rapidly evolving topology, like crack propagation Krongauz (1996). The first meshless method was presented in 1977 by Lucy (1977) and Monaghan (1982). It was called Smooth Particle Hydrodynamics and came from the field of theoretical astrophysics. It has later been applied to a wide range of phenomena, including fluid and solid mechanics. The method called Reproducing Kernel Particle Method (Liu and Chen, 1995) (Liu et al., 1995) arose from the SPH as a generalization of the former in which a correction function is added in order to provide linear consistency.

Coming from a different family, we find all the methods that derive from the Moving Least Squares Method. The idea behind this method is to obtain approximating functions around a given point from scattered nodal data which has been fitted via a moving least squares formulation. This method is used in 3D surface construction as a way to modify the sampling of a nodal cloud. Members of this family are the Diffuse Element Method (Nayroles et al., 1992), the Element-Free Galerkin Method (Belytschko et al., 1994) or its related counterparts (Atluri et al., 1999), (Atluri and Zhu, 2000), or the Hp-Clouds Method (Duarte and Oden, 1996a)(Duarte and Oden, 1996b). It has been independently developed by Duarte and Oden; and by Babuska and Menlek, that the main issue with these methods resided in the need to construct a partition of unity. The shape functions obtained by the general formulation of each of these methods are not strictly interpolant, meaning that shape functions do not evaluate to one at the the ascribed node and zero at any other (lack of fulfillment of the Kronecker delta property). As a result the approximated solution will pass through the nodal values. This situation constitutes a problem for the imposition of essential boundary conditions in many meshless methods. This problem has been studied and is partially solved for some methods (Belytschko et al., 1994) (Krongauz, 1996). Also new methods have been developed which circumvent this situation. To name but one, the approximation based upon maximum entropy schemes seems to be one of the most promising (Arroyo and Ortiz, 2006), (Cyron et al., 2009).

Natural Elements were born in the late 1990s and has been Zaragoza University one of the main research centers for the development of this method, see Cueto (2001), González (2004), Alfaro et al. (2006b) to name a few of these university contributions. An in-depth review of this method will be presented in Chapter 2 . Yet for now, suffice it to say that it provides a convenient way to implement the aforementioned updated Lagrangian approach for fluid simulation.

One of the most cited capabilities of meshless methods is that of simulating large deformation phenomena. With the possibility of simulating flows in an updated Lagrangian framework many works have been devoted to this end in the last years. The interested reader can consult, for instance, Martínez et al. (2004), Idelsohn et al. (2003), Idelsohn et al. (2004), González et al. (2007), among others. In these problems, we can add to the obvious advantages of updated Lagrangian meshless methods the absence of numerical diffusion associated with remeshing and the lack of convective terms in the formulations, that consequently do not need for any stabilization.

1.2 State-of-the-art in the simulation of free-surface flows

Regarding the treatment of free surfaces, these have been traditionally dealt with by either tracking methods or by surface capturing techniques. The Volume of Fluid (VoF) technique is an example of the later techniques, while the ALE formulation could be seen as an instance of the former. With traditional approaches is particularly noteworthy is the difficulty in the selection of mesh velocity in ALE formulations. In addition, tracking the free surface with boundary markers can be implemented in an elegant way in two dimensional problems —by employing a chain of markers and checking self-intersections of the chain to detect merging flows—, as in (Lewis et al., 1997), (Duchemin et al., 2002), for instance, but becomes much more intricate in three dimensions.

If one tries to avoid any form of meshing and only a set of nodes without explicit connectivity is employed, then finding the position of the free surface becomes a problem. In other words, the geometry of the domain must be extracted from the current, updated, position of the nodes, that move with the material velocity. To this end, various authors have employed Computational Geometry techniques. In particular, Cueto et al. (2000) seem to have been the first in employing *shape constructors* —α-shapes in this case—techniques to extract the geometry of the domain. Shape constructors are geometrical techniques that enable to find the shape of a cloud of nodes at each time step. α-shapes (Edelsbrunner et al., 1983), (Edelsbrunner and Mücke, 1994) have been employed in a number of previous works involving free surface flows, see for instance Idelsohn et al. (2004), Idelsohn and Oñate (2006), Martínez et al. (2004), González et al. (2007) and Birknes and Pedersen (2006), among others.

Different shape constructors have been proposed to extract the geometry of the domain. The Delaunay triangulation (Delaunay, 1934) is the base ingredient of these techniques, since it characterizes unequivocally the cloud of points —it is unique for each cloud. Different criteria have been proposed in order to select the triangles pertaining to the *shape* of the domain. The simplest one is maybe the α-shape technique, that proposes to eliminate all triangles (or tetrahedra) whose circumscribing radius is greater than a prescribed *level of detail*. One of the main drawbacks of the α-shape technique, as recognized in many works (see, for instance, Cazals et al. (2006)Teichmann and Capps (1998)) is precisely the choice of the α-value. In general, α-shapes work well for uniformly-distributed clouds of points, which generally does not constitute a problem for stationary problems. However, for our intended use it remains to be an issue that deserves further insight.

The jump of the aforementioned techniques to the field of Computational Mechanics has posed additional difficulties. By definition α-shapes are not able to detect holes or cavities of size smaller than α. This implies that contact between different surfaces is detected with an error $\mathcal{O}(\alpha) \approx \mathcal{O}(h)$, i.e., prior to the true expected contact (Teichmann and Capps, 1998). Precisely in this last reference a method is proposed to alleviate this drawback, but it needs information on the normal of the boundary at the sampling points. This is easy to achieve for three-dimensional scans of solids, for instance, but this kind of information is not readily available for the type of simulations we are interested in.

1.3 Structure of the thesis

Along with this first introductory chapter, this thesis is composed by five more chapters. In Chapter 2 a detailed exposition of the natural element method, together with its merits in the problem we face, will be addressed. Chapter 4 deals with the description of the free-surface problem. In it, a proposed solution for the problem of on-the-fly extraction of the shape (i.e., the boundary) of the domain will be described.

In Chapter 5 we will delve into the problem of simulating Non-Newtonian fluids, that add additional difficulties to the before presented problem. Examples of these fluids can be found almost anywhere, from body fluids as blood or mucus to man-made substances like paint, shampoo or molten plastics. Of course not all non-Newtonian fluids behave in the same manner so many different constitutive equations, with a wide range of complexity, have been developed to describe their mechanical response. The

great industrial importance of these fluids has originated a great deal of interest in its numerical simulation from a relatively long time. A great deal of success has been achieved, and we count with tools for solving a huge variety of complex fluids. Still the processes has been plagued with difficulties, many of which seem to come from the so called *high Weissenberg number problem*. Regardless of the employed numerical method or the viscoelastic constitutive equations used, when the elasticity of the fluid is increased by in a given measure, the simulation would suffer from a huge loss in accuracy leading to convergence problems. This situation limited the application spectrum of cases, rendering the numerical tools ineffective to deal with industrial grade problems. Amounting to this problem, the interesting free-surface phenomena that occur in some viscoelastic flows creates an unmissable setting for the applications tryout of the developed method.

Finally, a summary of all the main results and conclusions will be presented along with some remarks regarding the future lines that could be followed from this work.

Chapter 2

The Natural Element Method

The Natural Element Method (NEM) is a meshless Galerkin procedure based on the natural neighbor interpolation scheme, which in turn relies on the concepts of Voronoi diagrams and Delaunay triangulation to build Galerkin trial and test functions. All of these will be defined and explained in the following sections.

To explain the appeal of this method to our work, first we have to establish our motivation. Traditionally the fluid models are built using an eulerian frame of reference (Donea and Huerta, 2003). This framework is not particularly well suited for simulating free-surface flows, forcing different authors to tackle the problem with different methods. Several techniques have been developed to keep track of the evolving boundary (Duchemin et al., 2002). Other researchers have decided to change the formulation in order to work with pure or mixed lagrangian frames of reference. Our approach lies in this camp as we have adopted an updated Lagrangian fluid model which will be described thoroughly in chapter 3. We aim at simulating the fluid-structure interaction from a Lagrangian-Lagrangian standpoint considering it to be the most natural way to accomplish fluid-solid coupling.

The most prevalent method for solving partial differential equations in a Lagrangian approach is arguably the Finite Element Method. Yet this technique does not cope very well with the high deformation of the mesh which may occur in transient fluid problems. In this way seems natural to use a meshless method One of the main drawbacks of some of these methods is that the calculated variables do not correspond with the essential variables. This implies that the imposition of boundary conditions needs supplementary steps.

The natural element method allow us bypass the aforementioned problems. First, its meshless character allows us to employ a Lagrangian formulation in situations where we know there will be large deformation, yet the future position of the particles is not

known beforehand. We avoid the need for a conscious, and usually very time consuming, mesh generation. Second, by the properties of the method, the inter-domain data transfer can be accomplished without any extra load which also open the door to also tackle fluid-structure interaction problems, but more on this topic will be discussed in another section of this work

2.1 Natural neighbors

For the introduction of the geometrical concepts related to the Natural Element Method we will refer to a domain in $\Re^2$, explaining about a generalization to $\Re^n$ when necessary.

2.1.1 Voronoi Tesellation

Given a set of points $S = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, there exists a unique division of the plane such that every region T_i (called Thiessen or Voronoi polygon) is closer to a particular point $\boldsymbol{x}_i$ than to any other $\boldsymbol{x}_j$. The union of these areas tessellate the whole plane without gaps or overlapping. This division is called first-order Voronoi diagram (Voronoi, 1908), and is defined as

$$T_i = \{\boldsymbol{x} \in \Re^2 : d(\boldsymbol{x}, \boldsymbol{x}_i) < d(\boldsymbol{x}, \boldsymbol{x}_j) \forall j \neq i\}, \tag{2.1}$$

where $d(\cdot, \cdot)$ denotes the euclidean distance between two points in $\Re^2$. Figure 2.1 shows the Voronoi diagram of a set composed of 9 nodes. The cells T_i and T_j are divided by an hyperplane perpendicular to the line that passes through nodes i and j. The intersection between three or more hyperplanes is called Voronoi node or vertex, and is equidistant to all the nodes s_i to which the neighboring cells T_i are related. The regions can be either closed or unbounded, but always convex.

A higher-order Voronoi tessellation can be defined by including more nodes into the definition. In this manner, a second-order cell T_{ij} is defined as the locus of all the points whose distance to $\boldsymbol{x}_j$ is less than to any other $\boldsymbol{x}_k$ but more than the distance to $\boldsymbol{x}_i$. This can be written as

$$T_{ij} = \{\boldsymbol{x} \in \Re^2 : d(\boldsymbol{x}, \boldsymbol{x}_i) < d(\boldsymbol{x}, \boldsymbol{x}_j < d(\boldsymbol{x}, \boldsymbol{x}_k) \forall k \neq j, i\} \tag{2.2}$$

This set is non-empty only in the case where T_i borders with T_j. In this case, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are neighbors

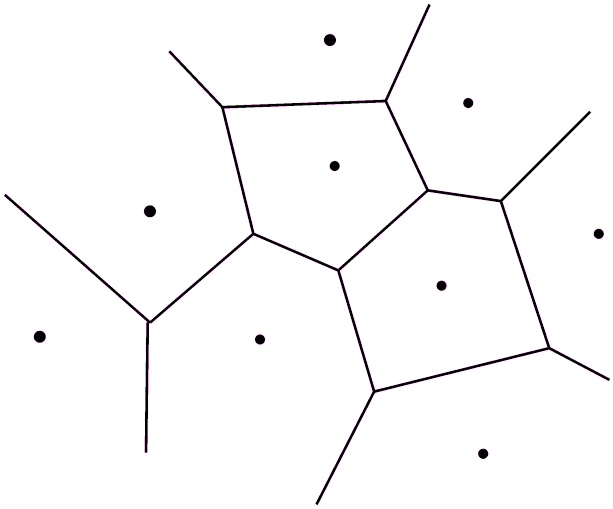


Figure 2.1. Voronoi Tessellation of a set S of points.

2.1.2 Delaunay Triangulation

The straight-line dual structure of the Voronoi tessellation is called Delaunay triangulation (Delaunay, 1934). It is constructed by connecting the nodes of S which are neighbors. Of all the possible three-node tesselations, the Delaunay triangles are the ones that maximize its minimal angle. This property make this graph very interesting for mesh generation and as such has been well studied (see Sukumar et al. (1998) and references therein). Figure 2.2 (b) depicts this triangulation. The Voronoi diagram of this set is shown in dashed lines.

If a circle passing through the three nodes of a triangle $DT(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$ (a circumcircle) is drawn, no other node of the set S will be enclosed by it. This property is called the empty circumcircle criterion (Fig 2.2 (a)). The circumcenter for each triangle is located at the Voronoi vertex where each node's cell converge. There exists a special case on which two (or more) triangles comply with this condition. This occurs when a number of nodes $k \geq d + 2$ are located on the same empty n-sphere, where $n = d - 1$. In two dimensions this means that at least 4 points are located in the same circle (1-sphere). Thus the Delaunay triangulation may not be unique for a given set of points (Fig 2.2 (c)).

The convex hull is the minimal external boundary of the set that contains all nodes of S. This concept has applications in numerous fields, ranging from image processing to GIS (Edelsbrunner et al., 1983). It can be obtained by means of the Delaunay triangulation as the union of all triangle segments that belong to only one triangle.

With the concepts of Voronoi tesselation and Delaunay triangulation, we can define

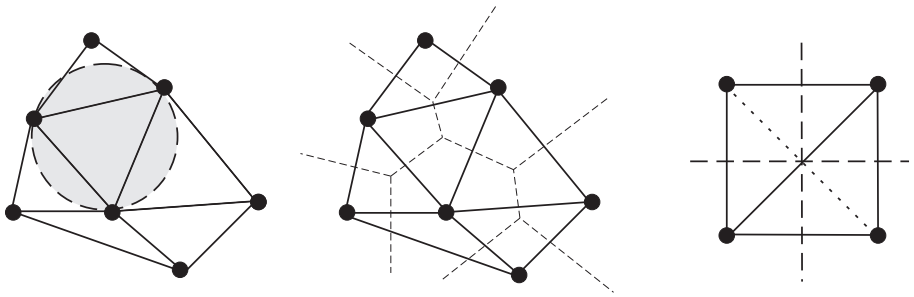


Figure 2.2. Empty circumcircle. Delaunay triangulation. Degenerated triangulation

the natural neighbors (n.n.) of a point $\boldsymbol{p}$ as those nodes whose cells limit with $T_{\boldsymbol{p}}$. As an alternative, they can also be defined as those nodes which share a triangle with $\boldsymbol{p}$, forming $DT(\boldsymbol{p}, \boldsymbol{x}_i, \boldsymbol{x}_j)$. The same definition applies whether $\boldsymbol{p}$ is a node of the set or an introduced point.

2.2 Natural Neighbour Interpolation

With the use of the second-order Voronoy diagram, the relation of neighborhood of an introduced point with the nearby nodes can be quantified. This is the principle used by Sibson (1980), when firstly proposing the natural neighbor interpolant, which is the basis of the NEM.

Years later, Belikov et al. (1997) showed the non-uniqueness of the natural neighbor interpolation schemes and introduced a new interpolant, known as non-Sibsonian (nS) or Laplacian interpolator. More recently, González et al. (2008) introduced a new class of interpolant constructed over a generalized de Boor algorithm. With this technique, it is possible to generate high-order consistency natural neighbor approximations.

2.2.1 Sibson Interpolant

Let $\boldsymbol{x}$ be a point introduced in a set of nodes $S = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$. And let $k_i(x)$ and $k(x)$ be the Lebesgue measures of the second order cell $T_{\boldsymbol{xi}}$ and the first order cell $T_{\boldsymbol{x}}$ respectively. Since the second order cell will be empty for every pair of nodes which are not neighbors, we have that $k_i(\boldsymbol{x}) = 0$ when i is not a natural neighbor of $\boldsymbol{x}$. From there we get that

$$\sum_{i=0}^{N} k_i(\boldsymbol{x}) = \sum_{i=0}^{n} k_i(\boldsymbol{x}) = k(\boldsymbol{x})$$

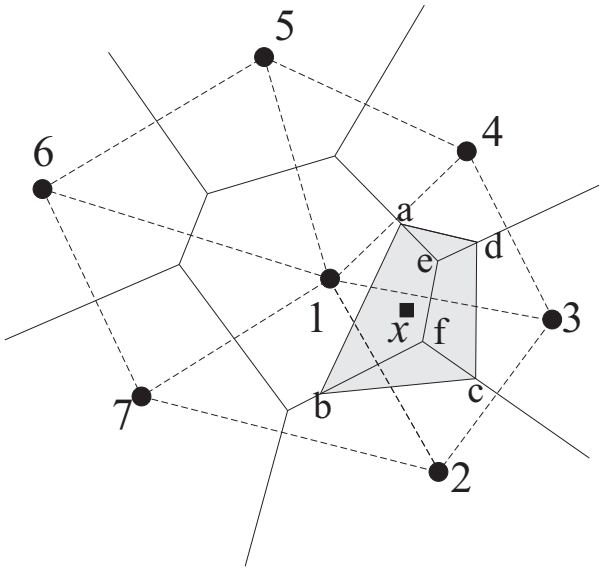


Figure 2.3. Modified Voronoi diagram by the inclusion of point $\boldsymbol{x}$

where n is the number of natural neighbors of point $\boldsymbol{x}$

The natural neighbor coordinate with respect to node i is defined as the ratio of $k_i(\boldsymbol{x})$ to $k(\boldsymbol{x})$. In two dimensions, the Lebesgue measure k is the area A of the cell.

$$\phi_i(\boldsymbol{x}) = \frac{k_i(\boldsymbol{x})}{k(\boldsymbol{x})} = \frac{A_{T_{\boldsymbol{xi}}}}{A_{T_{\boldsymbol{x}}}} \tag{2.3}$$

To illustrate the construction of the Sibson interpolant we will present an example. Figure 2.3 shows a set of 7 nodes to which a point $\boldsymbol{x}$ has been added. The coordinate respect to node 1 of this point is

$$\phi_1(x) = \frac{A_{abfe}}{A_{abcd}} \tag{2.4}$$

With the natural neighbor coordinates we can interpolate the value of some vectorial (or scalar) field $\boldsymbol{u}(x) : \Omega \subset \Re^2 \rightarrow \Re^2$ as

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{i=0}^{N} \phi_i(\boldsymbol{x})\boldsymbol{u}(i) = \sum_{i=0}^{n} \phi_i(\boldsymbol{x})\boldsymbol{u}(i) \tag{2.5}$$

where n is the number of natural neighbors of x.

Given the interpolant character of the natural neighbor coordinates and some other properties that will be treated next, they have been chosen as the shape functions in a Galerkin method, thus originating the Natural Element Method. Figure 2.4 depicts the Sibson shape function of a point centered in a four nodes square.

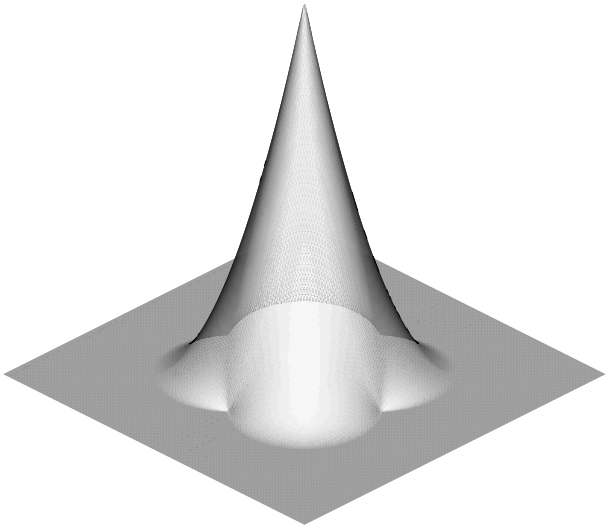


Figure 2.4. Sibson shape function. (Courtesy N. Sukumar)

2.2.2 Non-Sibson Interpolant

As mentioned before, Belikov et al. (1997) proposed a different interpolation scheme based on natural neighbors. This new interpolant required the calculation of Lebesgue measures in one dimension less than the working dimension. With this approach, to obtain the natural natural neighbor coordinates in $\Re^2$ only distances have to be calculated instead of areas, as required by the Sibson shape function. Since both schemes share most of their properties, the non-sibson interpolant presents itself as a very attractive option due to its ease of implementation and lower computational cost.

To define this interpolant we will assume again a finite set $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ in $\Re^n$. Based on the Voronoi cell, Eq. (2.1), and its closure $\bar{T}_i = T_i \cup \delta T_i$, we define $t_{ij} = \{\boldsymbol{x} \in \bar{T}_i \cap \bar{T}_j, i \neq j\}$. With $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as the distance between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the non-Sibson interpolator is

$$\phi_i(\boldsymbol{x}) = \frac{\frac{|t_{\boldsymbol{x}\boldsymbol{i}}|}{d(\boldsymbol{x}_i,\boldsymbol{x})}}{\sum_{j=1}^{m} \frac{|t_{\boldsymbol{x}\boldsymbol{j}}|}{d(\boldsymbol{x},\boldsymbol{x}_j)}} \tag{2.6}$$

where $|\cdot|$ denotes the Lebesgue measure in $\Re^{n-1}$. As with the second order Voronoi cell, t_{ij} is non-empty only for pairs of neighboring nodes.

Figure 2.5 shows a set comprised of four nodes in which a point $\boldsymbol{x}$ is inserted. The Voronoi cell $T_{\boldsymbol{x}}$ is also depicted. In this setting, $|t_{\boldsymbol{x}4}|$ is the Lebesgue measure of the edge $\boldsymbol{x}4$ and $d(\boldsymbol{x}, 4)$ is $2 \cdot h4$. In Belikov et al. (1997) the non-Sibson shape function is defined as

$$\phi_i(x) = \frac{\alpha_i(x)}{\sum_{j=1}^{n} \alpha_j(x)}, \quad \alpha_i(x) = \frac{s_i(x)}{h_i(x)} \tag{2.7}$$

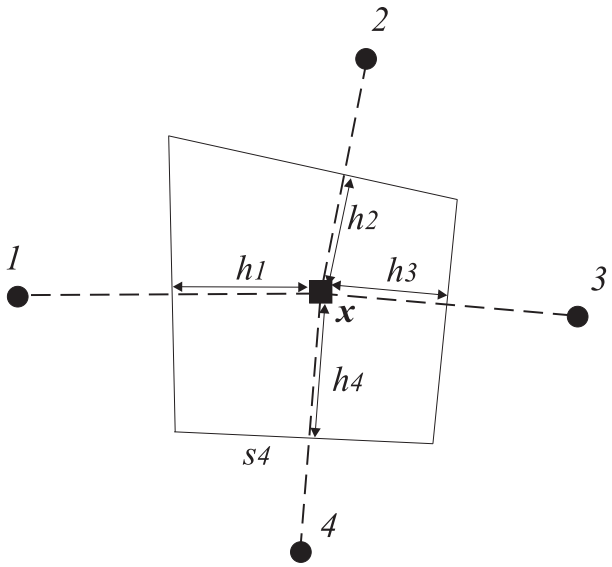


Figure 2.5. Laplace shape function construction.

In Fig. 2.6, a picture of the non-Sibson shape function is shown in the same situation as was presented the Sibson shape function.

2.3 Natural Element Shape Function Properties

The NEM has interesting qualities which derive from the use of the interpolation functions explained in the past sections. In this sections we will cover such properties in some detail. Unless stated otherwise, the notation will be the same used before.

2.3.1 Meshless Character

It is a known fact (Babuška and Aziz, 1976) that the accuracy of many approximation methods is dependent on the nodal distribution. As an example, the triangulation used in the FEM must comply with a minimum angle criterion in order to guarantee a minimum level of accuracy. However the NEM shape functions are not limited by such requirement, being independent of the nodal distribution. While this condition has not been formally demonstrated, studies in two (Sukumar, 1998) and in three (Cueto, 2001) dimensions have not found any dependence between the angles of the Delaunay triangles and the accuracy of the obtained results, hence implying a meshless character.

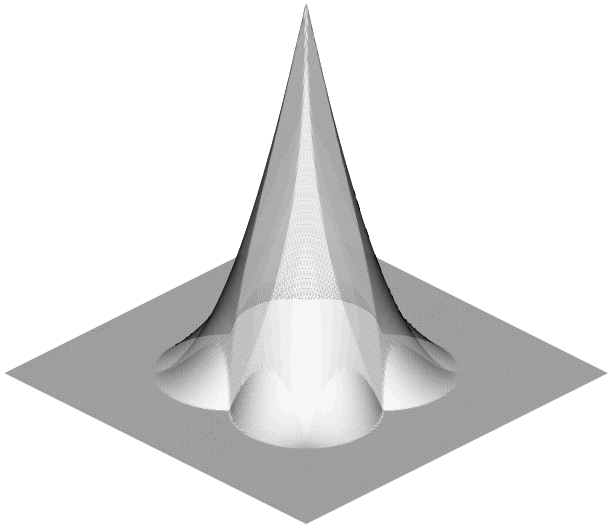


Figure 2.6. Laplace shape function (Courtesy N.Sukumar)

Indeed, in Alfaro et al. (2007) numerical tests are performed that show the superior accuracy of NEM over FEM when highly distorted meshes are employed.

2.3.2 Strictly Interpolant Character

Unlike most meshless methods, which are of approximate character, the NEM shape functions are strictly interpolant. That is, the approximated surface contains the nodal values. It satisfies the Kronecker delta condition at the nodes:

$$\phi_i(\boldsymbol{x}_j) = \delta_{ij} \tag{2.8}$$

Applying Eq. (2.5) to node i

$$\boldsymbol{u}^h(\boldsymbol{x}_i) = \sum_{j=1}^{n} \phi_i(\boldsymbol{x}_j)\boldsymbol{u}_j = \sum_{j=1}^{n} \delta_{ij}\boldsymbol{u}_j = \boldsymbol{u}_i,$$

which shows that the nodal parameters are directly the nodal variables

As a consequence of this property, there is no need to use enforcing techniques to impose essential conditions while solving PDEs. However, this property alone is not sufficient to properly impose Dirichlet conditions, for the nodal satisfaction at the boundary nodes does not imply full compliance within the boundary.

2.3.3 Approximation

Whereas in one dimension Sibson's shape functions equals that of the natural elements, non-Sibsonian shape functions are undefined (the Lebesgue measure of a point equals

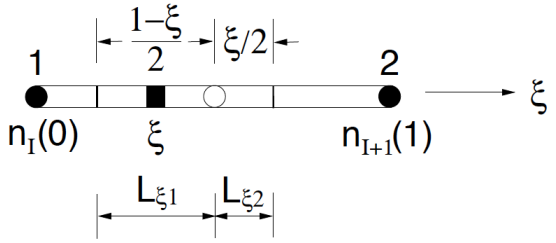


Figure 2.7. Natural neighbor coordinates in 1D.

zero). To prove it, let us consider a line segment of length L discretized in N elements. By definition the Voronoi nodes in one dimension will be located in the middle points of each element. The natural consequence of this is that each node will have only two natural neighbors. By establishing a coordinate system (see Fig. 2.7)

$$\xi = \frac{\boldsymbol{x} - \boldsymbol{x}_i}{\boldsymbol{x}_{i+1} - \boldsymbol{x}_i}, \tag{2.9}$$

with $\xi \in [0, 1]$. The shape functions for each element are written as

$$\phi_i(\xi) = \frac{L_{\xi i}}{L_{\xi 1} + L_{\xi 2}}, \tag{2.10}$$

with

$$L_{\xi 1} = \frac{1-\xi}{2} \quad \text{y} \quad L_{\xi 2} = \frac{\xi}{2},$$

from which we get to $\phi_1(\xi) = 1 - \xi$ and $\phi_2(\xi) = \xi$

In two dimensions the approximation depends on the number of natural neighbors. If the point has three n.n. the natural coordinates are equivalent to the barycentric coordinates that constitute the linear triangular FE shape functions (constant strain triangles, see Sukumar et al. (1998)). To prove this, lets take a point $\boldsymbol{x} = (x, y)$ with only three neighbors, numbered 1 to 3 with coordinates (x_i, y_i). Due to method's linear consistency, it is possible to write the following system

$$D(\boldsymbol{x}) = \begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} \begin{pmatrix} \phi_1(\boldsymbol{x}) \\ \phi_2(\boldsymbol{x}) \\ \phi_3(\boldsymbol{x}) \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ y \end{pmatrix} \tag{2.11}$$

which is solved by

$$\phi_1(\boldsymbol{x}) = \frac{D_1(\boldsymbol{x})}{D(\boldsymbol{x})} \tag{2.12}$$

$$\phi_2(\boldsymbol{x}) = \frac{D_2(\boldsymbol{x})}{D(\boldsymbol{x})} \tag{2.13}$$

$$\phi_3(\boldsymbol{x}) = \frac{D_3(\boldsymbol{x})}{D(\boldsymbol{x})} \tag{2.14}$$

being

$$D(\boldsymbol{x}) = \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} = 2A_{123} \tag{2.15}$$

and $D_1(\boldsymbol{x}) = 2A_1(\boldsymbol{x})$, $D_2(\boldsymbol{x}) = 2A_2(\boldsymbol{x})$ y $D_3(\boldsymbol{x}) = 2A_3(\boldsymbol{x})$, which are the FE shape functions that we were looking for.

For four nodes in a regular grid a bilinear approximation is obtained. For any other configuration or in the case of more neighbors, the shape functions have a quartic rational expression—see Sukumar et al. (1998) for proof.

In higher dimensions the Sibson shape function is a generalization of the univariate 1-D interpolation. The interpolant is built based on the appropriate dimension Lebesgue measure, which gives it very sound mathematical and geometric base.

2.3.4 Partition of Unity and Positivity

Because the NEM shape functions are constructed —Eqs. (2.3) and (2.7)—as ratios of a quantity related to each natural neighbor between the sum of all those quantities, it is direct to see that

$$\sum_{i=1}^{n} \phi_i(\boldsymbol{x}) = 1. \tag{2.16}$$

That is, the shape functions constitute a partition of unity.

Since the mentioned quantities are areas or distances, it is also easy to show that $\phi_i \geq 0$. Which means that the Natural Element Interpolation constitutes a convex combination of nodal coordinates (Arroyo and Ortiz, 2006).

2.3.5 Linear Consistency

As stated by Sibson (1980), the NEM shape functions satisfy the local coordinate property

$$\boldsymbol{x} = \sum_{i=1}^{n} \phi_i(\boldsymbol{x}) \boldsymbol{x_i} \tag{2.17}$$

which along with the partition of unity condition imply that the method posses linear consistency. Proof of this can be found on Sukumar (1998) for the Sibson interpolant and in Sukumar et al. (2001) for the non-Sibsonian approximation. Being able to reproduce a linear field, means that this shape functions can be used to solve PDEs of degree two, e.g., the elastostatic problem.

2.3.6 Smoothness

According to Sibson, NEM shape functions are infinitely differentiable everywhere inside its domain except at the nodes, where they are only continuous. Various modifications have been proposed in order to raise the differentiability class of this shape functions. Some of this approaches include the use a weighted least squares fit as a modifier in the original Sibson scheme (Sibson, 1981); embedding natural neighbor coordinates in the surface representation of a Bernstein-Bézier cubic simplex (Farin, 1990); or reformulating Sibson's interpolant to incorporate them into spline theory (Traversoni, 1994). Achieving to obtain a $\mathcal{C}^1(\Omega)$ class shape function allows to make use of the NEM in higher order problems.

2.4 Shape Function Construction

In this section we will discuss the two most extended methods for the shape function calculation. The Bowyer-Watson is based on the calculation of the natural neighbor coordinates by decomposing the Voronoi cells into triangles which are a subset of the Delaunay triangulation. The second, Laserre's algorithm performs the task by calculating areas of convex polygons.

2.4.1 Useful formulas

These are some geometric formulas which are used on the implementation of the following algorithms (Sukumar et al., 1998). We will consider three two-dimensional non-collinear points in global coordinates: $A(\boldsymbol{a}) = (a_1, a_2)$, $B(\boldsymbol{b}) = (b_1, b_2)$ and $C(\boldsymbol{c}) = (c_1, c_2)$ forming a triangle $t(A, B, C)$.

Area of a Triangle

The signed area of t_{abc} is given by

$$A = \frac{(a_1 - c_1)(b_2 - c_2) - (b_1 - c_1)(a_2 - c_2)}{2} \tag{2.18}$$

If $\boldsymbol{a}$ and $\boldsymbol{b}$ coordinates are dependent on $\boldsymbol{x}$, the derivatives of A is

$$\begin{aligned} A_{,i}(\boldsymbol{x}) = & \frac{(a_1(\boldsymbol{x}) - c_1)b_{2,i}(\boldsymbol{x}) + (b_2(\boldsymbol{x}) - c_2)a_{1,i}(\boldsymbol{x})}{2} \\ & - \frac{(b_1(\boldsymbol{x}) - c_1)a_{2,i}(\boldsymbol{x}) + (a_2(\boldsymbol{x}) - c_2)b_{1,i}(\boldsymbol{x})}{2} \end{aligned} \tag{2.19}$$

where $i = 1, 2$, denotes the spatial coordinate and a comma denotes differentiation in the corresponding direction. This definition can be extrapolated to higher dimensions.

Circumcenter Coordinates

The circumcenter of $t(A, B, C)$ denoted by $\boldsymbol{v} = (v_1, v_2)$ is obtained by:

$$\begin{aligned} v_1 &= \frac{(a_1^2 - c_1^2 + a_2^2 - c_2^2)(b_2 - c_2) - (b_1^2 - c_1^2 + b_2^2 - c_2^2)(a_2 - c_2)}{D}, \\ v_2 &= \frac{(b_1^2 - c_1^2 + b_2^2 - c_2^2)(a_1 - c_1) - (a_1^2 - c_1^2 + a_2^2 - c_2^2)(b_1 - c_1)}{D} \end{aligned} \tag{2.20}$$

where D equals four times the area of $t(A, B, C)$—Eq. (2.18).

To calculate the derivatives of the circumcenter's coordinates, make $\boldsymbol{c} = \boldsymbol{x} = (x_1, x_2)$. Assume $\boldsymbol{a}$ and $\boldsymbol{b}$ are independent of $\boldsymbol{x}$. Then

$$\begin{aligned} v_{1,1}(\boldsymbol{x}) &= \frac{(x_1 - v_1(\boldsymbol{x}))C_{,1}(\boldsymbol{x})}{D(\boldsymbol{x})}, \\ v_{1,2}(\boldsymbol{x}) &= \frac{(\alpha + x_2 D_{,1}(\boldsymbol{x}) - v_1(\boldsymbol{x})D_{,2}(\boldsymbol{x})}{D(\boldsymbol{x})}, \\ v_{2,1}(\boldsymbol{x}) &= \frac{(-\alpha + x_1 D_{,2}(\boldsymbol{x}) - v_2(\boldsymbol{x})D_{,1}(\boldsymbol{x}))}{D(\boldsymbol{x})}, \\ v_{2,2}(\boldsymbol{x}) &= \frac{(x_2 - v_2(\boldsymbol{x}))D_{,2}(\boldsymbol{x})}{D(\boldsymbol{x})} \end{aligned} \tag{2.21}$$

with $\boldsymbol{v}$ given by Eqs. (2.20), $D(\boldsymbol{x})$ equals four times the area of $t(A, B, X)$ and

$$D_{,1}(\boldsymbol{x}) = 2(a_2 - b_2), \tag{2.22}$$

$$D_{,2}(\boldsymbol{x}) = 2(b_1 - a_1) \tag{2.23}$$

and

$$\alpha = b_1^2 - a_1^2 + b_2^2 - a_2^2 \tag{2.24}$$

Circumradius

The square of the circumradius $R^2(\boldsymbol{x})$ is

$$R^2(\boldsymbol{x}) = (a_1 - v_1(\boldsymbol{x}))^2 + (a_2 - v_2(\boldsymbol{x}))^2 \tag{2.25}$$

2.4.2 Bowyer-Watson Algorithm

The Bowyer-Watson algorithm (Watson, 1981) is originally an incremental insertion algorithm to obtain the Delaunay triangulation of a set S. It starts with a super-triangle

that encompasses all the set. One node is added, the algorithm is carried on and then another node is added until all the set has been processed. Yet for obtaining the areas of the second order Voronoi cells, we will assume that the starting point already is a Delaunay triangulation.

At each step, a node $\boldsymbol{x}$ is inserted into the triangulation; a search is performed on a triangle-by-triangle basis to determine the natural neighbors of $\boldsymbol{x}$ using the empty circumcircle criterion.

For each of the n neighboring triangles t (with circumcenter $\boldsymbol{v}$), a new set of triangles $\{t_1, t_2, t_3\}$ converging at $\boldsymbol{x}$ is created. The circumcenters ($\boldsymbol{c}_1(\boldsymbol{x})$, $\boldsymbol{c}_2(\boldsymbol{x})$ and $\boldsymbol{c}_3(\boldsymbol{x})$) of the new sets are obtained and second subset of triangles is formed from the newly calculated points and $\boldsymbol{v}$. This triangles are: $t(\boldsymbol{c}_2(\boldsymbol{x}), \boldsymbol{c}_3(\boldsymbol{x}), \boldsymbol{v})$ $t(\boldsymbol{c}_3(\boldsymbol{x}), \boldsymbol{c}_3(\boldsymbol{x}), \boldsymbol{v})$ and $t(\boldsymbol{c}_1(\boldsymbol{x}), \boldsymbol{c}_2(\boldsymbol{x}), \boldsymbol{v})$. The areas of this triangles and its derivatives are calculated by Eqs. (2.18) and (2.19) respectively. This are partial results which can be written as $\alpha_{it}(\boldsymbol{x})$ and $\alpha_{it,m}(\boldsymbol{x})$, where $i = 1, 2, \ldots, n$ and $m = 1, 2$. Up to this point, care must be taken to maintain the node numbering always counterclockwise in each triangle in order to obtain areas properly signed. Note that some areas must not contribute to the total result, and those areas will be negative, but that is accounted for.

The partial results for each t are added on accumulator variables $\beta_i(\boldsymbol{x})$ and $\beta_{i,m}(\boldsymbol{x})$. Finally the area of the second order Voronoi cell $A_i(\boldsymbol{x})$ and its derivatives $A_{i,m}(\boldsymbol{x})$ are the values stored in the accumulator after the pass for all neighbor triangles.

The area of the first order Voronoi cell and its derivative are obtained as

$$A(\boldsymbol{x}) = \sum_{i=1}^{n} A_i(\boldsymbol{x}), \tag{2.26}$$

$$A_{,m}(\boldsymbol{x}) = \sum_{I=1}^{n} A_{i,m}(\boldsymbol{x}) \tag{2.27}$$

and the shape functions are obtained as per Eq. (2.4).

Regarding the non-sibsonian shape functions in $\Re^2$, the steps involving the area calculations can be skipped. Having the the circumcenters coordinates $\boldsymbol{c}_i(\boldsymbol{x})$ it is possible to obtain all the necessary distances to calculate the nS shape functions.

This algorithm fails if the inserted point falls on a Delaunay edge, as one of the triangles formed will have a non-unique circumcenter. In any case, since we only insert integration points —which are internal to the triangles—this problem will not appear.

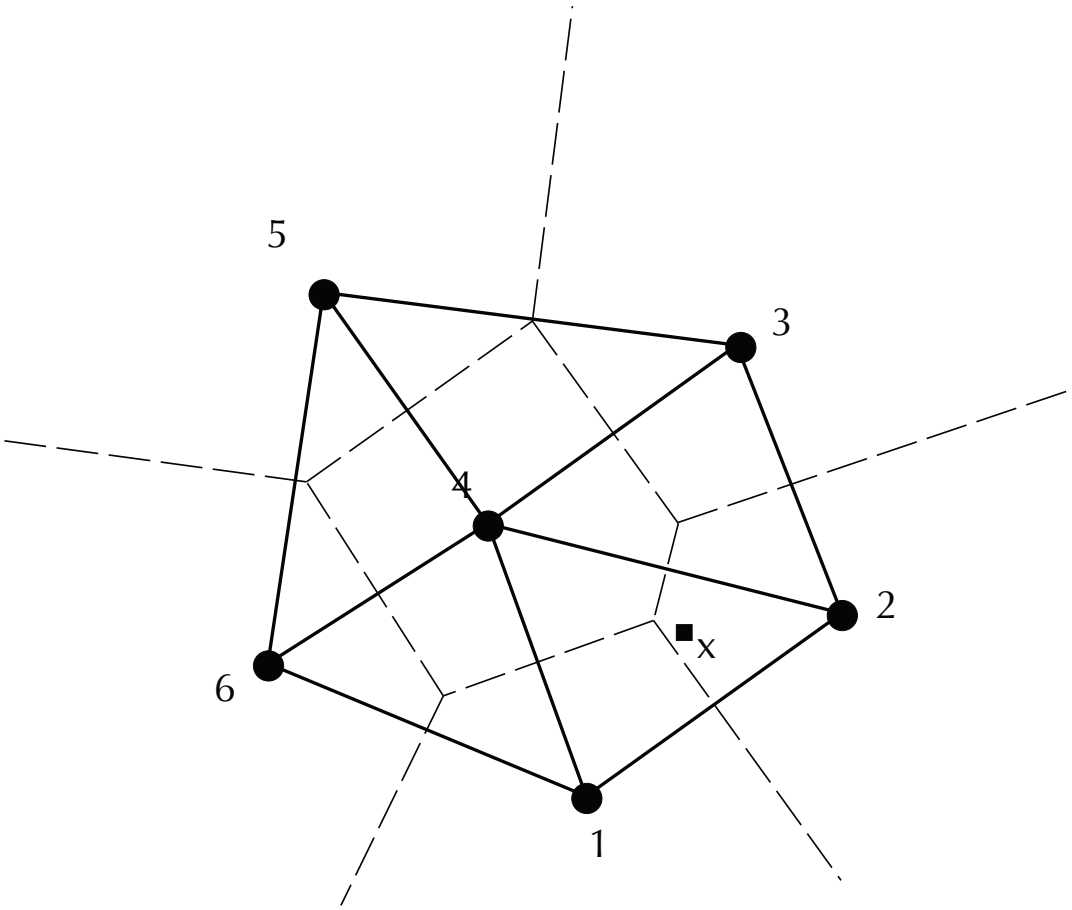


Figure 2.8. Starting point for the Bowyer-Watson algorithm.

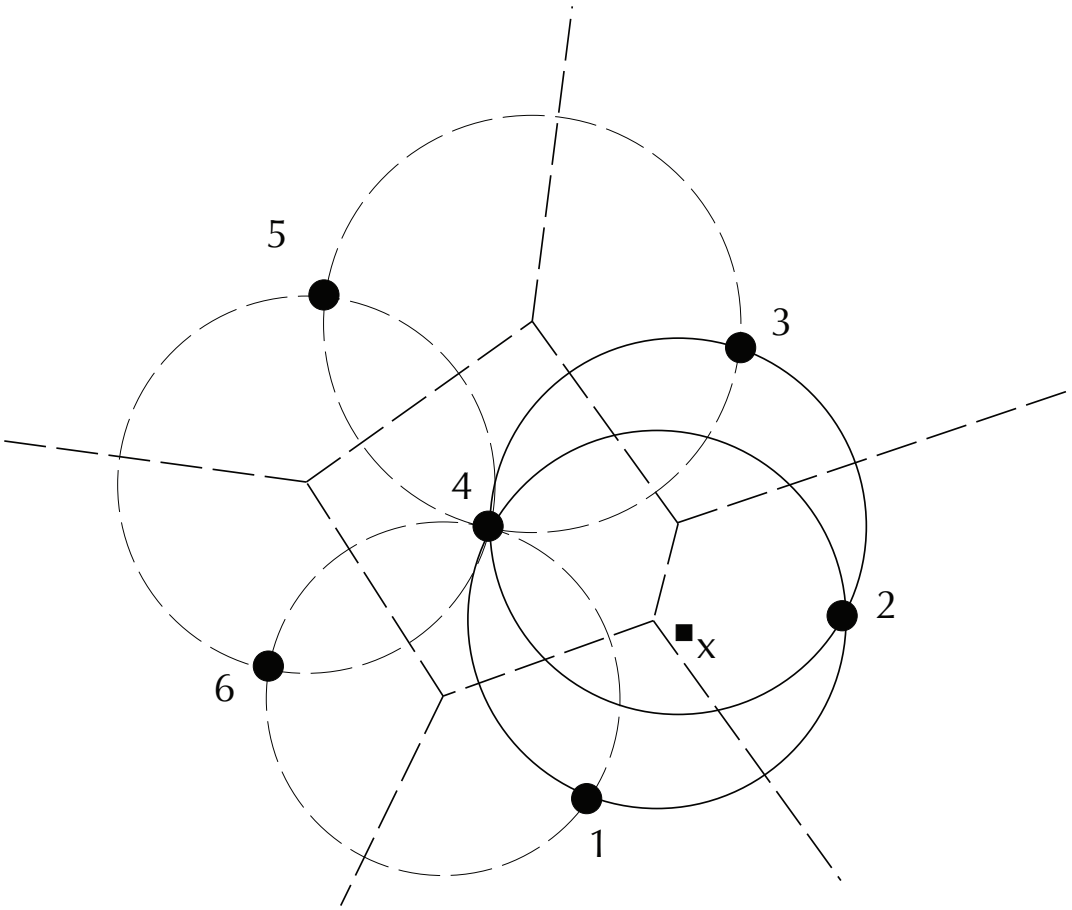


Figure 2.9. Natural neighbor search.

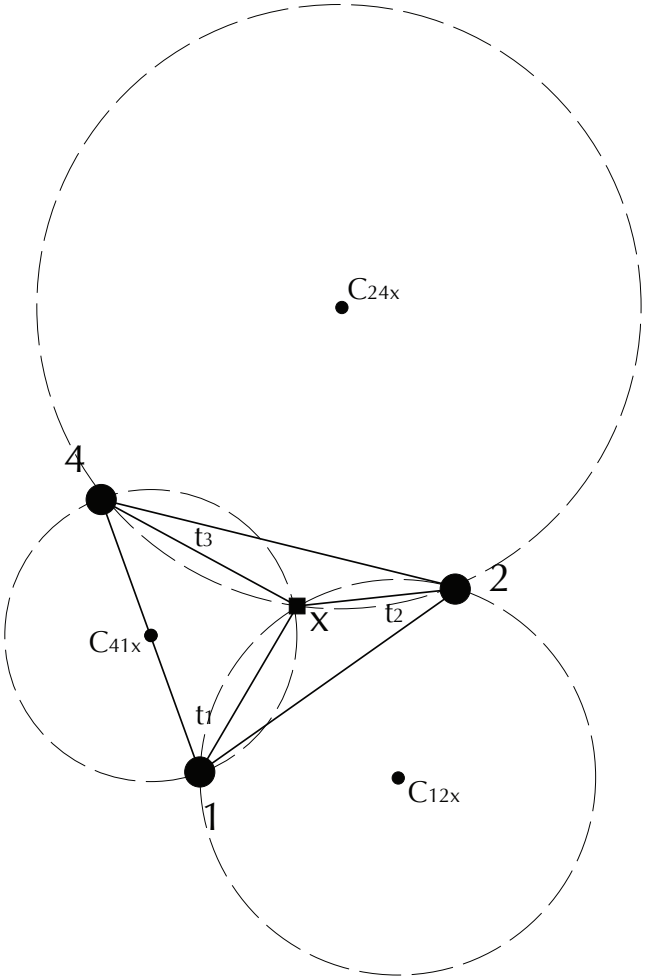


Figure 2.10. Triangles t_i and their circumcenters $\boldsymbol{c}_i(\boldsymbol{x})$.

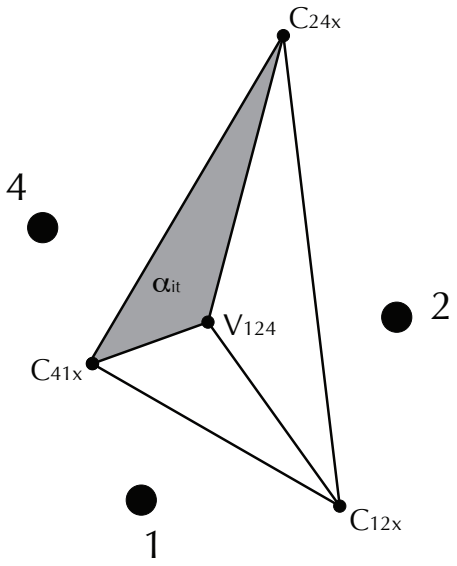


Figure 2.11. Subset of triangles($t(\boldsymbol{c}_i(\boldsymbol{x}), \boldsymbol{c}_j(\boldsymbol{x}), \boldsymbol{v})$) and their areas α_{it}.

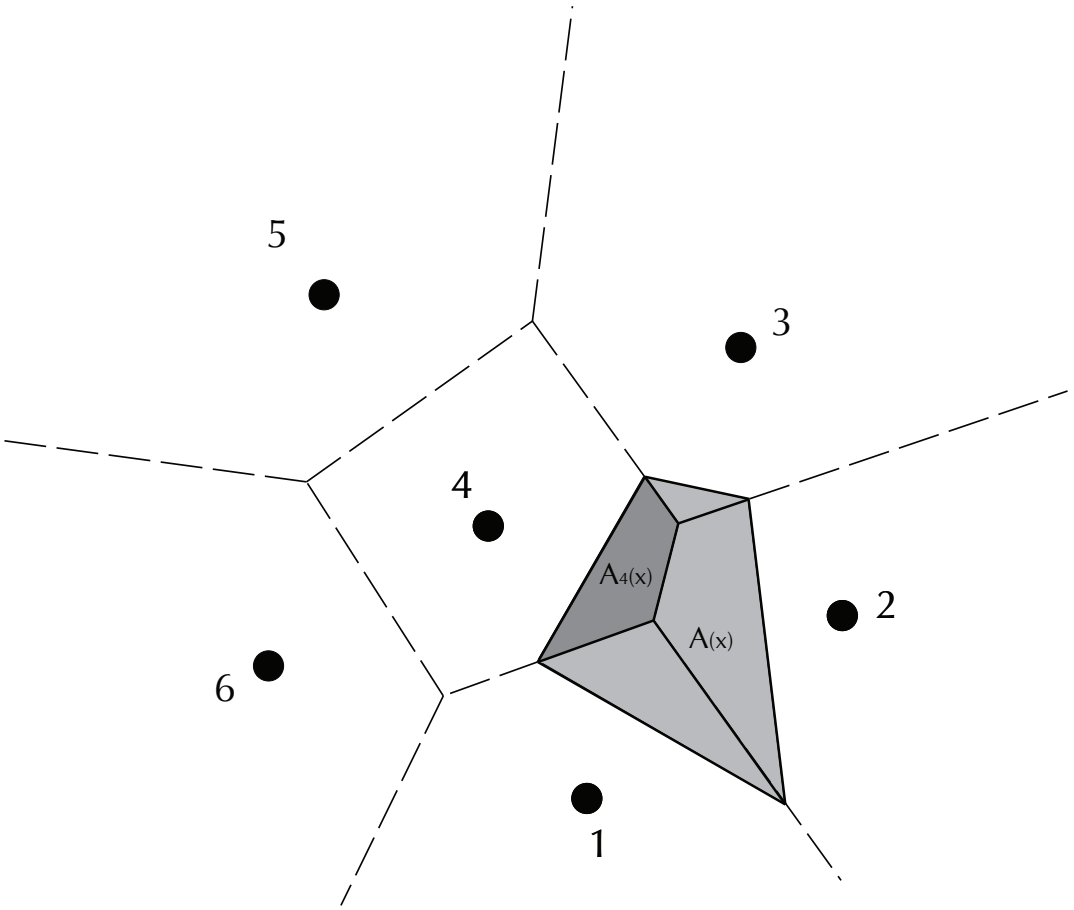


Figure 2.12. The sum of α_{it} over all t_is yields the area of the second order Voronoi cells.

2.4.3 Laserre's Algorithm

A different approach is to directly obtain the areas of the Voronoi cells without resorting to the duality with the Delaunay triangulation. In their implementation, Braun and Sambridge (1995) made use of the Lasserre algorithm (Lasserre, 1983) for obtaining the convex polygon areas (polyhedra volumes in $\Re^3$) that define natural neighbor coordinates. This algorithm works in a recursive way, presenting the polytope volume in $\Re^n$ as a function of volumes of polytopes in $\Re^{n-1}$.

The main virtues of this method is that is defined for $\Re^n$ and is independent on the relative position between the nodes and the evaluation point. However, has been found to be approximately twice as computationally expensive as the Bowyer-Watson algorithm.

The algorithm starts by defining the volume of a convex polytope as the inequality

$$\{c|\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\} \tag{2.28}$$

where $\boldsymbol{x}$ represents a point in $\Re^n$, $\boldsymbol{A}$ is a (m, n) matrix and $\boldsymbol{b}$ is a column vector of size m. Here n is the dimension on which the polytope is defined and m is the number of non-redundant constraints that define the volume. This enclosed volume is denoted as

$$V(n, \boldsymbol{A}, \boldsymbol{b}) \tag{2.29}$$

The i-th face of the polytope is defined as

$$\{\boldsymbol{x}|(\boldsymbol{a}_i \cdot \boldsymbol{x}) = b_i, \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\} \tag{2.30}$$

with $\boldsymbol{a}_i$ representing the i-th column of $\boldsymbol{A}$.

The polytope volume will be obtained as:

$$V(n, \boldsymbol{A}, \boldsymbol{b}) = \frac{1}{d}\sum_{i=1}^{m} d(\boldsymbol{a}, \boldsymbol{H}_i) \times V_i(n-i, \boldsymbol{A}, \boldsymbol{b}) \tag{2.31}$$

where $\boldsymbol{a}$ is the evaluation point, $\boldsymbol{H}_i$ is the hyperplane defined by the i-th constrain. $d(\boldsymbol{a}, \boldsymbol{H_i})$ is the distance from $\boldsymbol{a}$ to the hyperplane $\boldsymbol{H}_i$.

We now eliminate the t-th variable by solving $\boldsymbol{a}_i\boldsymbol{x} = b_i$. The reduced matrix obtained from $\boldsymbol{A}$ after the elimination is denoted by $\bar{\boldsymbol{A}}_{i,t}$. Likewise, the reduced $\boldsymbol{b}$ vector will be called $\bar{\boldsymbol{b}}_t$ and a_{it} is the t-th term of $\boldsymbol{a}_i$. With this notation, the formula for the volume calculation is

$$V(n, \boldsymbol{A}, \boldsymbol{b}) = \frac{1}{n}\sum_{i=0}^{m} \frac{b_i}{|a_{it}|} V'_{it}(n-1, \bar{\boldsymbol{A}}_{i,t}, \bar{\boldsymbol{b}}_t) \tag{2.32}$$

where $V_{it}^{'}$ is the volume of the polytope in the $(n-1)$-dimenson. In $\Re^3$, $V_{it}^{'}$ would be the area of the facets of the polyhedron, and in $\Re^2$ would be the length of the edges of the polygon. In their work, Braun and Sambridge Braun and Sambridge (1995) chose t such that $|a_{it}|$ is the maximum value of $\boldsymbol{a}_i$.

2.5 Imposition of Essential Boundary Conditions

The NEM shape functions will present different approximation properties depending on the type of boundaries present. Sukumar (1998), Sukumar et al. (2001) demonstrated that Sibson and non-Sibsonian shape functions have a linear behavior on convex boundaries, being able to reproduce a linear displacement field. This means that in those contours the essential boundary conditions can be imposed exactly as in finite elements.

In the case of non-convex boundaries, the Sibson interpolant does not comply with this condition. The natural coordinates at the contour will have non-zero contributions from the interior nodes. This contribution will cause the loss of the linear consistency. According to Sukumar (1998) the error incurred by the use of this interpolant may be of the order of 2 per cent, and can be mitigated by a higher nodal density on that boundary.

It was believed that this problem is not present when using the Laplace shape functions (Sukumar et al., 2001). Thus by choosing non-Sibsonian interpolants, essential boundary conditions could be imposed directly, in the same way as in the FEM. However, Cueto et al. (2003) demonstrated that this is not the case and proposed a different approach, discussed in the next section, which allows the proper imposition of the aforementioned conditions.

2.6 α-shape Based Natural Elements

A slight modification of the way in which the Natural Neighbour interpolant is built was proposed to achieve linear interpolation also over non-convex boundaries (Cueto, 2001)(Cueto et al., 2000). This modification was based on the concept of α-shapes, which will be exposed in more detail on Chapter 4. These are a generalization of the

concept of the convex hull of a cloud of points and are widely used in the field of scientific visualization and computational geometry to give a shape to a set of points.

It is a polytope that is not necessarily convex and that can be triangulated by a subset of the Delaunay triangulation, thereby maintaining the empty circumcircle criterion. It has been demonstrated (Cueto, 2001) how the construction of the interpolant over an appropriate α-shape of the domain gives rise to an exact imposition of essential boundary conditions over any kind of domain (convex or not.) In addition, it enables us to track the flow front position accurately.

This variant of the NEM is based on the fact that if a pair of nodes are neighbors, thew will share a Voronoi cell frontier an also will define the edge of a Delaunay triangle. The α-shape restricts this condition to all nodes which are apart by a distance defined by a parameter α.

The shape functions are build based on a modified definition of the Voronoi cell:

$$T_i = \{\boldsymbol{x} \in \Re^2 / d(\boldsymbol{x}, \boldsymbol{s}_i) < d(\boldsymbol{x}, \boldsymbol{s}_j) \wedge \sigma_T \in C_\alpha(S) \quad \forall j \neq i\} \tag{2.33}$$

where σ_T is a k-simplex formed by nodes s_i,s_j and any other node of the set S. $C_\alpha(S)$ is the set of all triangles of $DT(S)$ whose circumradius is less than a given α. The shape functions are obtained in the traditional way, either with Eq. (2.3) or (2.7). This functions will have linear consistency over all the contour, be it convex or not (Cueto et al., 2000).

2.7 High-Order NEM

Precisely the fact that the NEM possesses linear consistency and $\mathcal{C}^0$ continuity only is perhaps on the basis of its limited popularity, if compared with other meshless methods, which easily achieve higher-order consistency and even $\mathcal{C}^\infty$ continuity. Only one attempt has been made to overcome this difficulty, up to our knowledge, by applying a quadratic consistency and $\mathcal{C}^1$ interpolant based on natural neighbours that, however, does not seem to posses any further generalization (Sukumar and Moran, 1999). This interpolant can be used, for instance, for solving fourth-order partial differential equations such as those arising from the theory of Kirchhoff plate bending.

An attempt to overcome these NEM limitations was performed by González et al. (2008), by going back to the foundations of B-splines and how by linear combinations of linear interpolants, higher-order curves can be obtained. B-spline curves can be obtained by means of the so-called de Boor's algorithm (Farin, 2002). For the surface case,

tensor product B-spline surfaces were initially proposed in de Boor (1962). An extensive review of this topic can be found in Farin (2002, Ch. 16). Tensor product B-spline surfaces are, however, very rigid. For instance, no tensor product surface can have the connectivity of a double torus. This algorithm is here generalized, without the use of tensor products, to higher dimensions. This is done by employing different natural neighbour interpolation schemes.

Recently, the use of NURBS or B-splines as basis functions for a Finite Element-like simulation has been studied (Hughes et al., 2005). The main objective is clear: B-splines (or, more properly, NURBS) are the standard for CAD systems, and they reproduce the geometry of the domain exactly, which is not the case in Finite Element models. The use of the same approximation for both the construction of the geometry and the approximation of the essential field of the problem obviously simplifies the burdensome mesh generation process. Other interesting properties of NURBS like the so-called "variation diminishing" (Farin, 2002) property also apply. This means that, unlike high-order polynomials, B-splines or NURBS do not show the well-known Gibbs effect (Gibbs, 1898).

However, it remains unclear whether the use of B-splines-like finite elements leads to remeshing problems when large distortions of the mesh occur. Tensor-product B-splines, as mentioned before, are quite rigid. In addition, fulfillment of (inhomogeneous) essential (Dirichlet) boundary conditions should be done typically in an approximate sense, or by imposing them weakly. This same problem is common for many meshless methods.

Another aspect deserves some comments at this point. Stability restrictions imposed by the LBB condition (Babuška, 1973) when simulating incompressible media makes it interesting to have at hand high-order approximations that could help in verifying the LBB condition. As it is well-known, the higher the approximation is for displacements (or velocities) and lower it is for pressure, the more stable is the resulting approximation. Thus, the development of high-order natural elements is interesting also from the point of view of the problem here tackled, that of free-surface flow of incompressible fluids.

2.7.1 Revisiting the de Boor's algorithm for B-splines

The de Casteljau algorithm for Bézier curves states that such curves can be obtained by successive application of linear interpolation (Farin, 2002), i.e., given some points

$\boldsymbol{b}_0, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \Re^3$ and $t \in \Re$, the construction

$$\boldsymbol{b}_i^r(t) = (1-t)\boldsymbol{b}_i^{r-1}(t) + t\boldsymbol{b}_{i+1}^{r-1}(t) \text{ with } \begin{cases} r = 1, \ldots, n \\ i = 0, \ldots, n-r \end{cases}, \tag{2.34}$$

where $\boldsymbol{b}_i^0 = \boldsymbol{b}_i$, gives the desired Bézier curve.

The de Boor's algorithm generalizes this algorithm by introducing a parametric space, defined by an arbitrary sequence of knots u_0, u_1, u_2, u_3. A quadratic Bézier curve can thus be seen as parametrized by the series $0, 0, 1, 1$, for instance. The quadratic blossom $\boldsymbol{b}[u, u]$ can then be written as (see Fig. 2.13)

$$\begin{aligned} \boldsymbol{b}[u,u] &= \frac{u_2-u}{u_2-u_1}\boldsymbol{b}[u_1,u] + \frac{u-u_1}{u_2-u_1}\boldsymbol{b}[u,u_2] \\ &= \frac{u_2-u}{u_2-u_1}\left(\frac{u_2-u}{u_2-u_0}\boldsymbol{b}[u_0,u_1] + \frac{u-u_0}{u_2-u_0}\boldsymbol{b}[u_1,u_2]\right) \\ &+ \frac{u-u_1}{u_2-u_1}\left(\frac{u_3-u}{u_3-u_1}\boldsymbol{b}[u_1,u_2] + \frac{u-u_1}{u_3-u_1}\boldsymbol{b}[u_2,u_3]\right) \end{aligned} \tag{2.35}$$

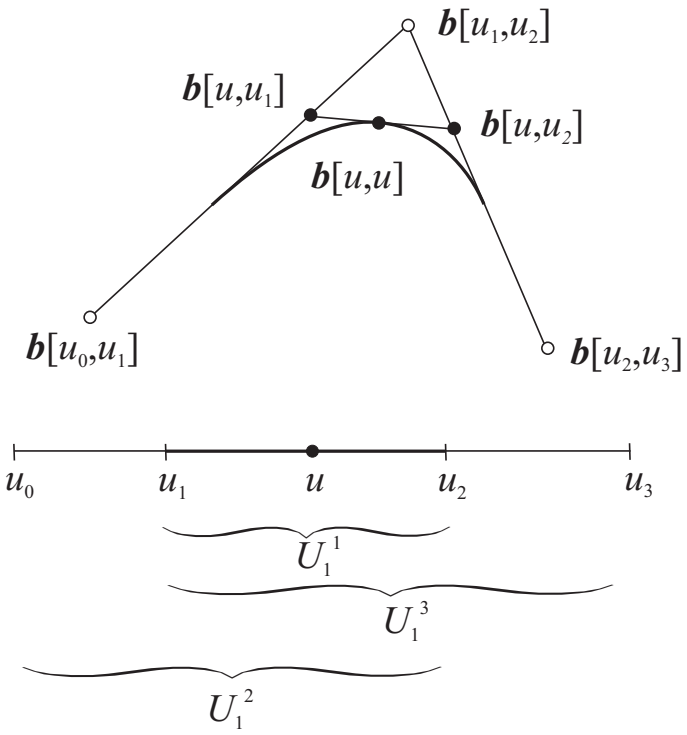


Figure 2.13. Schematic representation of the de Boor's algorithm.

The key aspect of the de Boor's algorithm is that it expresses u in terms of intervals of growing size. B-spline curves consist of a union of polynomial curve segments. Following the notation in Farin (2002), let U be an interval $[u_I, u_{I+1}]$ in the sequence of knots.

Then, there will be an ordered sequence of knots U_i^r, each containing u_I or u_{I+1}, such that U_i^r consists of $r+1$ successive knots and u_I is the $(r-i)$-th element of U_i^r.

A degree n curve segment corresponding to the interval U is then given by $n+1$ control points $\boldsymbol{d}_i$. Each intermediate control polygon leg $\boldsymbol{d}_i^r, \boldsymbol{d}_{i+1}^r$ can then be viewed as an affine image of U_{i+1}^{n-r+1}. The point $\boldsymbol{d}_i^{r+1}$ is the image of u under such an affine map.

It is well-known (Farin, 2002) that a non-parametric B-spline function $d(u)$ can be written as a parametric curve with control points

$$\boldsymbol{d}_i = \begin{bmatrix} \xi_i \\ d_i \end{bmatrix}, \text{ with } i = 0, \ldots, L$$

and $L = K - n + 1$, with K the number of knots and n the degree of the curve. In this case, the points ξ_i are called *Greville abscissae* and can be determined as:

$$\xi_i = \frac{1}{n}(u_i + \ldots + u_{i+n-1})$$

For $n = 2$ it is straightforward to prove that the Greville abscissae coincide with the Voronoi vertices of the knot sequence.

Working in non-parametric form, and using the equivalence between Sibson and linear interpolation in one dimension, this simple algorithm can alternatively be obtained by applying Natural Neighbour (Sibson) interpolation over segments U_i^r in which we eliminate $r-1$ of the closest neighbours of the point u:

$$\begin{aligned} \boldsymbol{b}[u,u] &= \phi_1(u)\boldsymbol{b}[u_1,u] + \phi_2(u)\boldsymbol{b}[u,u_2] \\ &= \phi_1(u)\left(\varphi_0^2(u)\boldsymbol{b}[u_0,u_1] + \varphi_2^2(u)\boldsymbol{b}[u_1,u_2]\right) \\ &+ \phi_2(u)\left(\varphi_1^2(u)\boldsymbol{b}[u_1,u_2] + \varphi_3^2(u)\boldsymbol{b}[u_2,u_3]\right) \end{aligned} \tag{2.36}$$

where $\phi_I(u)$ represent the natural neighbour coordinates of point u with respect to knot I and $\varphi_I^r(u)$ represent the natural neighbour coordinates of point u with respect to knot I, but computed over an interval U_i^r, i.e., by eliminating $r-1$ natural neighbours of the interval. The notation used is shown in Fig. 2.14.

2.7.2 B-spline surfaces constructed over natural neighbor interpolation

The de Boor's algorithm thus presented can be extended to higher-dimensional cases as follows. In the following development we employ Sibson coordinates, although

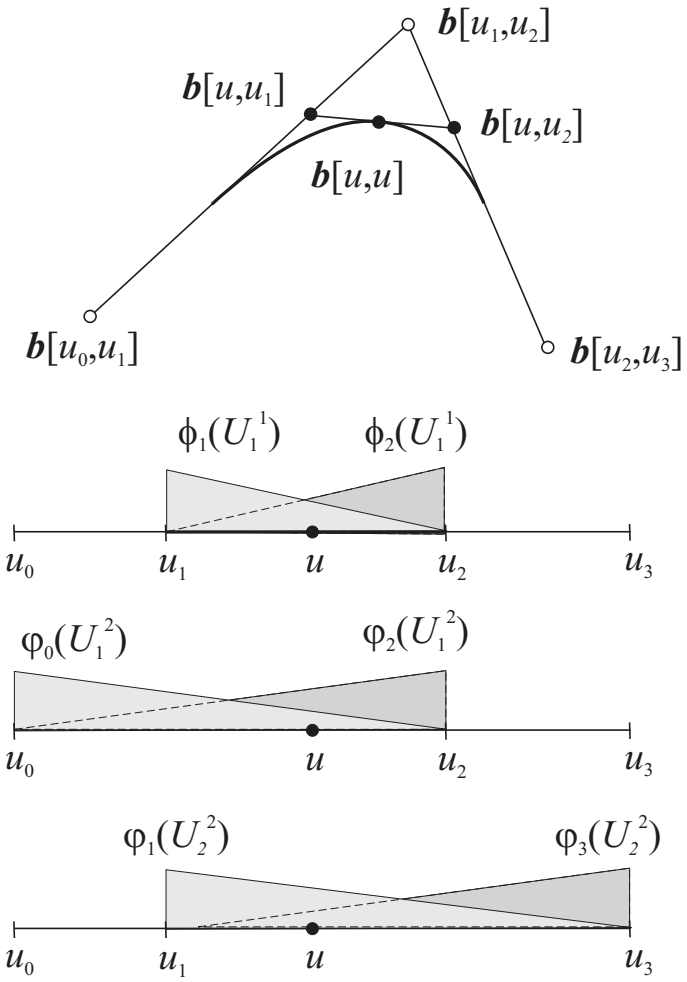


Figure 2.14. Schematic representation of the de Boor's algorithm employing natural neighbours. Between parentheses, the domain of each function.

the proposed algorithm can also be applied to Laplace interpolants, as will be shown later. Consider again, for simplicity, a set of nodes $\boldsymbol{N} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\} \subset \Re^2$ and a quadratic surface (the extension to three or higher dimensions and higher-order surfaces is straightforward). From now on, we will work in non-parametric form, since it is extremely hard to find the two-dimensional counterparts of the intervals U_i^r for irregularly scattered sites. Then, we define a new class of surfaces constructed in the way:

$$s(\boldsymbol{x}) = \sum_{I=1}^{n} \sum_{J=1}^{n^I} N_{IJ}(\boldsymbol{x}) d_{IJ}, \text{ with } d_{IJ} = d_{JI} \tag{2.37}$$

where n represents the number of neighbors of the point $\boldsymbol{x}$. In addition,

$$N_{IJ}(\boldsymbol{x}) = \phi_I(\boldsymbol{x}) \varphi_J^I(\boldsymbol{x}) \tag{2.38}$$

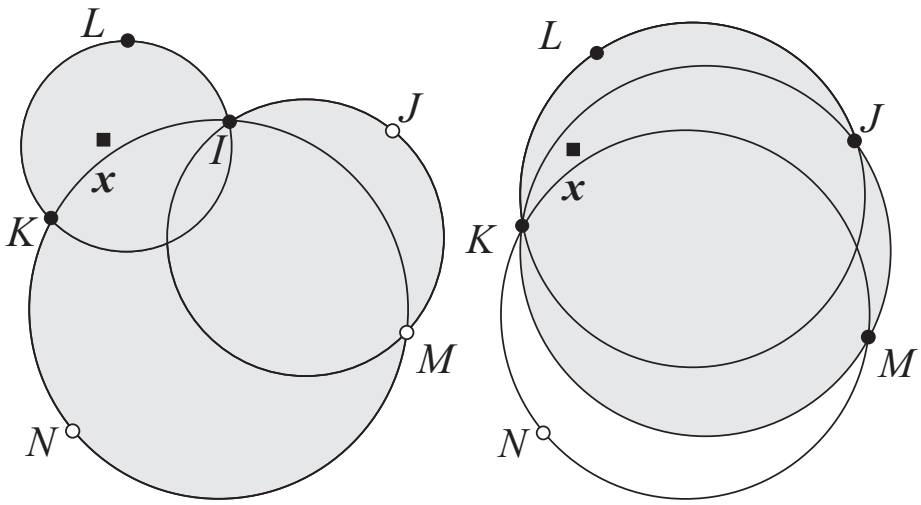


Figure 2.15. Schematic representation of the proposed algorithm. (a) Set of sites $\{I, \ldots, N\}$. We consider an evaluation point $\boldsymbol{x}$, whose neighbours are depicted as filled circles. The support of the function ϕ_I is highlighted. (b) After eliminating site I, the support of function φ_J^I is highlighted. Note the new set of neighbouring sites, $\{J, K, L, M\}$.

and d_{IJ} represent the control points in B-spline terminology (i.e., the degrees of freedom). $\phi_I(\boldsymbol{x})$ represents the natural neighbor (Sibson) coordinate of the point $\boldsymbol{x}$ with respect to site I. Functions $\varphi_J^I(\boldsymbol{x})$ represent the natural neighbor coordinates of point $\boldsymbol{x}$ with respect to site J, in the original cloud of points, but without the I-th site (see Fig. 2.15), in the sense described by the previous section. Finally, n^I is the number of natural neighbors of the point $\boldsymbol{x}$ when we eliminate the site I, similarly to the de Boor's algorithm. Note that the number of degrees of freedom of the proposed approximation is much less than $M^2/2$, since the sums in Eq. (2.37) extend only over natural neighbors of each node.

The typical shape of the functions N_{IJ} described before is shown in Fig. 2.16 for a general set of irregularly distributed sites.

Among the properties that can be cited about this type of approximation, we can cite the following (see González et al. (2008) for more details):

- Positiveness.
- They form a partition of unity.
- They span the space of linear and quadratic polynomials (and, by recursive application of the algorithm, polynomials of arbitrary degree).

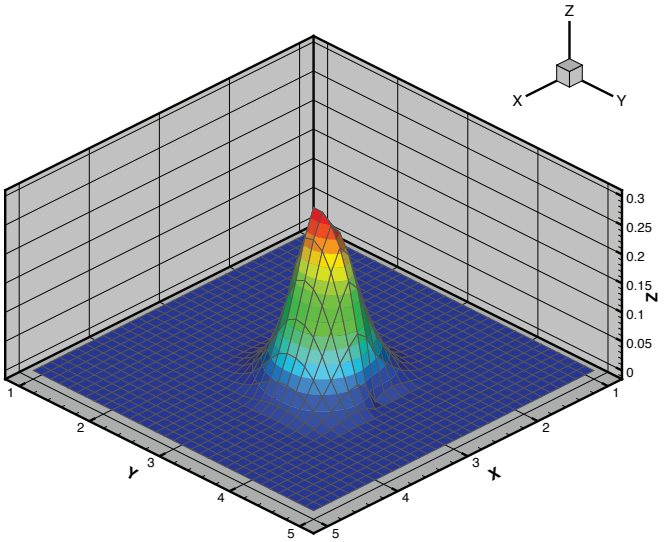


Figure 2.16. Shape of a typical function N_{IJ} for a set of irregularly distributed sites.

- Continuity is $\mathcal{C}^{p-1}$, where p stands for the order of consistency, except at lines joining neighboring nodes, where they are $\mathcal{C}^0$.
- Very much like B-splines, we can make the surfaces to be interpolant by repeating knots.

With this in mind, we face the challenge of simulating complex free-surface flows in the following chapter. In it, a thoroughly description of the techniques employed will be made, with particular emphasis in the development of a novel shape constructor algorithm that fits very well into the needs of this particular problem.

Chapter 3

An Updated Lagrangian Approach to Fluid Dynamics

In this chapter we will present the fluid formulation introduced in González (2004) and that will be subsequently enhanced in this thesis. We have adopted an updated Lagrangian fluid model which relies in turn on the method of characteristics to fully exploit the meshless characteristics of the Natural Element Method. We aim at simulating fluid flows with moving free surfaces where the fluid domain topology might be heavily altered along the time history of the process. This scenario posses two main obstacles to the use of traditional simulation methods, namely: we need to know the position of the boundaries at each time instant; and the fluid particles inside the domain move at different speeds and in different directions, which makes it difficult to know how their connectivity will evolve in time.

The way in which the fluid is represented by the computational method —be it by particles or elements—is heavily determined by the kinematic description of the movement adopted. This selection may either hamper or enable the solution of certain problems.

3.1 Kinematic Description of Fluid Flows

In continuum mechanics, two coordinate systems are used to describe a particle's movement. The first is the *material* frame, which is fixed to the particles place at all times and is defined by the basis vectors $\boldsymbol{I}_1, \ldots, \boldsymbol{I}_n$ in a n-dimensional space. The second reference frame, defined by vectors $\boldsymbol{i}_1, \ldots, \boldsymbol{i}_n$, is shifted and could be interpreted as a point where an observer is located watching all the particles that go by. This is called the *spatial* frame of reference (Donea and Huerta, 2003).

Let us consider a travelling material particle P in $\Re^n$. At the beginning of its movement (initial configuration), it can be localized by its position vector

$$\boldsymbol{X} = X_1 \boldsymbol{I}_n + \ldots + X_n \boldsymbol{I}_n \tag{3.1}$$

At a later time, in the ``deformed configuration'', the particle has moved to a new position whose spatial coordinates could be defined as

$$\boldsymbol{x} = x_1 \boldsymbol{i}_1 + \ldots + x_n \boldsymbol{i}_n \tag{3.2}$$

It is possible to define an application φ to map the reference configuration $R_{\boldsymbol{X}}$ to the spatial configuration $R_{\boldsymbol{x}}$ such that

$$\varphi : R_{\boldsymbol{X}} \times [t_0, t] \rightarrow R_{\boldsymbol{x}} \times [t_0, t] \tag{3.3}$$

$$(\boldsymbol{X}, t) \mapsto \varphi(\boldsymbol{X}, t) = (\boldsymbol{x}, t) \tag{3.4}$$

and thus, the relationship between the material and spatial coordinates is

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{X}, t), \quad t = t \tag{3.5}$$

According to the selected frame, the methods for describing the movement may be classified in Eulerian, Lagrangian and Arbitrary Lagrangian-Eulerian (ALE).

3.1.1 Eulerian Description

Probably the most common approach for the fluid simulation is the Eulerian approach. In these methods a set of spatial nodes is used to ``observe'' the evolution of the fluid domain. Since there is no mesh distortion, the problems related to particles moving in a very heterogeneous way are easily overcome. These techniques are specially well suited to address situations on which the boundaries are fixed, even if they are inflow or outflow boundaries. Also, these formulations are able to cope with large distortions in the fluid motion, indispensable quality in the treatment of turbulent flows. On the other hand, following moving surfaces can prove to be very challenging, requiring extra steps in order to be able locate the position of the fronts and adding difficulty to properly impose boundary conditions. One example of the ways to treat this problem are the Volume of fluid (VoF) techniques, see Duchemin et al. (2002) and references therein, which rely on the employ of an implicit function called *presence of fluid function*, that evaluates to one in the fluid region and zero at the empty zones. This function is advected with the velocity of the fluid throughout the computation. Other techniques,

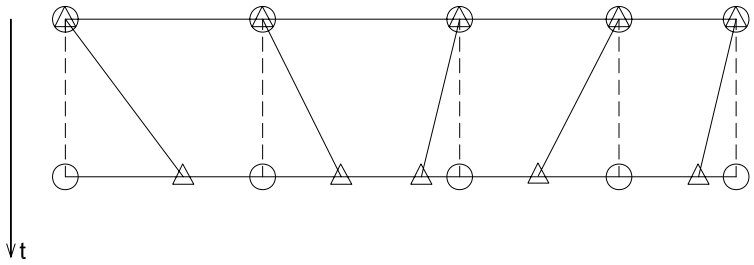


Figure 3.1. Eulerian approach. ◯ Numerical nodes. △ Material particles.

known as tracking methods (Crank, 1987) rely on the use of markers, whose position is updated with the just computed fluid velocity field.

Another effect of the Eulerian kinematic approach is that because of the relative motion between the material particles and the spatial nodes, convective terms will appear in the equations, leading some to numerical difficulties. Still, the wide use of this method have spurred the development of techniques that addressed the mentioned shortcomings as the Volume-of-Fluid or Level-set methods (Sethian, 1999) for following of free surfaces or several upwinding techniques for convection problems (Donea and Huerta, 2003).

3.1.2 Lagrangian Description

In Lagrangian methods, the computational domain follows the material particles, thus the mesh or cloud of nodes moves and deforms accordingly to the fluid movement. These techniques allow moving boundary tracking explicitly, not needing any interpolation technique to locate free surfaces or to impose boundary conditions. These formulations are the standard in solid finite element models, where not-so-large deformations are common and these techniques are very efficient. In time-dependent problems, as nodal position changes, the quality of the mesh might degrade, making it necessary to regenerate the grid in order to maintain accuracy. Depending on the application, remeshing can be performed automatically, but there are cases where this is not a possibility and manual mesh creation is very time consuming. When large deformations appear, this problem accentuates. The discussed problem has posed a

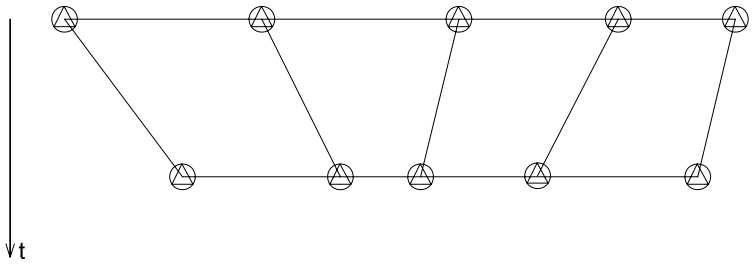


Figure 3.2. Lagrangian approach. ◯ Numerical nodes. △ Material particles

limitation for the application of Lagrangian formulations in fluid flow simulations even though it is not inherent to the approach but to the numerical method.

As meshless methods begun to appear, the interest in Lagrangian approaches resurfaced. Some early examples of this methods can be found in Belytschko et al. (1994) and Nayroles et al. (1992). More recent works on fluid simulations include for instance an application of the Meshless-Local Petrov-Galerkin method (Atluri et al., 1999) to nonlinear water-wave problems or Idelsohn et al. (2004) for an application of the particle finite element method to free-surface problems. As discussed in the last chapter, each of this methods present problems of their own, but open the door to feasible Lagrangian fluid simulations. In his thesis, González (2004) introduced a fluid model which tackles some of the problems posed by the previous methods.

3.1.3 Arbitrary Lagrangian-Eulerian (ALE)

The Arbitrary Lagrangian-Eulerian approach appeared in the ranks of the finite volume and finite differences methods as a means to take advantage of the Eulerian capabilities for treating large motions of the particles while at the same time being able to track moving boundaries in a Lagrangian way. In these methods, the numerical grid is not tied to the material particles nor is it fixed to any specific spatial point. Instead the mesh moves in a prescribed way in order to control its deformation in a new domain called a *referential configuration*. This allows us to follow the shape of the material domain while maintaining the best possible mesh to perform the calculations. See Donea and Huerta (2003) for a historical review of the development of these methods. These are

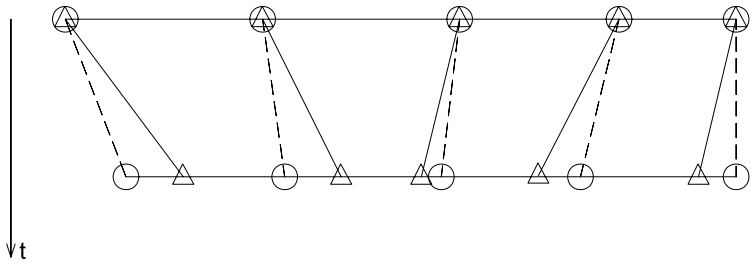


Figure 3.3. ALE approach. ◯ Numerical nodes. △ Material particles

specially useful in modelling large deformation in solids, free surface fluid flows and fluid-structure interaction.

Both Lagrangian and Eulerian approaches can be obtained as specific cases of the ALE formulation. In one case, if the mesh motion equals the material domain motion, the Lagrangian approach is obtained. If, on the opposite case, the referential configuration stays still, the kinematic description is Eulerian. The main drawback of the ALE approach is the need to determine a suitable mesh velocity. This can be done manually, although is a very difficult process for any but the simplest cases of motion. To avoid this, automatic mesh update strategies have been developed, like mesh regularization, to avoid as mush as possible the mesh distortion; or mesh adaptation, to focus on areas of steep gradients.

3.1.4 Material and Spatial Time Derivatives

In order to describe the influence of the chosen frame of reference on time derivatives we will follow Donea and Huerta (2003) and define two physical scalar quantities $f(\boldsymbol{x},t)$ and $f^{**}(\boldsymbol{X},t)$ on the spatial and material domains respectively. In this case, the asterisks denote that both quantities are different. For a moving particle, they are related by:

$$f^{**}(\boldsymbol{X},t) = f(\varphi(\boldsymbol{X},t),t) \quad \text{or} \quad f^{**} = f \circ \varphi.$$

The gradient of this expression can be obtained

$$\frac{\partial f^{**}}{\partial(\boldsymbol{X},t)}(\boldsymbol{X},t) = \frac{\partial f}{\partial(\boldsymbol{x},t)}(\boldsymbol{x},t)\frac{\partial \varphi}{\partial(\boldsymbol{X},t)}(\boldsymbol{X},t)$$

which in its matrix form is

$$\left(\frac{\partial f^{**}}{\partial \boldsymbol{X}} \; \frac{\partial f^{**}}{\partial t}\right) = \left(\frac{\partial f}{\partial \boldsymbol{X}} \; \frac{\partial f}{\partial t}\right) \begin{pmatrix} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} & \boldsymbol{v} \\ \boldsymbol{0}^T & 1 \end{pmatrix}$$

After matrix multiplication, we arrive at the following equations

$$\left(\frac{\partial f^{**}}{\partial \boldsymbol{X}}\right) = \left(\frac{\partial f}{\partial \boldsymbol{x}}\right)\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}\right), \tag{3.6}$$

$$\frac{\partial f^{**}}{\partial t} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial \boldsymbol{x}}\boldsymbol{v}. \tag{3.7}$$

on which the relationship between the material and spatial derivatives is expressed. This equations indicate that the physical quantity variation in time for a given material particle equals its local variation plus some quantity related to the relative movement between the material and spatial frames of reference, that is, a convection term. For the sake of notation, we will write

$$\left.\frac{\partial f}{\partial t}\right|_{\boldsymbol{X}} = \left.\frac{\partial f}{\partial t}\right|_{\boldsymbol{x}} + \boldsymbol{v}\cdot\nabla f \quad or \quad \frac{df}{dt} = \frac{\partial f}{\partial t} + \boldsymbol{v}\cdot\nabla f \tag{3.8}$$

where

$$\frac{d\cdot}{dt} := \left.\frac{\partial \cdot}{\partial t}\right|_{\boldsymbol{X}}$$

is called the material time derivative and

$$\frac{\partial\cdot}{\partial t} := \left.\frac{\partial \cdot}{\partial t}\right|_{\boldsymbol{x}}$$

is the spatial time derivative.

Another useful relation between material and spatial time derivatives is the Reynolds transport equation. Assuming the quantity f can be represented by a smooth motion over an arbitrary control volume, we arrive at

$$\frac{d}{dt}\int_{V_t} f(\boldsymbol{x},t)dV = \int_{V_c\equiv V_t} \frac{\partial f(\boldsymbol{x},t)}{\partial t}dV + \int_{S_c\equiv S_t} f(\boldsymbol{x},t)\boldsymbol{v}\cdot\boldsymbol{n}dS \tag{3.9}$$

Where V_c and S_c represent the control volume and its surface, and subscript t refer to the domain configuration at time t.

3.2 Governing Equations

In this section, the conservation equations are developed to arrive at the Navier-Stokes equations. We consider here the flow of an incompressible viscous fluid. In this kind of problems, the non linear convective terms mentioned in the last sections are usually a source of numerical problems. Since a Lagrangian scheme is employed in our method, these terms will not appear in the equations, bypassing those difficulties.

3.2.1 Conservation Equations

For the development of the governing equations, we consider, following Donea and Huerta (2003) closely, a fluid occupying a region Ω in $\Re^2$ or $\Re^3$, although in general we will say it occupies a volume V. The fluid presents a density ρ and a dynamic viscosity μ.

Mass Conservation

The conservation of mass through time in a given volume, assuming no mass is added or destroyed, is given by

$$\frac{dM}{dt} = \frac{d}{dt}\int_{V_t} \rho dV = 0 \tag{3.10}$$

By introducing Eq. (3.9), we obtain

$$\frac{dM}{dt} = \int_{V_t} \frac{\partial \rho}{\partial t} dV + \int_{S_t} \rho \boldsymbol{v} \cdot \boldsymbol{n} dS = \int_{V_t} \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) \right) dV = 0. \tag{3.11}$$

which holds for any V_t. This means that

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{3.12}$$

for all fluid particles. This is called the continuity equation and can also be expressed as

$$\frac{d\rho}{dt} + \rho \nabla \cdot \boldsymbol{v} = 0 \tag{3.13}$$

Momentum Conservation

Now, let we assume that the fluid particles are subjected to tensional state $\boldsymbol{\sigma}$ and distributed body forces $\rho \boldsymbol{b}$. The linear momentum of the body is defined as

$$P(t) = \int_V \rho \boldsymbol{v} dV \tag{3.14}$$

The linear momentum conservation principle states that the time variance of linear momentum equals the sum of all forces acting on the body, that is

$$\frac{d}{dt}\int_V \rho \boldsymbol{v} dV = \int_V \rho \frac{d\boldsymbol{v}}{dt} dV = \int_S \boldsymbol{\sigma} \cdot n dS + \int_V \rho \boldsymbol{b} dV \tag{3.15}$$

This integral relation holds for all material particles, and it is possible to write

$$\rho \frac{d\boldsymbol{v}}{dt} = \rho b + \nabla \cdot \boldsymbol{\sigma}, \tag{3.16}$$

The last equation is equivalent to

$$\rho \frac{d\boldsymbol{v}}{dt} + \rho(\boldsymbol{v} \cdot \nabla)\boldsymbol{v} = \rho \boldsymbol{b} + \nabla \cdot \boldsymbol{\sigma} \quad or \quad \frac{d\rho \boldsymbol{v}}{dt} = \rho \boldsymbol{b} + \nabla \cdot (\sigma + \rho \boldsymbol{v} \otimes \boldsymbol{v}). \tag{3.17}$$

3.2.2 Constitutive equations for a Newtonian fluid

Fluids are characterized for a continuous deformation under shear stresses and for as long as the stresses act. As a by-product of this property, resting fluids cannot undergo any shear stress at all. This is true for any arbitrary point in this fluid and also implies that the stress vector on any surface is proportional to the normal at that point but independent of its direction. We have then that

$$\boldsymbol{\sigma}_{ij}\boldsymbol{n}_j = -p\boldsymbol{n}_i \implies \boldsymbol{\sigma}_{ij} = -p\boldsymbol{\delta}_{ij}$$

where the proportionality constant p is called hydrostatic pressure. This shows that a fluid at rest is compressive in every direction and that p is the mean of the normal stress $\sigma_{ii}/3$.

For fluid in motion, the shear stresses are non zero and we get

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij} \tag{3.18}$$

where τ_{ij} is the viscous stress tensor and appears only as long as the fluid is moving. In this case the p is called thermodynamic pressure and is no longer one third of σ_{ii}

For developing the fluid constitutive equation, we must satisfy both rest and motion conditions. The term τ_{ij} must be a function of the deformation tensor, and in the Newtonian case, this relation is linear. The expression for the viscous stress tensor is

$$\tau_{ij} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \tag{3.19}$$

which leads to the constitutive equation

$$\sigma_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) = -p\delta_{ij} + 2\mu v_{(i,j)} \tag{3.20}$$

or

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\mu\nabla^s\boldsymbol{v} \tag{3.21}$$

which is known as Stokes' law.

3.2.3 Navier-Stokes Equations

The laminar flow of fluids subject to external and body forces can be described by solving the conservation equations coupled with a given constitutive relationship. This set of equations along with some boundary conditions are collectively known as the Navier-Stokes equations and they describe the motion in terms of particle velocities rather than displacements.

To formally enunciate the Navier-Stokes problem we will consider a closed finite fluid domain Ω in $\Re^n$ with $n = 2$ or 3 and a closed and sufficiently regular boundary $\Gamma = \partial\Omega$. The time-dependent flow of an incompressible viscous fluid must follow

$$\rho(\boldsymbol{v}_{,t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v}) = \nabla \cdot \boldsymbol{\sigma} + \rho\boldsymbol{b} \quad \text{in } \Omega \times (0, T), \tag{3.22}$$

$$\nabla \cdot \boldsymbol{v} = 0 \quad \text{in } \Omega \times (0, T) \tag{3.23}$$

$$\boldsymbol{v}(\boldsymbol{x}, t) = \boldsymbol{v}_D(\boldsymbol{x}, t), \quad \boldsymbol{x} \in \Gamma_D, \quad t \in (0, T) \tag{3.24}$$

$$\boldsymbol{n} \cdot \boldsymbol{\sigma}(\boldsymbol{x}, t) = \boldsymbol{t}(\boldsymbol{x}, t), \quad \boldsymbol{x} \in \Gamma_N, \quad t \in (0, T) \tag{3.25}$$

Eq. (3.23) is the incompressibility condition and implies that the elements of the fluid do not suffer any change in density when subjected to pressure changes.

3.3 The Method of Characteristics

The aforementioned equations define a convection problem that describe non-linear momentum transport phenomena. This is the standard Eulerian formulation for fluid mechanics. One of the methods for solving this equations is to convert them to an equivalent system in a Lagrangian reference frame. This is accomplished with the use of the characteristics lines concept which will be explained first by the use of a simpler linear hyperbolic PDE, following again Donea and Huerta (2003, Ch. 3). The strong form of this problem is

$$u_t + \nabla \cdot \boldsymbol{f}(u) = s(\boldsymbol{x}, t) \quad \text{in } \Omega \times]0, T[, \tag{3.26}$$

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}) \quad \text{on } \Omega|_{t=0}, \tag{3.27}$$

$$u = u_D \quad \text{on } \Gamma_D \times]0, T[, \tag{3.28}$$

$$-\boldsymbol{f} \cdot \boldsymbol{n} = h \quad \text{on } \Gamma_N \times]0, T[. \tag{3.29}$$

here u (the solution) and s (a source term) are functions of t and $\boldsymbol{x}$, and the Dirichlet and Neumann BCs are applied only in the inflow part of the boundary. The flux function is defined as

$$\boldsymbol{f}(u) = \boldsymbol{a}u \tag{3.30}$$

where $\boldsymbol{a}$ is the convection velocity. In this case, since the problem is linear, $\boldsymbol{a}$ is independent of u and represents the velocity at which the solution is propagated in time and space.

3.3.1 Characteristic Lines

After rewriting the Eq. (3.26) by using the convection velocity, we get

$$u_t + a u_x = s \tag{3.31}$$

and from it, the total derivative of u in the direction of slope $dx/dt = a$ equals s. This direction is called *characteristic direction* or simply *characteristic*. To show the transport of the solution along these lines, we follow the development explained in Donea and Huerta (2003), and take the homogeneous form of the above equation. First let us perform the following change of variables

$$\xi = x - at, \quad \eta = x + at$$

and the transformation

$$\left\{ \begin{array}{c} u_x \\ u_t \end{array} \right\} = \begin{pmatrix} \frac{\partial \xi}{\partial t} & \frac{\partial \eta}{\partial t} \\ \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \end{pmatrix} \left\{ \begin{array}{c} u_\xi \\ u_\eta \end{array} \right\} = \begin{pmatrix} -a & a \\ 1 & 1 \end{pmatrix} \left\{ \begin{array}{c} u_\xi \\ u_\eta \end{array} \right\}$$

so we arrive at

$$2 a u_\eta = 0$$

which is solved by any function such as

$$u = f(\xi) = f(\boldsymbol{x} - at),$$

By evaluating f at $t - \Delta t$ and $x - a\Delta t$ we observe that the solution is the same than when evaluating at t and x. This shows that the solution propagates the spatial profile of u along a line of slope a. Thus, we can backtrace the solution at any point to one of known value at t_0 or to the influx boundary, where the value of the solution is also known or can be determined via BC. Since the equation is linear, the characteristic lines are fixed in the $(\boldsymbol{x}, t)$ plane, regardless of the value of $u(\boldsymbol{x}, t)$. If the equation coefficients are constant, the characteristics are straight lines. In case a depends on u, the equation is non-linear.

3.3.2 Solution Strategies Based on Characteristic Lines

As mentioned in the previous section, to resolve the transport problem, one can transform the Eulerian equations to its Lagrangian equivalent by using the characteristic lines concept. This is performed by replacing the material derivative in Eq. (3.26) with a total time derivative in the Lagrangian sense.

Let us say we can determine trajectory of a material particle traversing a spatial point of coordinates $\boldsymbol{x}$ at time τ. This trajectory is the characteristic line $\boldsymbol{X} = \boldsymbol{X}(\boldsymbol{x}, \tau; \tau)$ and satisfies

$$\frac{d\boldsymbol{X}}{dt}(\boldsymbol{x}, \tau; t) = \boldsymbol{a}(\boldsymbol{X}(\boldsymbol{x}, \tau; t)), \tag{3.32}$$

$$\boldsymbol{X}(\boldsymbol{x}, \tau; \tau) = \boldsymbol{x}. \tag{3.33}$$

This equation is typically non-linear, but solving this problem allows to reduce the linear unsteady convection equation to an ordinary differential equation.

Along the characteristic $\boldsymbol{X}$ the material derivative

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \boldsymbol{a} \cdot \nabla u, \tag{3.34}$$

reduces to a simple time derivative.

It is possible now to transform the original problem into its *characteristic form*. The characteristic line that passes at space-time point $(\boldsymbol{x}, \tau)$ will be denoted as $\boldsymbol{X}(t)$, the value of the transported quantity (the solution) along the that line will be $U(t) := u(\boldsymbol{X}(\boldsymbol{x}, \tau; t), t)$ and the source term $S(t) = s(\boldsymbol{X}(t), t)$. The problem is written now as

$$\frac{dU}{dt} = S(t), \tag{3.35}$$

subject to the initial condition

$$U(t_\Gamma) = u_D(\boldsymbol{X}_\Gamma, t_\Gamma) \tag{3.36}$$

where t_Γ is the time at which the characteristic line $\boldsymbol{X}$ intersects the Dirichlet boundary. The intersection is denoted by $\boldsymbol{X}_\Gamma = \boldsymbol{X}(\boldsymbol{x}, \tau; t_\Gamma)$.

Solving this equation for time τ, we have two situations: if the characteristic line intersects the Dirichlet boundary the solutions is

$$u(\boldsymbol{x}, \tau) = u_D(\boldsymbol{X}_\Gamma, t_\Gamma) + \int_{t_\Gamma}^{\tau} S(t)dt, \tag{3.37}$$

or,

$$u(\boldsymbol{x}, \tau) = u_0(\boldsymbol{X}(0)) + \int_{0}^{\tau} S(t)dt, \tag{3.38}$$

if the characteristic line passed through a point of known solution at $t = 0$.

We will treat now convection problems with a variable convection field $\boldsymbol{a}(\boldsymbol{x}, t)$. As suggested before, in these problems the solution is constant along characteristic lines. This fact is used either in semi-Lagrangian or in Lagrange-Galerkin approaches to solve the transport equations.

The semi-Lagrangian methods are very well suited to treat unsteady convection problems. They are extensively used in the field of meteorological forecasting and in environmental flows in non uniform cartesian meshes. This is due to the excellent accuracy and efficiency yield in problems that exhibit a low to medium wave number. In these methods, the characteristic lines—Eq. (3.32)—over the interval $]t, t^{n+1}[$ are approximated by a mid-point rule. Then, the characteristic form of Eq. (3.35) is solved to obtain the values at t^{n+1} of the solution u. This method implies evaluating $u^n(\boldsymbol{x})$ and $s^n(\boldsymbol{x})$ at points different from the nodes, which is done by interpolating the nodal values.

3.4 Lagrange-Galerkin Methods Based on Characteristic Lines

The Lagrange-Galerkin methods are similar to the semi-Lagrangian but instead of differentiating along the characteristic lines, these methods employ a spatial discretization based on Galerkin projections.

Two different approaches fall in the Lagrange-Galerkin type of methods. The first consists in directly integrating along the characteristics. For this, and assuming the convection velocity $\boldsymbol{a}(\boldsymbol{x}, t)$ is known, the characteristic lines for all nodes are calculated and then the integrals in Eqs. (3.37) and (3.38) are solved via a Galerkin method.

The second approach consists in the use of a variational formulation, which is the method employed in this work. Briefly put, the Navier-Stokes problem is recast into its weak form and then solved by integrating in space and time along the characteristic lines.

3.4.1 Weak Formulation

For the presentation of the weak formulation we will consider now the full problem that is being solved instead of the simpler equations used until know. To recall, we are treating with the dynamic response of incompressible viscous fluids described by

$$\nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{b} = \rho \left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \nabla \cdot \boldsymbol{v} \right) = \rho \frac{d\boldsymbol{v}}{dt}, \tag{3.39}$$

$$\nabla \cdot \boldsymbol{v} = 0, \tag{3.40}$$

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\mu \nabla^s \boldsymbol{v} = -p\boldsymbol{I} + \boldsymbol{s} \tag{3.41}$$

After introducing the constitutive equation, Eq. (3.41), in the momentum conservation equation, Eq. (3.39), and multiplying for a suitable test function, we obtain

$$\int_\Omega 2\mu \boldsymbol{D} : \boldsymbol{D}^* d\Omega - \int_\Omega p\boldsymbol{I} : \boldsymbol{D}^* d\Omega = -\int_\Omega \rho \boldsymbol{b}\boldsymbol{v}^* d\Omega + \int_\Omega \rho \frac{d\boldsymbol{v}}{dt}\boldsymbol{v}^* d\Omega, \tag{3.42}$$

$$\int_\Omega \nabla \cdot \boldsymbol{v} p^* d\Omega = 0 \tag{3.43}$$

where $\boldsymbol{D} = \nabla^s \boldsymbol{v}$ represents the strain rate tensor and $\boldsymbol{b}$ are the body forces. The test functions are $\boldsymbol{v}^*$ and p^* which correspond to the different essential variables, namely: the velocity and pressure.

3.4.2 Time Discretization

The convective terms of Eq. (3.17) were replaced by a Lagrangian temporal derivative term, which is a material derivative along the fluid particle trajectories. This term, the second in the r.h.s. of Eq. (3.42) gathers the inertial effects of the flow. Using a standard first order temporal discretization, and knowing the solution at time $t^{n-1} = (n-1)\Delta t$, we can calculate the solution at $t^n = n\Delta t$ by means of

$$\int_\Omega \rho \frac{d\boldsymbol{v}}{dt}\boldsymbol{v}^* d\Omega = \int_\Omega \rho \frac{\boldsymbol{v}^n(\boldsymbol{x}) - \boldsymbol{v}^{n-1}(\boldsymbol{X}(t^{n-1}))}{\Delta t}\boldsymbol{v}^* d\Omega, \tag{3.44}$$

In this equation $\boldsymbol{X}(t)$ represents the position at time t along the characteristic line that passes through point $\boldsymbol{x}$ at time t^n, so $\boldsymbol{X}(t^{n-1})$ is the position the material particle occupied during the last time step. For an even simpler notation, we will denote this point as $\boldsymbol{X}_{n-1}$. Using this notation the particle position can be written as

$$\boldsymbol{x} = \boldsymbol{X}_{n-1} + \boldsymbol{v}^{n-1}(\boldsymbol{X}_{n-1})\Delta t \tag{3.45}$$

This was the temporal discretization employed in González (2004). With it, the weak form of the Navier-Stokes equations is

$$\int_\Omega 2\mu \boldsymbol{D} : \boldsymbol{D}^* d\Omega - \int_\Omega p\boldsymbol{I} : \boldsymbol{D}^* d\Omega - \int_\Omega \rho \frac{\boldsymbol{v}\boldsymbol{v}^*}{\Delta t} = -\int_\Omega \rho \boldsymbol{b}\boldsymbol{v}^* d\Omega - \int_\Omega \rho \frac{\boldsymbol{v}^{n-1}\boldsymbol{v}^*}{\Delta t} d\Omega, \tag{3.46}$$

$$\int_\Omega \nabla \cdot \boldsymbol{v} p^* d\Omega = 0 \tag{3.47}$$

When a more accurate scheme is needed, it is possible to employ a second order time discretization. In Boukir et al. (1997) it is shown that a higher order time discretization not only increases accuracy but also relaxes the spatial-temporal restrictions imposed by the CFL conditions. This is also the experience obtained in the GEMM group

with the usage of the first-order in time characteristic method developed in González et al. (2007). This first-order scheme very often lead to numerical difficulties and lack of convergence in finding the root of the characteristic line that made it non suitable for fluid-structure interaction problems. In this thesis a second-order in time approach is suggested instead:

$$\int_\Omega \rho \frac{d\boldsymbol{v}}{dt}\boldsymbol{v}^* d\Omega = \int_\Omega \rho \frac{3\boldsymbol{v}^n(\boldsymbol{x}) - 4\boldsymbol{v}^{n-1}(\boldsymbol{X}(t^{n-1})) + \boldsymbol{v}^{n-2}(\boldsymbol{X}(t^{n-2}))}{2\Delta t}\boldsymbol{v}^* d\Omega, \tag{3.48}$$

leaving the weak form of the problem as

$$\int_\Omega 2\mu \boldsymbol{D} : \boldsymbol{D}^* d\Omega - \int_\Omega p\boldsymbol{I} : \boldsymbol{D}^* d\Omega - \frac{3}{2}\int_\Omega \rho \frac{\boldsymbol{v}\boldsymbol{v}^*}{\Delta t} =$$
$$- \int_\Omega \rho \boldsymbol{b}\boldsymbol{v}^* d\Omega - 2\int_\Omega \rho \frac{\boldsymbol{v}^{n-1}\boldsymbol{v}^*}{\Delta t} d\Omega + \frac{1}{2}\int_\Omega \rho \frac{\boldsymbol{v}^{n-2}\boldsymbol{v}^*}{\Delta t}, \tag{3.49}$$

$$\int_\Omega \nabla \cdot \boldsymbol{v} p^* d\Omega = 0 \tag{3.50}$$

Please note that in the weak form equations, the past velocities have been written as $\boldsymbol{v}^{n-i}$ but they are still evaluated at points $\boldsymbol{X}_{n-i}$.

3.4.3 Algorithmic Issues

The most difficult terms to evaluate in Eq. (3.46) or Eq. (3.49) are those which refer to the particle position at past times. The numerical integration of this terms depends on the quadrature scheme employed. If we employ traditional Gauss-based quadratures on the Delaunay triangles, it will be necessary to find the position at time t^{n-1} (and at t^{n-2}) of the material particle now (at time t^n) occupying the position of the integration point in question.

To explain this problem and the employed solution method, we will refer to the first order time discretization, Eq. (3.46). The integration of the term in question is performed according to

$$\int_\Omega \rho \frac{\boldsymbol{v}^{n-1}\boldsymbol{v}^*}{\Delta t} d\Omega = \sum_k \rho \frac{\boldsymbol{v}^{n-1}(\boldsymbol{\Xi}_k)\boldsymbol{v}(\boldsymbol{\xi}_k)}{\Delta t} w_k, \tag{3.51}$$

where w_k represent the weight associated to integration point k. That point occupies the position $\boldsymbol{\xi}_k$ at the current time instant, and the characteristic line $\boldsymbol{X}(\boldsymbol{\xi}_k, t^n; t)$ takes the value $\boldsymbol{\Xi}_k$ at t^{n-1}.

Two main difficulties need to be addressed: finding $\boldsymbol{\Xi}_k$ and evaluating $\boldsymbol{v}^{n-1}(\boldsymbol{\Xi}_k)$. The later may be performed by interpolation between neighbouring nodes, given that

the past nodal velocities are stored, which is fairly easy to assure. It is also necessary to know which nodes were near Ξ, so we will assume the nodal connectivity remains constant within two subsequent time increments. Although this is known to be false, in general, is a reasonable assumption for small time increments. It would be more accurate to store and manipulate all the information regarding nodal connectivity for past steps, but the possible gains do not compensate the computational costs in the cases where that is even possible —closed codes could probably deny that possibility at all. It can occur that some of the nodes neighbouring the integration point at time t were not actually its neighbors at time t^{n-1}, but this does not constitute a problem, since the number of natural neighbours of a point is usually high (much more than three), so the quality of the interpolation is thus guaranteed.

Regarding the search for Ξ, we utilize a two-step iterative procedure. First we project the integration point $\boldsymbol{\xi}$ backwards in time by

$$\Xi_i = \boldsymbol{\xi} - \boldsymbol{v}^{n-1}(\Xi_{i-1})\Delta t, \tag{3.52}$$

for the first iteration we interpolate the past velocities of the neighbouring nodes at t^n to make the projection.

In the second step, we evaluate the velocity $\boldsymbol{v}^{n-1}(\Xi_i)$ calculate the projection in t^n of Ξ_i:

$$\boldsymbol{\xi}_i = \Xi_i + \boldsymbol{v}^{n-1}(\Xi_i)\Delta t \tag{3.53}$$

Until $\boldsymbol{\xi} \approx \boldsymbol{\xi}_i$. González et al. (2007) reports convergence in two or three iterations with an error of the order of 10^{-8}, see also González (2004) .

When employing a second-order time discretization, the process will be performed in a recursive way, solving for the t^{n-1} and t^{n-2} time steps. In this case

$$\Xi = \Xi' + \boldsymbol{v}^{n-2}(\Xi')\Delta t \tag{3.54}$$

$$\boldsymbol{\xi} = \Xi' + \boldsymbol{v}^{n-2}(\Xi')\Delta t + \boldsymbol{v}^{n-1}(\Xi)\Delta t \tag{3.55}$$

where $\Xi' = \boldsymbol{X}(\boldsymbol{\xi}, t^n; t^{n-2})$.

If we employ some type of nodal integration, as in Chen et al. (2001) or González (2004), this procedures becomes unnecessary as we only need to store nodal velocities at time steps t^{n-1} and t^{n-2}.

3.5 Numerical Examples

The first-order version of this formulation has been tested in González (2004), González et al. (2007) and has shown a very good performance in the simulation of free surface

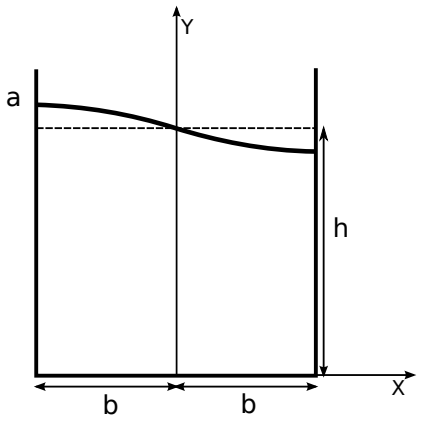


Figure 3.4. Initial configuration for a 2-D sloshing problem.

flows. We refer the reader to these works for more details on the topic. However, it has demonstrated problems, and very often lack of convergence in the characteristic root finding, when applied to fluid-structure interaction problems. In this section we will show the behavior of the higher-order time discretization in a situation where the original formulation was unable to yield the appropriate results.

3.5.1 Sloshing

We will consider a two-dimensional sloshing problem with small deformations as proposed by Ramaswamy (1990). The initial setting is shown in Fig. 3.4. In this problem a stationary fluid in an unstable initial condition is set free to move under the gravity influence alone. The fluid is contained in a 2-dimensional tank. The idea is to observe the wave amplitude as the fluid motion stops. The initial surface elevation in given by

$$\eta_o = a \; cos \; k_2(x + l/2) \tag{3.56}$$

with

$$k_n = n\pi/l \tag{3.57}$$

where n represents the number of waves present in a tank of with l. a represents the initial maximum amplitude measured from the mean level h.

An analytical solution was proposed by Prosperetti for certain parameters for which

is possible to obtain a solution for the amplitude. In this case the solution is given by

$$a(t) = \frac{4v^2k^4}{8v^2k^4 + w_0^2} a_0 erfc\left(\sqrt{vk^2 t}\right) + \sum_{i=1}^{4} \frac{z_i}{Z_i}\left(\frac{w_0^2 a_0}{z_i^2 - vk^2}\right) exp[(z_i^2 - vk^2)t] erfc(z_i\sqrt{t}), \tag{3.58}$$

where a_0 is the initial amplitude, w_0 is the natural invicid frequency: $w_0^2 = gk + \gamma k^3$; the z_i are the four roots of

$$z^4 + 2k^2 v z^2 + 4(k^2 v^3)^{3/2} + v^2 k^4 + w_0^2 = 0 \tag{3.59}$$

and $Z_1 = (z_2 - z_1)(z_3 - z_1)(z_4 - z_1)$. Z_2, Z_3, Z_4 are obtained by a permutation of the indices.

For the tank boundaries free-slip conditions were imposed along the walls. Note there is no need to impose conditions on the free surface, nor to perform any special boundary tracking of if.

For discretization purposes, a Sibson approximation was chosen for the velocity field, whereas piece-wise constant (Thiessen) approximation was chosen for the pressure. This kind of approximation is known not to fulfil the inf-sup condition associated with the incompressibility condition, see González et al. (2004), but nevertheless still provides with stable results in the vast majority of cases, very rarely producing volumetric locking. This is the reason why an as much simple as possible approach has been preferred.

In the shown case a time increment $\Delta t = 0.005s$ was used and the Reynolds number resulted to be of 3200. The gravity force had an acceleration of $g = 9.8m/s^2$. Fig. 3.5 shows the performance of the numerical solution (in red) compared against the analytical solution (purple line), where a very good agreement can be appreciated.

In Fig. 3.6 both first-order and-second order schemes can be compared. it is possible to see how the first order scheme was unable to replicate the diminishing amplitude. At larger time increments the solution even diverges, while the second order scheme was able to still give reliable results.

In Figs. 3.7 to 3.9 the velocity field for different time steps is shown. Vectors show the direction of the velocity, while the colors show its magnitude. It can be appreciated the deceleration as the fluid height on one side decreases until finally the velocity field reverses.

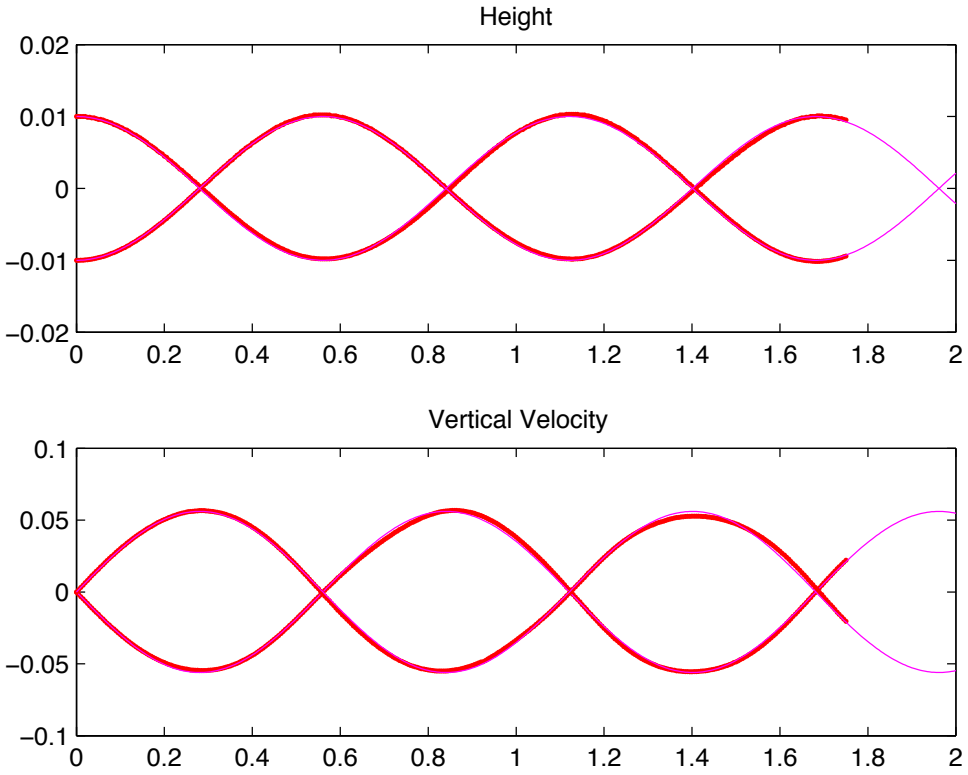


Figure 3.5. Evolution in time of the vertical displacement and velocities at both edges of the free surface. The numerical solution is shown in red, in purple the analytic solution.

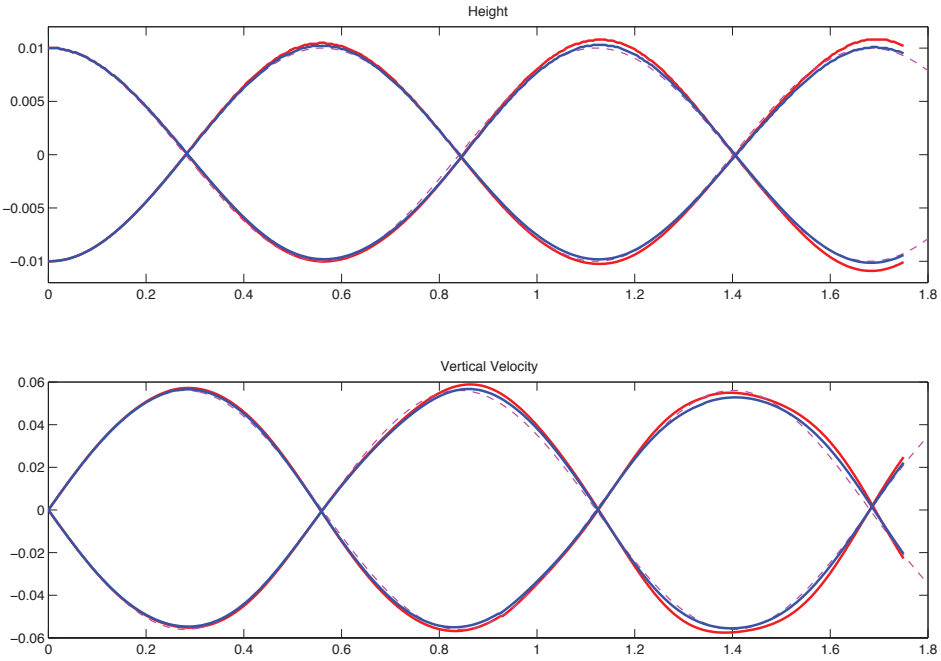


Figure 3.6. Vertical displacement and velocities at both edges of the free surface. The second-order solution is shown in blue, the first-order solution in red and the analytic solution in a purple line.

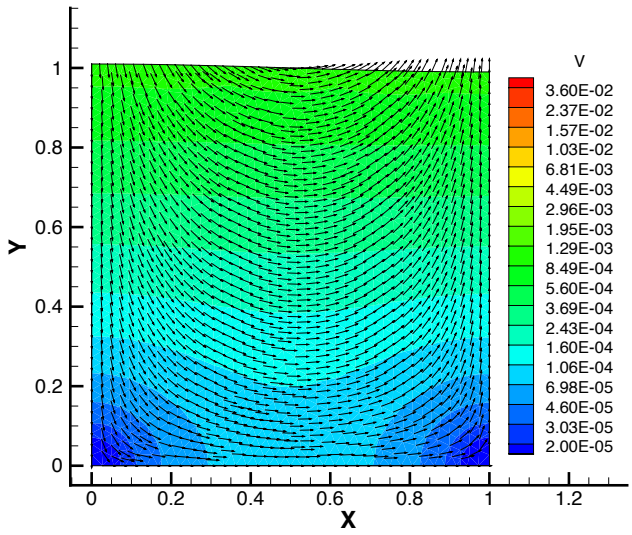


Figure 3.7. Velocity field at time $t = 0.005$.

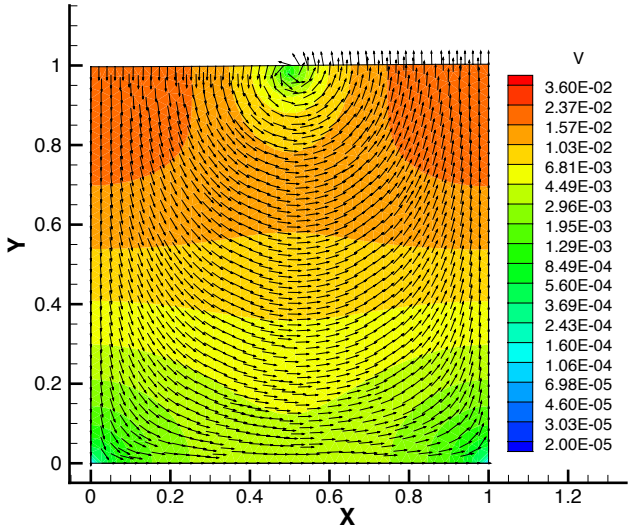


Figure 3.8. Velocity field at time $t = 0.455$.

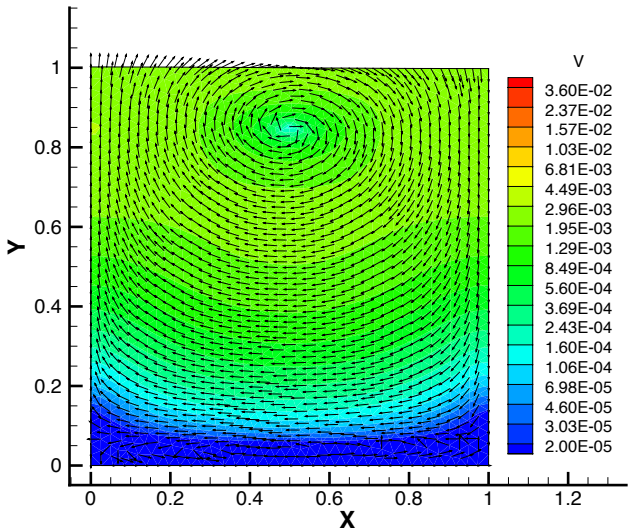


Figure 3.9. Velocity field at time $t = 1.205$.

3.6 Conclusions

In this chapter a second-order in time Natural Neighbour Lagrange-Galerkin scheme has been introduced. The modified technique improves the quality of the obtained solutions over the previous technique; even allowing to simulate problems that were not previously possible. By incorporating more historical information about the previous flow steps, it was possible to reduce error and reach better approximations. The improved accuracy comes at a cost of increased resource requirements (CPU time, memory and storage) but nevertheless the increase is not in general excessive. The characteristic root finding algorithm has shown to converge in a very reduced number of iterations for the vast majority of cases tested. In the next chapters applications of the recently developed scheme will be studied, with special emphasis in non-Newtonian free-surface flows. But for the time being the free-surface detection algorithm is still to be presented.

Chapter 4

Free Surface Treatment

In this chapter we will introduce the concept of *shape constructors* and review their use. Particular interest will be put in α-shape techniques for the simulation of free-surface flow problems. These procedures, in conjunction with meshless methods, allow for the simulation of such problems in an updated Lagrangian approach without the need for an explicit description of the boundary of the domain. At each time step, the shape of the domain is extracted automatically by the proposed method. However, it is well know that standard α-shape techniques present some drawbacks. The first is the choice of the α parameter, related to the level of detail to which the domain is represented. Also contact detection of free surfaces (auto-contact) or between the free surface and a rigid boundary, for instance, is often detected with an error of the order $\mathcal{O}(h)$ —the nodal spacing parameter—in the gap distance. A heuristic technique for the choice of the α parameter is proposed and a novel methodology for an improved detection of contact or merging flows is developed. The proposed technique is illustrated with the help of some examples in solid and fluid mechanics.

One of the most cited capabilities of meshless methods is that of simulating large deformation phenomena without degrading accuracy, as opposed to Finite Element Methods, if no remeshing is performed. This opens the possibility of simulating free surface flows, for instance, in an updated Lagrangian framework, and many works have been devoted to this end in the last years. The interested reader can consult, for instance, Martínez et al. (2004), Idelsohn et al. (2003), Idelsohn et al. (2004) or González et al. (2007), among others. These free surface problems are different in nature. The reader may imagine readily waves breaking, but not only dynamical problems can be solved with such a treatment. Many forming processes, for instance, can be also treated in an updated Lagrangian setting advantageously, see Alfaro et al. (2006a) or Alfaro et al. (2006b). Forging or casting and, obviously, mould filling, are among these processes that present free or internal surfaces, like phase boundaries.

In these problems, the obvious advantages of updated Lagrangian meshless methods over Eulerian or Arbitrary Lagrangian Eulerian (ALE) methods —in which an artificial velocity is added to the mesh—, for instance, are the absence of remeshing nor the associated numerical diffusion, or the lack of convective terms in the formulations, that consequently do not need for any stabilization. Note that connectivity between nodes is computed by the different meshless methods in a process transparent to the user, as the cloud of nodes evolves, convected by the material velocity.

A particularly elegant analysis of the difficulties associated to an Eulerian/Lagrangian treatment of the equations arising from free surface flows can be found in Lewis et al. (1997). Particularly noteworthy is the difficulty in the selection of mesh velocity in ALE formulations, in which the mesh moves with a velocity different to the material one, in order to minimize mesh distortion. Also, in Eulerian (fixed mesh) approaches, some marking technique should be used in order to track the evolution of the free surface. The Volume of Fluid (VoF) technique is an example of these techniques. In Lewis et al. (1997) an interesting mixed Eulerian/updated Lagrangian technique is developed.

As stated before, meshless, or particle (those in which a mass is linked to each node) methods have avoided the need to perform such complicated treatments. Nevertheless, new difficulties arise. For instance, the nodal connectivity in meshless methods is not dictated by geometrical reasons (the best available triangle in terms of internal angles, for instance, in FE mesh generation) but by algorithmic reasons. In the Element Free Galerkin method (Belytschko et al., 1994), for instance, the connectivity is dictated primarily by the need of a support (radius of the shape function) big enough to encompass a sufficiently large number of nodes so as to make a matrix invertible. Remarkably, this is not related to the geometry of domain. That reason precludes the nodal connectivity to be used directly to determine the shape of the domain, as in FE methods. Nothing similar to an *isoparametric* representation exists in meshless methods.

In addition, tracking the free surface with boundary markers can be implemented in an elegant way in two dimensional problems —by employing a chain of markers and checking self-intersections of the chain to detect merging flows—, as in Lewis et al. (1997), for instance, but becomes much more intricate in tree dimensions.

If one tries to avoid any form of meshing, and only a set of nodes, with no connectivity between them, is employed, it then becomes difficult to find the position of the free surface. In other words, the geometry of the domain should be extracted in any way from the current, updated, position of the nodes, that move, as stated before, with the material velocity.

To this end, various authors have employed Computational Geometry techniques. In particular, Cueto et al. (2000) seems to have been the first in employing *shape constructors* —α-shapes in this case—techniques to extract the geometry of the domain. Shape constructors are geometrical techniques that enable to find the *shape* of the domain at each time step. α-shapes (Edelsbrunner and Mücke, 1994) have been employed in a number of previous works involving free surface flows, see for instance Idelsohn et al. (2004), Idelsohn and Oñate (2006), Martínez et al. (2004), González et al. (2007) or Birknes and Pedersen (2006), among others.

Also, different shape constructors have been proposed after α-shapes, see Cazals et al. (2006), Amenta et al. (1998b), Amenta et al. (1998a), Giesen and John (2003) to name a few. In order to extract the geometry of the domain, in general, these methods propose a *filtration* of the Delaunay triangulation of the cloud of points. The Delaunay triangulation is the base ingredient of these techniques, since it characterizes unequivocally the cloud of points —it is unique for each cloud. Different criteria are proposed in order to select the triangles pertaining to the *shape* of the domain. The simplest one is maybe the α-shape technique, that proposes to eliminate all triangles (or tetrahedra) whose circumscribing radius (or, equivalently in finite element terminology, their associated mesh size, h) is greater than a prescribed *level of detail* for the geometry, α. α-shapes have generated a great interest on ``provable'' shape reconstruction arguments. We mean that, under certain, usually very weak, assumptions on the size of the cloud of points, we obtain geometric and topologically accurate descriptions of the domain under consideration.

One of the main drawbacks of the α-shape technique, as recognized in many works (see, for instance, Cazals et al. (2006), Teichmann and Capps (1998)) is precisely the choice of the α-value. In addition, α-shapes work well only for uniformly-distributed cloud of points. This generally does not constitute a problem for stationary problems. For initial-value problems, the choice of a uniform nodal sampling on the initial geometry, in the absence of any information on the final geometry of the domain, seems to be judicious.

The jump of the mentioned techniques to the field of Computational Mechanics has posed additional difficulties. It is well-known that α-shapes are not able to detect holes or cavities of size smaller than α, by definition. This implies that contact between different surfaces is detected with an error $\mathcal{O}(\alpha) \approx \mathcal{O}(h)$, i.e., prior to the true expected contact (Teichmann and Capps, 1998). Precisely in Teichmann and Capps (1998) a method is proposed to alleviate this drawback, but it needs information on the normal of the boundary at the sampling points. This is easy to achieve for three-dimensional scans

of solids, for instance, but this kind of information is not readily available in the class of simulations we are interested in.

In this chapter we present a technique, well suited for the numerical simulation of free-surface flows, that avoids the before mentioned problems. The proposed technique is based in performing an additional filtration to the Delaunay triangulation (tetrahedrization) of the cloud of points. After the α-filtration, we perform an additional filtration based on the information provided by nodal velocities at the last converged time step, and the gradient of velocities. The tests performed during this work have provided excellent results over problems where traditional α-shapes have revealed deficiencies.

4.1 Theory of α-shapes

As mentioned before, the idea of α-shapes in particular, and shape constructors in general, is to extract the shape of a domain described only by a set of nodes. While an easy task to the human eye, there is no formal definition of *shape* in the mathematical literature. α-shapes were first established by Edelsbrunner and co-workers Edelsbrunner et al. (1983) Edelsbrunner and Mücke (1994). Other shape constructors giving homotopy-equivalent shapes have been recently proposed (Dey et al., 2003). Given a finite set of points (that will be the nodes employed in the approximation of the problems described in the previous section), there exist a finite set of shapes described by all the possible combination of points, edges, triangles and tetrahedra (if we consider three-dimensional spaces) forming *simplicial complexes*.

A k-simplex σ_T with $0 \leq k \leq 3$ is defined as the convex hull of a subset $T \subseteq N$ of size $\mid T \mid = k+1$. A three-dimensional simplicial complex is a collection, $\mathcal{C}$, of closed k-simplexes ($0 \leq k \leq 3$) that satisfies:

(i) If $\sigma_T \in \mathcal{C}$ then $\sigma_{T'} \in \mathcal{C}$ for every $T' \subseteq T$.

(ii) The intersection of two simplexes in $\mathcal{C}$ is empty or is a face of both.

The particular complexes considered in the theory of α-shapes have vertices in the node set and simplexes from the Delaunay triangulation of the set, which is unique, as it is well known. The formal definition of the set of α-shapes of the cloud of nodes follows.

4.1.1 Definition of the family of α-shapes

α-shapes define a one-parameter family of shapes $\mathcal{S}_\alpha$ (α being the parameter), ranging from the ``coarsest'' to the ``finest'' level of detail. α can be seen, precisely, as a measure of this level of detail.

Let $\boldsymbol{N}$ be our finite set of points in $\Re^3$ and α a real number, with $0 \leq \alpha < \infty$. Let b be an α-ball, that is, an open ball of radius α. A k-simplex σ_T is said to be α-exposed if there exist an empty α-ball b with $T = \partial b \bigcap N$ where ∂ means the boundary of the ball. In other words, a k-simplex is said to be α-exposed if an α-ball that passes through its defining points contains no other point of the set N.

Thus, we can define the family of sets $F_{k,\alpha}$ as the sets of α-exposed k-simplices for the given set N. This allows us to define an α-shape of the set N as the polytope whose boundary consists on the triangles in $F_{2,\alpha}$, the edges in $F_{1,\alpha}$ and the vertices or nodes in $F_{0,\alpha}$.

Each k-simplex σ_T included in the Delaunay triangulation, $\mathcal{D}$, defines an open ball b_T whose bounding spherical surface (in the general case) ∂b_T passes through the $k+1$ points of the simplex. Let ϱ_T be the radius of that bounding sphere, then, the family $G_{k,\alpha}$, is formed by all the k-simplexes $\sigma_T \in \mathcal{D}$ whose ball b_T is empty and $\varrho_T < \alpha$. The family $G_{k,\alpha}$ does not necessarily form simplicial complexes, so Edelsbrunner and Mücke (1994) defined the α-complex, $\mathcal{C}_\alpha$, as the simplicial complex whose k-simplexes are either in $G_{k,\alpha}$, or else they bound $(k+1)$-simplexes of C_α . If we define the underlying space of $\mathcal{C}_\alpha$, $|\mathcal{C}_\alpha|$, as the union of all simplexes in $\mathcal{C}_\alpha$, the following relationship between α-shapes and α-complexes is found:

$$\mathcal{S}_\alpha = |\mathcal{C}_\alpha| \quad \forall 0 \leq \alpha < \infty \tag{4.1}$$

α-shapes provide a means so as to eliminate from the triangulation those triangles or tetrahedra whose size is bigger than the before-mentioned level of detail, α. Thus, we make a *filtration* of the triangles.

In Fig. 4.1 an example of the previously presented theory is presented. It represents some instances of the finite set of shapes for a cloud in a intermediate step of the simulation of a wave breaking at a beach.

4.1.2 How to choose the α-value

Many authors claim that the main difficulty with the α-shape technique is related to the choice of the α-value (Mandal and Murthy, 1997). In this section we provide a prac-

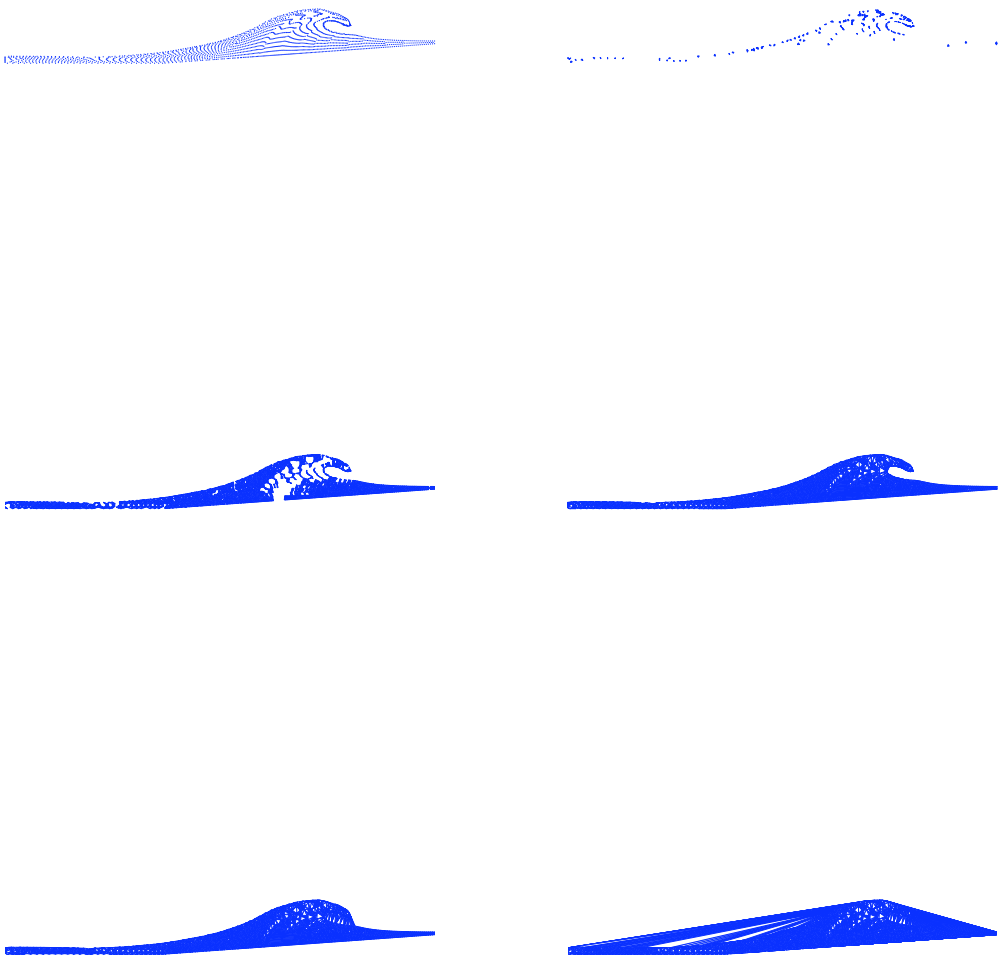


Figure 4.1. Evolution of the family of α-shapes of a cloud of points representing a wave breaking on a beach. Shapes $\mathcal{S}_0$ or cloud of points (a), $\mathcal{S}_{0.5}$ (b), $\mathcal{S}_{1.0}$ (c), $\mathcal{S}_{2.0}$ (d), $\mathcal{S}_{3.0}$ (e) and $\mathcal{S}_\infty$ (f) are depicted.

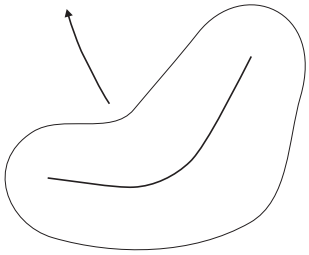


Figure 4.2. Medial axis of a two-dimensional curve.

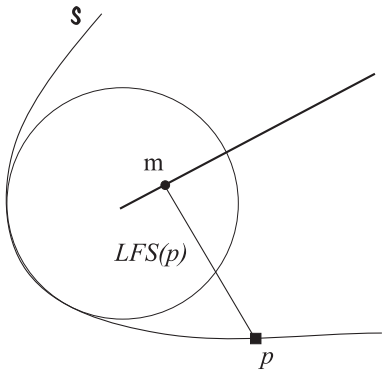


Figure 4.3. Computation of the LFS at a point p.

tical means to do so in the type problems we are dealing with. To this end, it will be necessary to give some prior definitions.

The *medial axis* (see for instance Amenta et al. (1998b) and references therein) of a $d-1$ dimensional, twice-differentiable, surface $\Gamma = \partial\Omega$ in $\Re^d$ is the closure of the set of points which have two or more closest points in Γ. An example of medial axis of a curve is shown in figure 4.2.

The *local feature size* (Amenta et al., 1998b), $LFS(p)$, of a point $p \in \Gamma$ is defined as the Euclidean distance from p to the closest point m on the medial axis. In Fig. 4.3 the computation of the LFS at a point is shown. Observe the difference between this concept and the radius of curvature of the curve at that point, which is different at different directions.

In mesh generation, the medial axis of a surface has been used to account for a measure of the desired point density in a region (see Armstrong et al. (1995)). To this end, it is useful to define a measure of the sampling density of the curve.

The surface Γ is said to be ε-*sampled* by a subset $\{n_I\}_{I=0}^{m}$ of the set of nodes $\boldsymbol{N}$ if every point $p \in \Gamma$ is within a distance $\varepsilon \cdot LFS(n_I)$ of a sample point $n_I \in \Gamma$.

In practical situations, it is common to have an explicit description of the boundary of the domain at the initial time step, or reference configuration —this will not be the

case for all the subsequent time steps, as mentioned before, since we try to avoid the use of boundary markers or similar techniques. At this configuration, we proceed by constructing an ε-sampling of the boundary curve or surface. Note that it should be twice differentiable in order to guarantee a non-vanishing LFS. In other words, it will not be possible to represent a sharp (concave) corner in the domain without the help of a segment chain (in 2D) or boundary triangulation (in 3D).

It is therefore important to choose the *level of detail* up to which we represent the initial configuration of the domain. Details of size lower than the chosen *discrete* LFS will not be represented by the method. In fact this is similar to the situation found when meshing a mechanical part, for instance. Many analysts choose to eliminate some details of the geometry irrelevant for the results.

Once we chose the desired level of detail for representing the initial configuration of the domain, we construct an ε-sampling of the boundary (with $\varepsilon < 1$) and extend the cloud of nodes to the interior of the domain, taking always the nodal distance measure, $h \approx \varepsilon \cdot LFS$.

As dictated by the preceding definitions, the choice of α such that $h < \alpha < LFS$ will provide a good approximation of the initial domain. In this way, triangles pertaining to the obtained shape of the domain will be bounded from above by the chosen LFS and from below by h. Thus, no triangle will overlap concave portions of the domain's boundary, nor spurious holes will appear. There exist, in addition, theoretical proofs of the convergence of the shape of the domain to the actual one with increasing nodal distributions, see for instance Mandal and Murthy (1997).

As the domain evolves, no further explicit definition of the boundary will be available, and the resulting shapes will never reproduce details of LFS lower than α, as is obvious (those triangles will be eliminated from the triangulation). However, for nodal discretizations fine enough, this technique provides very good results, with excellent mass conservation properties, see Martínez et al. (2004), González et al. (2007), Alfaro et al. (2006a).

4.1.3 Problems with the α-shape technique

There remain, however, some important problems in the application of α-shape techniques to updated Lagrangian simulations of flows with free surfaces. Maybe the most important is that, when contact between two portions of the domain, or auto-contact occurs, the LFS of portions of the boundary —precisely those getting into contact—decreases, and can be, during some time steps, below the threshold value α. This is precisely the

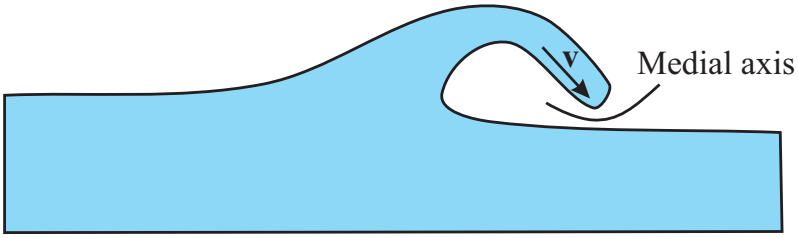


Figure 4.4. Evolution of the LFS at the neighbourhood of two surfaces getting into contact. A portion of the medial axis of points in the neighbourhood of the contating area is depicted. Remember that the LFS is the distance between the boundary and the medial axis. Thus, it vanishes rapidly in this situation.

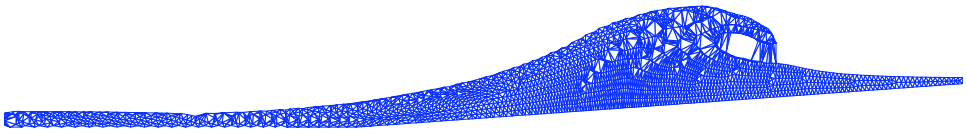


Figure 4.5. Spurious detection of contact at the crest of the wave.

situation that will happen shortly after the time step depicted in Fig. 4.1, see Fig. 4.4. If this happens, contact will be spuriously detected by the standard α-shape technique once the LFS is below α. In Fig. 4.5 an example is provided for the previous problem of spurious detection of auto-contact between the breaking wave and the surface of the sea. Note that contact is detected some time steps prior to its actual occurrence.

In the next section we propose two additional filtrations to be done after the α-shape filtration in order to improve the behaviour of the method.

4.2 Proposed algorithm

The proposed algorithm makes use of the information provided by previous time steps on the shape of the domain and, through the computed velocity field, on its future

shape. Thus, we will make use of the essential variable fields to improve the behaviour of the α-shape technique by performing a modified filtration processes over the Delaunay triangulation of the set of points.

In order to discern different parts of a body or different bodies getting into contact, we assume that all particles belonging to the same body should behave in a somewhat similar way. In our case, they all should move roughly with the same velocity or, more precisely, without jumps in the velocity nor steep gradients (this is true only for moderate Reynolds numbers in the flow, thus the proposed technique is not valid for turbulent flows). In this way, the k-simplexes found to be constituted by nodes that exhibit highly dissimilar characteristics should be regarded as *invalid* and filtered out of the α-shape.

For each k-simplex, a modified circumcircle criterion is employed. This modification includes a deformation parameter based on the differences between the associated nodal velocities. This parameter is used to alter the metric space. Elongating the Euclidean distance measured proportionally to the velocity differences causes the *invalid* simplexes to appear larger and therefore fail the circumcircle test.

To determine the deformation parameter in our case, a comparison is made between the different velocity vector directions. To this end, we first compute a principal direction $\boldsymbol{d}$, which is found as the *local normal direction* at the considered k-simplex (Teichmann and Capps, 1998)

$$\boldsymbol{d} = \sum_{i=1}^{k+1} s_i \boldsymbol{v}_i \text{ such that } ||\boldsymbol{d}|| = \max_{s_i=\pm 1} ||\sum_{i=1}^{k+1} s_i \boldsymbol{v}_i||, \tag{4.2}$$

where $\boldsymbol{v}_i$ represents each of the nodal velocities associated to the k-simplex, and $||\cdot||$ denotes the norm associated to the metric space.

We define the angle β as the one formed by each velocity vector with the principal direction $\boldsymbol{d}$. A deformation factor f_β is then obtained according to

$$f_\beta = 1 - \frac{|\beta_{\text{max}} - \beta_{\text{min}}|}{\pi}. \tag{4.3}$$

This factor allows to filter those k-simplexes formed by nodes of opposing or diverting velocities. Note that only if the simplex is ``large'' (according with an user provided measure, α) and its nodes move with very dissimilar velocities, it will be eliminated from the triangulation. If the triangle is small enough it will be most likely representing a recirculation in the flow, for instance, and will still be maintained in the model.

There are, however, cases in which only one of the bodies (or only some sub-region of the model) is moving and the previous filter alone would still detect a spurious contact. In that situation one or more nodes will not be taken into account by the above

factor, yet those simplexes need to be filtered. The need arises to take into account the gradient of those velocities, and calculate a deformation factor f_{mod} as

$$f_{\text{mod}} = 1 - \frac{\|v_i\|_{\text{max}} - \| v_j \|_{\text{min}}}{\| v_i \|_{\text{max}}} \tag{4.4}$$

Once the deformation factors are obtained we proceed to alter the metric tensor, assuming it constant at each simplex. The distance between two points X and Y, with coordinates $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively, can be defined as

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})\boldsymbol{M}(\boldsymbol{x} - \boldsymbol{y})^T} \tag{4.5}$$

where $\boldsymbol{M}$ represents the metric tensor. We define a ``modified'' metric tensor $\boldsymbol{M}$ with $\frac{1}{(f^a_{\text{mod}} f^b_\beta)}$ on the diagonal, where a and b are user defined parameters that allow adjusting the penalty owing to each factor depending on the nature of the simulation.

The newly deformed circumradius is used to check the α-shape test, usually making the unwanted simplexes fail. This process is performed on a simplex by simplex basis.

4.2.1 Choosing a and b.

The selected values for the new parameters a and b will depend on the nature of the problem under consideration. Two main factors will influence the selection: the ratio of α to the original simplex size and the relative difference between the nodal property values. Fig. 4.6 depicts in a simplified way the relationships between the mentioned factors, graphing the deformed size ratio d'/α versus the percentage of property variation. In this chart, the element will fail the test when d'/α is greater than one.

For these charts a 1-dimensional element of unitary size was assumed and only one property for each node —this property could represent the particle speed or the velocity direction—, so we only have to choose one parameter to calculate the deformation factor f. In Fig. 4.6(a), it is shown the effect of varying f for a fixed $\alpha = 4$, (so this element would easily pass a standard filtration). Notice how raising the deformation parameter (thus decreasing f) means that the percentage of difference in properties will make the test fail sooner. Thus if small differences in values are expected, f should be made accordingly small to be able to discern the invalid simplexes.

In Fig. 4.6(b), f was held constant at 2 and the α to element size ratio was varied from 2 to 32, to show that when this ratio is large, a higher f needs to be chosen to avoid making the test too strict.

While computing the Delaunay triangulation is necessary when dealing with Natural Element methods, it is not with the rest of meshless methods. It adds a little bit of

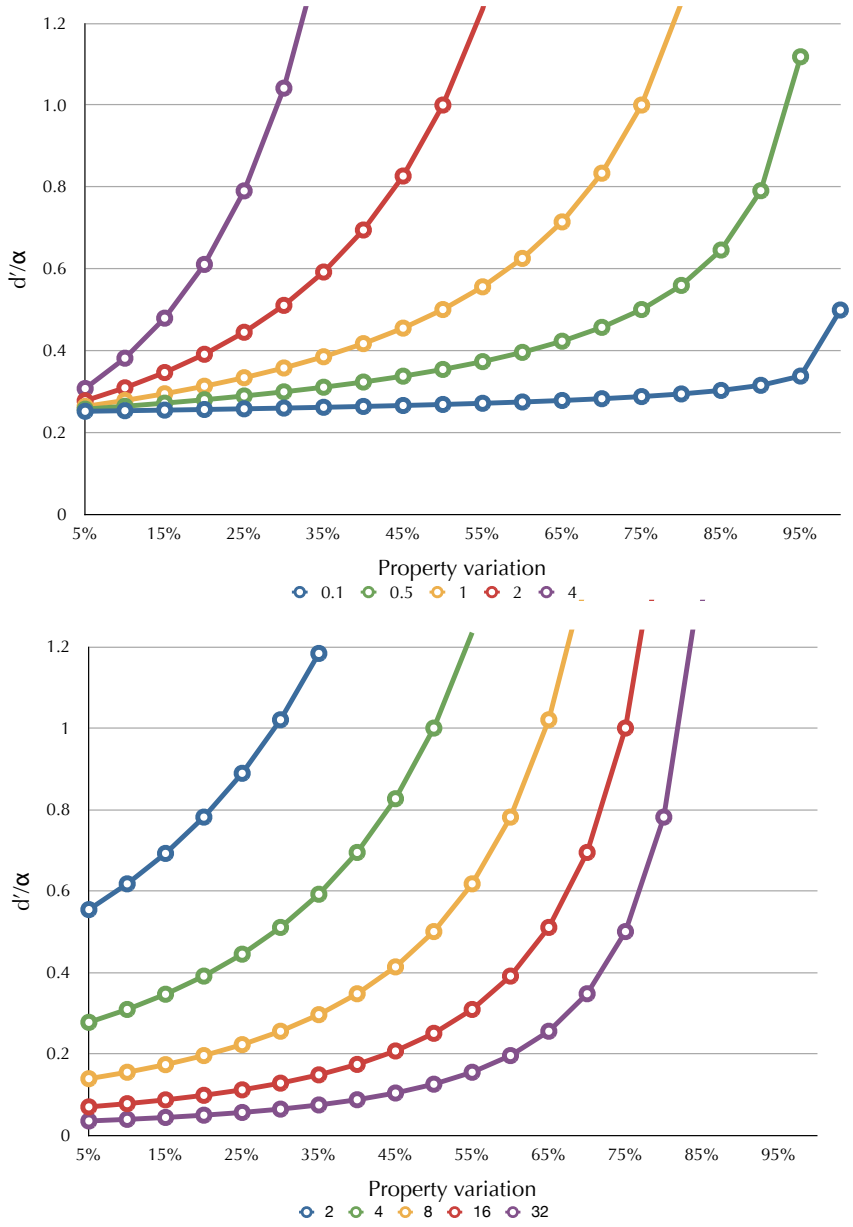


Figure 4.6. Relationship between the modifying factors. Variation in any of the deformation factors (a) and in alpha value (b).

CPU time to the simulation, that in general is negligible. Very efficient algorithms exist in the literature (see, for instance the *Qhull* software (Barber et al., 1996), which is free and very efficient). It is able to triangulate 1000 nodes in 0.016 CPU seconds on a laptop equipped with a Centrino processor and 500 Mb of RAM memory. The proposed filtration adds some very little extra CPU time to this, since it can be implemented within the Delaunay algorithm, or by adding a single `do while` loop to the code over all the triangles.

4.3 Examples

There is a wide variety of problems involving the presence of free or internal surfaces. Typically, Navier-Stokes equations in the presence of such boundary conditions are maybe the most ubiquitous example. But we do not restrict ourselves to Navier-Stokes equations. Even without the presence of inertia terms, many forming processes can be formulated in the so-called *flow formulation* (Zienkiewicz et al., 1978), (Zienkiewicz and Godbolet, 1974), if a rigid-(visco)plastic constitutive equation is assumed. Most of these forming processes (extrusion, forging, ...) imply the presence of free-surfaces, and very often the precise location of them, together with accurate determination of contact, auto-contact, etc. is of utmost importance.

We refer ourselves mainly to these last two examples: Navier-Stokes equations, as treated in Chapter 3 and the flow formulation of a rigid-plastic metal, briefly reviewed hereafter. Other problems are also suitable for the formulation here proposed.

4.3.1 Benchmarking

In order to validate the proposed method, it was employed in two classes of idealized cases of a 2D drop falling as a rigid body towards a wall, Fig 4.7. On the first class problems, see Fig. 4.7(a), the ball was dropped over a plane surface moving in the same direction at less speed than the ball. On the second family of cases, Fig. 4.7(b) both bodies move at the same speed but in different direction. The α parameter on all cases was chosen deliberately larger than actually needed, so that the α-shape would be a complete convex hull encompassing both bodies. That resulted in a triangulation that included several *invalid* triangles, shown in Fig. 4.8(b). These triangles could constitute an important error source due to the effects of a non-existing contact.

The first setting allows to test the effect of the gradient of velocities, taken into account by f_{mod}. At the limit case, the speed of the plane is null, so the deformation factor

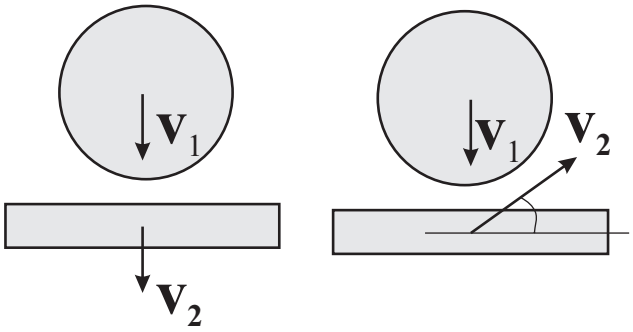


Figure 4.7. Method validation. Cases studied of a 2D ball drop over a flat surface. (a), $0 \leq \boldsymbol{v}_2 < \boldsymbol{v}_1$ and (b), $\boldsymbol{v}_2 = \boldsymbol{v}_1,\ 0 \leq \beta \leq \pi/2$.

goes to infinite, therefore the size of α becomes unimportant as the triangles composed by nodes from the two different surfaces will always fail the test. In this case α was set to 5—thus taken deliberately large—, a to 10 and b to 0.

Less extreme cases where tested, on which the surface was not completely still, but moving at less speed in the same direction of the ball. All cases resulted in successful filtrations. Fig. 4.9 shows a detail of the area on which both surfaces nearly touch. Triangles eliminated by the proposed filtration are shown in light grey. Specially noteworthy is the difference between the element sizes between the drop and the plate. Without an external filtration, there is no α which could manage to obtain a reasonable α-shape, given that the plate element size is more than five times the element size of the drop and the difference with the gap between both bodies is even more drastic. Density based filtrations could be made to recognize both areas, yet the case would still prove to be challenging if possible at all.

On the second family of cases the surface moves at the same speed but in different direction, still usually towards the ball. This exercise allows to check the performance of the filtration due to f_β. The case in which both bodies approach directly to each other is also an extreme situation where none of the offending triangles will ever pass the test regardless of the chosen α. In this case, α, a and b were 5 —again deliberately large—, 0 and 1, respectively. The angle difference has been tested up to the case where the bodies moved in a perpendicular way. In all the conditions both bodies could be recognized by adjusting the b factor only. In this case the difference between element sizes at the drop and the plate is also noteworthy.

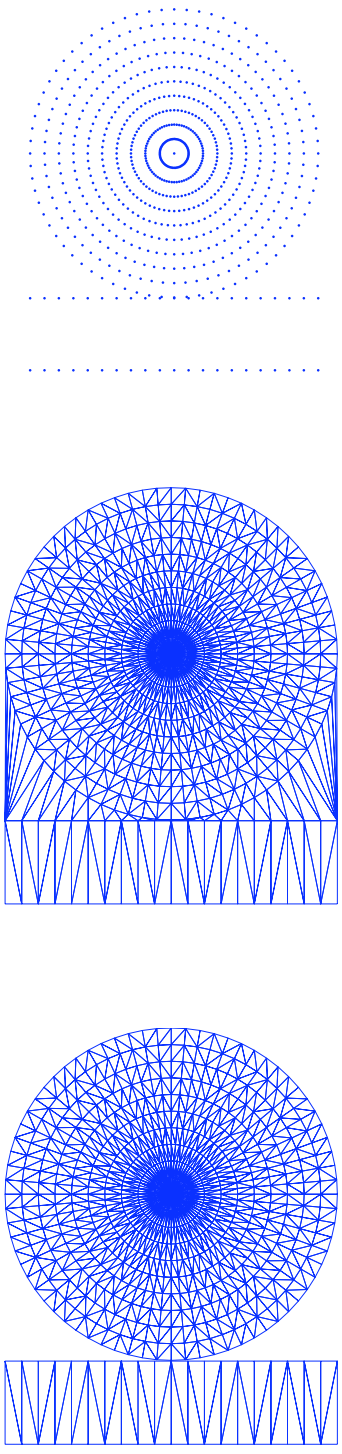


Figure 4.8. Drop approaching a surface. Both families of cases studied were tested on the same set of nodes (a).The resultant geometry of the domain provided the standard α-shape (b) and the modified method (c) are depicted.

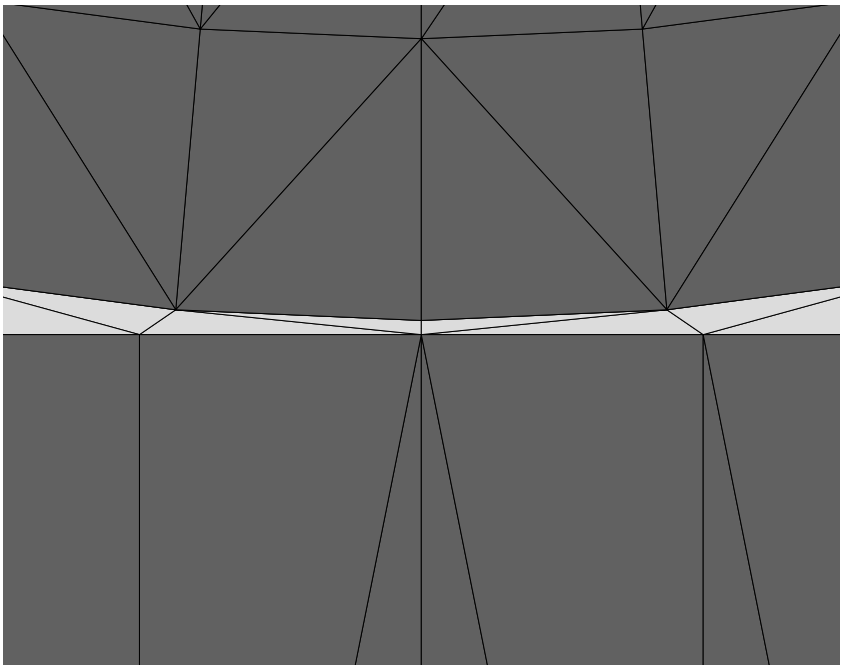


Figure 4.9. Drop approaching a still plate (detail of the contact zone). In light grey the triangles filtrated by the proposed techniques are shown.

Figure 4.10. Detail of the velocity field at the wave crest

4.3.2 2D wave breaking

A third type of test was performed with the wave problem shown in Figs. 4.1(a) to (f) and 4.5 in which we could check the performance on a *real* 2D case. The velocity field on the crest of the wave is shown in Fig. 4.10. It can be noticed how the vectors are roughly aligned in the same direction, thus resembling the first family of cases in the preceding section. Even though the velocity vectors seem to be very similar, the difference is so that the filtration is successful at the crest. Again, a reasonable value for the parameters a and b seems to be 10 and 1, respectively, and our experience dictates that this is so for a general problem presenting this kind of difficulty.

The results of the proposed technique are shown in Fig. 4.11. In this case the proposed method is able to discern between the crest and the trough of the wave. Again, the α value was taken deliberately too high, to show that even a poor choice of α will lead to a proper result.

Mass (volume) conservation is analyzed in Fig. 4.12. In this case, the predicted volume of the whole domain is analysed, taking into account that obtained by standard α-shape techniques and the one obtained by the proposed method. A sudden rise in volume implies spurious contacts. As can be noticed, the proposed method gives more accurate results, with less than 1% error in volume. The gain in volume due to spurious contact detection for the α-shape technique raises up to 5% for the final time steps, even if the contact region in the model is concentrated near the wave crest.

In this test we applied the modified filtration to the velocity fields obtained in a simulation performed with the standard α-shape method. This implies that after the spu-

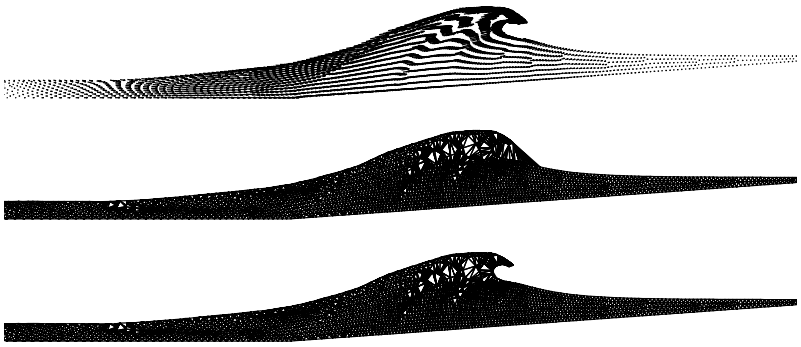


Figure 4.11. Wave before breaking. Velocity vectors (a), α-shape without additional filtration (b) and shape reconstructed with the new approach (c). In this cases the parameters used where $\alpha = 9$, $a = 10$, $b = 1$.

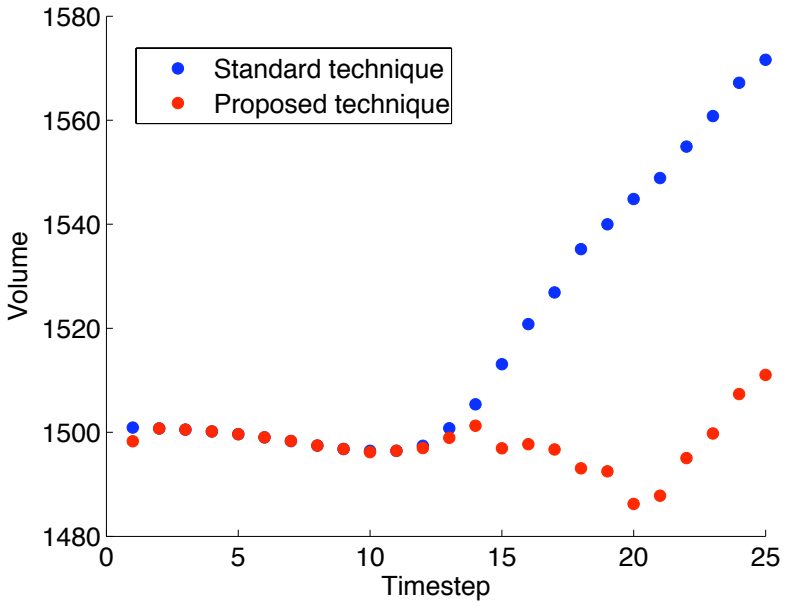


Figure 4.12. Volume conservation for the standard and the proposed technique.

rious contacts occurred the velocities at the crest where already compromised. This, however allowed us to test the capabilities of the proposed methodology to *fix* spurious shapes.

4.3.3 An extension to 3D problems. Aluminium extrusion

What follows is a short explanation of the extrusion problem, which was the last test case for this method and a proof of applicability for the 3D extension.

Flow formulations of rigid-plastic solids

As mentioned before, many forming processes can be formulated as free-surface problems under very standard assumptions. Although, to some extent, an elastic recovery exists at the end of many metal forming process, this is often neglected. In addition, the Cauchy stress is usually related to the strain rate tensor. This leads to a formulation that closely resembles that of non-Newtonian fluids, and hence the term *flow formulation* (Zienkiewicz and Godbolet, 1974).

Thus, the equations governing the metal deformation can be expressed in terms of velocities rather than displacements. Stresses produced in the forming process can be set in a simple form as

$$\boldsymbol{\sigma} = \boldsymbol{D}(\boldsymbol{d}, T) \cdot \boldsymbol{d}, \tag{4.6}$$

where $\boldsymbol{d}$ represents again the strain rate tensor (symmetric part of the velocity gradient) and T the temperature. Depending of the particular constitutive equation chosen for the metal, we thus obtain different formulations. In Alfaro et al. (2006b) and Alfaro et al. (2006a) a Sellars-Tegart temperature-dependent constitutive model was implemented in this framework.

Constitutive equations for aluminium

We considered a rigid-viscoplastic constitutive law for the aluminium, allowing for a *flow formulation* for the problem (Zienkiewicz and Godbolet, 1974). In essence, we neglect inertia terms in the Navier-Stokes equations and considered a non-linear constitutive law for the aluminium in the form

$$\boldsymbol{s} = 2\frac{\eta\bar{d} + \sigma_y(\bar{d})}{3\bar{d}}\boldsymbol{d}, \tag{4.7}$$

where $\boldsymbol{s}$ represents the deviatoric part of he strain rate tensor and σ_y represents the yield stress. $\overline{d}$ represents the equivalent strain rate. Note that, depending on the η value, the return to the yield surface is done with different velocity. Since it is common to describe aluminium behaviour as rigid-plastic (rather than viscoplastic) we employ null viscosity, so as to enforce $Y = \overline{\sigma} - \sigma_y = 0$, leading to

$$\boldsymbol{s} = \frac{2\sigma_y}{3\overline{d}}\boldsymbol{d}. \tag{4.8}$$

Finally, the constitutive equation, accounting the incompressibility of plastic flow results:

$$\boldsymbol{\sigma} = 2\mu\boldsymbol{d} - p\boldsymbol{I}, \text{ with } \mu = \frac{\sigma_y}{3\overline{d}}. \tag{4.9}$$

Linearized form of the variational problem

If we write the incremental variational equation at time $t + \Delta t$ we arrive to:

$$\int_{\Omega(t+\Delta t)} \Big(-(p^t + \Delta p)\boldsymbol{I} + 2\mu(\boldsymbol{d}^t + \Delta\boldsymbol{d})(\boldsymbol{d}^t + \Delta\boldsymbol{d})\Big) : \boldsymbol{d}^* d\Omega = 0. \tag{4.10}$$

Domain updating is done in an explicit procedure, given the last converged velocity field, but due to the non-linear character of the constitutive equations, an iterative approach has been applied to the conservation equations, using the Newton-Raphson scheme, thus leading to

$$\begin{aligned}\int_{\Omega(t+\Delta t)} \Big(-\Delta\Delta p\boldsymbol{I} + 2\mu\Big(\frac{\partial\mu(\boldsymbol{d}_k^{t+\Delta t})}{\partial\boldsymbol{d}} : \Delta\Delta\boldsymbol{d}\Big)\boldsymbol{d}_k^{t+\Delta t} + \\ +2\mu(\boldsymbol{d}_k^{t+\Delta t})\Delta\Delta\boldsymbol{d}\Big) : \boldsymbol{d}^* d\Omega = \\ = -\int_{\Omega(t+\Delta t)} (-p_k^{t+\Delta t}\boldsymbol{I} + 2\mu(\boldsymbol{d}_k^{t+\Delta t})\boldsymbol{d}_k^{t+\Delta t}) : \boldsymbol{d}^* d\Omega,\end{aligned} \tag{4.11}$$

where the subscript k indicates the iteration within a time increment. The incremental form of the incompressibility condition results

$$\int_{\Omega(t+\Delta t)} \boldsymbol{\nabla}\cdot(\Delta\Delta\boldsymbol{v})\; p^* d\Omega = -\int_{\Omega(t+\Delta t)} \boldsymbol{\nabla}\cdot(\boldsymbol{v}_k^{t+\Delta t}) p^* d\Omega. \tag{4.12}$$

If we approximate the velocities and pressures, as well as their variations, by employing a finite-dimensional set of basis functions, we arrive to a discrete form of the previous equations (Bubnov-Galerkin method)

$$\Delta\Delta\boldsymbol{v}^h(\boldsymbol{x}) = \sum_{I=1}^{n} \phi_I(\boldsymbol{x})\Delta\Delta\boldsymbol{v}_I \tag{4.13}$$

$$\Delta\Delta p^h(\boldsymbol{x}) = \sum_{I=1}^{n} \psi_I(\boldsymbol{x})\Delta\Delta p_I, \tag{4.14}$$

where n represents the number of nodes considered in the approximation. Laplace interpolations (Sukumar et al., 2001) are employed in this work to interpolate the velocity field, while Thiessen interpolation (piece-wise constant on each Voronoi cell) is used for pressures. Any other form of meshless approximations could also be employed as well. More details on the derivation of the model can be found in Alfaro et al. (2006b) and Alfaro et al. (2006a).

Performance of the proposed technique

We considered the simulation of the extrusion of a hollow cylinder. Tube extrusion is especially difficult to simulate from the geometrical point of view, since the diverted metal flow must converge before going through the last section of the extrusion die. A schematic representation of the geometry of the die is shown in Fig. 4.13, where only a quarter of the domain was represented. By invoking appropriate symmetry conditions, this same quarter of the domain was employed for simulation. Some snapshots of the flow of aluminium during this extrusion process are shown in Fig. 4.14, where post-processing has been employed for clarity, in order to show the whole geometry of the domain.

The domain is marked in red lines on Fig. 4.15. This figure also shows a particular time step where using regular α-shapes results in spurious contact detection. This is also notorious in Fig. 4.14(c). The *invalid* tetrahedra can be recognized by their size, larger than the nodal spacing in that area, giving a jagged feel to the resulting solid.

The model is divided in sections of different nodal densities, being the sparsest part at the top. Using the standard method special care is needed to avoid spurious contact. It is possible to define different α-values for each region to address this problem. Still a different problem arises when the last nodes of the model reach a section of small α, because the nodal spacing is then too large and all the elements get filtered.

With the new approach, a single, deliberately big, α-value can be defined and still obtain good results. In Fig. 4.15(b) a snapshot corresponding to the same time step on which traditional α-shapes failed to avoid the spurious contact is shown. In this case flow fronts are clearly kept apart from each other until actual contact occurs.

4.4 Conclusions

An improved α-shape technique for domain tracking in updated Lagrangian simulations of free surface flows was introduced in this chapter. This improvement is based

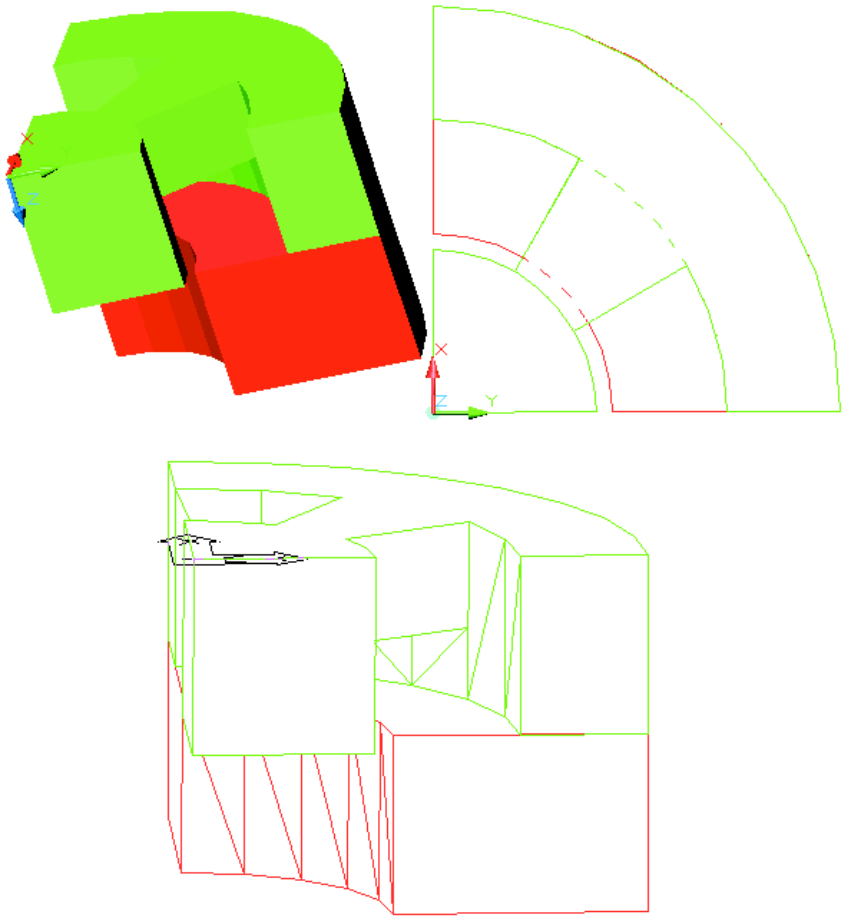


Figure 4.13. Schematic geometry of the die for the extrusion of a hollow profile. Note the special characteristics of the flow, that must divide to pass trough the green region and then re-join to flow out of the die (red region).

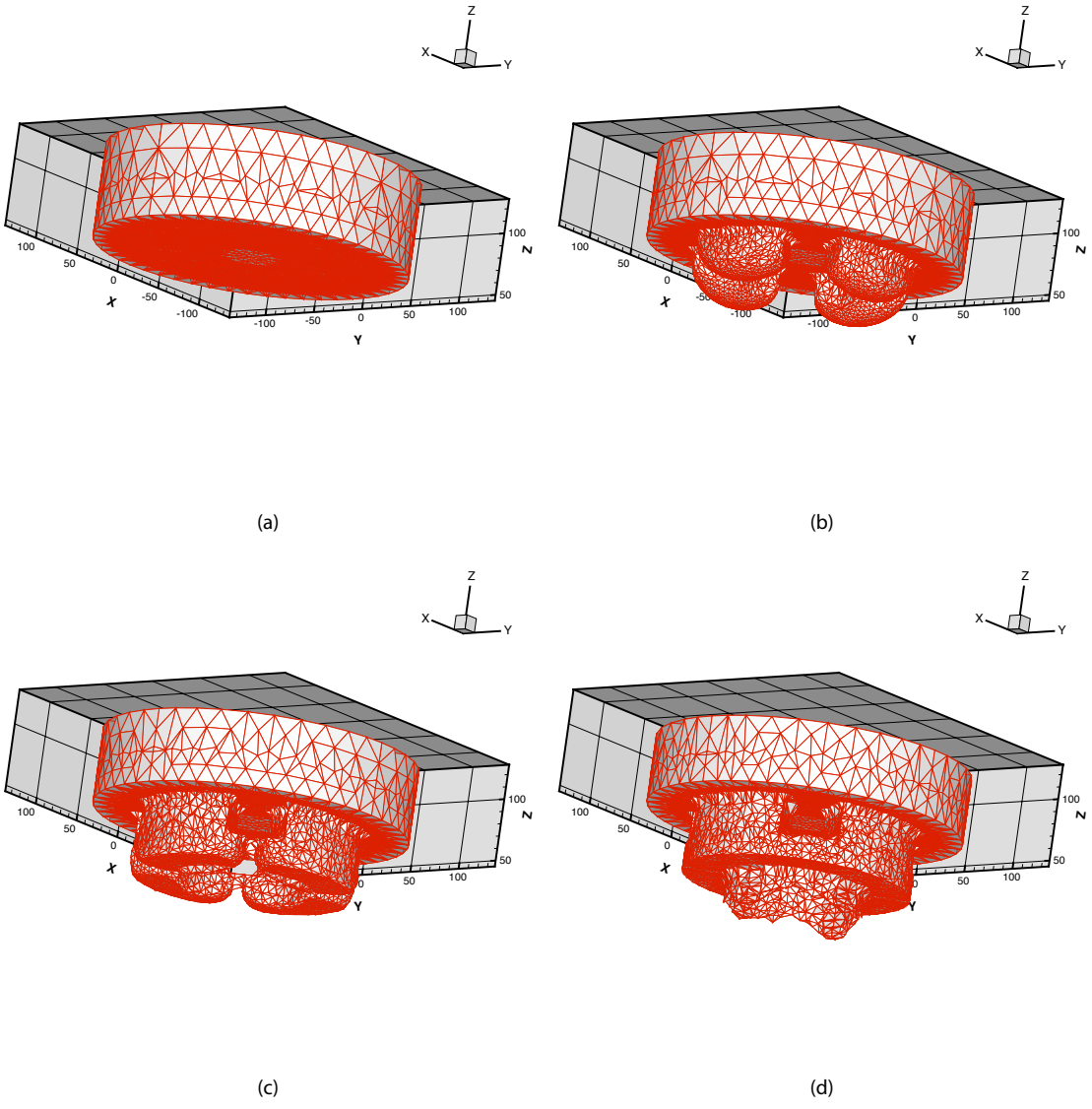


Figure 4.14. Sequence of aluminium flow at the early stages of the extrusion of a hollow cylindrical profile.

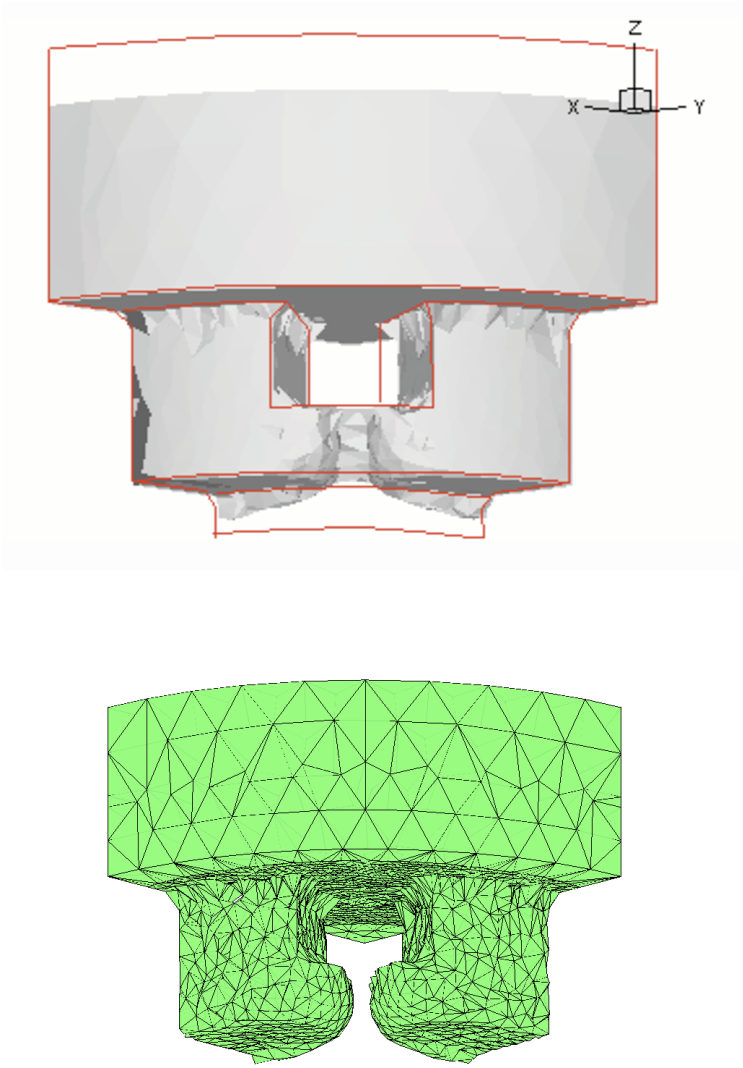


Figure 4.15. Extrusion process. Instant before contact of the two metal flows. (a) Simulation domain (red lines) and snapshot showing spurious contact between flows at an intermediate time step. (b) Spurious tetrahedra are removed from the triangulation ($\alpha = 8, a = 0.1, b = 2$).

on the addition a new filtration to the standard α-shape technique. This new filtration takes into account the velocity field of the flow, so as to predict in some sense its future geometry. We have introduced a way to use non-geometric information inherent to our model, as a tool to filter α-shapes and being able to obtain good surface definition, avoiding traditional problems associated to this method, as spurious contacts. Even in cases where there is large nodal density differences, the flow provides enough information to recognize, even for rough tuning of α-values, different regions in the model that pertain either to zones getting in contact or to different bodies in the simulation.

Despite the inclusion of two new user-defined parameters, the proposed methodology is flexible enough to face the geometry changes that occur with moderate Reynolds number flows, as covered by the presented formulation. The main conclusion is that it is considerably easier to find the three parameters α, a and b, than to find the single α value for some special, delicate cases —notably some time steps prior to contact—in standard α-shape technique. We have shown how, even for a poor selection of α, the proposed technique is able to correctly filtrate the actual geometry of the domain. Values of the parameters are much less sensible to modifications than α for standard α-shapes, and thus the ease of use and good results of the proposed technique.

Chapter 5

Applications to Non-Newtonian fluid flows

As could be grasped by the reader during the explanation of the Newtonian fluid constitutive equation in Chapter 3, not all fluids present a linear stress-deformation relationship. Those that do not are collectively named Non-Newtonian, and represent a broad class of very important and interesting fluids both for the academic and industrial worlds. Examples of these fluids can be found almost anywhere, from body fluids as blood or mucus to man-made substances like paint, shampoo or molten plastics. Of course not all non-Newtonian fluids behave in the same manner so many different constitutive equations, with a wide range of complexity, have been developed to describe their mechanical response.

In this thesis we have but scratched the surface of this broad topic while addressing the matter of simulating viscoelastic fluid flows. The great industrial importance of these fluids has originated a great deal of interest in its numerical simulation from a relatively long time. A great deal of success has been achieved, and nowadays we count with tools to solve a huge deal of non-trivial flows. Abundant literature can be found on the topic. The interested reader can be bothered to check Owens and Phillips (2002) and Crochet and Walters (1983) as a primer for techniques introduced prior 1983. Still the processes have been plagued with difficulties, many of which seem to come from the so called *high Weissenberg number problem* (the Weissemberg number is a non-dimensional measure of fluid elasticity). Regardless of the employed numerical method or the viscoelastic constitutive equations used, either differential or integral, when the elasticity of the fluid increased by a little measure, the simulation would suffer from a huge loss in accuracy leading to convergence problems. This situation limited the application spectrum of cases, rendering the numerical tools ineffective to deal

with industrial grade problems. The initial suspects of this situation were bifurcations in numerical solutions, the possible 3-dimensional effects in 2-dimensional flows and inability to cope with changes in the type of governing equations (Crochet et al., 1984).

At present, the most widely accepted causes for the high We problem are numerical approximation errors. In Owens and Phillips (2002), they are attributed mainly to three sources: First, errors caused by inaccurate integration schemes employed on the coupled non-linear elliptic-hyperbolic equations system governing viscoelastic flows. Second, numerical oscillations caused by trying to solve an ill-posed problem due to the badly chosen approximation spaces for the essential variables. This problem relates to the LBB or inf-sup condition, and has been addressed in the solution scheme employed in this thesis. Lastly, there is the issue of steep boundary layers not being solved in an adequate fashion due to coarse spatial discretization or a misrepresentation of the domain near singularities. This problem should be mitigated by the meshless character of the method here presented, as well as the possibility to easily add particles at critical points. Given that our method addresses some of the suspected causes of the high We problem, it would seem plausible to think that we could gain some ground in this field, as an alternative to the upwinding techniques (Marchal and Crochet, 1987) or discontinuous Galerkin methods (Lesaint and Raviart, 1974) currently employed to cope with this problem.

Amounting the already discussed, the interesting free surface phenomena that occur in some viscoelastic flows creates an unmissable setting for the applications tryout of the developed method. In all the developments that follow, the proposed second order in time natural neighbor Lagrange-Galerkin method developed in this thesis has been employed. As will be noticed, the methods shows excellent accuracy in problems where traditional, state-of-the-art techniques fail in some sense.

5.1 Governing Equations: The Oldroyd-B Fluid Model

In 1950, Oldroyd developed a constitutive equation that while simple, is useful in describing the general flow behavior of dilute polymeric solutions. This model can be obtained as an empirical generalization of the linear viscoelastic equation. For this, the constitutive relation is written in tensorial form and some admissibility conditions are enforced. Another —and probably more popular—way to derive this model is to consider a suspension of Hookean dumbbells in a Newtonian solvent and study it from a

molecular point of view. These dumbbells represent the polymer chains suspended in the Newtonian medium. They will react to the flow and to other dumbbells, and will add some extra resistance to the viscous character of the solvent. When the fluid stops the springs will remember their initial configuration, hence representing the elastic component of the model. Probably the most interesting part about this approach, is the way to obtain a —somewhat —accurate macroscopic model based on molecular assumptions. The Oldroyd-B model present a constant shear viscosity like that observed on Bogger fluids, which are highly elastic non-shear thinning fluids. Other fluids which exhibit non-shear thinning elastic properties are low molecular weight polymethysiloxanes (PDMS), polycarbonates and solutions of glass fibers in viscous Newtonian fluids. Still, this model is useful only at low shear rates. In extensional flows, the infinite extensibility of the hookean springs in the dumbbells yield an extensional viscosity that tends to infinity at a finite extensional rate.

5.1.1 Model Derivation

The afore mentioned dumbbells idealize the behavior of a polymer chain and consist of two beads connected by an ideal spring which obeys Hooke's Law (Figure 5.1).

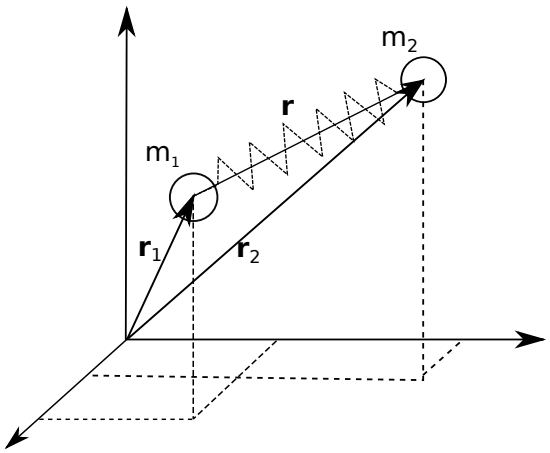


Figure 5.1. A sample dumbbell.

Now, following Owens and Phillips (2002), let us consider a single dumbbell immersed in a moving Newtonian fluid. There are three effects acting on each bead: A drag force, caused by the movement relative to the fluid around it; impact forces between beads, due to Brownian motion; and the spring force, representing the resistance of the molecule to be perfectly stretched.

The beads masses are denoted by m_i and both have a radius a. Each bead has a position vector r_i with $i = 1, 2$ relative to a fixed coordinate frame and $\boldsymbol{r} = \Delta r_i$ the dumbbell length. At this scale, the fluid flow field $\boldsymbol{u}$ around the beads is supposed to be homogeneous so that the rate of strain $\dot{\boldsymbol{\gamma}}$ is constant. In this situation, $\dot{\boldsymbol{\gamma}} = (\boldsymbol{\nabla u})^T$ and we can write that

$$\boldsymbol{u} = \boldsymbol{u_0} + (\boldsymbol{\nabla u})^{\boldsymbol{T}} \boldsymbol{r}, \tag{5.1}$$

at any point located at a position $\boldsymbol{r}$ and where $\boldsymbol{u_0}$ is a constant vector.

For each bead, Newton's second law is written as

$$m_i \frac{d^2 \boldsymbol{r_i}}{dt^2} = -\zeta_i \left(\frac{d\boldsymbol{r_i}}{dt} - (\boldsymbol{u_0} + (\boldsymbol{\nabla u})^T \boldsymbol{r_i}) \right) + \boldsymbol{B_i} + \boldsymbol{F_i}, \quad i = 1, 2 \tag{5.2}$$

The term $\boldsymbol{F_i}$ refers to the force that the coil exerts on each bead and $\boldsymbol{F}_1 = -\boldsymbol{F}_2 = \boldsymbol{F} = \lambda \Delta \boldsymbol{r}_i$, where λ is the spring constant, and is obtained from polymer parameters and thermal forces as

$$\lambda = \frac{3kT}{a^2}. \tag{5.3}$$

Here T denotes the fluid temperature and k is a proportionality constant.

The drag force on the bead is proportional to the velocity difference between the solvent ($\boldsymbol{u}$) and the bead ($d\boldsymbol{r}_i/dt$). At this scale, any hydrodynamic effects on the solvent caused by the presence of any other dumbbell in the vicinity are neglected. Gravity and other inertial effects are also left aside so we can make use of the Stokes equations. Under this conditions, the proportionality constant is $\zeta_i = 6\pi\eta_s a_i$

The term $\boldsymbol{B}_i$ refers to the impacts on the polymer chain due to Brownian motion and is written as

$$\boldsymbol{B}_i = -kT \frac{\partial}{\partial \boldsymbol{r}_i} ln\psi, \tag{5.4}$$

where ψ is a probability density function (pdf) which yields the probability that any given dumbbell length $\boldsymbol{r}$ is in the range $\boldsymbol{r}$ to $\boldsymbol{r} - d\boldsymbol{r}$ at some time t. This probability is independent of the position of the dumbbell and is given by $\psi(\boldsymbol{r}, t)d\boldsymbol{r}$.

Now, returning to the movement Eq. (5.1), in a strongly damped system the average particle velocity is almost constant so we can neglect the acceleration term on the left-hand side. After dividing by ζ_i and subtracting the two components of the system we

get

$$0 = \frac{d\boldsymbol{r}_1}{dt} - \frac{d\boldsymbol{r}_2}{dt} + (\nabla \boldsymbol{u})^T \boldsymbol{r}_2 - (\nabla \boldsymbol{u})^T \boldsymbol{r}_1 + \frac{kT}{\zeta_1}\frac{\partial}{\partial \boldsymbol{r}_1} ln\psi - \frac{kT}{\zeta_2}\frac{\partial}{\partial \boldsymbol{r}_2} ln\psi + \frac{\boldsymbol{F}_2}{\zeta_2} - \frac{\boldsymbol{F}_1}{\zeta_1} \quad (5.5)$$

It is possible to write that

$$\begin{aligned} \frac{kT}{\zeta_1}\frac{\partial}{\partial \boldsymbol{r}_1} \ln \psi - \frac{kT}{\zeta_2}\frac{\partial}{\partial \boldsymbol{r}_2} ln\psi &= -\frac{kT}{\zeta_1}\frac{\partial}{\partial \boldsymbol{r}} ln\psi - \frac{kT}{\zeta_2}\frac{\partial}{\partial \boldsymbol{r}} ln\psi \\ &= -kT\left(\frac{1}{\zeta_1} + \frac{1}{\zeta_2}\right)\frac{\partial}{\partial \boldsymbol{r}} ln\psi \end{aligned}$$

so

$$\frac{d\boldsymbol{r}}{dt} = (\nabla \boldsymbol{u})^T \boldsymbol{r} - kT\zeta_{12}\frac{\partial}{\partial \boldsymbol{r}} ln\psi - \zeta_{12}\boldsymbol{F}, \quad (5.6)$$

Here the relationship

$$\zeta_{12} = \frac{1}{\zeta_1} + \frac{1}{\zeta_2}$$

was used.

The probability balance equation between ψ and the probability flux vector $\boldsymbol{J} \equiv \dot{\boldsymbol{r}}\psi$ is

$$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial \boldsymbol{r}} \cdot \boldsymbol{J} = 0. \quad (5.7)$$

Multiplying Eq. (5.6) by ψ, differentiating with respect to $\boldsymbol{r}$ and using Eq. (5.7) we obtain the so-called *Smoluchowski equation*:

$$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial \boldsymbol{r}} \cdot \left[(\nabla \boldsymbol{u})^T \boldsymbol{r}\psi - \psi kT\zeta_{12}\frac{\partial}{\partial \boldsymbol{r}} ln\psi - \psi\zeta_{12}\boldsymbol{F}\right] = 0, \quad (5.8)$$

$$\Rightarrow \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial \boldsymbol{r}} \cdot \left[(\nabla \boldsymbol{u})^T \boldsymbol{r}\psi - kT\zeta_{12}\frac{\partial \psi}{\partial \boldsymbol{r}} - \psi\zeta_{12}\boldsymbol{F}\right] = 0, \quad (5.9)$$

since

$$\psi\frac{\partial}{\partial \boldsymbol{r}} ln\psi = \psi\frac{1}{\psi}\frac{\partial \psi}{\partial \boldsymbol{r}} = \frac{\partial \psi}{\partial \boldsymbol{r}}.$$

Now,

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{r}} \cdot \left((\nabla \boldsymbol{u})^T \boldsymbol{r}\psi\right) &= \frac{\partial}{\partial \boldsymbol{r}} \cdot \left((\nabla \boldsymbol{u})^T \boldsymbol{r}\right)\psi + \left((\nabla \boldsymbol{u})^T \boldsymbol{r}\right) \cdot \frac{\partial \psi}{\partial \boldsymbol{r}}, \\ &= (\nabla \cdot \boldsymbol{u})\psi + \left((\nabla \boldsymbol{u})^T \boldsymbol{r}\right) \cdot \frac{\partial \psi}{\partial \boldsymbol{r}}, \\ &= \left((\nabla \boldsymbol{u})^T \boldsymbol{r}\right) \cdot \frac{\partial \psi}{\partial \boldsymbol{r}}, \end{aligned}$$

so we finally arrive at the diffusion equation for ψ, which is the already mentioned Fokker-Plank equation (Owens and Phillips, 2002):

$$\frac{\partial \psi}{\partial t} + \left((\nabla \boldsymbol{u})^T \boldsymbol{r}\right) \cdot \frac{\partial \psi}{\partial \boldsymbol{r}} - kT\zeta_{12}\frac{\partial^2 \psi}{\partial \boldsymbol{r}^2} - \zeta_{12}\frac{\partial}{\partial \boldsymbol{r}} \cdot (\psi \boldsymbol{F}) = 0. \quad (5.10)$$

The ensemble average $\langle\cdot\rangle$ for any function f of $\boldsymbol{r}$ is defined as

$$\langle f(\boldsymbol{r})\rangle = \int_{\Re^3} f(\boldsymbol{r})\,\psi(\boldsymbol{r},t)\,d\boldsymbol{r}, \tag{5.11}$$

and it is possible to express the extra-estress tensor $\boldsymbol{T}$ [1] to the ensemble average of the dyadic product $\boldsymbol{rF}$ via the Kramers expression:

$$\boldsymbol{T} = -nkT\boldsymbol{I} + \eta_s\dot{\boldsymbol{\gamma}} + n\langle\boldsymbol{rF}\rangle, \tag{5.12}$$

where n denotes the number density of dumbbells.

In order to reach the extra-stress constitutive equation, one must multiply the Fokker-Plank Eq. (5.10) by $\boldsymbol{rr}$, integrate over $\Re^3$ and use the divergence theorem, noting also that $\psi \to 0$ as $|\boldsymbol{r}|$ tends to the maximum permissible polymer length. We get to

$$\overset{\nabla}{\boldsymbol{rr}} = 2kT\zeta_{12}\boldsymbol{I} - 2\zeta_{12}\langle\boldsymbol{rF}\rangle, \tag{5.13}$$

Where $\overset{\nabla}{\boldsymbol{rr}}$ is the upper-convected derivative of $\langle\boldsymbol{rr}\rangle$, defined as

$$\overset{\nabla}{\boldsymbol{rr}} = \frac{D}{Dt}\langle\boldsymbol{rr}\rangle - (\nabla\boldsymbol{u})^T\langle\boldsymbol{rr}\rangle - \langle\boldsymbol{rr}\rangle(\nabla\boldsymbol{u}). \tag{5.14}$$

Using this expression into Eq. (5.12) we obtain the *Giesekus expression* for the stress tensor:

$$\boldsymbol{T} = \eta_s\dot{\boldsymbol{\gamma}} - \frac{n}{2\zeta_{12}}\overset{\nabla}{\boldsymbol{rr}}. \tag{5.15}$$

Since the model includes a Hookean spring, we have that

$$\boldsymbol{F} = H\boldsymbol{r}, \tag{5.16}$$

where H is a positive constant parameter. This means that Eq.(5.12) can be rewritten as

$$\boldsymbol{T} = -nkT\boldsymbol{I} + \eta_s\dot{\boldsymbol{\gamma}} + nH\langle\boldsymbol{rr}\rangle. \tag{5.17}$$

Taking the upper convected derivative, in Eq. (5.15) and noting that

$$\overset{\nabla}{\boldsymbol{I}} = -\nabla\boldsymbol{u} - (\nabla\boldsymbol{u})^T = -\dot{\boldsymbol{\gamma}} \tag{5.18}$$

is possible to eliminate $\langle\boldsymbol{rr}\rangle$ from the stress expressions and arrive at

$$\boldsymbol{T} - \eta_s\dot{\boldsymbol{\gamma}} = -\frac{1}{2H\zeta_{12}}\left[\overset{\nabla}{\boldsymbol{T}} - nkT\dot{\boldsymbol{\gamma}} - \eta_s\overset{\nabla}{\dot{\boldsymbol{\gamma}}}\right]. \tag{5.19}$$

[1] This tensor is related to the total fluid stress $\boldsymbol{\sigma}$ by $\sigma_{ij} = -p\delta_{ij} + T_{ij}$

This expression can be written in terms of the following polymer physical parameters: polymer viscosity (η_p), characteristic relaxation time for the fluid (λ_1) and characteristic retardation time (λ_2); each defined as

$$\eta_p = \frac{nkT}{2H\zeta_{12}}, \tag{5.20}$$

$$\lambda_1 = \frac{1}{2H\zeta_{12}}, \tag{5.21}$$

$$\lambda_2 = \frac{\eta_s}{2(\eta_p+\eta_s)H\zeta_{12}} = \frac{\eta_s\lambda_1}{(\eta_p+\eta_s)}; \tag{5.22}$$

so that Eq. (5.19) now reads

$$\boldsymbol{T} + \lambda_1 \overset{\nabla}{\boldsymbol{T}} = \eta_0 \left(\dot{\boldsymbol{\gamma}} + \lambda_2 \overset{\nabla}{\dot{\boldsymbol{\gamma}}} \right), \tag{5.23}$$

with $\eta_0 = \eta_s + \eta_p$ as the total fluid viscosity. Eq. (5.23) is the Oldroyd constitutive equation.

We now separate the stress in the solvent and polymeric components as

$$\boldsymbol{T} = \eta_s \dot{\boldsymbol{\gamma}} + \boldsymbol{\tau}, \tag{5.24}$$

and substitute into (5.23) to get

$$\boldsymbol{\tau} + \lambda_1 \overset{\nabla}{\tau} = \eta_p \dot{\boldsymbol{\gamma}}, \tag{5.25}$$

which is the constitutive equation for the elastic stress. It is possible to note from this equation that there is no need to solve the Fokker-Plank equation in oder to find the extra-stress. When $\eta_e \to 0$, $\boldsymbol{T} \equiv \boldsymbol{\tau}$ and this model reduces to the Upper Convected Maxwell Model.

5.1.2 Model Implementation

Let us recall the the Navier-Stokes problem as defined in Ch. 3

$$\rho(\boldsymbol{v}_{,t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}) = \nabla\cdot\boldsymbol{\sigma} + \rho\boldsymbol{b} \quad \text{in } \Omega\times(0,T), \tag{5.26}$$

$$\nabla\cdot\boldsymbol{v} = 0 \quad \text{in } \Omega\times(0,T) \tag{5.27}$$

$$\boldsymbol{v}(\boldsymbol{x},t) = \boldsymbol{v}_D(\boldsymbol{x},t), \quad \boldsymbol{x}\in\Gamma_D, \quad t\in(0,T) \tag{5.28}$$

$$\boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{x},t) = \boldsymbol{t}(\boldsymbol{x},t), \quad \boldsymbol{x}\in\Gamma_N, \quad t\in(0,T), \tag{5.29}$$

and introduce the Oldroyd B constitutive equation. Equation (5.26) will change to

$$\rho(\boldsymbol{v}_{,t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}) = \nabla\cdot\boldsymbol{\sigma}_n + \nabla\tau + \rho\boldsymbol{b}, \tag{5.30}$$

where $\boldsymbol{\sigma}_n$ denotes the Newtonian component of the stress: $-p\boldsymbol{I} + \eta_s\dot{\boldsymbol{\gamma}}$.

We expand now the upper convected derivative in Eq. (5.25) to get

$$\boldsymbol{\tau} + \lambda_1 \left(\frac{D\boldsymbol{\tau}}{Dt} - (\nabla \boldsymbol{u})^T \boldsymbol{\tau} - \boldsymbol{\tau} (\nabla \boldsymbol{u}) \right) = \eta_p \dot{\boldsymbol{\gamma}}, \tag{5.31}$$

from which

$$\frac{D\boldsymbol{\tau}}{Dt} = \frac{\eta_p \dot{\boldsymbol{\gamma}} - \boldsymbol{\tau}}{\lambda_1} + (\nabla \boldsymbol{u})^T \boldsymbol{\tau} + \boldsymbol{\tau} (\nabla \boldsymbol{u}) \tag{5.32}$$

and we will employ a first-order time discretization

$$\frac{D\boldsymbol{\tau}}{Dt} = \frac{\boldsymbol{\tau}^{n+1} + \boldsymbol{\tau}^n}{\Delta t}. \tag{5.33}$$

In the implementation here developed it is assumed that the extra-stress at $t = t_n$ is known, so when solving for $u|_{t=n+1}$ all terms in Eq.(5.30) are known. After the velocity field has been obtained, the velocity derivatives, which are known for the integration points, are projected to the nodes (this intermediate step is by no menas necessary if Stabilized Conforming Nodal Integration is used). This intermediate variable along with $\boldsymbol{\tau}_n$ is used to calculate $\boldsymbol{\tau}_{n+1}$ according to

$$\boldsymbol{\tau}^{n+1} = \boldsymbol{\tau}^n + \Delta t \left(\frac{\eta_p \dot{\boldsymbol{\gamma}}^n - \boldsymbol{\tau}^n}{\lambda_1} + (\nabla \boldsymbol{u}^n)^T \boldsymbol{\tau}^n + \boldsymbol{\tau}^n (\nabla \boldsymbol{u}^n) \right). \tag{5.34}$$

Notably, if the method here presented is compared to that of Tome et al. (2007), or in general any based upon finite difference or finite volume schemes, it must be highlighted that no special treatment is necessary to compute the stress tensor along the boundaries.

5.1.3 Model Validation: Fully developed flow Inside a Pipe

Aiming at checking the accuracy of the proposed technique, a developed flow was simulated inside a completely full pipe of length L and radius $R = 1$. In this case we used an axisymmetric representation of the problem with the symmetry axis set on $x = 0$.

On the Pipe walls ($r = R$) a no-slip boundary condition was applied ($v_r = v_z = 0$) while no special outflow conditions are necessary. Any particle that crossed the boundary set on $z = L$, was eliminated from the simulation. Similarly, the flow is forced by entering a set of particles through the inflow boundary. At the entrance, velocity conditions were those of a fully developed flow, that is,

$$v_r = 0 \tag{5.35}$$

$$v_z = \frac{2U(R^2 - r^2)}{R^2} \tag{5.36}$$

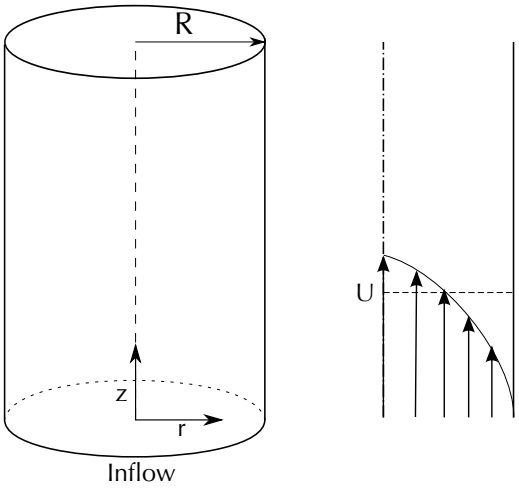


Figure 5.2. Flow schematics

where U is the average velocity.

Two non-dimensional numbers are widely used to characterize the flow of non-Newtonian fluids. The first is the Reynolds number Re, defined as

$$Re = \frac{\rho_0 U L}{\mu_0}, \tag{5.37}$$

and

$$We = \frac{\lambda_0 U}{L} \tag{5.38}$$

which is called the Weissenberg number. This dimensionless number is the ratio of the relaxation time of the fluid and a specific process time and represents a non-dimensional measure of the fluid's elasticity.

Since the non Newtonian behavior depends on the history of the flow and we assume that the particles are already moving at $t < 0$, it is necessary for each of them to know its past extra-stress tensor. For this reason the initial velocity and stress condition in all the domain were imposed in the same way as is done at the inflow boundary.

For a node cloud composed initially by 3999 nodes (remember that the simulation is left until the steady-state is reached, while nodes enter and leave the domain through the inflow and outflow boundaries), the solution was stable from the first step, and after 500 time steps, the extra-stress field was as shown in Figs. 5.3 and 5.4. The τ_{xx} component is not shown here because it vanishes for the whole domain.

As can be noticed in Figs. 5.5 through 5.7, where the analytical solution for τ_{xy}, τ_{yy} and $\boldsymbol{v}_y$ is shown by a red line; the agreement between the expected and behavior and the results obtained is remarkable. The $\mathcal{L}_2$-norm of the errors are $Err_{\tau_{xy}} = 6.3027 \cdot 10^{-06}$, $Err_{\tau_{yy}} = 1.4730 \cdot 10^{-06}$ and $Err_{V_y} = 2.3987 \cdot 10^{-07}$.

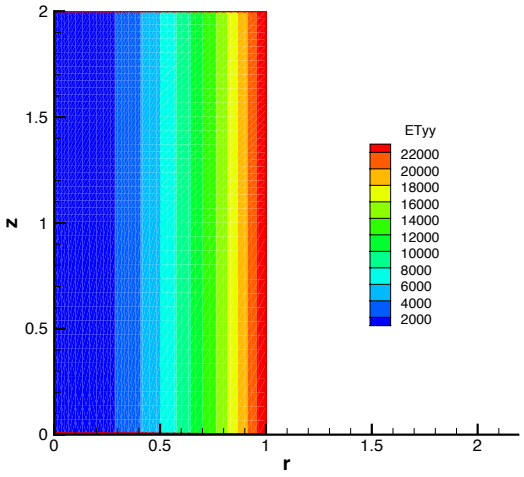


Figure 5.3. Extra-stress field for a fully developed pipe flow: τ_{yy}

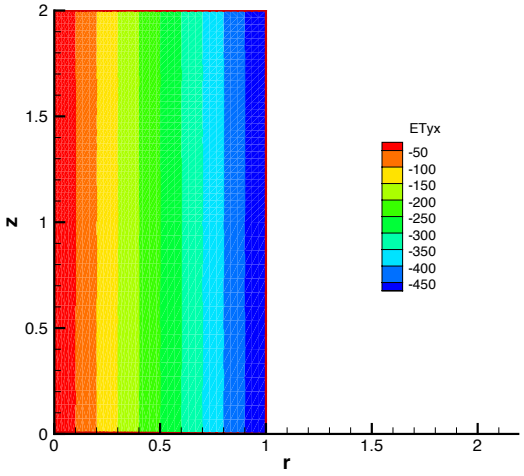


Figure 5.4. Extra-stress field for a fully developed pipe flow: τ_{xy}

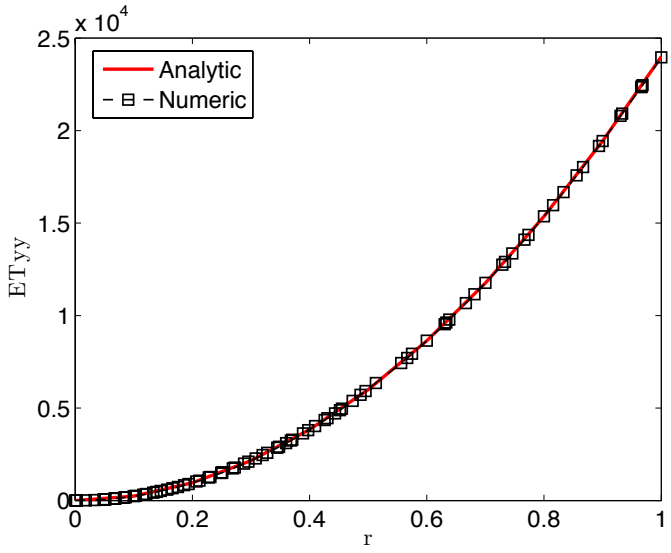


Figure 5.5. Extra-stress field for a fully developed pipe flow: τ_{yy} at $y = 1$

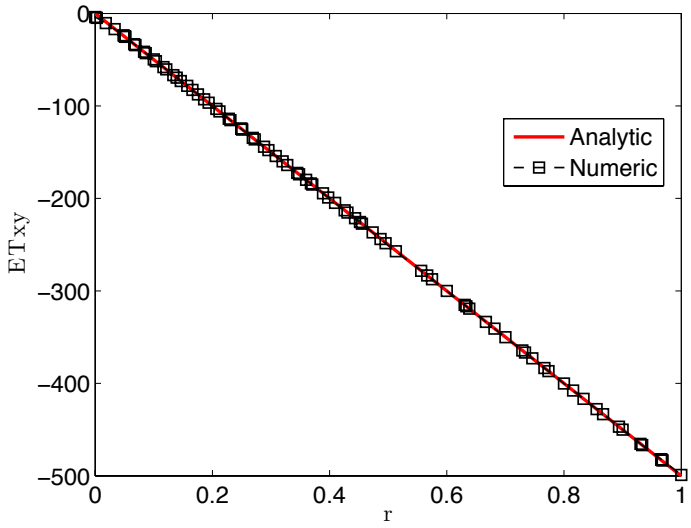


Figure 5.6. Extra-stress field for a fully developed pipe flow: τ_{xy} at $y = 1$

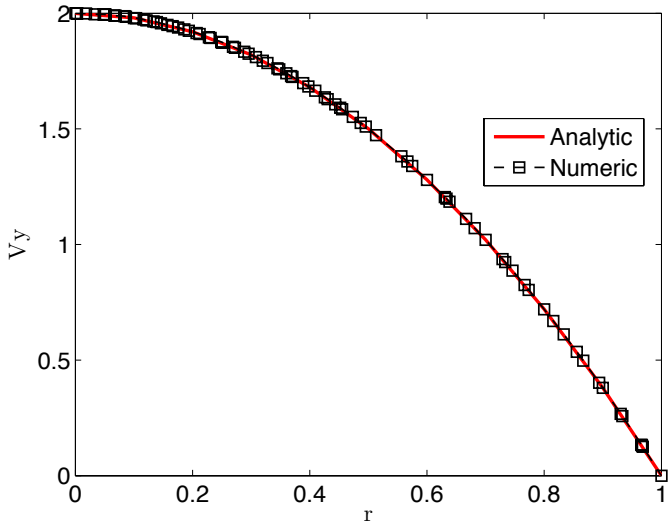


Figure 5.7. Velocity profile for a fully developed pipe flow: $\boldsymbol{V}_y$ at $y = 1$

5.2 Other, more complex, non-Newtonian Examples

In order to demonstrate the method's ability to reproduce the motion of non-Newtonian flows we simulated an extrusion process which causes the *die swelling* effect, on one hand, and the impact of a splashing drop on the other. Each problem has been studied by many authors and almost constitutes a benchmark for the simulation of non-Newtonian flows in the presence of free surfaces.

5.2.1 Die Swelling

A visco-elastic fluid jet presents a characteristic behavior known as *extrudate swell* that consist in the jet expansion in the direction perpendicular to the stream after leaving the extrusion die. This phenomenon is also known as *die swelling*. It is a very important effect in the polymer industry because many processes involve the extrusion of viscoelastic fluids, for example plastic in their molten state. The phenomenon of die swell may be explained by elastic recovery. The molecules are stretched by the shear forces in the pipe and the average axial stress at the exit is a tension.

In this problem we simulated the exit of an Oldroyd-B fluid trough a circular extrusion die of radius R and length L. The swell ratio S_r, which is defined as the ratio of the

maximum diameter of the jet and the diameter of the die, was measured as a way to quantify the swelling effect.

The problem setup is similar to the pipe flow, a fully developed flow passes through a pipe which imposes a no-slip condition to the particles in contact with it as in the previous example. The only difference is that the pipe has a finite length, that in this case causes all nodes reaching the end of the pipe to be free of any condition. Gravity and inertial effects were neglected.

Following the steps of Tome et al. (2007), different flows were simulated with the following parameters: $R = 0.1m$, $U = 1m/s$, $\mu_0 = 0.01m^2s^{-1}$ and $\lambda_1 = 0.01$. The scaling parameters were R, U, μ_0 and λ_1. Therefore $Re = 1$ and $We = 1$. The ratio λ_2/λ_1 took values of 0 (Newtonian case), 0.1, 0.5, 0.7 and 0.9. The effective Weissenberg number, defined as

$$We_{ef} = \left(1 - \frac{\lambda_2}{\lambda_1}\right) We, \tag{5.39}$$

was therefore $We_{ef} = 0.9, 0.5, 0.3$ and 0.1.

The swelling ratios obtained were lower than those obtained by Tomé Tome et al. (2007) , who reported values of $S_r = 2.13, 1.88$ and 1.37 for $We_{ef} = 0.9, 0.5$ and 0.1. Our model resulted in $S_r = 1.504, 1.435, 1, 236$ and 1.133 for $We_{ef} = 0.9, 0.5, 0.3$ and 0.1 respectively. Figure 5.8 presents the fluid contours at a point at which the die swelling for each fluid has already stabilized. It is possible to notice the deviation from the Newtonian behavior as We increases.

In a later section of the same work, a comparison is made against the works of Crochet and Keunings (1982) and the analytical solution to this problem by Tanner (2005). These tests allow to check the proposed method in a more meaningful way. In his work, Tanner used the *recoverable shear* which is a non-dimensional number defined as

$$S_R = \frac{\tau_{zz}}{2\left[\tau_{rz} + \frac{2\lambda_2}{Re\lambda_1}\dot{\gamma}_{rz}\right]}, \tag{5.40}$$

evaluated at the pipe wall.

Since the flow is fully developed inside the pipe, we use Eqs. (5.35) and (5.36) so the relevant terms in equation (5.40) in their non-dimensional form are

$$\begin{aligned}
\tau_{zz} &= 2We\tau_{rz}\frac{\partial v_z}{\partial r}, \\
\tau_{rz} &= \frac{1}{Re}\left(1 - \frac{\lambda_2}{\lambda_1}\right)\frac{\partial v_z}{\partial r}, \\
\dot{\gamma}_{rz} &= \frac{1}{2}\frac{\partial v_z}{\partial r}.
\end{aligned}$$

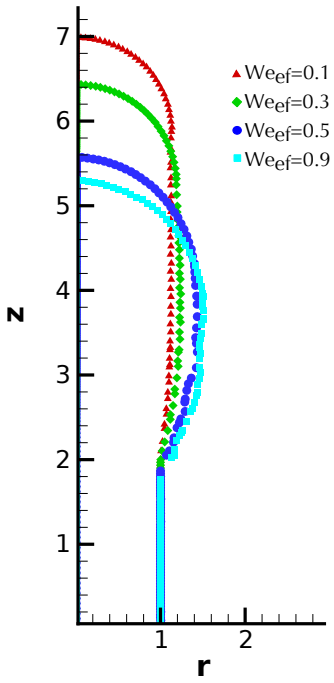


Figure 5.8. Swelling comparison between fluids with different Weissenberg numbers

thus

$$S_R = \frac{2We\tau_{rz}(\partial v_z/\partial r)}{2\left[\frac{1}{Re}\left(1-\frac{\lambda_2}{\lambda_1}\right)\frac{\partial v_z}{\partial r}+\frac{1}{Re}\left(\frac{\lambda_2}{\lambda_1}\right)\frac{\partial v_z}{\partial r}\right]} = We\left(1-\frac{\lambda_2}{\lambda_1}\right)\frac{\partial v_z}{\partial r},$$

and since in this case

$$\left|\frac{\partial v_z}{\partial r}\right|_{r=\pm R} = 4 \implies S_R = 4\left(1-\frac{\lambda_2}{\lambda_1}\right)We$$

The theoretic swelling ratio S_r of an axisymmetric jet can be predicted in a simplified manner via the equation (Tanner, 2005)

$$\frac{D_{max}}{D} = 0.14 + \left[1+\frac{S_R^2}{2}\right]^{\frac{1}{6}} \tag{5.41}$$

Cases were run using the following parameters: $R = 1m, U = 1m/s, v_0 = 4m^2/s,$ yielding $Re = 0.25, We = \lambda_1$. A fixed ratio $\lambda_2/\lambda_1 = 1/9$ was used as in Crochet and Keunings (1982) and Tome et al. (2007). Weissenberg numbers were varied from 0.125 to 1.125 with increments of 0.125. This represented a *recoverable shear* range between 0.44 and 4, both inclusive.

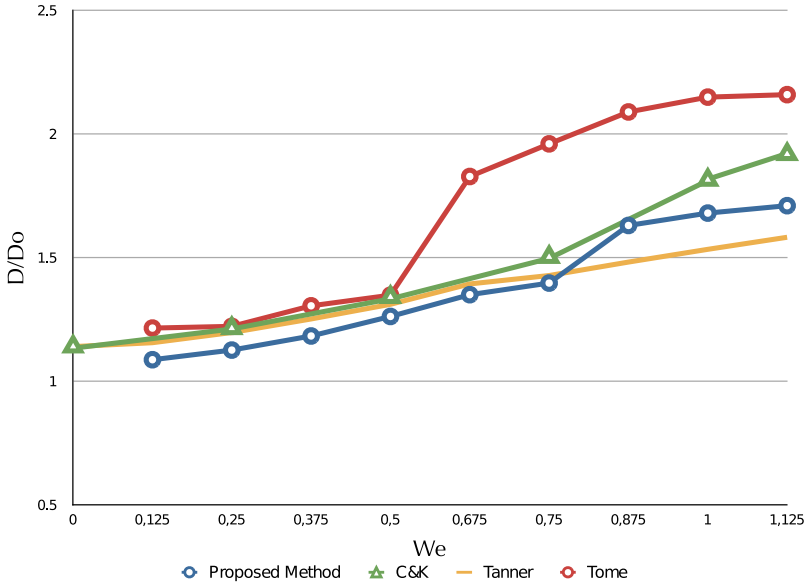


Figure 5.9. Swelling ratio against We. Different numerical models compared against an approximated theoretical solution by Tanner (2005).

In Fig. 5.9 results obtained with the proposed technique are presented and compared against the cited works. The yellow line represents the theoretical solution as obtained by Tanner (2005). It is possible to observe that the presented technique (blue line) yields results that very much agree with Tanners theory for a greater We range. Even though the results for near Newtonian ranges present a higher error, the swelling ratios obtained are still in the same order and the increased range of applicability amounts to the merit of the approach here presented.

5.2.2 Drop Splashing. Worthington Jet

The third problem tackled was the simulation of a drop (both Newtonian and Oldroyd B fluids) impacting the free surface of a reservoir of the same fluid. At certain velocity this impact produces a crater around which a crown is usually formed. The crater is subsequently refilled and the filling fluid starts building up until a jet is formed. This splash was fist studied by M.A. Worthington (1877), who photographed low-viscosities Newtonian fluid splashes caused both by droplets and solid balls. Fig. 5.10 shows pictures taken at different stages of the experiment. In these it is possible to observe the crater, crown and satellite drops caused by the impact; and the jet formed shortly afterward. In this case the resulting jet was discontinuous, forming a droplet.

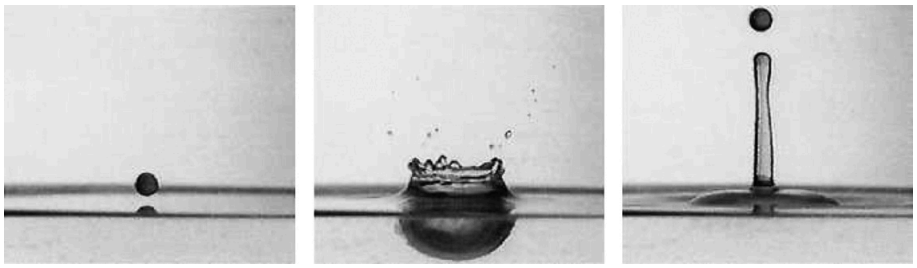


Figure 5.10. Drop falling, example of a Worthington Jet formation. (Taken from Cheny and Walters (1999))

When studying the drop of a ball, Worthington was able to identify two kind of splashes, depending on whether the ball had a smooth surface or if it was rough. The drop of a small polished dry ball would slip in the fluid without almost any disturbance, this was denominated a "smooth splash". On the other hand, "rough splashes" would appear when the ball was ground with a coarse sand paper or when it was left wet. In this case the sphere would produce the aforementioned crater and and jet. He also determined other factors which would govern the kind of splash formed. For instance,

he showed that as the impact velocity increased a gradual transition from the smooth to rough splash occurred.

In the case of falling drops, Worthington classified the resulting splash as rough. However not all drops would produce a splash. Some drops simply produce vortex rings in the reservoir but nothing else. In this case the size of the droplet along with the impact velocity are the factor which determine the kind of behavior that would take place.

In their work on the subject, Cheny and Walters (1999) addressed the question on whether the surface tension influenced the splash by carrying on a series of experiments on Newtonian fluids with different surface tension but being equal all other parameters. They were able to conclude that the maximum jet height did not depend on the surface tension (Cheny and Walters, 1999). Another finding of this job was that a small increase in polymer concentration (thus augmenting the elasticity of the solution) reduced drastically the maximum jet height. It is precisely this behavior which constitutes the objective of this section. They also determined the necessary conditions to ensure that the splash experiment could be free of the influence of the reservoir walls (Cheny and Walters, 1996).

The experiment setup included a drop of radius $r_d = 0.5$ cm falling on a circular tank of radius $r_t = 10$ cm and height $h_t = 10$ cm. As can be seen on Cheny and Walters (1996), this tank dimension ensure that the non Newtonian drop will behave as if it were falling on an infinite reservoir. For the Newtonian case the reservoir might create some interference according to the data published, yet those experiments were performed with solid spheres of a diameter 50 percent larger than our drop. This give us confidence that this factor will not affect us. Still the height of the jet should be very close to the maximum reachable even in the case of some wall interference.

The initial configuration can be seen in Fig. 5.11(a). The simulation starts one step prior to the impact, at a time when the drop is traveling with a speed of $200cm/s$. The nodal density was increased in near the impact zone and inside de drop in order to have enough particles to properly describe the crown and jet.

We performed a series of simulations to be able to appreciate the viscoelastic effects due to an increase in the Weissenberg number. This was an attempt to simulate the different behavior observed in Cheny and Walters (1999) for fluids with the same viscosity i.e., flows at the same Reynolds number, but containing polymers of different stiffness. Following the steps of Tome et al. (2007), a kinematic viscosity $\nu_0 = 0.2cm^2/s$ was considered which, after defining the Reynolds number as $Re = Ur_t/\nu_0$ would yield $Re = 500$. The cited work repeated the simulations at $Re = 1000$ and 2000 but showed

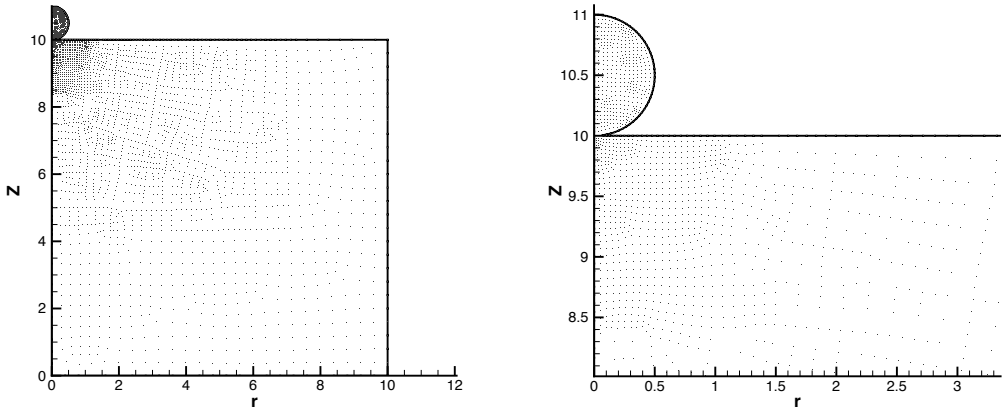


Figure 5.11. Initial configuration for the falling drop (left) and detail in the vicinity of the drop (right).

that the only real difference at this ranges was that the jet heights were higher as Re increased.

Figure 5.12 shows the apex reached by each of the test fluids. In this case the non-Newtoninan fluids had a We_{ef} of 0.1 and 0.5 respectively. As expected, the lower obtained jet heights corresponded to more non-Newtonian behaviors. Even though the obtained jets are lower than those observed in the laboratory, the results still agree qualitatively with the actual behavior. Figures 5.13 and 5.18 show the evolution of the splash as well as allow us to contrast the process both for a Newtonian fluid and an Olroyd B fluid.

5.3 Conclusions

The technique introduced in this chapter represents an alternative to simulate non Newtonian flows in situations where a traditional Eulerian approach would call for the meshing of empty spaces while trying to predict the flow pattern. Furthermore, since the method follows fluid particles, it presents itself as an excellent option for keeping the history of extra-stresses without the need to resource to interpolation techniques or the need to calculate in points other than the nodes. In fact this might be the most important facet of this experience. It has been shown that the NEM is totally able to successfully work with nodal properties in situations where the cloud configuration change is large.

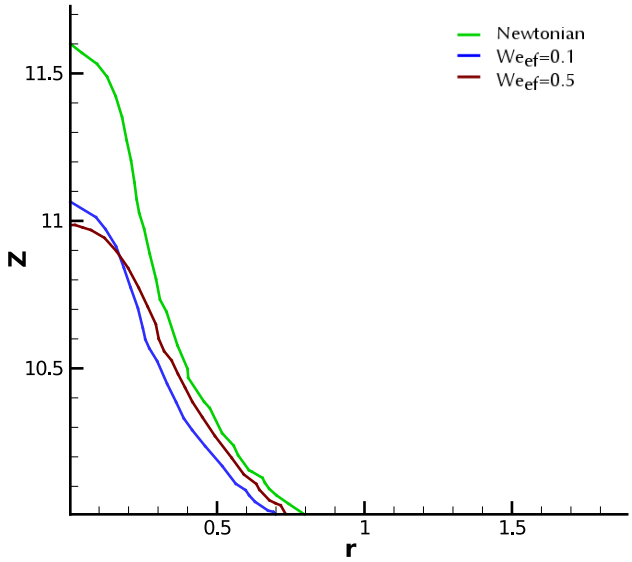


Figure 5.12. Maximum height reached by the Newtonian fluid and Oldroyd fluids of $We_{ef} = 0.1$ and $We_{ef} = 0.5$

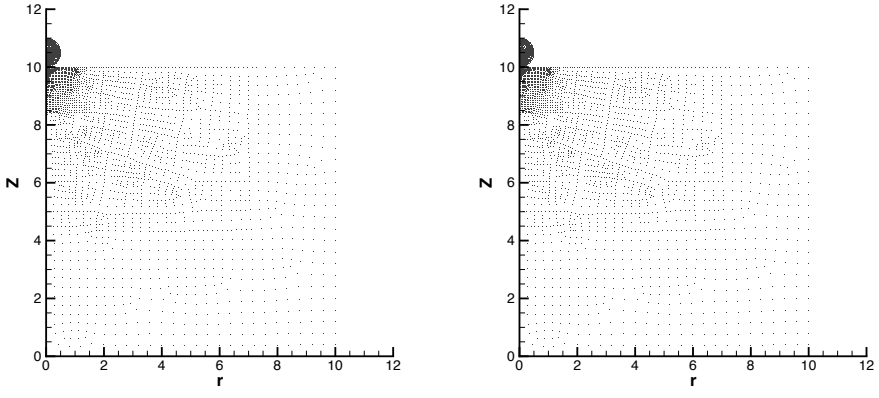


Figure 5.13. Comparision between Newtonian fluid (left) and Oldroyd-B fluid (right) with $We = 0.5$. $t = 0$.

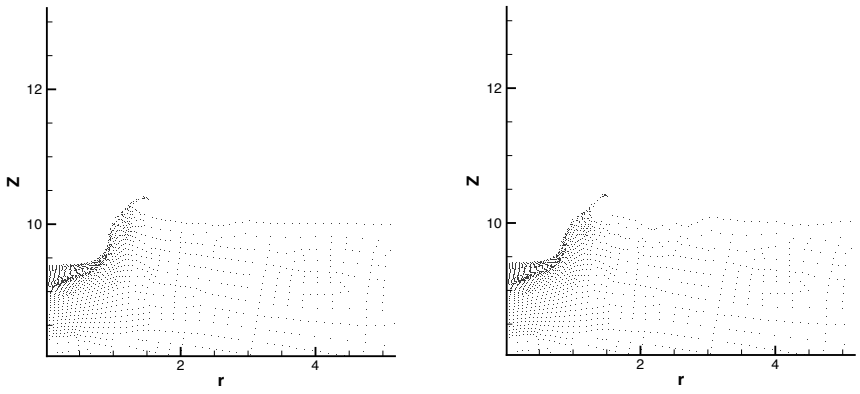


Figure 5.14. t=10 ms. The crown is visible at this point.

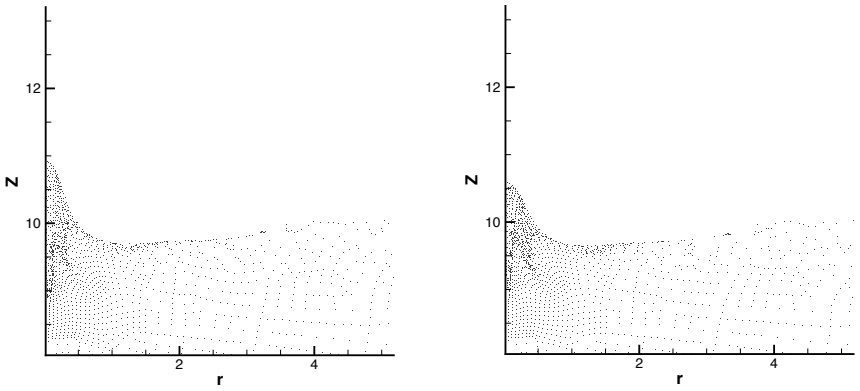


Figure 5.15. t=75 ms. The jet is already formed.

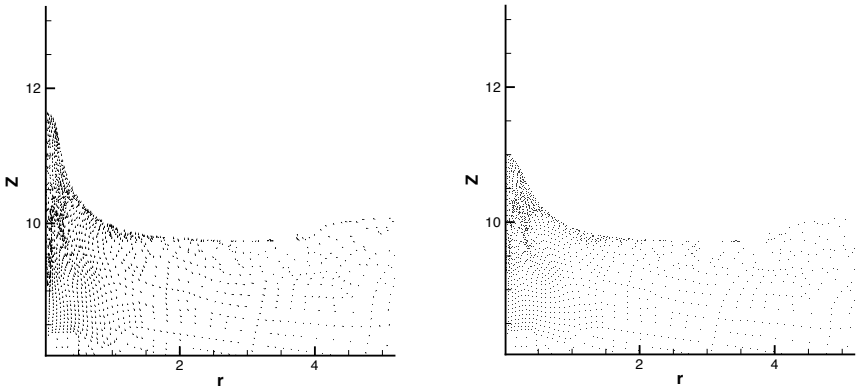


Figure 5.16. t=105 ms. Oldroyd fluid reaches its maximum height.

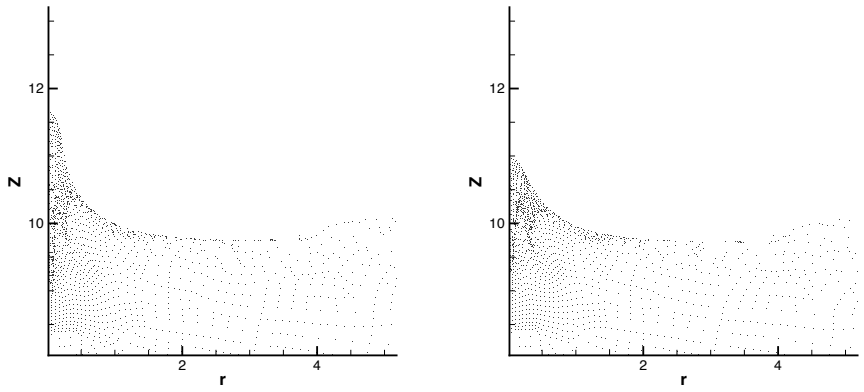


Figure 5.17. t=110 ms. Newtonian fluid reaches its maximum height.

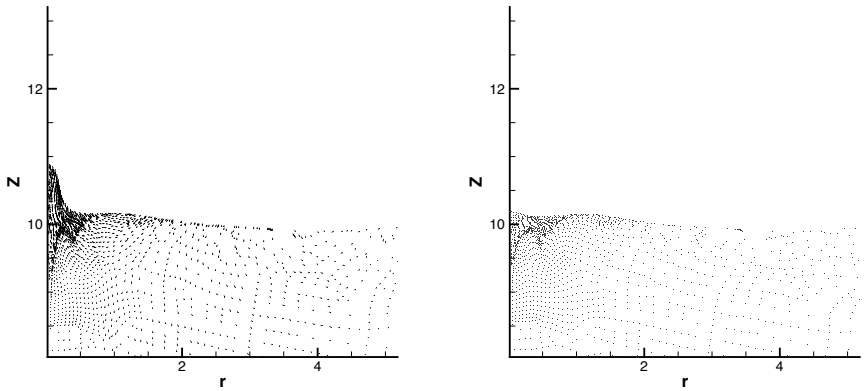


Figure 5.18. t=150 ms. Both jets are already decreasing.

The proposed method is able to reproduce with great accuracy analytical solutions where available. In addition, it works well for moderate We numbers, better than existing state-of-the-art techniques. However, it is still necessary to study the source of the error observed for the Worthington jet problem. A possible solution could be implement a more robust constitutive model, since the Olroyd-B is one of the simplest for non-Newtonian behavior. At least in the splashing problem, where the extensional characteristics of the flow play an important role, the model might have been an inadequate choice. Another area that presented special difficulty and a probable source or error was the shape recognition algorithm. While the method introduced in the last chapter certainly improved the situation, it is still an open field of study.

Chapter 6

Ending Remarks

Developing a technique that could accurately simulate the intricate phenomena present on free surface flows has been the main objective pursued by this work.

The work started with the examination of the theory regarding the Natural Element Method, which was the main tool employed on this thesis. The information gathered on this phase was summarized in Chapter 2. The natural element method was chosen due to a combination of several reasons. Firstly, this thesis grew up in the context of a national project devoted to the meshless simulation of free surface fluid-structure interaction phenomena. To this end, it was assumed that the ability of the natural element method to exactly impose essential boundary conditions, very much like finite elements, was of utmost importance for ulterior coupling with finite element meshes of the solid under consideration. Secondly, the vast experience accumulated in the Group of Structural Mechanics and Material Modeling of the University of Zaragoza with respect to this particular method made it ideal for its application to this particular problem. This thesis somewhat closes a period of more than ten years devoted in the GEMM to the natural element method.

The following step was to give solution to problems found in the original form of the updated-Lagrangian approach. To this end a second order in time numerical scheme was introduced. The modified technique improved the quality of the obtained solutions over the previous technique, developed in the doctoral thesis of David González, that showed deficiencies for some particular examples, the sloshing problem being the most noteworthy example, and an excellent accuracy for others, apparently similar. With the higher order approximation it was possible to accurately simulate problems that were not feasible previously . Although incorporating more historic information about the previous flow steps reduced error and yielded better approximations, the improved accuracy came at the cost of increased resource requirements (CPU time,

memory and storage). It is recommended that higher order approximations should be reserved for times when a first order scheme proves to be inadequate or insufficient.

The updated Lagrangian scheme resulted very convenient for transient fluid simulations by not having to deal with convective terms from the Navier-Stokes equations. These terms are well-known due to their complexity and the need for a suitable stabilization. In addition, being able to follow fluid particles allowed the simulation of free surfaces without the need to track the liquid interphase. Yet the lack of explicit nodal connectivity proper of meshless methods implies that the shape of the domain must be extracted from the nodal set. With this in mind and knowing the difficulties in the current used method, an improved α-shape technique specific for domain tracking in updated Lagrangian simulations of free surface flows was developed. A new filtration was incorporated into the standard α-shape technique. This new filtration takes into account the velocity field of the flow as a region discriminator to help determine the domain's future geometry. A way to use non-geometric information inherent to our model was introduced, working as a tool to further filter α-shapes and to obtain a better surface definition. Traditional problems associated to this method, such as spurious contacts , were avoided thanks to the proposed improvement. Even in cases where there is large nodal density differences, the flow provided enough information to recognize, even for rough tuning of α-values, different regions in the model that pertain either to zones getting in contact or to different bodies in the simulation.

Despite the inclusion of two new user-defined parameters, the proposed methodology resulted flexible enough to face the geometry changes that occur with moderate Reynolds number flows, as covered by the presented formulation. The main conclusion was that it is considerably easier to find the three parameters α, a and b, than to find the single α value for some special, delicate cases—notably some time steps prior to contact—in standard α-shape technique. We have shown how, even for a poor selection of α, the proposed technique is able to correctly filtrate the actual geometry of the domain. Values of the parameters are much less sensible to modifications than α for standard α-shapes, and thus the ease of use and good results of the proposed technique.

Being able to simulate in a satisfactory manner some Newtonian flows, the next step was to extend the model's applicability range to the very interesting viscoelastic fluids. These present some curious free surface features and a set of challenges that enticed the author. The technique developed in this work represents an alternative to model non-Newtonian flows in situations where a traditional Eulerian approach would call for the meshing of empty spaces while trying to predict a flow pattern. Furthermore, since

the required extra-stress information used in the Oldryd-B model is a quantity associated to material particles moving along characteristic lines, the presented method performs a fairly goo job at keeping the historic stress without the need to resource to interpolation techniques or the need to calculate in points other than nodes. It has been shown that the NEM is able to successfully work with nodal properties in situations where the particle configuration change is large.

Although we have been able to reproduce qualitative results, it is still necessary to study the source of the error observed. A possible solution could be to implement a more robust model, since the Olroyd-B is one of the simplest for non-Newtonian behavior. At least in the splashing problem, where the extensional characteristics of the flow play an important role, the model might have been an inadequate choice. On the other hand, since part of error can be attributed to the commonly observed (yet not well understood) high We problem, we are satisfied as the proposed technique has been able to provide consistent results for a We range wider than previous (influential) published works.

6.1 Original developments in this thesis

While there is still a huge amount of work to be performed on the simulation of free surface flows, at the end of this work it has been ossible to contribute to the actual state of the art at least on the following aspects points:

- An improvement on the Updated-Lagrangian approach developed by González (2004) has been made by formulating a second order in time approximation, which allowed to solve problems whose solution would not converge under the original schema.

- An Updated-Lagrangian model was developed an implemented for non-Newtonian fluids following the Oldroyd-B formulation. This model yielded excellent approximations on the benchmark tests and its performance was sound on the other applications tested. It is plausible to say that through the application of this model, some advance has been made on the high We problem compared with earlier works on the field.

- A new geometric technique has been introduced to deal with the shape recognition problem from a cloud of nodes. This technique was designed to be used on

data sets that include physical information (particularly, velocities) used to characterize different regions, and specifically to be applied on the computational simulation of large deformation problems.

These developments have been published in a number of journal papers and conference proceedings. Some of them are include in the Appendix section for completeness:

- A. Galavis, D. Gonzalez, I. Alfaro, E. Cueto. *Improved boundary tracking in meshless simulations of free-surface flows.* Computational Mechanics, 42, 467-479, 2008.
- A. Galavis, D. Gonzalez, E. Cueto, F. Chinesta, M. Doblare. *A Natural Element updated Lagrangian approach for modelling Fluid-Structure interactions*. European Journal of Computational Mechanics (2006) 16:323–336.
- A. Galavis, D. Gonzalez, E. Cueto, F. Chinesta, M. Doblare. *A Natural Neighbour characteristics-Galerkin method for Fluid-Structure interaction problems*. Journées AUM / Association Francaise de Mecanique. Groupe de Recherche Interaction Fluide-Structure CNRS. La Rochelle, France, 2006.
- A. Galavis, D. Gonzalez, E. Cueto, F. Chinesta. *Una aproximacion Lagrangiana para problemas de Interaccion Fluido-Estructura basada en un metodo de Elementos Naturales y Caracteristicas*. Congreso Metodos Numericos en la Ingenieria (SEMNI-APMTAC). Oporto, Portugal, 2007.
- A. Galavis, D. Gonzalez, E. Cueto, F. Chinesta. *An Updated Lagrangian Approach for Fluid-Structure Problems based on Natural Elements and the Method of Characteristics*. World Congress on Computational Mechanics, WCCM 08. Venice, Italy, 2008.
- A. Galavis, D. Gonzalez, E. Cueto. *A natural element approach for non-newtonian, free-surface flows at high We numbers*. International Journal for Numerical Methods in Fluids, submitted, 2010.

6.2 Future work

Future work to improve the already accomplished should be focused on improving and developing better shape construction techniques. That has been an area that presented special difficulty and a probable source or error. While the method introduced in the last chapter certainly improved the situation, it is still an open field of study.

Also some attention should be paid on the computational efficiency improvements that can be incorporated in to the NEM. It is clear that this method is more resource consuming than the FEM or the Finite Volume Method and even though the nodal connectivity is obtained in a process transparent to the end user, can be also demanding. In addition, a large number of nodes must be used in order to accurately capture the most interesting flow features, although this is a common issue of all meshless methods, and the price to pay to avoid to deal with a mesh. Thus if any future is going to have this scheme for this application, the efficiency issue must be dealt with.

As a natural next step we also foresee the application of the proposed method to fluid-structure simulation in scenarios other than the simple ones considered in this work. The updated-Lagrangian NEM's ability to track free surfaces, along with the possibility to couple the nodal cloud to FEM structure models provides the setting to solve a wide array of interesting problems. Wave breaking on mooring structures or submersed structures deformation are only two of a myriad of possible industrial applications. Some groundwork has been laid on this field by the author although it 's still vary early for any useful results.

The application of the here developed technique to more sophisticated non-Newtonian fluid is also the topic of current research in the GEMM. Particularly, models based upon kinetic theory of fluids are under consideration and its application to Finitely Extensible Non-Linear Elastic dumbbells (FENE models) has rendered very promising results in problems such as ink-jet printing.

Bibliography

Bibliography

Alfaro, I., Bel, D., Cueto, E., Doblaré, M., and Chinesta, F. (2006a). Three-dimensional simulation of aluminium extrusion by the alpha-shape based natural element method. *Computer Methods in Applied Mechanics and Engineering*, 195(33-36):4269--4286.

Alfaro, I., González, D., Bel, D., Cueto, E., Doblaré, M., and Chinesta, F. (2006b). Recent advances in the meshless simulation of aluminium extrusion and other related forming processes. *Archives of Computational Methods in Engineering*, 13(1):3--44.

Alfaro, I., Yvonnet, J., Chinesta, F., and Cueto, E. (2007). A study on the performance of Natural Neighbour-based Galerkin Methods. *International Journal for Numerical Methods in Engineering*, 71:1436?1465.

Amenta, N., Bern, M., and Eppstein, D. (1998a). The crust and the beta-skeleton: combinatorial curve reconstruction. *Graphical Models and Image Processing*, 60/2:2:125--135.

Amenta, N., Bern, M., and Kamvysselis, M. (1998b). A new voronoi-based surface reconstruction algorithm. In *Siggraph 98*, pages 415--421.

Armstrong, C. G., Robinson, D., McKeag, R., Li, T., Bridgett, S., Donaghy, R., and McGleenan, C. (1995). Medials for meshing and more. In *4th International Meshing Roundtable. Sandia National Laboratories*, pages 277--288.

Arroyo, M. and Ortiz, M. (2006). Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. *International Journal for Numerical Methods in Engineering*, 65(13):2167--2202.

Atluri, S., Kim, H. G., and Cho, J. Y. (1999). A critical assesment of the truly Meshless Local Petrov-Galerkin and Local Boundary Integral Equation methods. *Computational Mechanics*, 24:348--372.

Atluri, S. and Zhu, T. (2000). New concepts in meshless methods. *International Journal for Numerical Methods in Engineering*, 47:537--556.

Babuška, I. (1973). The Finite Element Method with Lagrange multipliers. *Numerische Mathematik*, 20:179--192.

Babuška, I. and Aziz, A. (1976). On the angle condition in the finite element method. *SIAM J. Numer. Anal.*, 13:214--227.

Babuška, I. and Melenk, J. M. (1996). The partition of unity finite element method: Basic theory and applications. *Comp. Meth. in Appl. Mech. and Eng.*, 4:289--314.

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. T. (1996). The quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software*, 22(4):469--483.

Belikov, V. V., Ivankov, V. D., Kontorovich, V. K., Korytnik, S. A., and Semenov, A. Y. (1997). The non-Sibsonian interpolation: A new method of interpolation of the values of a function on arbitrary set of points. *Computational Mathematics and Mathematical Physics*, 37(1):9--15.

Belytschko, T., Lu, Y. Y., and Gu, L. (1994). Element-Free Galerkin Methods. *International Journal for Numerical Methods in Engineering*, 37:229--256.

Birknes, J. and Pedersen, G. (2006). A particle finite element method applied to long wave run-up. *International Journal For Numerical Methods In Fluids*, 52:237--261.

Boukir, K., Maday, Y., Métivet, B., and Razafindrakoto, E. (1997). A high-order characteristic/finite element method for the incompressible navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 25:1421--1454.

Braun, J. and Sambridge, M. (1995). A numerical method for solving partial differential equations on highly irregular evolving grids. *Nature*, 376:655--660.

Cazals, F., Giesen, J., Pauly, M., and Zomorodian, A. (2006). The conformal alpha-shape filtration. *The Visual Computer*, 22(8):531--540.

Chen, J.-S., Wu, C.-T., Yoon, S., and You, Y. (2001). A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50:435--466.

Cheny, J. M. and Walters, K. (1996). Extravagant viscoelastic effects in the worthington jet experiment. *Journal of Non-Newtonian Fluid Mechanics*, 67:125 -- 135. Papers Presented at the Workshop on 'Unresolved Experimental Dilemmas in the Dynamics of Complex Fluids'.

Cheny, J. M. and Walters, K. (1999). Rheological influences on the splashing experiment. *Journal of Non-Newtonian Fluid Mechanics*, 86(1-2):185 -- 210.

Crank, J. (1987). *Free and Moving Boundary Problems*. Oxford University Press.

Crochet, M., Davies, A., and Walters, K. (1984). *Numerical Simulation of Non-Newtonian Flow*. Elsevier.

Crochet, M. and Keunings, R. (1982). Finite element analysis of die swell of a highly elastic fluid. *Journal of Non-Newtonian Fluid Mechanics*, 10(3-4):339 -- 356.

Crochet, M. and Walters, K. (1983). Numerical methods ind non-newtonian fluid mechanics. *Anual Review of Fluid Mechanics*, 15:241--260.

Cueto, E. (2001). *El Método de los Elementos Naturales basado en formas α: Aplicación a la simulación de la remodelación interna de fracturas de cadera con sistema Éxeter*. PhD thesis, Universidad de Zaragoza, Zaragoza, España.

Cueto, E., Cegoñino, J., Calvo, B., and Doblaré, M. (2003). On the imposition of essential boundary conditions in Natural Neighbour Galerkin methods. *Communications in Numerical Methods in Engineering*, 19(5):361--376.

Cueto, E., Doblaré, M., and Gracia, L. (2000). Imposing essential boundary conditions in the Natural Element Method by means of density-scaled α-shapes. *International Journal for Numerical Methods in Engineering*, 49-4:519--546.

Cyron, C. J., Arroyo, M., and Ortiz, M. (2009). Smooth, second order, non negative meshfree approximants selected by maximum entropy. *International Journal for Numerical Methods in Engineering*, accepted for publication.

de Boor, C. (1962). Bicubic Spline Interpolation. *J. Math and Physics*, 41:212--218.

Delaunay, B. (1934). Sur la SphËre Vide. A la memoire de Georges Voronoi. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7:793--800.

Dey, T. K., Giesen, J., and John, M. (2003). Alpha-shapes and flow shapes are homotopy equivalent. In *Proc. Sympos. Theory Computing (STOC 2003)*, pages 493--502.

Donea, J. and Huerta, A. (2003). *Finite Element Methods for Flow Problems*. John Wiley and sons.

Duarte, C. and Oden, J. T. (1996a). H-p Clouds, An h-p Meshless Method. *Numerical Methods for Partial Differential Equations*, 12:673--705.

Duarte, C. A. M. and Oden, J. T. (1996b). An h-p Adaptive Method using Clouds. *Computer Methods in Applied Mechanics and Engineering*, 139:237--262.

Duchemin, L., Popinet, S., Josserand, C., and Zaleski, S. (2002). Jet formation in bubbles bursting at a free surface. *Physics of Fluids*, 14:3000--3008.

Edelsbrunner, H., Kirkpatrick, D. G., and Seidel, R. (1983). On the Shape of a Set of Points in the Plane. *IEEE Transactions on Information Theory*, IT-29(4):551--559.

Edelsbrunner, H. and Mücke, E. (1994). Three dimensional alpha shapes. *ACM Transactions on Graphics*, 13:43--72.

Farin, G. (1990). Surfaces over Dirichlet Tesselations. *Computer Aided Geometric Design*, 7(1-4):281--292.

Farin, G. (2002). *Curves and surfaces for CAGD*. Morgan Kaufmann.

Gibbs, J. W. (1898). Fourier Series. *Nature*, 59.

Giesen, J. and John, M. (2003). The Flow Complex: A data structure for geometric modelling. In *Proceedings of the 14th Annual ACM-SIAM symposium on DiscreteAlgorithms*, pages 285--294.

González, D. (2004). *Aproximaciones Mixtas en Métodos de Lagrange-Galerkin de Vecindad Natural. Aplicación a la Mecánica de Sólidos y de Fluidos*. PhD thesis, Universidad de Zaragoza.

González, D., Cueto, E., Chinesta, F., and Doblaré, M. (2007). A natural element updated Lagrangian strategy for free-surface fluid dynamics. *Journal of Computational Physics*, 223(1):127--150.

González, D., Cueto, E., and Doblaré, M. (2004). Volumetric locking in Natural Neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering*, 61(4).

González, D., Cueto, E., and Doblaré, M. (2008). Higher-order natural element methods: Towards an isogeometric meshless method. *Int. J. Numer. Meth. Engng.*, 74(13):1928--1954.

Hughes, T. J. R., Cottrell, J. A., and Bazilevs, Y. (2005). Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135--4195.

Idelsohn, S. R. and Oñate, E. (2006). To mesh or not to mesh: that is the question. *Computer Methods in Applied Mechanics and Engineering*, 195(37--40):4681--4696.

Idelsohn, S. R., Oñate, E., Calvo, N., and del Pin, F. (2003). The meshless finite element method. *International Journal for Numerical Methods in Engineering*, 58:893--912.

Idelsohn, S. R., Oñate, E., and del Pin, F. (2004). The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International Journal for Numerical Methods in Engineering*, 61 (7):964--989.

Krongauz, Y. (1996). *Application of Meshless Methods to Solid Mechanics*. PhD thesis, Northwestern University, Evanston, IL.

Lasserre, J. B. (1983). An analytical expression and an algorithm for the volume of a convex polyhedron in $\Re^n$. *Journal of Optimization Theory and Applications*, 39(3):363--377.

Lesaint, P. and Raviart, P. (1974). On a finite element method for solving the neutron transport equation. *Mathematical Aspectst of Finite Elements in Partial Diferencial Equations*, pages 89--123.

Lewis, R. W., Navti, S. E., and Taylor, C. (1997). A mixed lagrangian-eulerian approach to modelling fluid flow during mould filling. *International Journal for Numerical Methods in Fluids*, 25:931--952.

Liu, W. K. and Chen, Y. (1995). Wavelet and Multiple Scale Reproducing Kernel Methods. *International Journal for Numerical Methods in Fluids*, 21:901--931.

Liu, W. K., Jun, S., Li, S., Adee, J., and Belytschko, T. (1995). Reproducing Kernel Particle Methods. *International Journal for Numerical Methods in Engineering*, 38:1655--1679.

Lucy, L. B. (1977). A Numerical Approach to the Testing of the Fission Hypothesis. *The Astronomical Journal*, 82(12):1013--1024.

Mandal, D. P. and Murthy, C. A. (1997). Selection of alpha for alpha-hull in $\Re^2$. *Pattern Recognition*, 10:1759--1767.

Marchal, J. and Crochet, M. (1987). A new mixed finite element method for calculating viscoelastic flow. *Journal of Non-Newtonian Fluid Mechanics*, 20:187--207.

Martínez, M. A., Cueto, E., Alfaro, I., Doblaré, M., and Chinesta, F. (2004). Updated Lagrangian free surface flow simulations with Natural Neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering*, 60(13):2105--2129.

Monaghan, J. J. (1982). Why particle methods work. *SIAM J. Sci. Stat. Comput.*, 3(4):422--433.

Nayroles, B., Touzot, G., and Villon, P. (1992). Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307--318.

Owens, R. and Phillips, T. (2002). *Computational Rheology*. Imperial College PRess.

Ramaswamy, B. (1990). Numerical Simulation of Unsteady Viscous Free Surface Flow. *Journal of Computational Physics*, 90:396--430.

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.

Sibson, R. (1980). A Vector Identity for the Dirichlet Tessellation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 87:151--155.

Sibson, R. (1981). A brief description of natural neighbour interpolation. In *Interpreting Multivariate Data. V. Barnett (Editor)*, pages 21--36. John Wiley.

Sukumar, N. (1998). *The Natural Element Method in Solid Mechanics*. PhD thesis, Northwestern University, Evanston, Illinois.

Sukumar, N. and Moran, B. (1999). C^1 Natural Neighbour Interpolant for Partial Differential Equations. *Numerical Methods for Partial Differential Equations*, 15(4):417--447.

Sukumar, N., Moran, B., and Belytschko, T. (1998). The Natural Element Method in Solid Mechanics. *International Journal for Numerical Methods in Engineering*, 43(5):839--887.

Sukumar, N., Moran, B., Semenov, A. Y., and Belikov, V. (2001). Natural Neighbor Galerkin Methods. *International Journal for Numerical Methods in Engineering*, 50(1):1--27.

Tanner, R. I. (2005). A theory of die-swell revisited. *Journal of Non-Newtonian Fluid Mechanics*, 129(2):85 -- 87.

Teichmann, M. and Capps, M. (1998). Surface reconstruction with anisotropic density-scaled alpha shapes. In *Proceedings of the 1998 IEEE Visualization Conference*.

Tome, M., Grossi, L., Castello, A., Cuminato, J., McKee, S., and Walters, K. (2007). Die-swell, splashing drop and a numerical technique for solving the oldroyd b model for axisymmetric free surface flows. *Journal of Non-Newtonian Fluid Mechanics*, (141):148--166.

Traversoni, L. (1994). Natural Neighbour Finite Elements. In *Intl. Conference on Hydraulic Engineering Software. Hydrosoft Proceedings*, pages 291--297. Computational Mechanics publications.

Voronoi, G. M. (1908). Nouvelles Applications des Parametres Continus a la Theorie des Formes Quadratiques. Deuxieme Memoire: Recherches sur les parallelloedres Primitifs. *J. Reine Angew. Math.*, 134:198--287.

Watson, D. (1981). Computing the n-dimensional Delaunay Tessellation with Application to Voronoi Polytopes. *The Computer Journal*, 24(2):162--172.

Worthington, A. M. (1877). On the forms assumed by drops of liquid falling on a horizontal plate. *Proc. R. Soc. London*, 25:261.

Zienkiewicz, O. C. and Godbolet, P. N. (1974). Flow of plastic and visco-plastic solids with special reference to extrusion and forming processes. *International Journal for Numerical Methods in Engineering*, 8:3--16.

Zienkiewicz, O. C., Pain, P. C., and Oñate, E. (1978). Flow of solids during forming and extrusion: some aspects of numerical solutions. *International Journal of Solids and Structures*, 14:15--38.

Appendix

Appendix A

Publications in peer-reviewed journals

A.1 Computational Mechanics (2008) 42:467–479

Comput Mech (2008) 42:467–479
DOI 10.1007/s00466-008-0263-5

ORIGINAL PAPER

Improved boundary tracking in meshless simulations of free-surface flows

A. Galavís · D. González · I. Alfaro · E. Cueto

Received: 18 October 2007 / Accepted: 11 February 2008 / Published online: 7 May 2008

Abstract In this paper we review the use of *shape constructors*, particularly α-shapes, for the simulation of free-surface flow problems. This technique, in conjunction with meshless methods, allows for the simulation of such problems in an updated Lagrangian approach without the need for an explicit description of the boundary of the domain. At each time step, the shape of the domain is extracted automatically. However, it is well know that α-shape techniques present some drawbacks. The first is the choice of the α parameter, related to the level of detail to which the domain is represented. Also contact detection of free surfaces (auto-contact) or between the free surface and a rigid boundary, for instance, is often detected with an error of the order $\mathcal{O}(h)$, the nodal spacing parameter, in the gap distance. We propose an heuristic technique for the choice of the α parameter and develop a novel methodology for an improved detection of contact or merging flows. The proposed technique is illustrated with the help of some examples in solid and fluid mechanics.

Keywords Free surface · Meshless · Updated Lagrangian · Boundary tracking · Shape constructors

1 Introduction

Meshless methods [7,25] opened a very active decade of research in the middle nineties. Today, more than ten years after, the area is still active, and has provided some very useful techniques for the Computational Mechanics community.

One of the most cited capabilities of meshless methods is that of simulating large deformations phenomena without lack of accuracy, as opposed to Finite Element Methods, if no remeshing is performed. This opens the possibility of simulating free surface flows, for instance, in an updated Lagrangian framework, and many works have been devoted to this end in the last years. The interested reader can consult, for instance, [17,19,20,24], among others. These free surface problems are different in nature. The reader may imagine readily waves breaking, but not only dynamical problems can be solved with such a treatment. Many forming processes, for instance, can be also treated in an updated Lagrangian setting advantageously, see [1,2]. Forging or casting and, obviously, mould filling, are among these processes that present free or internal surfaces (like phase boundaries).

The obvious advantages of updated Lagrangian meshless methods for this class of problems, if compared to Eulerian or Arbitrary Lagrangian Eulerian (ALE) methods—in which an artificial velocity is added to the mesh—for instance, are the absence of remeshing nor the associated numerical diffusion, or the lack of convective terms in the formulations, that consequently do not need of any stabilization. Note that connectivity between nodes is computed by the different meshless methods in a process transparent to the user, as the cloud of nodes evolves, convected by the material velocity.

A particularly elegant analysis of the difficulties associated to an Eulerian/Lagrangian treatment of the equations arising from free surface flows can be found in [21]. Particularly noteworthy is the difficulty in the selection of mesh velocity in ALE formulations, in which the mesh moves with a velocity different to the material one, in order to minimize mesh distortion. Also, in Eulerian (fixed mesh) approaches, some marking technique should be used in order to track

Work supported by the Spanish Ministry of Education and Science through grant number CICYT-DPI2005-08727-C02-01.

A. Galavís · D. González · I. Alfaro · E. Cueto (✉)
Group of Structural Mechanics and Materials Modelling,
Aragón Institute of Engineering Research, University of Zaragoza,
Betancourt Building, María de Luna, 5, 50018 Zaragoza, Spain
e-mail: ecueto@unizar.es

the evolution of the free surface. The Volume of Fluid (VoF) technique is an example of these techniques. In [21] a mixed Eulerian/updated Lagrangian technique is developed.

As stated before, meshless, or particle (those in which a mass is linked to each node) methods have avoided the need to perform such complicated treatments. Nevertheless, new difficulties arise. For instance, the nodal connectivity in meshless methods is not dictated by geometrical reasons (the best available triangle in terms of internal angles, for instance, in FE mesh generation) but by algebraic reasons. In the Element Free Galerkin method [7], for instance, the connectivity is dictated primarily by the need of a support (radius of the shape function) big enough to encompass a sufficiently large number of nodes so as to make a matrix invertible. Remarkably, this is not related to the geometry of domain. That reason precludes the nodal connectivity to be used directly to determine the shape of the domain, as in FE methods. Nothing similar to an *isoparametric* representation exists in meshless methods.

In addition, tracking the free surface with boundary markers can be implemented in an elegant way in two dimensional problems—by employing a chain of markers and checking self-intersections of the chain to detect merging flows—as in [21], for instance, but becomes much more intricate in tree dimensions.

If one then tries to avoid any form of meshing, and only a set of nodes, with no connectivity between them, is employed, it then becomes difficult to find the position of the free surface. In other words, the geometry of the domain should be extracted in any way from the current, updated, position of the nodes, that move, as stated before, with the material velocity.

To this end, various authors have employed Computational Geometry techniques. In particular, [11] seems to have been the first in employing *shape constructors*—α-shapes in this case—techniques to extract the geometry of the domain. Shape constructors are geometrical techniques that enable to find the *shape* of the domain at each time step. α-Shapes [13] have been employed in a number of previous works involving free surface flows, see for instance [9,17,18,20,24], among others.

Also, different shape constructors have been proposed after α-shapes, see [3,4,10,15] to name a few. In order to extract the geometry of the domain, in general, these methods propose a *filtration* of the Delaunay triangulation of the cloud of points. The Delaunay triangulation is the base ingredient of these techniques, since it characterizes univocally the cloud of points—it is unique for each cloud. Different criteria are proposed in order to select the triangles pertaining to the *shape* of the domain. The simplest one is maybe the α-shape technique, that proposes to eliminate all triangles (or tetrahedra) whose circumscribing radius (or, equivalently in finite element terminology, their associated mesh size, h) is greater than a prescribed *level of detail* for the geometry, α. α-Shapes have generated a great interest on "provable" shape reconstruction arguments. We mean that, under certain, usually very weak, assumptions on the size of the cloud of points, we obtain geometric- and topologically accurate descriptions of the domain under consideration.

One of the main drawbacks of the α-shape technique, as recognized in many works (see, for instance, [10,28]) is precisely the choice of the α-value. In addition, α-shapes work well only for uniformly distributed cloud of points. This does not constitute a problem, generally, for the problems being considered. Since we deal with initial value problems, the choice of a uniform nodal sampling on the initial geometry, in the absence of any information on the final geometry of the domain, seems to be judicious.

The jump of the before mentioned techniques to the field of Computational Mechanics has posed additional difficulties. It is well-known that α-shapes are not able to detect holes or cavities of size smaller than α, by definition. This implies that contact between different surfaces is detected with an error $\mathcal{O}(\alpha) \approx \mathcal{O}(h)$, i.e., prior to the true expected contact [28]. Precisely in [28] a method is proposed to alleviate this drawback, but it needs information on the normal of the boundary at the sampling points. This is easy to achieve for three-dimensional scans of solids, for instance, but this kind of information is not suitable from the class of simulations we are interested in.

In this paper we propose a new technique, well suited for the numerical simulation of free-surface flows, that avoids the before mentioned problems. The proposed technique is based in performing an additional filtration to the Delaunay triangulation (tetrahedrization) of the cloud of points. After the α-filtration, we perform an additional filtration based on the information provided by nodal velocities at the last converged time step, and the gradient of velocities. The tests performed during this work have provided excellent results over problems where traditional α-shapes have revealed deficiencies.

The outline of the paper is as follows. First, we pose the formulation of the problem, taking into account the wide scope of "free-surface" phenomena, possibly involving dynamic evolution. We then review the basics of α-shape theory and show its inherent limitations. The proposed method is described in Sect. 4. The paper is completed with some two- and three-dimensional examples showing the performance of the method in Sect. 5.

2 Problem settings

There is a wide variety of problems involving the presence of free or internal surfaces. Typically, Navier–Stokes equations in the presence of such boundary conditions are maybe the most ubiquitous example. But we do not restrict ourselves to

Comput Mech (2008) 42:467–479 469

Navier–Stokes equations. Even without the presence of inertia terms, many forming processes can be formulated in the so-called *flow formulation* [29,30], if a rigid-(visco)plastic constitutive equation is assumed. Most of these forming processes (extrusion, forging,…) imply the presence of free-surfaces, and very often the precise location of them, together with accurate determination of contact, auto-contact, etc. is of utmost importance.

We refer ourselves mainly to these last two examples: Navier–Stokes equations and the flow formulation of a rigid-plastic metal. Other problems are also suitable for the formulation here proposed.

2.1 Updated Lagrangian formulation for Navier–Stokes equations

We review here a formulation for the numerical solution of Navier–Stokes equations previously presented in [17]. Other formulations also exist and work properly (see, for instance, the implicit three-step fractional method presented in [20]), but we believe that this one exploits particularly well the updated Lagrangian setting of the method.

Consider a fluid in a region Ω of the space $\mathbb{R}^2$ or $\mathbb{R}^3$. The fluid flow is governed by the following mass and momentum conservation equations:

$$\rho(\boldsymbol{v}_{,t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}) = \nabla\cdot\boldsymbol{\sigma} + \rho\boldsymbol{b} \quad \text{in } \Omega\times(0,T), \tag{1}$$

$$\nabla\cdot\boldsymbol{v} = 0 \quad \text{in } \Omega\times(0,T) \tag{2}$$

where $\boldsymbol{v}$ represents the fluid velocity, $\boldsymbol{\sigma}$ the stress tensor, ρ represents fluid density and $\boldsymbol{b}$ the volumetric forces acting on the fluid.

The constitutive equation for a Newtonian fluid is given by:

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{\tau} = -p\boldsymbol{I} + 2\mu\boldsymbol{d} + \lambda(\nabla\cdot\boldsymbol{v})\boldsymbol{I}, \tag{3}$$

where $\boldsymbol{d}$ is the strain rate tensor, p the pressure, μ is the dynamic viscosity of the fluid and λ the second coefficient of viscosity. For incompressible fluids $\nabla\cdot\boldsymbol{v} = 0$ and consequently the before-mentioned Eq. (3), is reduced to the so-called Stokes law

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\boldsymbol{d}. \tag{4}$$

Substituting into Eqs. (1)–(2) we arrive to

$$\rho\left(\boldsymbol{v}_{,t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}\right) - 2\mu\nabla\cdot\boldsymbol{d} + \nabla p = \rho\boldsymbol{b}. \tag{5}$$

It is usual to rewrite this last equation as:

$$\rho\left(\boldsymbol{v}_{,t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}\right) - \mu\nabla^2\boldsymbol{v} - \mu\nabla(\nabla\cdot\boldsymbol{v}) + \nabla p = \rho\boldsymbol{b}. \tag{6}$$

Under the incompressibility assumption (2), this last Eq. (6) is transformed into

$$\rho\left(\boldsymbol{v}_{,t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}\right) - \mu\nabla^2\boldsymbol{v} + \nabla p = \rho\boldsymbol{b}, \quad \text{in } \Omega\times(0,T). \tag{7}$$

To solve the problem we must prescribe an initial state as well as boundary conditions given by

$$\boldsymbol{v}(\boldsymbol{x},t) = \boldsymbol{v}_D(\boldsymbol{x},t), \quad \boldsymbol{x}\in\Gamma_D, \quad t\in(0,T), \tag{8}$$

$$\boldsymbol{\sigma}(\boldsymbol{x},t)\cdot\boldsymbol{n} = \boldsymbol{t}(\boldsymbol{x},t), \quad \boldsymbol{x}\in\Gamma_N, \quad t\in(0,T), \tag{9}$$

where Γ_D stands for the Dirichlet (essential) portion of the boundary and Γ_N represents the Neumann or natural portion of the boundary.

2.1.1 Time discretization

The motion equations can be grouped to

$$\nabla\cdot\boldsymbol{\sigma} + \rho\boldsymbol{b} = \rho\frac{d\boldsymbol{v}}{dt} = \rho\left(\frac{\partial\boldsymbol{v}}{\partial t} + \boldsymbol{v}\nabla\cdot\boldsymbol{v}\right), \tag{10}$$

$$\nabla\cdot\boldsymbol{v} = 0, \tag{11}$$

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\boldsymbol{d}. \tag{12}$$

The weak form of the problem associated to Eqs. (10), (11) and (12) is:

$$\int_\Omega 2\mu\boldsymbol{d}:\boldsymbol{d}^*\,d\Omega - \int_\Omega p\boldsymbol{I}:\boldsymbol{d}^*\,d\Omega = -\int_\Omega \rho\boldsymbol{b}\cdot\boldsymbol{v}^*\,d\Omega + \int_\Omega \rho\frac{d\boldsymbol{v}}{dt}\cdot\boldsymbol{v}^*\,d\Omega, \tag{13}$$

and

$$\int_\Omega \nabla\cdot\boldsymbol{v}\; p^*\,d\Omega = 0, \tag{14}$$

where ":" denotes the tensor product twice contracted. $\boldsymbol{d}^*$ represents and admissible variation of the strain rate tensor, whereas $\boldsymbol{v}^*$ represents equivalently an admissible variation of the velocity.

The second term in the right-hand side of Eq. (13) represents the inertia effects. Time discretization of this term represents the discretization of the material derivative along the nodal trajectories, which are precisely the characteristic lines related to the advection operator. Thus, assuming known the flow kinematics at time $t^{n-1} = (n-1)\Delta t$, we proceed as follows:

$$\int_\Omega \rho\frac{d\boldsymbol{v}}{dt}\boldsymbol{v}^*\,d\Omega = \int_\Omega \rho\frac{\boldsymbol{v}^n(\boldsymbol{x}) - \boldsymbol{v}^{n-1}(\boldsymbol{X})}{\Delta t}\boldsymbol{v}^*\,d\Omega, \tag{15}$$

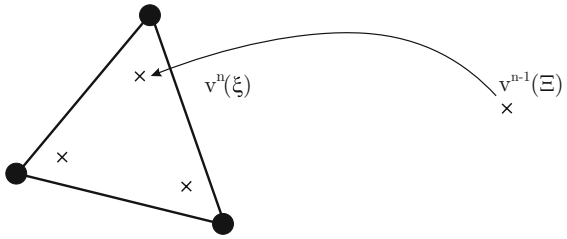


Fig. 1 Determination of the position of quadrature points at time step t^{n-1}

where $\boldsymbol{X}$ represents the position at time t^{n-1} occupied by the particle located at position $\boldsymbol{x}$ at present time t^n, i.e.:

$$\boldsymbol{x} = \boldsymbol{X} + \boldsymbol{v}^{n-1}(\boldsymbol{X})\Delta t. \tag{16}$$

So we arrive at

$$\int_\Omega 2\mu \boldsymbol{d} : \boldsymbol{d}^* \, d\Omega - \int_\Omega p\boldsymbol{I} : \boldsymbol{d}^* \, d\Omega - \int_\Omega \frac{\boldsymbol{v}\cdot\boldsymbol{v}^*}{\Delta t} \, d\Omega = -\int_\Omega \rho \boldsymbol{b}\cdot\boldsymbol{v}^* \, d\Omega - \int_\Omega \rho \frac{\boldsymbol{v}^{n-1}\cdot\boldsymbol{v}^*}{\Delta t} \, d\Omega, \tag{17}$$

and

$$\int_\Omega \nabla\cdot\boldsymbol{v} \; p^* \, d\Omega = 0, \tag{18}$$

where we have dropped the superindex in all the variables corresponding to the current time step.

2.1.2 Algorithmical issues

The most difficult term in Eq. (17) is the second term of the right-hand side. The numerical integration of this term depends on the particular quadrature scheme employed.

If we employ traditional Gauss-based quadrature on triangles, it will be necessary to find the position at time t^{n-1} of the point occupying at time t^n the position of the integration point $\boldsymbol{\xi}_k$ (see Fig. 1):

$$\int_\Omega \rho \frac{\boldsymbol{v}^{n-1}\cdot\boldsymbol{v}^*}{\Delta t} \, d\Omega = \sum_k \rho \frac{\boldsymbol{v}^{n-1}(\boldsymbol{\Xi}_k)\cdot\boldsymbol{v}^*(\boldsymbol{\xi}_k)}{\Delta t}\omega_k, \tag{19}$$

where ω_k represent the weights associated to integration points $\boldsymbol{\xi}_k$, and $\boldsymbol{\Xi}_k$ corresponds to the position occupied at time t^{n-1} by the quadrature point $\boldsymbol{\xi}_k$, see Fig. 1.

If we employ some type of nodal integration, as in [16], this procedure becomes straightforward, with the only need to store nodal velocities at time step t^{n-1}. We discuss here the procedure to follow when employing Gauss quadratures on the Delaunay triangles. We proceed iteratively. Denoting by i the current iteration, we apply

$$\boldsymbol{x}_k = \boldsymbol{X}_k^i + \boldsymbol{v}^{n-1}(\boldsymbol{X}_k^{i-1})\Delta t, \quad \text{with } \boldsymbol{x}_k = \boldsymbol{X}_k^0; \quad i \geq 1$$

until $\boldsymbol{X}_k^i \approx \boldsymbol{X}_k^{i-1}$.

Since we are using an updated Lagrangian strategy, the computation of the term $\boldsymbol{v}^{n-1}(\boldsymbol{X}_k^{i-1})$ requires a projection from the stored nodal velocities at time t^{n-1}.

The velocity and pressure variables of the problem can now be approximated using any of the meshless techniques of approximation, see, for instance, Moving Least Squares methods [8,25], Reproducing Kernel Particle Methods [22] or Natural Elements [17,26], to name a few. Of course, care must be paid to the fulfilment of the LBB condition, but the type of interpolation chosen is not relevant for the purposes of the method here developed.

In the result present in this paper, we have employed natural neighbour approximation in a Galerkin framework. See [2,17] for more details on the formulation.

2.2 Flow formulations of rigid-plastic solids

As mentioned before, many forming processes can be formulated as free-surface problems under very standard assumptions. Although, to some extent, an elastic recovery exists at the end of many metal forming process, this is often neglected. In addition, the Cauchy stress is usually related to the strain rate tensor. This leads to a formulation that closely resembles that of non-Newtonian fluids, and hence the term *flow formulation* [29].

Thus, the equations governing the metal deformation can be expressed in terms of velocities rather than displacements. Stresses produced in the forming process can be set in a simple form as

$$\boldsymbol{\sigma} = \boldsymbol{D}(\boldsymbol{d}, T)\cdot\boldsymbol{d}, \tag{20}$$

where $\boldsymbol{d}$ represents again the strain rate tensor (symmetric part of the velocity gradient) and T the temperature. Depending of the particular constitutive equation chosen for the metal, we thus obtain different formulations. In [1,2] a Sellars–Tegart temperature-dependent constitutive model was implemented in this framework.

3 Theory of α-shapes

As mentioned in Sect. 1, the idea of α-shapes in particular, and shape constructors in general, is to extract the shape of a domain described by a set of nodes only. The human eye can do this easily, but there is no formal definition of *shape* in the mathematical literature. α-shapes were first established by Edelsbrunner and Mcke [13] and Edelsbrunner et al. [14].

Comput Mech (2008) 42:467–479 471

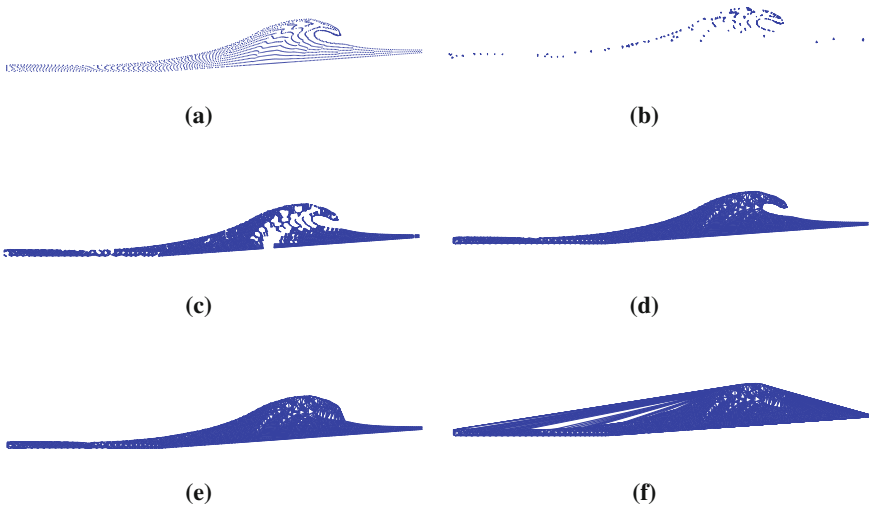


Fig. 2 Evolution of the family of α-shapes of a cloud of points representing a wave breaking on a beach. Shapes $\mathcal{S}_0$ or cloud of points (**a**), $\mathcal{S}_{0.5}$ (**b**), $\mathcal{S}_{1.0}$ (**c**), $\mathcal{S}_{2.0}$ (**d**), $\mathcal{S}_{3.0}$ (**e**) and $\mathcal{S}_\infty$ (**f**) are depicted

Other shape constructors giving homotopy-equivalent shapes have been recently proposed [12]. Given a finite set of points (that will be the nodes employed in the approximation of the problems described in the previous section), there exist a finite set of shapes described by all the possible combination of points, edges, triangles and tetrahedra (if we consider three-dimensional spaces) forming simplicial complexes.

A k-simplex σ_T with $0 \leq k \leq 3$ is defined as the convex hull of a subset $T \subseteq N$ of size $\mid T \mid = k+1$. A three-dimensional simplicial complex is a collection, $\mathcal{C}$, of closed k-simplices ($0 \leq k \leq 3$) that satisfies:

(i) If $\sigma_T \in \mathcal{C}$ then $\sigma_{T'} \in \mathcal{C}$ for every $T' \subseteq T$.
(ii) The intersection of two simplexes in $\mathcal{C}$ is empty or is a face of both.

The particular complexes considered in the theory of α-shapes have vertices in the node set and simplices from the Delaunay triangulation of the set, which is unique, as it is well known. The formal definition of the set of α-shapes of the cloud of nodes follows.

3.1 Definition of the family of α-shapes

α-Shapes define a one-parameter family of shapes $\mathcal{S}_\alpha$ (α being the parameter), ranging from the "coarsest" to the "finest" level of detail. α can be seen, precisely, as a measure of this level of detail.

Let $\boldsymbol{N}$ be our finite set of points in $\mathbb{R}^3$ and α a real number, with $0 \leq \alpha < \infty$. Let b be an α-ball, that is, an open ball of radius α. A k-simplex σ_T is said to be α-exposed if there exist an empty α-ball b with $T = \partial b \bigcap N$ where ∂ means the boundary of the ball. In other words, a k-simplex is said to be α-exposed if an α-ball that passes through its defining points contains no other point of the set N.

Thus, we can define the family of sets $F_{k,\alpha}$ as the sets of α-exposed k-simplices for the given set N. This allows us to define an α-shape of the set N as the polytope whose boundary consists on the triangles in $F_{2,\alpha}$, the edges in $F_{1,\alpha}$ and the vertices or nodes in $F_{0,\alpha}$.

Each k-simplex σ_T included in the Delaunay triangulation, $\mathcal{D}$, defines an open ball b_T whose bounding spherical surface (in the general case) ∂b_T passes through the $k+1$ points of the simplex. Let ϱ_T be the radius of that bounding sphere, then, the family $G_{k,\alpha}$, is formed by all the k-simplexes $\sigma_T \in \mathcal{D}$ whose ball b_T is empty and $\varrho_T < \alpha$. The family $G_{k,\alpha}$ does not necessarily form simplicial complexes, so Edelsbrunner and Mcke [13] defined the α-complex, $\mathcal{C}_\alpha$, as the simplicial complex whose k-simplexes are either in $G_{k,\alpha}$, or else they bound $(k+1)$-simplexes of C_α . If we define the underlying space of $\mathcal{C}_\alpha$, $|\mathcal{C}_\alpha|$, as the union of all simplexes in $\mathcal{C}_\alpha$, the following relationship between α-shapes and α-complexes is found:

$$\mathcal{S}_\alpha = |\mathcal{C}_\alpha| \quad \forall 0 \leq \alpha < \infty. \tag{21}$$

α-Shapes provide a means so as to eliminate from the triangulation those triangles or tetrahedra whose size is bigger than the before-mentioned level of detail, α. Thus, we make a *filtration* of the triangles.

In Fig. 2 an example of the previously presented theory is presented. It represents some instances of the finite set of shapes for a cloud in a intermediate step of the simulation of a wave breaking at a beach.

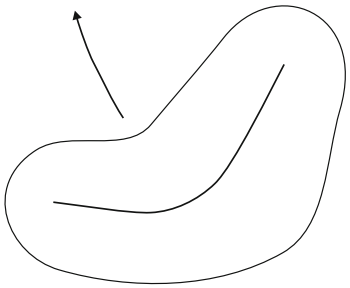


Fig. 3 Medial axis of a two-dimensional curve

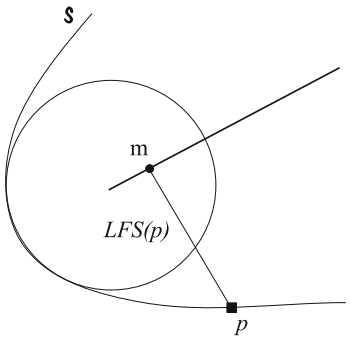


Fig. 4 Computation of the LFS at a point p

3.2 How to choose the α-value

Many authors claim that the main difficulty with the α-shape technique is related to the choice of the α-value [23]. In this section we provide a practical means to do so in the type problems we are dealing with. To this end, it will be necessary to give some prior definitions.

Definition 1 The medial axis (see for instance [4] and references therein) of a $d-1$ dimensional, twice-differentiable, surface $\Gamma = \partial\Omega$ in $\mathbb{R}^d$ is the closure of the set of points which have two or more closest points in Γ. An example of medial axis of a curve is shown in Fig. 3.

Definition 2 The *local feature size* [4], $LFS(p)$, of a point $p \in \Gamma$ is defined as the Euclidean distance from p to the closest point m on the medial axis. In Fig. 4 the computation of the LFS at a point is shown. Observe the difference between this concept and the radius of curvature of the curve at that point, which is different at different directions.

In mesh generation, the medial axis of a surface has been used to account for a measure of the desired point density in a region (see [5]). To this end, it is useful to define a measure of the sampling density of the curve.

Definition 3 (ε-sampling) The surface Γ is said to be ε-sampled by a subset $\{n_I\}_{I=0}^{m}$ of the set of nodes $\boldsymbol{N}$ if every point $p \in \Gamma$ is within a distance $\varepsilon \cdot LFS(n_I)$ of a sample point $n_I \in \Gamma$.

In practical situations, it is common to have an explicit description of the boundary of the domain at the initial time step, or reference configuration—this will not be the case for all the subsequent time steps, as mentioned before, since we try to avoid the use of boundary markers or similar techniques. At this configuration, we proceed by constructing an ε-sampling of the boundary curve or surface. Note that it should be twice differentiable in order to guarantee a non-vanishing LFS. In other words, it will not be possible to represent a sharp (concave) corner in the domain without the help of a segment chain (in 2D) or boundary triangulation (in 3D).

It is therefore important to choose the *level of detail* up to which we represent the initial configuration of the domain. Details of size lower than the chosen *discrete* LFS will not be represented by the method. In fact this is similar to the situation found when meshing a mechanical part, for instance. Many analysts choose to eliminate some details of the geometry irrelevant for the results.

Once we chose the desired level of detail for representing the initial configuration of the domain, we construct an ε-sampling of the boundary (with $\varepsilon < 1$) and extend the cloud of nodes to the interior of the domain, taking always the nodal distance measure, $h \approx \varepsilon \cdot LFS$.

As dictated by the preceding definitions, the choice of α such that $h < \alpha < LFS$ will provide a good approximation of the initial domain. In this way, triangles pertaining to the obtained shape of the domain will be bounded from above by the chosen LFS and from below by h. Thus, no triangle will overlap concave portions of the domain's boundary, nor spurious holes will appear. There exist, in addition, theoretical proofs of the convergence of the shape of the domain to the actual one with increasing nodal distributions, see for instance [23].

As the domain evolves, no further explicit definition of the boundary will be available, and the resulting shapes will never reproduce details of LFS lower than α, as is obvious (those triangles will be eliminated from the triangulation). However, for nodal discretizations fine enough, this technique provides very good results, with excellent mass conservation properties, see [1,17,24].

3.3 Problems with the α-shape technique

There remain, however, some important problems in the application of α-shape techniques to updated Lagrangian simulations of flows with free surfaces. Maybe the most important is that, when contact between two portions of the domain, or auto-contact occurs, the LFS of portions of the boundary—precisely those getting into contact—decreases,

Comput Mech (2008) 42:467–479 473

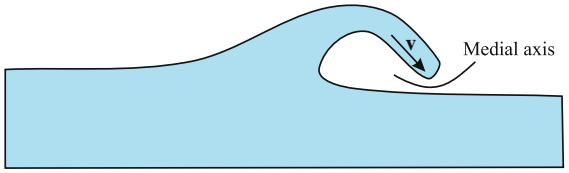


Fig. 5 Evolution of the LFS at the neighbourhood of two surfaces getting into contact. A portion of the medial axis of points in the neighbourhood of the contacting area is depicted. Remember that the LFS is the distance between the boundary and the medial axis. Thus, it vanishes rapidly in this situation

Fig. 6 Spurious detection of contact between the breaking wave and the still water

and can be, during some time steps, below the threshold value α. This is precisely the situation that will happen shortly after the time step depicted in Fig. 2, see Fig. 5. If this happens, contact will be spuriously detected by the standard α-shape technique once the LFS is below α. In Fig. 6 an example is provided for the previous problem of spurious detection of auto-contact between the breaking wave and the surface of the sea. Note that contact is detected some time steps prior to its actual occurrence.

In the next section we propose two additional filtration to be done after the α-shape filtration in order to improve the behaviour of the method.

4 Proposed algorithm

The proposed algorithm makes use of the information provided by previous time steps on the shape of the domain and, through the computed velocity field, on its future shape. Thus, we will make use of the essential variable fields to ameliorate the behaviour of the α-shape technique by performing a modified filtration processes over the Delaunay triangulation of the set of points.

In order to discern different parts of a body or different bodies getting into contact, we assume that all particles belonging to the same body should behave in a somewhat similar way. In our case, they all should move roughly with the same velocity or, more precisely, without jumps in the velocity nor steep gradients (this is true only for moderate Reynolds numbers in the flow, but the proposed technique is not valid for turbulent flows). In this way, the k-simplexes found to be constituted by nodes that exhibit highly dissimilar characteristics should be regarded as *invalid* and filtered out of the α-shape.

For each k-simplex, we employ a modified circumcircle criteria which includes a deformation parameter based on the differences between the associated nodal velocities. This parameter is used to alter the metric space. Elongating the Euclidean distance measured proportionally to the velocity differences causes the *invalid* simplices to appear larger and therefore fail the circumcircle test.

In order to determine the deformation parameter in our case, we compare the different velocity vector directions. To this end, we first compute a principal direction $\boldsymbol{d}$, which is found as the *local normal direction* at [28]

$$\boldsymbol{d} = \sum_{i=1}^{k+1} s_i \boldsymbol{v}_i \text{ such that } ||\boldsymbol{d}|| = \max_{s_i=\pm 1} \left\| \sum_{i=1}^{k+1} s_i \boldsymbol{v}_i \right\|, \tag{22}$$

where $\boldsymbol{v}_i$ represent each of the nodal velocities associated to the k-simplex, and $||\cdot||$ denotes the norm associated to the metric space.

We define the angle β as the one formed by each velocity vector with the principal direction d. A deformation factor f_β is then obtained according to

$$f_\beta = 1 - \frac{|\beta_{\max} - \beta_{\min}|}{\pi}. \tag{23}$$

This factor allows to filter those k-simplexes formed by nodes of opposing or diverting velocities. Note that only if the simplex is "large" (according with an user provided measure, α) and their nodes move with very dissimilar velocities, it will be eliminated from the triangulation. If the triangle is small enough it will be most likely representing a recirculation in the flow, for instance, and will still be maintained in the model.

There are, however, cases in which only one of the bodies (or only some sub-region of the model) is moving and the previous filter alone would still detect a spurious contact. In that situation one or more of the nodes will not be taken into account by the above factor, yet those simplexes need to be filtered. The need arises to take into account the gradient of those velocities, and calculate a deformation factor f_{mod} as

$$f_{\text{mod}} = 1 - \frac{\| v_i \|_{\max} - \| v_j \|_{\min}}{\| v_i \|_{\max}}. \tag{24}$$

Once the deformation factors are obtained we proceed to alter the metric tensor, assuming it constant at each simplex. The distance between two points can be defined as

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})\boldsymbol{M}(\boldsymbol{x} - \boldsymbol{y})^T}, \tag{25}$$

where $\boldsymbol{M}$ represents the metric tensor. We define a "modified" metric tensor $\boldsymbol{M}$ with $1/(f_{\text{mod}}^a f_\beta^b)$ on the diagonal, where a

and b are user defined factors that allow adjusting the method depending on the nature of the simulation.

The new deformed circumradius is used to check the α-shape test, usually making the unwanted simplexes fail. This process is performed on a simplex by simplex basis.

While the Delaunay triangulation is necessary when dealing with Natural Element methods, it is not with the rest of meshless methods. It adds a little bit of CPU time to the simulation, that in general is negligible. Very efficient algorithms exist in the literature (see, for instance the *Qhull* software [6], which is free and very efficient). It is able to triangulate 1000 nodes in 0.016 CPU seconds on a laptop equipped with a Centrino processor and 500 Mb of RAM memory. The proposed filtration adds some very little extra CPU time to this, since it can be implemented within the Delaunay algorithm, or by adding a single `do while` loop to the code over all the triangles.

5 Examples

5.1 Benchmarking

In order to validate the proposed method, it was employed in two classes of idealized cases of a 2D drop falling as a rigid body towards a wall, Fig. 7. On the first class problems, see Fig. 7a, the ball was dropped over a plane surface moving at the same direction at less speed than the ball. On the second family of cases, Fig. 7b both bodies move at the same speed but in different direction. The α parameter on all cases was chosen deliberately larger than actually needed, so that the α-shape would be a complete convex hull encompassing both bodies. That resulted in a triangulation that includes several *invalid* triangles, shown in Fig. 8b. These triangles could constitute an important error source due to the effects of a non-existing contact.

The first setting allows to test the effect of the gradient of velocities, taken into account by f_{mod}. At the limit case, the speed of the plane is null, so the deformation factor goes to infinite, therefore the size of α becomes unimportant as the triangles composed by nodes from the two different surfaces will always fail the test. In this case α was 5—thus taken deliberately large—a was 10 and b was 0.

Less extreme cases where tested, on which the surface was not completely still, but moving at less speed in the same direction of the ball. All cases resulted in successful filtrations. Figure 9 shows a detail of the area on which both surfaces nearly touch. Triangles eliminated by the proposed filtration are shown in light grey. Specially noteworthy is the difference between the element sizes between the drop and the plate. Without an external filtration, there is no α which could manage to obtain a reasonable α-shape, given that the plate element size is more than five times the element size of

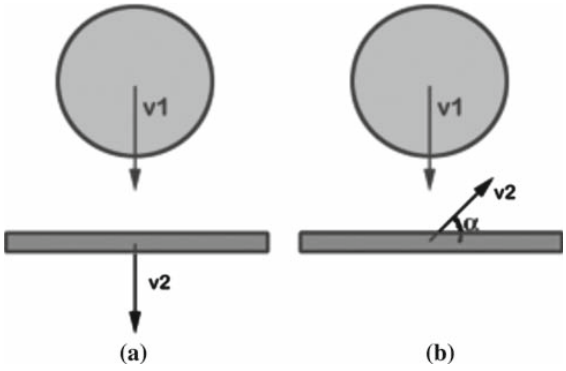
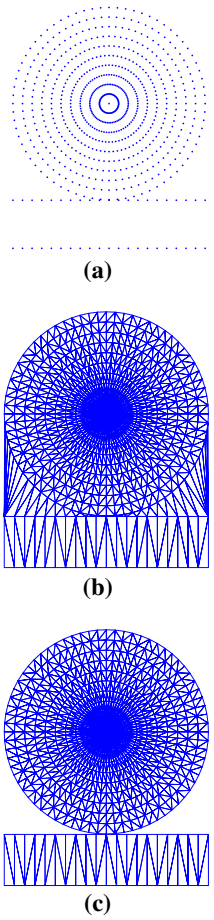


Fig. 7 Method validation. Cases studied of a 2D ball drop over a flat surface. **a** $0 \leq \boldsymbol{v}_2 < \boldsymbol{v}_1$ and **b** $\boldsymbol{v}_2 = \boldsymbol{v}_1,\ 0 \leq \beta \leq \pi/2$

Fig. 8 Drop approaching a surface. Both families of cases studied were tested on the same set of nodes (**a**). The resultant geometry of the domain provided the standard α-shape (**b**) and the modified method (**c**) are depicted



the drop and the difference with the gap between both bodies is even more drastic. Density based filtrations could be made to recognize both areas, yet the case would still prove to be challenging if possible at all.

On the second family of cases the surface moves at the same speed but in different direction, still usually towards

Comput Mech (2008) 42:467–479

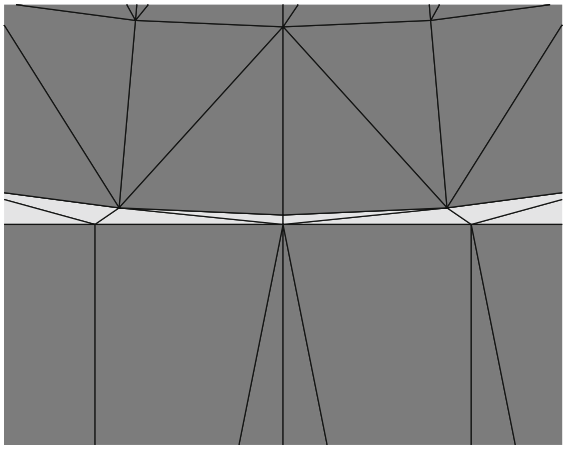


Fig. 9 Drop approaching a still plate (detail of the contact zone). In light grey the triangles filtrated by the proposed techniques are shown

Fig. 10 Detail of the velocity field at the wave crest

the ball. This exercise allows to check the performance of the filtration due to f_β. The case in which both bodies approach directly to each other is also an extreme situation where none of the offending triangles will ever pass the test regardless of the chosen α. In this case, α, a and b were 5—again deliberately large—0 and 1, respectively. The angle difference has been tested up to the case where the bodies moved in a perpendicular way. In all the conditions both bodies could be recognized by adjusting the b factor only. In this case the difference between element sizes at the drop and the plate is also noteworthy.

5.2 2D wave breaking

A third type of test was performed with the wave problem shown in Figs. 2a–f and 6 in which we could check the performance on a *real* 2D case. The velocity field on the crest of the wave is shown in Fig. 10. It can be noticed how the vectors are roughly aligned in the same direction, thus resembling the first family of cases in the preceding section. Even though the velocity vectors seem to be very similar, the difference is so that the filtration is successful at the crest. Again, a reasonable value for the parameters a and b seems to be 10 and 1, respectively, and our experience dictates that this is so for a general problem presenting this kind of difficulty.

The results of the proposed technique are shown in Fig. 11. In this case the proposed method is able to discern between the crest and the trough of the wave. Again, the α value was taken deliberately too high, to show that even a poor choice of α will lead to a proper result.

Mass (volume) conservation is analysed in Fig. 12. In this case, the predicted volume of the whole domain is analysed, taking into account that obtained by standard α-shape techniques and the one obtained by the proposed method. As can be noticed, the proposed method gives more accurate results, with less than 1% error in volume. The gain in volume due to spurious contact detection for the α-shape technique raises up to 5% for the final time steps, even if the contact region in the model is concentrated near the wave crest.

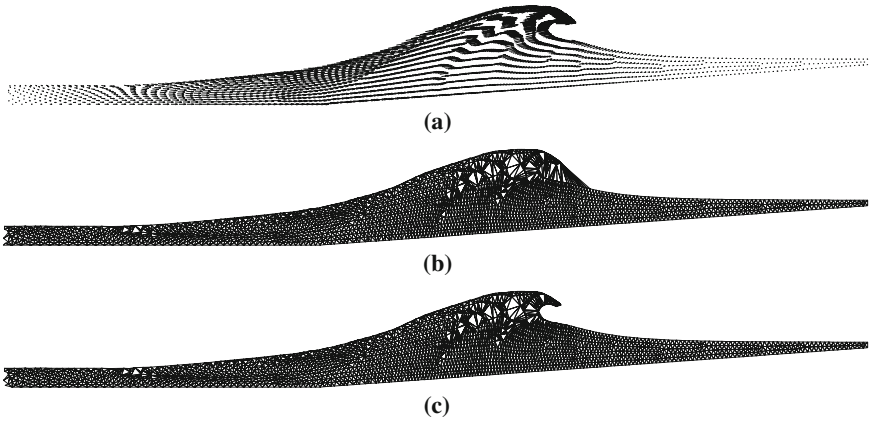


Fig. 11 Wave before breaking. Velocity vectors (**a**), α-shape without additional filtration (**b**) and shape reconstructed with the new approach (**c**). In this cases the parameters used where $\alpha = 9, a = 10, b = 1$

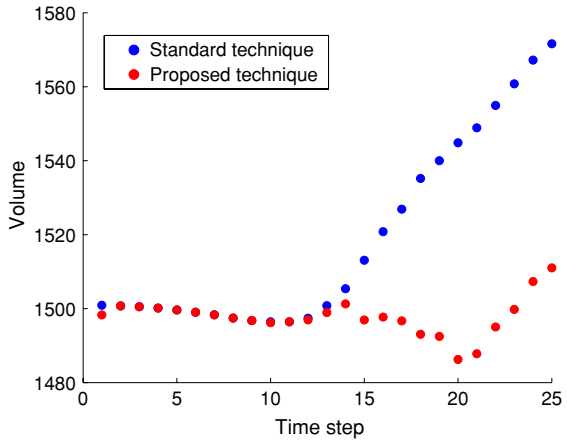


Fig. 12 Volume conservation for the standard and the proposed technique

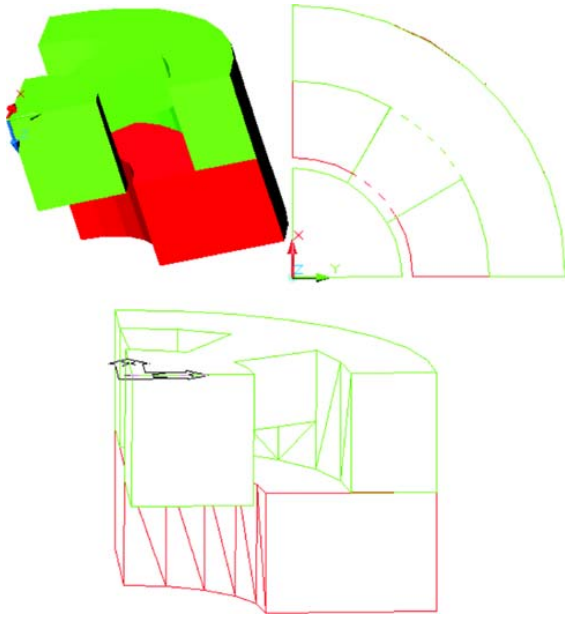


Fig. 13 Schematic geometry of the die for the extrusion of a hollow profile. Note the special characteristics of the flow, that must divide to pass trough the green region and then re-join to flow out of the die (*red region*)

5.3 An extension to 3D problems. Aluminium extrusion

5.3.1 Constitutive equations for aluminium

In this last example we considered a rigid-viscoplastic constitutive law for the aluminium, allowing for a *flow formulation* for the problem [29]. In essence, we neglect inertia terms in Eqs. (10)–(12) and considered a non-linear constitutive law for the aluminium in the form

$$\boldsymbol{s} = 2\frac{\eta\overline{d} + \sigma_y(\overline{d})}{3\overline{d}}\boldsymbol{d}, \tag{26}$$

where $\boldsymbol{s}$ represents the deviatoric part of he strain rate tensor and σ_y represents the yield stress. $\overline{d}$ represents the equivalent strain rate. Note that, depending on the η value, the return to the yield surface is done with different velocity. Since it is common to describe aluminium behaviour as rigid-plastic (rather than viscoplastic) we employ null viscosity, so as to enforce $Y = \overline{\sigma} - \sigma_y = 0$, leading to

$$\boldsymbol{s} = \frac{2\sigma_y}{3\overline{d}}\boldsymbol{d}. \tag{27}$$

Finally, the constitutive equation, accounting the incompressibility of plastic flow results:

$$\boldsymbol{\sigma} = 2\mu\boldsymbol{d} - p\boldsymbol{I}, \quad \text{with } \mu = \frac{\sigma_y}{3\overline{d}}. \tag{28}$$

5.3.2 Linearized form of the variational problem

If we write the incremental variational equation at time $t+\Delta t$ we arrive to:

$$\int_{\Omega(t+\Delta t)} \left(-(p^t + \Delta p)\boldsymbol{I} + 2\mu(\boldsymbol{d}^t + \Delta\boldsymbol{d})(\boldsymbol{d}^t + \Delta\boldsymbol{d})\right) : \boldsymbol{d}^* d\Omega = 0. \tag{29}$$

Domain updating is done in an explicit procedure, given the last converged velocity field, but due to the non-linear character of the constitutive equations, an iterative approach has been applied to the conservation equations, using the Newton–Raphson scheme, thus leading to

$$\begin{aligned}\int_{\Omega(t+\Delta t)} &\left(-\Delta\Delta p\boldsymbol{I} + 2\mu\left(\frac{\partial\mu(\boldsymbol{d}_k^{t+\Delta t})}{\partial\boldsymbol{d}} : \Delta\Delta\boldsymbol{d}\right)\boldsymbol{d}_k^{t+\Delta t}\right. \\ &\left. + 2\mu(\boldsymbol{d}_k^{t+\Delta t})\Delta\Delta\boldsymbol{d}\right) : \boldsymbol{d}^* d\Omega \\ &= -\int_{\Omega(t+\Delta t)} (-p_k^{t+\Delta t}\boldsymbol{I} + 2\mu(\boldsymbol{d}_k^{t+\Delta t})\boldsymbol{d}_k^{t+\Delta t}) : \boldsymbol{d}^* d\Omega,\end{aligned} \tag{30}$$

Comput Mech (2008) 42:467–479 477

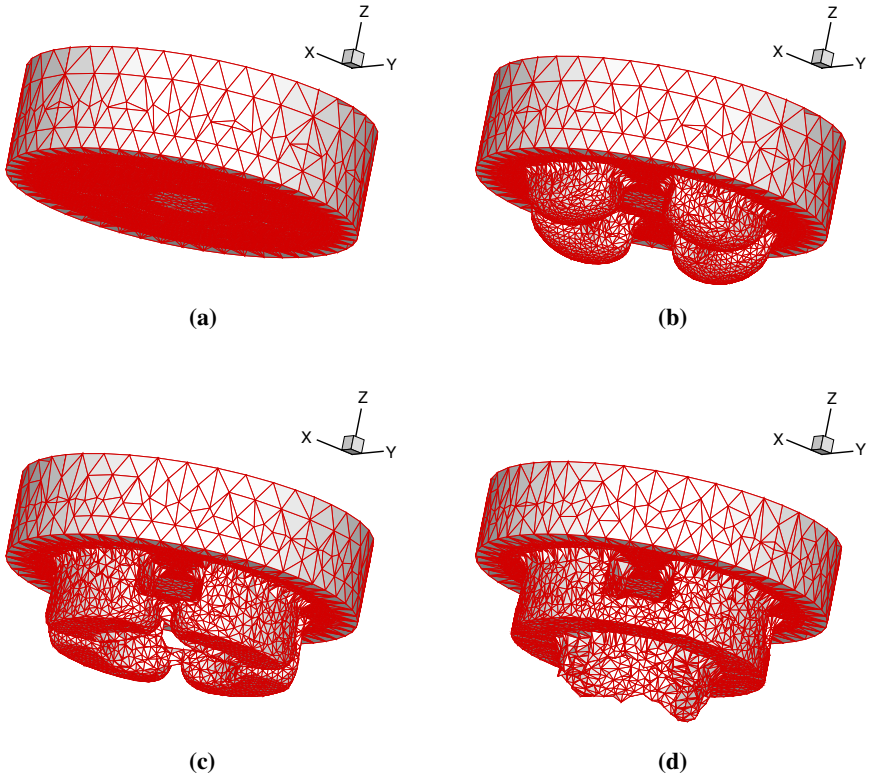


Fig. 14 Sequence of aluminium flow at the early stages of the extrusion of a hollow cylindrical profile

where the subscript k indicates the iteration within a time increment. The incremental form of the incompressibility condition results

$$\int_{\Omega(t+\Delta t)} \nabla \cdot (\Delta\Delta \boldsymbol{v})\ p^* d\Omega = - \int_{\Omega(t+\Delta t)} \nabla \cdot (\boldsymbol{v}_k^{t+\Delta t}) p^* d\Omega . \tag{31}$$

If we approximate the velocities and pressures, as well as their variations, by employing a finite-dimensional set of basis functions, we arrive to a discrete form of the previous equations (Bubnov–Galerkin method)

$$\Delta\Delta \boldsymbol{v}^h(\boldsymbol{x}) = \sum_{I=1}^{n} \phi_I(\boldsymbol{x}) \Delta\Delta \boldsymbol{v}_I , \tag{32}$$

$$\Delta\Delta p^h(\boldsymbol{x}) = \sum_{I=1}^{n} \psi_I(\boldsymbol{x}) \Delta\Delta p_I , \tag{33}$$

where n represents the number of nodes considered in the approximation. Natural neighbour approximation (Laplace interpolations [27]) is employed in this work to interpolate the velocity field, while Thiessen interpolation (piecewise constant on each Voronoi cell) is used for pressures. Any other form of meshless approximations could also be employed as well. More details on the derivation of the model can be found in [1,2].

5.3.3 Performance of the proposed technique

We considered the simulation of the extrusion of a hollow cylinder. Tube extrusion is especially difficult to simulate from the geometrical point of view, since the diverted metal flow must converge before going through the last section of the extrusion die. A schematic representation of the geometry of the die is shown in Fig. 13, where only a quarter of the domain was represented. By invoking appropriate symmetry conditions, this same quarter of the domain was employed for simulation. Some snapshots of the flow of aluminium during this extrusion process are shown in Fig. 14, where post-processing has been employed for clarity, in order to show the whole geometry of the domain.

The domain is marked in red lines on Fig. 15. This figure also shows a particular time step where using regular α-shapes results in spurious contact detection. This is also notorious in Fig. 14c. The *invalid* tetrahedra can be recognized by their size, larger than the nodal spacing in that area, giving a jagged feel to the resulting solid.

478 Comput Mech (2008) 42:467–479

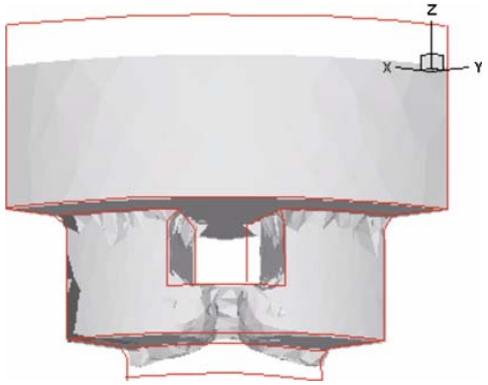


Fig. 15 Extrusion process. Simulation domain (*red lines*) and snapshot showing spurious contact between flows at an intermediate time step

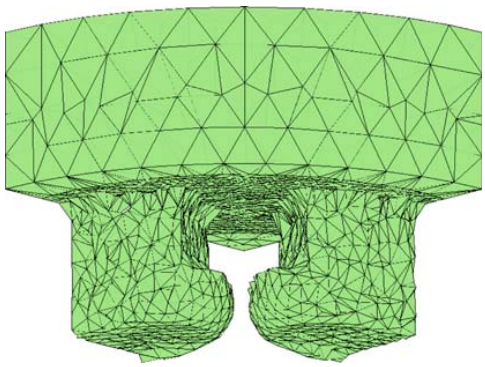


Fig. 16 Extrusion process. Instant before contact of the two metal flows. Spurious tetrahedra are removed from the triangulation ($\alpha = 8$, $a = 0.1$, $b = 2$)

The model is divided in sections of different nodal densities, being the sparsest part at the top. Using the standard method special care is needed to avoid spurious contact. It is possible to define different α-values for each region to address this problem.

With the new approach, a single, deliberately big, α-value can be defined and still obtain good results. In Fig. 16 a snapshot corresponding to the same time step on which traditional α-shapes failed to avoid the spurious contact is shown. In this case flow fronts are clearly kept apart from each other until actual contact occurs.

6 Conclusions

An improved α-shape technique is introduced for domain tracking in updated Lagrangian simulations of free surface flows. This improvement is based on the addition a new filtration to the standard α-shape technique. This new filtration takes into account the velocity field of the flow, so as to predict in some sense its future geometry. We have introduced a way to use non-geometric information inherent to our model, as a tool to filter α-shapes and being able to obtain good surface definition, avoiding traditional problems associated to this method, as spurious contacts. Even in cases where there is large nodal density differences, the flow provides enough information to recognize, even for rough tuning of α-values, different regions in the model that pertain either to zones getting in contact or to different bodies in the simulation. Despite the inclusion of two new user-defined parameters, our methodology is flexible enough to face the geometry changes that occur with moderate Reynolds number flows, as covered by the presented formulation. The main conclusion is that it is considerably easier to find the three parameters α, a and b, than to find the single α value for some special, delicate cases—notably some time steps prior to contact—in standard α-shape technique. We have shown how, even for a poor selection of α, the proposed technique is able to correctly filtrate the actual geometry of the domain. Values of the parameters are much less sensible to modifications than α for standard α-shapes, and thus the ease of use and good results of the proposed technique.

Acknowledgments The support of the Spanish Ministry of Education and Science, through the project CICYT-DPI2005-08727-C02-01, is gratefully acknowledged.

References

1. Alfaro I, Bel D, Cueto E, Doblaré M, Chinesta F (2006a) Three-dimensional simulation of aluminium extrusion by the alpha-shape based natural element method. Comput Methods Appl Mech Eng 195(33–36):4269–4286
2. Alfaro I, Gonzalez D, Bel D, Cueto E, Doblaré M, Chinesta F (2006b) Recent advances in the meshless simulation of aluminium extrusion and other related forming processes. Arch Comput Methods Eng 13(1):3–44
3. Amenta N, Bern M, Eppstein D (1998a) The crust and the beta-skeleton: combinatorial curve reconstruction. Graph Models Image Process 60/2(2):125–135
4. Amenta N, Bern M, Kamvysselis M (1998b) A new voronoi-based surface reconstruction algorithm. In: Siggraph 98, pp 415–421
5. Armstrong CG, Robinson D, McKeag R, Li T, Bridgett S, Donaghy R, McGleenan C (1995) Medials for meshing and more. In: 4th International Meshing Roundtable. Sandia National Laboratories, pp 277–288
6. Barber CB, Dobkin DP, Huhdanpaa HT (1996) The quickhull algorithm for convex hulls. ACM Trans Math Softw 22(4):469–483
7. Belytschko T, Lu YY, Gu L (1994) Element-free Galerkin methods. Int J Numer Methods Eng 37:229–256
8. Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P (1998) Meshless methods: an overview and recent developments. Comput Methods Appl Mech Eng 139:3–47
9. Birknes1 J, Pedersen G (2006) A particle finite element method applied to long wave run-up. Int J Numer Methods Fluids 52:237–261
10. Cazals F, Giesen J, Pauly M, Zomorodian A (2006) The conformal alpha-shape filtration. Visual Comput 22(8):531–540

11. Cueto E, Doblaré M, Gracia L (2000) Imposing essential boundary conditions in the Natural Element Method by means of density-scaled α-shapes. Int J Numer Methods Eng 49(4):519–546
12. Dey TK, Giesen J, John M (2003) Alpha-shapes and flow shapes are homotopy equivalent. In: Proc. Sympos. Theory Computing (STOC 2003), pp 493–502
13. Edelsbrunner H, Mücke E (1994) Three dimensional alpha shapes. ACM Trans Graph 13:43–72
14. Edelsbrunner H, Kirkpatrick DG, Seidel R (1983) On the shape of a set of points in the plane. IEEE Trans Inform Theory IT 29(4): 551–559
15. Giesen J, John M (2003) The flow complex: a data structure for geometric modelling. In: Proceedings of the 14th Annual ACM-SIAM symposium on Discrete Algorithms, pp 285–294
16. Gonzalez D, Cueto E, Martinez MA, Doblaré M (2004) Numerical integration in natural neighbour Galerkin methods. Int J Numer Methods Eng 60(12):2077–2104
17. González D, Cueto E, Chinesta F, Doblaré M (2007) A natural element updated Lagrangian strategy for free-surface fluid dynamics. J Comput Phys 223(1):127–150
18. Idelsohn SR, Oñate E (2006) To mesh or not to mesh: that is the question. Comput Methods Appl Mech Eng 195(37–40):4681–4696
19. Idelsohn SR, Oñate E, Calvo N, del Pin F (2003) The meshless finite element method. Int J Numer Methods Eng 58:893–912
20. Idelsohn SR, Oñate E, del Pin F (2004) The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. Int J Numer Methods Eng 61(7):964–989
21. Lewis RW, Navti SE, Taylor C (1997) A mixed lagrangian-eulerian approach to modelling fluid flow during mould filling. Int J Numer Methods Fluids 25:931–952
22. Liu WK, Jun S, Li S, Adee J, Belytschko T (1995) Reproducing kernel particle methods. Int J Numer Methods Eng 38:1655–1679
23. Mandal DP, Murthy CA (1997) Selection of alpha for alpha-hull in $\mathbb{R}^2$. Pattern Recognit 10:1759–1767
24. Martinez MA, Cueto E, Alfaro I, Doblaré M, Chinesta F (2004) Updated Lagrangian free surface flow simulations with Natural Neighbour Galerkin methods. Int J Numer Methods Eng 60(13):2105–2129
25. Nayroles B, Touzot G, Villon P (1992) Generalizing the finite element method: diffuse approximation and diffuse elements. Comput Mech 10:307–318
26. Sukumar N, Moran B, Belytschko T (1998) The natural element method in solid mechanics. Int J Numer Methods Eng 43(5):839–887
27. Sukumar N, Moran B, Semenov AY, Belikov VV (2001) Natural neighbor Galerkin methods. Int J Numer Methods Eng 50(1):1–27
28. Teichmann M, Capps M (1998) Surface reconstruction with anisotropic density-scaled alpha shapes. In: Proceedings of the 1998 IEEE Visualization Conference
29. Zienkiewicz OC, Godbolet PN (1974) Flow of plastic and visco-plastic solids with special reference to extrusion and forming processes. Int J Numer Methods Eng 8:3–16
30. Zienkiewicz OC, Pain PC, Oñate E (1978) Flow of solids during forming and extrusion: some aspects of numerical solutions. Int J Solids Struct 14:15–38

A.2 European Journal of Computational Mechanics (2006) 16:323–336

A Natural Element updated Lagrangian approach for modelling Fluid-Structure interactions

A. Galavís[1], D. González[1], E. Cueto[1], F. Chinesta[2], M. Doblaré[1]

[1]*Aragón Institute of Engineering Research. University of Zaragoza. Edificio Betancourt. María de Luna, 7. E-50018 Zaragoza, Spain.*

E-mail : ecueto@unizar.es

[2]*Laboratoire de mécanique des systèmes et des procédés, UMR CNRS-ENSAM-ESEM, Ecole Nationale Supérieure d'Arts et Métiers, 151 boulevard de l'hôpital, 75013 Paris, France.*

ABSTRACT. *In this paper we present a novel methodology for the numerical simulation of fluid-structure interactions in the presence of free surfaces. It is based on the use of the Natural Element Method (NEM) in an updated Lagrangian framework, together with the integration of the Navier-Stokes equations by employing a Galerkin-characteristics formulation. Tracking of the free-surface is made by employing shape constructors, in particular α-shapes. A theoretical description of the method is made and also some examples of the performance of the technique are included.*

KEYWORDS: *Fluid-structure interaction, Meshless methods, Natural Element Method, α-shapes.*

1. Introduction

The fact that meshless methods (Belytschko *et al.*, 1994) (Liu *et al.*, 1995) do not suffer of mesh distortion opened a renewed interest in the last decade in Lagrangian formulations for some problems, being free-surface flows a typical example. Thus, it is possible to employ an updated Lagrangian strategy for the fluid domain, while employing a total or updated Lagrangian strategy for the solid. This approach is very convenient for some classes of problems, especially those involving drastic changes in the fluid domain geometry. Both domains are then formulated in similar frameworks and the coupling between them becomes more direct than in ALE formulations (see (Donea, 1983) or (Donea and Huerta, 2003))

In this paper we describe mainly the fluid flow formulation proposed in the context of an updated Lagrangian strategy. We employ the α-shape-based Natural

REMN – 15/2006. Calculs avec méthodes sans maillage, pages 1 à X

Element Method (α-NEM) (Cueto *et al.*, 2002) (Cueto *et al.*, 2003) to this end. At present the solid is assumed rigid, being prescribed its kinematics.

This formulation posses some advantages, that include an exact interpolation along the boundary (Cueto *et al.*, 2001), that allows for a standard, FE-like, treatment of the fluid-solid interface conditions. We firstly describe the bases of the α-NEM and then introduce the proposed numerical scheme for the integration of the Navier-Stokes equations. Finally, we include some examples that demonstrate the accuracy of the proposed scheme and also prove the potential of the technique.

2. The Natural Element Method

2.1. *Standard formulation*

The NEM (Sukumar *et al.*, 1998) (Cueto *et al.*, 2003) is a Galerkin procedure based on the natural neighbor interpolation scheme, which in turn relies on the concepts of Voronoi diagrams and Delaunay triangulations (see Figure 1), to build Galerkin trial and test functions. These are defined as the Natural Neighbor coordinates of the point under consideration, that is, with respect to Figure 2, the value in the point x of the shape function associated with the node 1 is (Sibson, 1980) (Sibson, 1981)

$$\phi_1(x) = \frac{A_{abfe}}{A_{abcd}} \qquad [1]$$

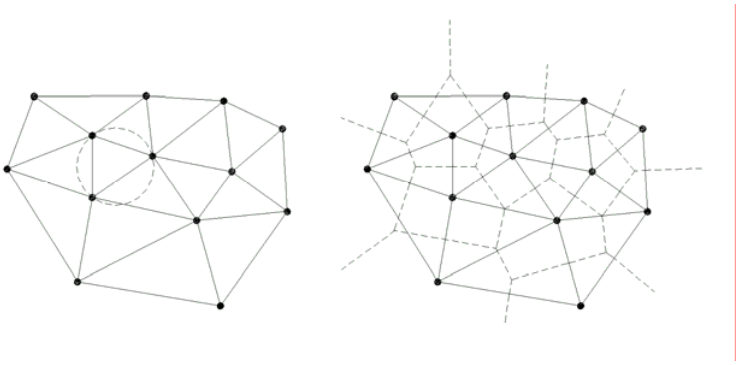


Figure 1. *Delaunay triangulation and Voronoi diagram of a set of points.*

In addition, the NEM has other interesting properties such as linear consistency and smooth shape functions (C^1 everywhere except of the nodes). These functions are dependent on the position and density of nodes, leading to standard FE constant strain triangle shape functions, bilinear shape functions or rational quartic functions in different situations (see Fig. 3 for a typical shape function). These properties permit an exact reproduction of linear displacement fields on the boundary of convex domains.

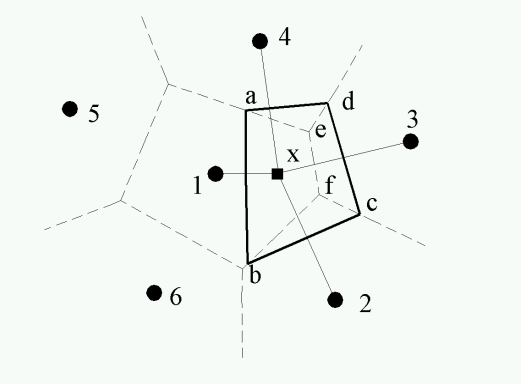


Figure 2. *Definition of natural neighbour coordinates.*

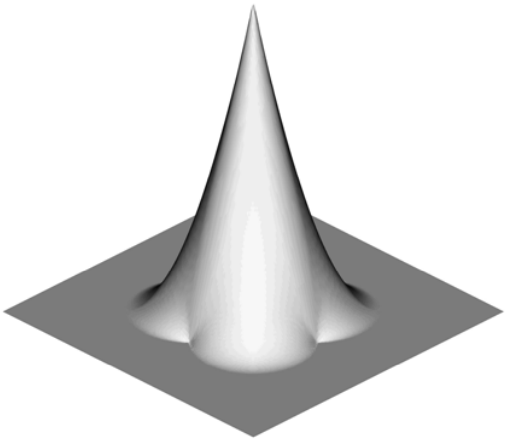


Figure 3. *Typical Sibson shape function (courtesy N. Sukumar)*

2.2. *α-shape formulation*

A slight modification of the way in which the Natural Neighbour interpolant is built was proposed to achieve linear interpolation also over non-convex boundaries (Cueto *et al.*, 2001). This modification was based on the concept of α-shapes. These are a generalization of the concept of the convex hull of a cloud of points and are widely used in the field of scientific visualization and computational geometry to give a shape to a set of points. Alpha-shapes give shape to a cloud of points and are widely used in Computational Geometry despite having been developed quite recently. They were first introduced in two-dimensions by Edelsbrunner in 1983,

and not generalised in three-dimensions until (Edelsbrunner and Mücke, 1994) An α-shape is a generalisation of the convex hull of a cloud of points. It is a polytope that is not necessarily convex and that can be triangulated by a subset of the Delaunay triangulation, thereby maintaining the empty circumcircle criterion.

In what follows, we introduce the formal definition of a complete family of α-shapes for a given set of points *N*, as in Edelsbrunner and Mücke, 1994. Let *N* be a finite set of points in $\Re^3$ and α a real number with $0 \le \alpha \le \infty$. A *k-simplex* σ_T with $0 \le k \le 3$, is defined as the convex hull of a subset $T \subseteq N$ of size $|T| = k+1$. Let *b* be an *α-ball*, that is, an open ball of radius α. A *k*-simplex σ_T for $0 \le k \le 2$ is said to be *α-exposed* if there exists an empty α-ball *b* with $T = \partial b \bigcap N$, where ∂ indicates the boundary of the ball or, more properly, the sphere or plane bounding *b*. That is, a *k*-simplex is α-exposed if an α-ball whose boundary passes through its defining points contains no other point of the set *N*. In this way, we can define a family of sets $F_{k,\alpha}$ as the sets of α-exposed *k*-simplices for the given set *N*, fixed α and $0 \le k \le 2$.

Based on these concepts, the α-shape of *N*, $\mathcal{S}_\alpha$, is defined as the polytope whose boundary consists of the triangles in $F_{2,\alpha}$, the edges in $F_{1,\alpha}$ and the points or vertices in $F_{0,\alpha}$. As the α value decreases, the α-shape shrinks by the progressive development of cavities or holes. For this to occur, one or more α-balls can occupy the interior of a simplex. The α value clearly gives an intuitive measure of the maximum curvature in a region of the domain. The α-shape concept is also a generalisation of the convex hull since the α-shape for value $\alpha = 0$ is identical to the initial set of points, i.e., $\mathcal{S}_0 = N$, and the α-shape for sufficiently high values of α is the convex hull of the given set.

An example of some α-shapes of the complete family for a given set of points distributed over the geometry of a human jaw can be seen in Figure 4.

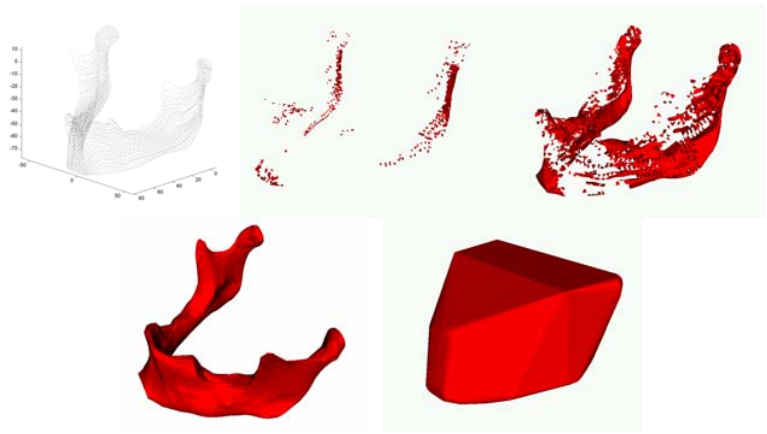


Figure 4. *Five elements of the complete family of α-shapes of a cloud of points distributed over the geometry of a human jaw. Increasing values of α from 0 (top line, left) to infinity (bottom line, right)*

It has been demonstrated (Cueto *et al.*, 2001) how the construction of the interpolant over an appropriate α-shape of the domain gives rise to an exact imposition of essential boundary conditions over any kind of domain (convex or not.) In addition, it enables us to track the flow front position accurately.

3. A natural neighbour updated Lagrangian Strategy for the fluid domain.

In this section we review the time integration scheme developed in (Gonzalez *et al.*, 2006), that will be applied in the integration of the fluid flow equations. It is based on a Galerkin-characteristics formulation of the Navier-Stokes equations.

3.1. *Governing equations.*

We consider here the problem of Fluid Dynamics at moderate Reynolds number. Thus, the governing equations can be set as follows. Consider a fluid in a region Ω of the space $\mathbb{R}^2$ or $\mathbb{R}^3$. The fluid flow is governed by the following momentum and mass balance equations:

$$\rho(\mathbf{v}_{,t} + (\mathbf{v}\cdot\nabla)\mathbf{v}) = \nabla\boldsymbol{\sigma} + \rho\mathbf{b} \text{ in } \Omega\times[0,t] \quad [2]$$

$$\nabla\cdot\mathbf{v} = 0 \text{ in } \Omega\times[0,t] \quad [3]$$

where $\boldsymbol{v}$ represents the fluid velocity, $\boldsymbol{\sigma}$ the stress tensor, ρ represents fluid density and $\mathbf{b}$ the volumetric forces acting on the fluid. The constitutive equation for a newtonian fluid is given by:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau} = -p\mathbf{I} + 2\mu\mathbf{D} \quad [4]$$

where $\mathbf{D}$ is the strain rate tensor, p the pressure and μ the dynamic viscosity of the fluid. To solve the problem we must prescribe an initial state as well as boundary conditions, as usual.

3.2. *Time discretisation*

The motion equations can be grouped to

$$\nabla\cdot\boldsymbol{\sigma} + \rho\mathbf{b} = \rho\frac{d\mathbf{v}}{dt} = \rho\left(\frac{\partial\mathbf{v}}{\partial t} + (\mathbf{v}\cdot\nabla)\mathbf{v}\right) \quad [5]$$

$$\nabla\cdot\mathbf{v} = 0 \quad [6]$$

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\mathbf{D} \quad [7]$$

The weak form of the problem associated to Eqs. [5], [6] and [7] is:

$$\int_\Omega 2\mu\mathbf{D}:\mathbf{D}^* d\Omega - \int_\Omega p\mathbf{I}:\mathbf{D}^* d\Omega = -\int_\Omega \rho\mathbf{b}\mathbf{v}^* d\Omega + \int_\Omega \rho\frac{d\mathbf{v}}{dt}\mathbf{v}^* d\Omega$$
$$\int_\Omega \nabla\cdot\mathbf{v}p^* d\Omega = 0 \qquad [8]$$

The second term in the right-hand side of Eq. [8] represents the inertia effects. Time discretization of this term represents the discretization of the material derivative along the nodal trajectories, which are precisely the characteristic lines related to the advection operator. Thus, assuming known the flow kinematics at time $t_{n-1} = (n-1)\Delta t$, we proceed as follows:

$$\int_\Omega \rho\frac{d\mathbf{v}}{dt}\mathbf{v}^* d\Omega = \int_\Omega \rho\frac{\mathbf{v}^n(\mathbf{x}) - \mathbf{v}^{n-1}(\mathbf{X})}{\Delta t}\mathbf{v}^* d\Omega \qquad [9]$$

where $\mathbf{X}$ represents the position at time t_{n-1} occupied by the particle located at position $\mathbf{x}$ at present time t_n, i.e.:

$$\mathbf{x} = \mathbf{X} + \mathbf{v}^{n-1}(\mathbf{X})\Delta t \qquad [10]$$

So we arrive to:

$$\int_\Omega 2\mu\mathbf{D}:\mathbf{D}^* d\Omega - \int_\Omega p\mathbf{I}:\mathbf{D}^* d\Omega - \int_\Omega \rho\frac{\mathbf{v}\mathbf{v}^*}{\Delta t} d\Omega = -\int_\Omega \rho\mathbf{b}\mathbf{v}^* d\Omega - \int_\Omega \rho\frac{\mathbf{v}^{n-1}\mathbf{v}^*}{\Delta t} d\Omega \qquad [11]$$

where we have dropped the superscript in all the variables corresponding to the current time step.

3.3. *Algorithmical issues*

The most difficult term in Eq. [11] is the second term of the right-hand side. The numerical integration of this term depends on the quadrature scheme employed. If we employ traditional Gauss-based quadratures on the Delaunay triangles, it will be necessary to find the position at time t_{n-1} of the point occupying at time t_n the position of the integration point ξ_k :

$$\int_\Omega \rho\frac{\mathbf{v}^{n-1}\mathbf{v}^*}{\Delta t} d\Omega = \sum_k \rho\frac{\mathbf{v}^{n-1}(\Xi_k) - \mathbf{v}^*(\xi_k)}{\Delta t}\omega_k \qquad [12]$$

where ω_k represent the weight associated to integration point ξ_k, and Ξ_k corresponds to the position occupied at time t_{n-1} by the quadrature point ξ_k. If we employ some type of nodal integration, as in (Chen *et al.*, 2001) (Gonzalez *et al.*, 2004a), this procedure becomes straightforward, with the only need to store nodal velocities at time step t_{n-1}.

We discuss here the procedure to follow when employing Gauss quadratures on the Delaunay triangles. We proceed iteratively. Denoting by i the current iteration, ($i \geq 1$), we apply

$$\mathbf{x}_k = \mathbf{X}_k^i + \mathbf{v}_{n-1}(\mathbf{X}_k^{i-1})\Delta t \text{ with } \mathbf{x}_k = \mathbf{X}_k^0 \quad [13]$$

until $\mathbf{X}_k^i \approx \mathbf{X}_k^{i-1}$ within a prescribed tolerance.

We have assumed that the number of natural neighbours of a given integration point does not change during a time step, thus needing the storage of nodal velocities at time *t-1* only. It can occur that some of the nodes neighbouring the integration point at time *t* were not actually its neighbours at time *t-1*, but this does not constitute a problem, since the number of natural neighbours of a point is usually high (much bigger than three), so the quality of the interpolation is thus guaranteed. In fact, this procedure has shown to converge at a high speed, with no more than 3 iterations, at least for reasonable time steps.

4. Numerical examples.

4.1. Broken dam problem

The broken dam problem is classic when testing the performance of integration methods for free surface flows. We consider a rectangular column of water, initially retained by a door that is instantaneously removed at time *t=0* (see Fig. 5). When the door is removed, water flows under the action of gravity, considered as 9.81 m/s^2. Density of water is 10^3 kg/m^3, and a viscosity of 0.1 $Pa{\cdot}s$ was assumed as in other numerical simulations performed using different numerical strategies (see Duchemin *et al.*, 2002 and references therein, for instance). The discrete model was composed of 3364 nodes. No remeshing, addition or deletion of nodes was performed throughout the computation.

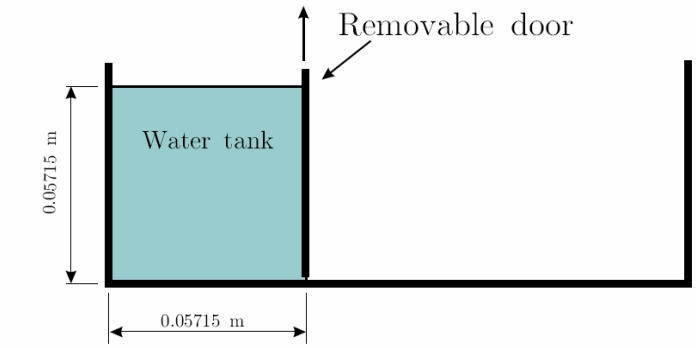


Figure 5. *Geometry of the broken dam problem.*

Fig. 6 shows a comparison between numerical results and experimental ones, obtained from the literature (Martin and Moyce, 1952). As can be noticed, an excellent agreement was found between experimental and numerical results, despite the distortion of the triangulation. In Fig. 7 the error in mass conservation is depicted, which remained always below 3%. The influence of the relationship between the parameter α and the nodal parameter h on this error was deeply analyzed in (Martinez *et al.*, 2004).

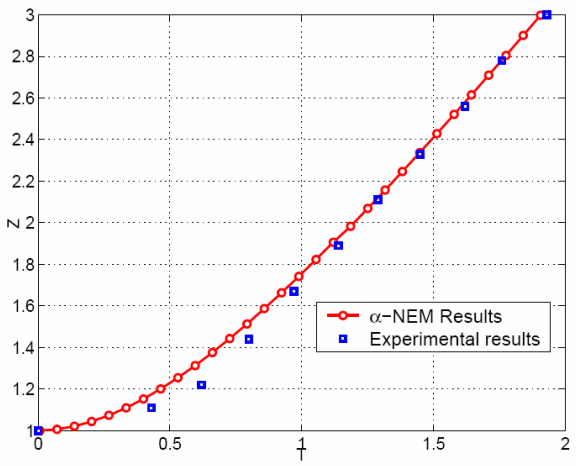


Figure 6. *Front position (in non-dimensional form) in time. Numerical results vs. experimental ones.*

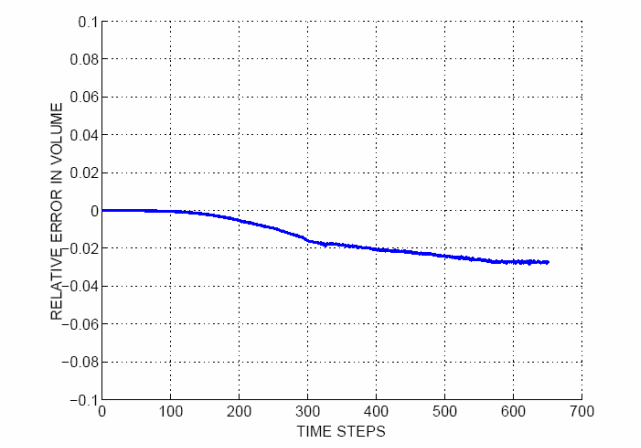
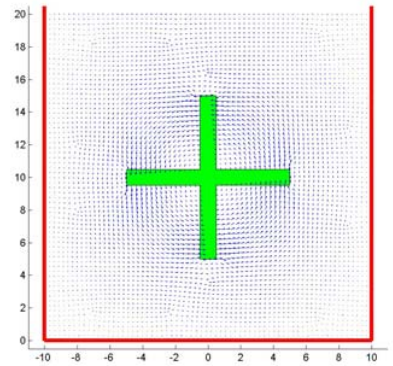


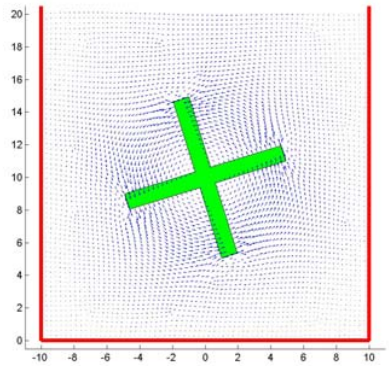
Figure 7. *Error in mass conservation.*

4.2. Water mill

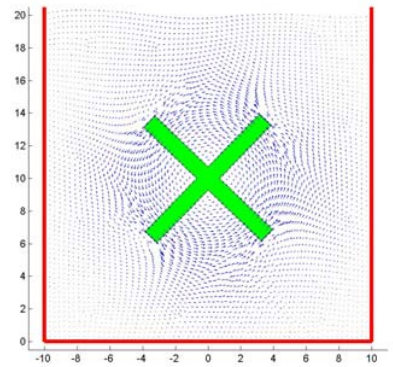
In this example we study the flow generated by the movement of a water mill. The geometry of the container and the dimensions of the mill are shown in Fig. 8(a). The model is composed of 4698 nodes, distributed uniformly at the initial time in a square domain of dimension 20×20 *cm*. The sail is 10 *cm* long, with unit thickness. The sail rotates with constant angular speed of 0.5 *rad/s*.



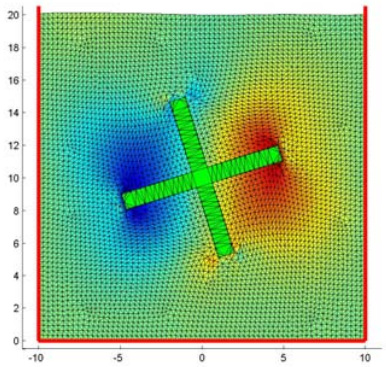
(a) 0^{th} time step.



(b) 100^{th} time step.



(c) 200^{th} time step.



(d) 100^{th} time step.

Figure 8. *(a) to (c) vector plot of the velocity field at three different time steps. (d) Contour plot of the velocity field.*

Stick boundary conditions were assumed on the reservoir walls, being the upper water surface a free boundary which evolves slightly during the simulation as

noticed in Fig. 8(c). Time increment was set to 0.005*s*. being the fluid viscosity of 0.01 *Pa·s*.

The ability of the proposed method for describing flows in the framework of updated Lagrangian description is then fully proved.

4.3. Water mill partially submerged

The proposed method seems particularly well adapted for dealing with free-surface flows. If the sail is only partially submerged, then large-amplitude waves are expected, justifying the interest of the present simulation. For this purpose, we consider the same geometry as in the previous example, but maintaining the sail only partially submerged, as shown in Fig. 9.

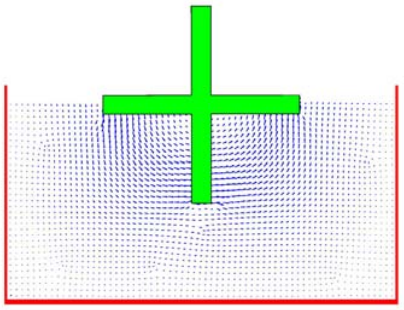
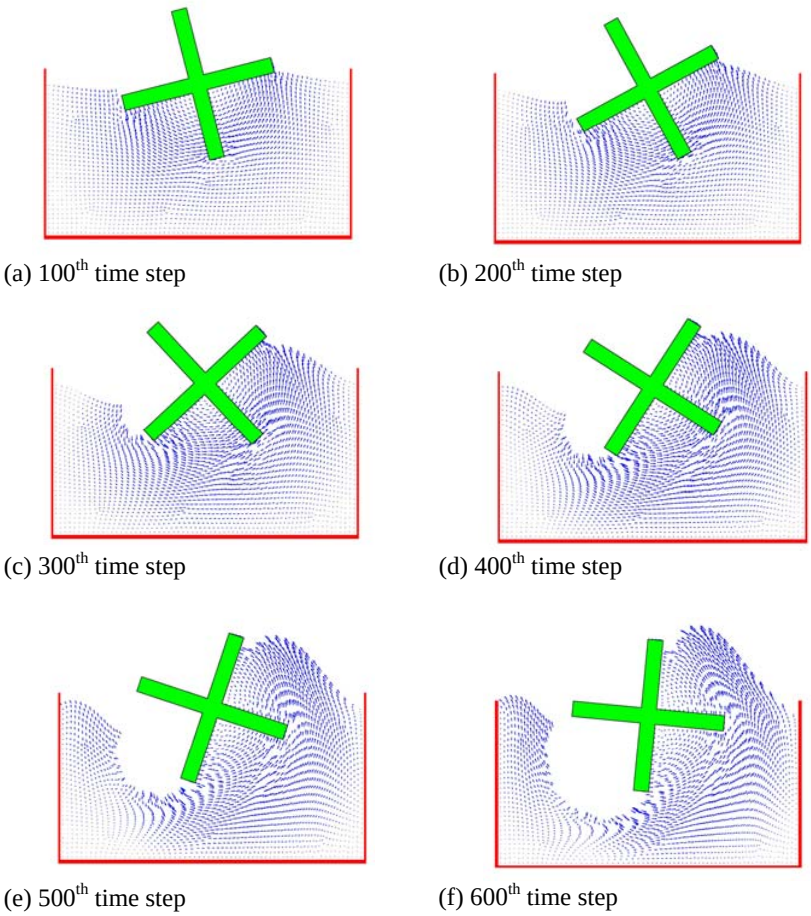


Figure 9. *Initial geometry of the mill problem for the partially submerged configuration.*

Material parameters were chosen as in the previous example. In this case the time step was set to 0.03s. This test can be found in other references, see, for instance (Idelsohn *et al.*, 2004). Note the appearance of a large amplitude wave on the free surface of the liquid. The geometry of the fluid and the eventual generation of drop and jets can be accurately described by the α-shapes. A deep study on this topic has been recently presented in (Martinez *et al.*, 2004) and (Gonzalez *et al.*, 2006).



(a) 100th time step (b) 200th time step

(c) 300th time step (d) 400th time step

(e) 500th time step (f) 600th time step

Figure 10. *Six snapshots of the generation of a wave during the rotation of the mill.*

5. Conclusions.

This paper proposes a Galerkin-characteristics updated-Lagrangian fluid flow formulation for simulation of fluid-structure interaction problems. The fields approximation is based on the use of the Natural Element Method which makes it possible to work with the same cloud of nodes which moves with the material velocity, avoiding remeshing stages. The use of Lagrangian descriptions in both the

solid and fluid domains greatly simplifies the formulation and numerical resolution of fluid-structure interaction problems, especially those involving free-surfaces.

The application of the proposed scheme to real FSI problems (i.e., those in which the movement of the solid is coupled with the fluid one) is currently the aim of our research.

6. Acknowledgement

The support from the Spanish Ministry of Education and Science, through project DPI-2005-08727-C02-01 is gratefully acknowledged.

7. Bibliography

T. Belytschko, Y. Y. Lu, and L. Gu. « Element-Free Galerkin Methods ». *International Journal for Numerical Methods in Engineering*, 37:229-256, 1994.

Jiun-Shyan Chen, Cheng-Tang Wu, Sangpil Yoon, and Yang You. « A stabilized conforming nodal integration for galerkin mesh-free methods ». *International Journal for Numerical Methods in Engineering*, 50:435-466, 2001.

E. Cueto, B. Calvo, and M. Doblare. « Modeling three-dimensional piece-wise homogeneous domains using the α-shape based Natural Element Method ». *International Journal for Numerical Methods in Engineering*, 54:871-897, 2002.

E. Cueto, J. Cegoñino, B. Calvo, and M. Doblare. « On the imposition of essential boundary conditions in Natural Neighbour Galerkin methods ». *Communications in Numerical Methods in Engineering*, 19(5):361-376, 2003.

E. Cueto, M. Doblare, L. Gracia. « Imposing essential boundary conditions in the Natural Element Method by means of density-scaled alpha-shapes ». *International Journal for Numerical Methods in Engineering*, 49-4:519-546, 2000.

E. Cueto, N. Sukumar, B. Calvo, M. A. Martinez, J. Cegoñnino, and M. Doblare. « Overview and recent advances in Natural Neighbour Galerkin methods ». *Archives of Computational Methods in Engineering*, 10(4):307-384, 2003.

J. Donea. « Arbitrary Lagrangian-Eulerian finite element methods ». In T. Belytschko and T. J. R. Hughes, editors, *Computer methods for transient analyses*, pages 474-516. Elsevier-Amsterdam, 1983.

J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley and sons, 2003.

L. Duchemin, S. Popinet, C. Josserand, and S. Zaleski. « Jet formation in bubbles bursting at a free surface ». *Physics of Fluids*, 14:3000-3008, 2002.

H. Edelsbrunner and E. Muecke. « Three dimensional alpha shapes ». *ACM Transactions on Graphics*, 13:43-72, 1994.

D. Gonzalez, E. Cueto, M. A. Martinez, and M. Doblare. « Numerical integration in Natural Neighbour Galerkin methods ». *International Journal for Numerical Methods in Engineering*, 60(12):2077-2104, 2004.

D. Gonzalez, E. Cueto, M. Doblare. « Volumetric locking in Natural Neighbour Galerkin methods ». *International Journal for Numerical Methods in Engineering*, 61(4):611-632, 2004.

D. González, E. Cueto, F. Chinesta, M. Doblaré. « A Natural Element updated Lagrangian strategy for free-surface Fluid Dynamics » Submitted, 2006.

S. R. Idelsohn, E. Oñate, and F. del Pin. « The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves ». *International Journal for Numerical Methods in Engineering*, 61 (7):964-989, 2004.

W. K. Liu, S. Jun, S. Li, J. Adee, and T. Belytschko. « Reproducing kernel particle methods ». *International Journal for Numerical Methods in Engineering*, 38:1655-1679, 1995.

J.C. Martin and W.J. Moyce. « Part IV. an experimental study of the collapse of liquid columns on a rigid horizontal plane ». *Phil. Tran. R. Soc. London*, 244:312, 1952.

M. A. Martinez, E. Cueto, I. Alfaro, M. Doblare, and F. Chinesta. « Updated Lagrangian free surface flow simulations with Natural Neighbour Galerkin methods ». *International Journal for Numerical Methods in Engineering*, 60(13):2105-2129, 2004.

R. Sibson. « A Vector Identity for the Dirichlet Tesselation ». *Mathematical Proceedings of the Cambridge Philosophical Society* , 87:151-155, 1980.

R. Sibson. « A brief description of natural neighbour interpolation ». In *Interpreting Multivariate Data*. V. Barnett (Editor), pages 21-36. John Wiley, 1981.

N. Sukumar, B. Moran, and T. Belytschko. « The Natural Element Method in Solid Mechanics ». *International Journal for Numerical Methods in Engineering*, 43(5):839-887, 1998.

N. Sukumar, B. Moran, A. Yu Semenov, and V. V. Belikov. « Natural Neighbor Galerkin Methods ». *International Journal for Numerical Methods in Engineering*, 50(1):1-27, 2001.