LEILA HASHEMI BENI

DEVELOPMENT OF A 3D KINETIC DATA STRUCTURE ADAPTED FOR A 3D SPATIAL DYNAMIC FIELD SIMULATION

Thèse présentée
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de doctorat en Sciences Géomantiques
pour l'obtention du grade de Philosophiae Doctor (Ph.D.)

DÉPARTEMENT DES SCIENCES GÉOMATIQUES
FACULTÉ DE FORESTERIE ET DE GÉOMATIQUE
UNIVERSITÉ LAVAL
QUÉBEC

2009

Résumé

Les systèmes d'information géographique (SIG) sont employés couramment pour la représentation, la gestion et l'analyse des données spatiales dans un grand nombre de disciplines, notamment les sciences de la terre, l'agriculture, la sylviculture, la météorologie, l'océanographie et plusieurs autres. Plus particulièrement, les géoscientifiques utilisent de plus en plus ces outils pour l'intégration et la gestion de données dans différents types d'applications environnementales, allant de la gestion des ressources en eau à l'étude du réchauffement climatique. Au delà de ces possibilités, les géoscientifiques doivent modéliser et simuler des champs spatiaux dynamiques et 3D et intégrer aisément les résultats de simulation à d'autres informations spatiales associées afin d'avoir une meilleure compréhension de l'environnement. Cependant, les SIG demeurent extrêmement limités pour la modélisation et la simulation des champs spatiaux qui sont habituellement tridimensionnels et dynamiques. Ces limitations sont principalement reliées aux structures de données spatiales actuelles des SIG qui sont bidimensionnelles et statiques et ne sont pas conçues pour aborder le 3D et les aspects dynamiques des champs spatiaux 3D.

Par conséquent, l'objectif principal de ce travail de recherche est d'améliorer la capacité actuelle des SIG concernant la modélisation et la simulation des champs spatiaux dynamiques et 3D par le développement d'une structure de données spatiale 3D cinétique. Selon notre revue de littérature, la tetraèdrisation Delaunay dynamique 3D (DT) et sa structure duale, le diagramme Voronoi 3D (VD), ont un potentiel intéressant pour manipuler la nature tridimensionnelle et dynamique de ce genre de phénomène. Cependant, en raison de l'échantillonnage particulier des données utilisées dans les applications en géosciences, la tetraèdrisation Delaunay de telles données est souvent inadéquate pour l'intégration et la simulation numériques de champs dynamiques. Par exemple, dans une simulation hydrogéologique, les données sont réparties irrégulièrement i.e. verticalement denses et horizontalement clairsemées, ce qui peut résulter en une tessellation inadéquate dont les éléments seront soit très grands, soit très petits, soit très minces. La taille et la forme des éléments formant la tessellation ont un impact important sur l'exactitude des résultats de la simulation, ainsi que sur les coûts de calcul qui y sont reliés. Par conséquent, la première étape de notre travail de recherche est consacrée au développement d'une méthode de raffinement adaptative basée sur la structure de données Delaunay dynamique 3D et à la construction d'une tessellation 3D adaptative pour la représentation et la simulation de champs dynamiques. Cette tessellation s'ajuste à la complexité des champs, en considérant les discontinuités et les critères de forme et de taille.

Afin de traiter le comportement dynamique des champs 3D dynamiques dans SIG, nous étendons dans la deuxième étape de cette recherche le VD 3D dynamique au VD 3D cinématique pour pouvoir mettre à jour en temps réel la tessellation 3D lors des procédés de simulation dynamique. Puis, nous montrons comment une telle structure de données spatiale peut soutenir les éléments en mouvement à l'intérieur de la tessellation ainsi que leurs interactions. La structure de données cinétique proposée dans cette recherche permet de gérer de manière élégante les changements de connectivité entre les éléments en mouvement dans la tessellation. En outre, les problèmes résultant de l'utilisation d'intervalles de temps fixes, tels que les dépassements et les collisions non détectées, sont abordés en fournissant des mécanismes très flexibles permettant de détecter et contrôler différents changements (événements) dans la tessellation Delaunay 3D.

Enfin, nous étudions le potentiel de la structure de données spatiale cinétique 3D pour la simulation de champs dynamiques dans l'espace tridimensionnel. À cette fin, nous décrivons en détail les différentes étapes menant à l'adaptation de cette structure de données, de sa discrétisation pour des champs 3D continus à son intégration numérique basée sur une méthode événementielle. Nous démontrons également comment la tessellation se déplace et comment la topologie, la connectivité, et les paramètres physiques des cellules de la tessellation sont localement mis à jour suite à un événement topologique survenant dans la tessellation. Trois études de cas sont présentées dans la thèse pour la validation de la structure de données spatiale proposée, et de son potentiel pour la simulation de champs spatiaux 3D et dynamiques. Selon nos observations, pendant le procédé de simulation, la structure de données est préservée et l'information 3D spatiale est gérée adéquatement. En outre, les résultats calculés à partir des expérimentations sont très satisfaisants et sont comparables aux résultats obtenus à partir d'autres méthodes existantes, pour la simulation des mêmes champs dynamiques. En conclusion, certaines des limites de l'approche proposée liées au développement de la structure de données 3D cinétique et à son adaptation pour la représentation et la simulation d'un champ spatial 3D et dynamique sont discutées, et quelques solutions sont suggérées pour l'amélioration de l'approche proposée.

Abstract

Geographic information systems (GIS) are widely used for representation, management and analysis of spatial data in many disciplines including geosciences, agriculture, forestry, metrology and oceanography etc. In particular, geoscientists have increasingly used these tools for data integration and management purposes in many environmental applications ranging from water resources management to global warming study. Beyond these capabilities, geoscientists need to model and simulate 3D dynamic spatial fields and readily integrate those results with other relevant spatial information in order to have a better understating of the environment. However, GIS are very limited for modeling and simulation of spatial fields which are mostly three dimensional and dynamic. These limitations are mainly related to the existing GIS spatial data structures which are 2D and static and are not designed to address the 3D and dynamic aspects of continuous fields.

Hence, the main objective of this research work is to improve the current GIS capabilities for modeling and simulation of 3D spatial dynamic fields by development of a 3D kinetic data structure. Based on our literature review, 3D dynamic Delaunay tetrahedralization (DT) and its dual, 3D Voronoi diagram (VD), have many interesting potentials for handling the 3D and dynamic nature of those kind of phenomena. However, because of the special configurations of datasets in geosciences applications, the DT of such data is often inadequate for numerical integration and simulation of dynamic field. For example, in a hydrogeological simulation, the data form highly irregular set of points aligned in vertical direction and very sparse horizontally which may result in very large, small or thin tessellation elements. The size and shape of tessellation elements have an important impact on the accuracy of the results of the simulation of a field as well as the related computational costs. Therefore, in the first step of the research work, we develop an adaptive refinement method based on 3D dynamic Delaunay data structure, and construct a 3D adaptive tessellation for the representation and simulation of a dynamic field. This tessellation is conformed to represent the complexity of fields, considering the discontinuities and the shape and size criteria.

In order to deal with the dynamic behavior of 3D spatial fields in a moving framework within GIS, in the second step, we extend 3D dynamic VD to 3D kinetic VD in the sense of being capable of keeping update the 3D spatial tessellation during a dynamic simulation process. Then, we show how such a spatial data structure can support moving elements within the tessellation and their interactions. The proposed kinetic data structure provides an elegant way for the management of the connectivity changes between moving elements within the tessellation. In addition, the problems resulting from using a fixed time step,

such as overshoots and undetected collisions, are addressed by providing very flexible mechanisms to detect and manage different changes (events) in the spatial tessellation by 3D DT.

Finally, we study the potentials of the kinetic 3D spatial data structure for the simulation of a dynamic field in 3D space. For this purpose, we describe in detail different steps for the adaption of this data structure from its discretization for a 3D continuous field to its numerical integration based on an event driven method, and show how the tessellation moves and the topology, connectivity, and physical parameters of the tessellation cells are locally updated following any event in the tessellation. For the validation of the proposed spatial data structure itself and its potentials for the simulation of a dynamic field, three case studies are presented in the thesis. According to our observations during the simulation process, the data structure is maintained and the 3D spatial information is managed adequately. Furthermore, the results obtained from the experimentations are very satisfactory and are comparable with results obtained from other existing methods for the simulation of the same dynamic field. Finally, some of the limitations of the proposed approach related to the development of the 3D kinetic data structure itself and its adaptation for the representation and simulation of a 3D dynamic spatial field are discussed and some solutions are suggested for the improvement of the proposed approach.

Acknowledgments

This thesis would not have been possible without the invaluable assistance and support of many people.

I would like to express my sincere gratitude to my supervisor Dr. Mir Abolfazl Mostafavi for his guidance, support and his insights into my PhD topic. I appreciate his comments, patience and understanding during the various phases of this research work. Special thanks to my cosupervisor, Dr. Jacynthe Pouliot, for her help, advice and availability. I want to thank her for the constant support and encouragement throughout my PhD period.

I am grateful to the members of my PhD committee who have contributed valuable feedback on this thesis, namely Prof. Mike Worboys, Dr. René Therrien and Dr. Jean-Loup Robert.

I would thank to Dr. Marina L. Gavrilova (Calgary University), Dr. Majid Mohammadian (MIT), Dr. Daniel Russell (University of California), Dr. Herbert Edelsbrunner (Duke University), and Dr. James Campbell (Cranfield University) for all of the helpful comments, suggestions and advice during the development of the data structure, its adaption for the simulation purpose and the validation. Special thanks to Dr. Christopher Gold and his research team in University of Glamorgen, specially Dr. Hugo Ledoux, for providing us with the sources code for 3D Delaunay triangulation and Voronoi diagram.

This research work is a part of the GEOIDE GeoTopo3D project entitled: “Development of 3D Predictive Modeling Platform for Exploration, Assessment and Efficient Management of Mineral Petroleum and Groundwater Resources.” I acknowledge GEOIDE network for financial support of this research work. Also, I want to thank again Dr. Jacynthe Pouliot, GeoTopo3D project leader, for her vision and leadership in the project, and for her great support and advices.

I would like to thank all administrative and technical staff members of the department of Geomatics and CRG who have been kind enough to advise and help in their respective roles. Special thanks to Madam Carmen Couture, for her valuable help and kindness.

I also would like to thank all my office associated with whom I have had the pleasure to interact during my studies, especially, Omid, Karine, Daniela, Mehrdad and Eric. I deeply thank my caring friend, Madam Andrée Tremblay, for her unfailing spiritual support, and encouragement throughout my PhD period. I owe her a huge debt of gratitude.

Finally, my special thanks to my family, especially my parents for all of their encouragement and support throughout my life, I feel fortunate to have such a loving and caring family. My sincere thanks from the bottom of my heart go to my husband, Reza.

Without his wonderful love, patience and encouragement, I would not have been able to finish this thesis.

"The good life is one inspired by love and guided by knowledge."
Bertrand Russell

To my beloved parents and Reza

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 Introduction

Spatial modeling and simulation of 3D dynamic continuous phenomena are very important for various applications including protection and management of water resources, meteorological prevision, environmental modeling, urban dynamics monitoring and management. A realistic representation of a 3D dynamic phenomenon allows users and decision-makers to better understand, analyze and predict its behavior (Goodchild 2005, Rivington et al. 2007, Fedra 2006). For example, the simulation of 3D groundwater flow can adequately provide answers to the questions such as: "where does ground water come from?" or “how does it travel through a 3D complex geological system?" The reliable answers will be useful to exploit and effectively protect groundwater resources by planning for constructing or situating wells, agricultural activities, other potential groundwater contamination sources, etc. On the other hand, in the case of groundwater contamination, the simulation results may indicate “how water contaminant propagates within an aquifer” which would be valuable in monitoring and

remediation planning. The simulation of urban or wild fire is another example which helps modelers to predict the propagation of fire under various conditions and scenarios such as different climatic and environmental state (Kose et al. 2008). The results of such simulations are essential not only for decision support in case of an ongoing fire but also for planning and prevention.

In both examples, we notice that the phenomena (groundwater and fire) are continuous and multidimensional. The 3D spatial and dynamic properties of the phenomena are among their most important properties and must be considered for modeling and simulation purposes. For example, Groundwater flow travels within a 3D continuous space composed of geological objects such as fractures, faults, and layers- and its dynamic behavior (the direction and speed of flow) depends on the physical properties of these objects such as porosity and permeability. Also, the fire propagates in a 3D space through 3D complex spatial objects such as buildings in urban areas and its dynamic behavior depends on the wind around the 3D buildings. Therefore, for the realistic modeling and simulation of such dynamic fields, there is a strong need for managing their spatial and dynamic data in 3D.

Most of the efforts put into the modeling and simulation of 3D dynamic fields are done by computer-based mathematical models called, in this thesis, simulation tools. MODFLOW (USGS) and POM (Princeton Ocean Model) are two examples of tools for hydrogeological and ocean modeling (Bekey et al. 2001). While, these simulation tools have been very efficient for the description of the dynamic behavior of the fields, but they generally suffer from insufficient spatial analytical components and are not convenient for the representation of the spatial properties of the fields (Maguire et al. 2005). However, as previously mentioned, dynamic fields have inherently spatially distributed properties and their spatial representation is critical. In addition, these tools are handicapped at considering and integration of other spatial data of the simulation domain such as digital maps, satellite images, and aerial photos. For instance, as illustrated in figure 1-1, the simulation of dispersion of smoke in an urban area in the spatial context presents more realistic

results in comparison with the results of the simulation done in the absence of other spatial information of the city such as buildings.

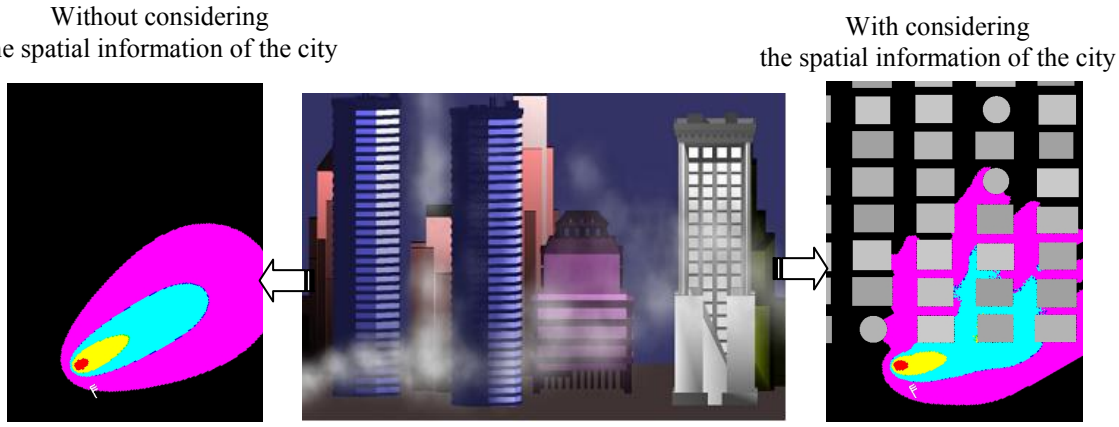


Figure 1-1 Smoke dispersion simulation for urban security[1].

Geographic information system (GIS) can provide simulation tools with strong computing platforms for data management, visualization, querying, and analysis (Burrough et al. 1988, Goodchild 1992, Maguire et al. 2005). In particular, GIS is an effective tool to integrate heterogeneous and non-spatial information from different sources (Flowerdew 1991, Maguire 2005). However, dynamic and three-dimensional complexity of fields goes well beyond the basic analytical capabilities of traditional GIS systems (Fedra 2006). Hence, the integration of simulation tools and GIS has been considered by many researchers in order to reduce the stated limitations and make them more robust by their linkage and co-evolution (Bivand and Lucas 2000, Chapman and Thornes 2003, Fotheringham and Wegener 2000, Sui and Maggio 1999, Valavanis 2002).

An overview of the existing approaches reveals that there are three main approaches to couple GIS and simulation tools, namely loose coupling, tight coupling and full coupling, which have raised several conceptual and technical questions (Gimblett 2002):

[1] Ref.:http://www.cs.sunysb.edu/~vislab/projects/urbansecurity/Gallery.html

A *Loose coupling* approach is defined as data exchange between GIS and the simulation tool (fig. 1-2). In this level of integration, a simulation is done independently of GIS. However, the input into the simulation tool is done through a GIS data processing and the output of the tool is exported back to GIS for analysis and visualization purposes. Thus, standard file formats such as ASCII or binary data format must be used for file transfer. These approaches sometimes use some GIS functions such as interpolation, overlay, slope calculation and proximity buffer to analyze and produce inputs for the simulation tool. The advantage of this approach is that redundant programming can be avoided and also simulation tools can be linked with different GIS (portability). However, data conversion between different packages can be tedious, time consuming and error prone (Sui and Maggio 1999), especially, when simulations need to be run for different scenarios. Examples of this type of coupling are the linking of AERMOD (air dispersion model) to ArcGIS (Maantay et al. 2009), SIMWE (erosion and deposition) to GRASS (Mitas and Mitasova 1998), DHSVM (Distributed Hydrology Soil Vegetation Model) to Arc/Info, and GLEAMS (Groundwater Loading Effects of Agricultural Management System) to Arc/Info (Stallings et al. 1992).

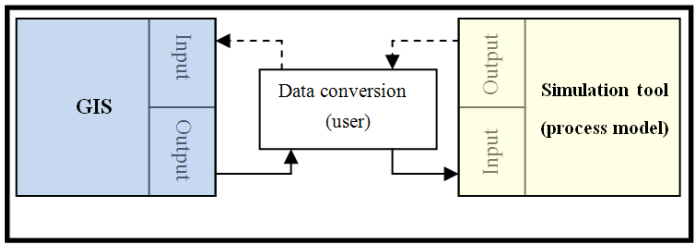


Figure 1-2 Loose coupling of a simulation tool and GIS.

A *Tight coupling* approach is defined as linking simulation tools to GIS via either GIS macro or script programming which is hidden from user (fig.1-3). In this approach, similar to the loose approach, the simulation is developed outside of the GIS, but with this difference that the data is exchanged between those automatically (Burrough 1996). The advantage of a tight coupling approach is the use of the interface of the GIS which increases the interactivity of the system. However,

complex programming for data exchange and data management are required in this level of integration. Furthermore, interoperability is also restricted i.e. the simulation tool cannot be linked with different GIS (Mitasova and Mitas 1998). Coupling of SWAT (Soil and Water Assessment Tool) (Di Luzio et al. 2004), MIKE SHE (hydrological modeling system) (Zhang et al. 2008) and HEC-HMS (Hydrologic Model System) to ArcView (Obenour 2002, USACE 2003) are examples of these approaches.

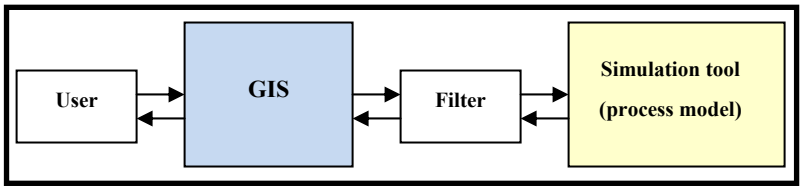


Figure 1-3 Tight coupling of a simulation tool and GIS.

A *Full coupling* approach is defined as a complete embedding of the simulation process within a GIS (fig. 1-4). This approach involves implementing a simulation model within the GIS and taking full advantage of the built-in GIS functionalities. Moreover, users are able to make on-the-fly modifications and perform different `what if?' scenarios to better understand and analysis the field behavior. This allows a GIS to go beyond being a simple data management tool and offer more sophisticated analyses and simulation capabilities for the representation of spatial dynamic phenomena (Sui and Maggio, 1999).

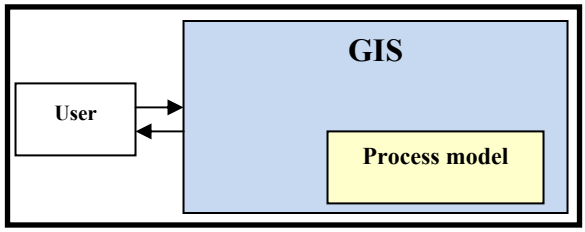


Figure 1-4 Full coupling of a simulation model and GIS.

A few efforts to full coupling such as LISFLOOD have been reported which has been written using the PCRaster GIS to simulate river discharge in a drainage basin

as a function of spatial data on topography, soils and land cover (Van Der Knijff and De Roo 2008). PCRaster models a process as a sequence of snapshot raster layers and includes a range of functions for spatio-temporal environmental modeling in 2D (or 2.5D) (Karssenberg and de Jong 2002). However, this type of integration requires much more effort to extend GIS functions which can support 3D spatio-temporal data. This research work is an attempt for this purpose.

1.2 Requirements for simulation of 3D dynamic phenomena within GIS

Simulation of dynamic phenomena within GIS generally consists of two fundamental elements: 3D spatial modeling of phenomena and process models (Mostafavi 2002, Brown et al. 2005). Spatial modeling is related to the modeling of the spatial aspects of spatial entities such as shape, size, location and their spatial relationships while, process models describe the dynamic behavior of the phenomena in 3D space (Brown et al. 2005, Goodchild 2005).

Real-world phenomena exist in 3D environment, thus we require an appropriate GIS for modeling, representing and storing data associated with these phenomena in a 3D environment. Regarding the measurement techniques, such as seismic profiling, weather balloons, probes, systematic survey and borehole drilling, almost all data are 3D points with irregular distribution. Each data point is defined by its geometric position in 3D space (x, y, z) and can have one or more attributes attached to it. To represent a continuous field from these discrete samples, it is necessary to tessellate the field into finite elements. The elements, having some attributes such as mass or pressure, must represent the spatial information and physical properties of the field. For this purpose, the elements of the tessellation must have some specifications. First, they must be 3-dimensional and well-shaped to provide conformity to the outline of the data points or in other word complex geometries. The size and shape of the elements affect the accuracy of a simulation

as well as the cost of the computation (Mansell et al., 2002). Therefore, for specific regions with either high variations of the field properties or discontinuities, a tessellation with higher resolution is needed. For example, a hydrogeological simulation requires a fine tessellation in areas where the hydraulic gradient is high such as in the vicinity of a pumping or injection wells. Second, the tessellation must be flexible to modifications and editing (Shewchuk 1997). This ability allows the tessellations to be dynamic and to be modified locally and consequently offers the user the ability to interact with the environment and to extract information. Finally, the tessellation must represent the dynamic behavior of the field during a given period of time (Soni 2000). For instance, in the case of flood simulation, the water land boundary is an important feature which is irregular and constantly changing as the flood progresses. Tidal flows in coastal regions, free-surface tracking in a hydrogeological system and forest fire monitoring are other application examples in which the spatial information (geometry and topology) of the fields and their attributes may change over time.

As mentioned before, the dynamic behavior of a field is usually described by a set of partial differential equations (PDEs). The PDEs are normally solved either on a fixed tessellation or on a moving tessellation. In the former, the dynamic behavior of the field is modeled as the variation of the physical properties of each element of a fixed tessellation over time (Eulerian approaches). In the latter, the dynamic behavior of the field is modeled as changes in the position, shape, or properties of each element of a kinetic (moving) tessellation (Lagrangian approaches) (Mostafavi 2002, Gold 2006, Brimicombe 2003, Hashemi et al. 2008). This type of simulation is of great interest in a wide variety of applications such as the tracking of pollution plumes in the ocean or dispersion models in meteorology. Considering these requirements for GIS-based simulation, the central question is: “do the current GISs have the capability to adequately represent the three dimensional and dynamic aspects of a dynamic field?”

1.3 Problems

Simulation of a 3D dynamic field is particularly difficult using GIS, since GIS are not well designed for handling either time or the interaction between the continuously changing elements of a tessellation describing the dynamic behavior of fields. Moreover, the role of GISs remains primarily in the area of 2D data processing, analysis, visualization and management. It is impossible to include a process model directly within GISs and 3D simulation results must often be sliced into several 2D datasets for visualization and analysis. This problem originates from the fact that current GISs data structures are 2D and static and are not designed to address the three-dimensionality nor the dynamic nature of 3D dynamic fields. Therefore, the specific problems that will be considered in this research work is the lack of proper spatial data models and data structures in current GIS to represent 3D spatial fields and their changes over time and space:

a) Problem with the representation of 3D spatial fields

Fields are continuous while their data are usually sampled by irregular 3D unconnected points (Davis and Davis 1988, Gold et al. 2004, Ledoux and Gold 2008). To represent the continuous field from these discrete samples (objects), a 3D spatial data structure is required to ideally be able to represent 3D objects and fields- two various ways of discretizing space. However, integrating field and object representation is still challenging in the context of GIS data structures. Raster data structures discretize the field region into a regular tessellation having field-like properties where the topology between the elements (not between the objects) is implicit in the data structure. Although raster data structure is best suited for continuous field representation (because raster elements cover the whole space), they are not computationally efficient to represent the variations of the heterogeneous fields over the 3D space. Vector data structures, such as triangular irregular network (3D TIN), can discretize the field region into irregular tessellation which is adapted to the special configuration of points, consequently, it is more flexible to represent the complexity of a field. However, the topology between 3D

elements needs to be computed and stored explicitly while 3D topology construction can be a difficult task for 3D GIS which are still in their infancy (Ellul and Haklay 2006, Zlatanova et al. 2002). Another important problem of vector models is that the variation of the field properties during the simulation process is undefined across the edges or elements of this model (Goodchild 2008). Considering the mentioned limitations of current raster and vector data structures, the key to 3D spatial representation of heterogeneous fields is a 3D adaptive tessellation which has both raster and vector properties.

b) Problem with the representation of the dynamic behavior of fields

Time discretization- in a dynamic field, the properties values of the field change over space for each moment in time, then, the data associated with such a field becomes function of location and time. Atmospheric pressure and temperature or soil moisture are examples of dynamic fields in which their data may change over time and space. For simulation purposes, modeling of such spatio-temporal data within GIS is critical. Most research efforts on spatio-temporal modeling try to extend current GIS spatial modeling by simply adding temporal information in the database (Yuan 2007). These approaches represent temporal changes of fields based on a temporal series of spatially registered snap-shots which is not an efficient way for handling the continuous dynamics behavior of fields over time (Brown et al. 2005). Indeed, an improved method based on event-driven approach is necessary to describe the spatio-temporal variations of a field or in better words, the dynamics and interactions of the elements of the tessellation in a 3D space. This means that all interesting changes in a field spatial model or its property value must be detected and stored as a sequence of events and time is discretized based on these events.

Topology management- In a dynamic field simulation in a dynamic system (e.g. free-Lagrangian approaches) (Goodchild et al. 2007), the position, shape and connectivity of the tessellation elements change with respect to space and time and we need a 3D “kinetic” spatial data structure to maintain a valid 3D spatial

tessellation during the simulation process. Current raster and vector data structures are not adequate for this purpose. The problem with raster data structure is related to the representation and management of moving elements because the raster grid is static by nature and cannot move or be adaptive to represent the behavior of a dynamic field. In the vector data structure, topological relations need to be updated continuously which is a complex task. Since it is not easy to define the region affected by a single change and thus, a batch operation is used for topology construction. Therefore, any change in the relation between elements requires a global reconstruction of topology which is not an efficient approach.

1.4 Objectives

The main objective of this research work is to improve GIS capabilities for the modeling and simulation of 3D spatial dynamic and continuous phenomena in a dynamic system. To meet the requirements of this type of simulation, it is essential to solve the problems previously discussed which lead us to identify three specific objectives:

a) ***Constructing an adaptive tessellation for fields modeling in a three dimensional space by developing an adaptive refinement method based on 3D dynamic Voronoi data structure***

Based on a literature review, the dynamic Voronoi diagram (Gold 1991) has several interesting properties that make it a good candidate for our purpose. The Voronoi diagram is defined by a region of influence for each sampled point in d-dimensional spaces. Then, Voronoi diagram provides an adequate tessellation for representation of a continuous space (see Aurenhammer 1991, Okabe 2001, Mostafavi 2002 and Ledoux and Gold 2008). Voronoi polyhedra can be defined by points with an arbitrary distribution, creating elements of different sizes and shapes which can fill any arbitrary surface or volume and can thus adapt to complex variation of field properties. Each Voronoi polyhedron can have an arbitrary number of neighbors

which their connectivity with the given element is clearly defined and can be explicitly retrieved if needed. There has also been work in 2D (such as Mostafavi 2002) and 3D (such as Ledeoux 2006) that allow these spatial data structures to be "dynamic". A dynamic Voronoi diagram offers the local editing and manipulating possibility of the tessellation after adding/deleting a data point within the spatial tessellation. The first objective of my research work is therefore taking into account this elegant capability to develop an interactive and adaptive method for the refining of the tessellation without having to rebuild the whole tessellation. This method allows users to modify and adjust the density of the tessellation elements locally, i.e. on-the-fly, producing small elements in areas with a higher variation of field, and to reduce elements density in smooth regions.

b) Handling a moving tessellation in a 3D dynamic environment by developing a 3D kinetic Voronoi data structure

To simulate and represent dynamic fields, the second objective is to extend dynamic 3D Voronoi data structure to "kinetic" 3D data structure in the sense of being capable of keeping up to date the 3D spatial tessellation during a dynamic simulation process. The development of this 3D kinetic spatial data structure is based on previous results in 2D and on theoretical research work in 3D. Kinetic VD data structures in 2D have been practically shown to elegantly solve many problems associated with traditional GISs data structures for time and topology management in 2D (Mostafavi 2002). The problem of fixed time steps (snapshot methods) is addressed by providing very flexible mechanisms to detect and manage different changes (events) in the spatial tessellation. It also provides an elegant way for the local on-the-fly updating management of the connectivity changes between moving elements within the tessellation. For this reason, the second objective is to bring the same benefits for three-dimensional space. For these purposes, the development of the necessary operations, metric and topological, is required.

c) Evaluating the feasibility of the application of the three dimensional kinetic Voronoi diagram for a dynamic field simulation

Two- and three dimensional VD have been shown to provide adequate discretization of the space for the simulation of dynamic fields such as fluid flow in a dynamic system, ensuring that physically realistic results are obtained (Fritts et al. 1985, Campbell and Shashkov 2003, Ghemyr et al. 1997). However, since these methods use a fixed time-step method during the simulation process, the whole topological relations must be reconstructed after each time-step which considerably reduces the efficiency of the methods. To solve this problem, the use of a kinetic VD as underlying tessellation has been proposed (Gold and Condal 1995, Gavrilova 1998, Mostafavi and Gold 2004). Therefore, the tessellation is allowed to move during the simulation and the connectivity of the tessellation elements as well as the field properties are updated based on local adaptive time-steps. Based on this proposition, Mostafavi and Gold (2004) adapted a 2D kinetic VD for the numerical integration of dynamic continuous fields described by a set of PDEs for the simulation of dynamic fields and validated the application of 2D kinetic VD for the tidal simulation on the globe. Their results confirmed the advantages of the 2D kinetic VD for simulation of the dynamic field. On this basis, the third objective of the research work is the study of the potentials of the kinetic 3D spatial data structure for the simulation of a dynamic field in 3D space. For this purpose, the 3D kinetic data structure will be adapted for the numerical integration of dynamic continuous models (PDEs) of fields in 3D such as 3D hydrodynamic flow and investigated for representing the field evolution over 3D space and time.

By defining these objectives, this research work aims to persuade simulation specialists that GIS is not only a good tool for spatial data integration and management but also can be a powerful simulation tool.

1.5 Methodology

This research work is a part of a GEOIDE[2] project called GeoTopo3D[3] that proposes the development of a 3D predictive modeling platform for exploration, assessment and efficient management of mineral, petroleum and groundwater resources. GeoTopo3D is organized around three topics, presented at figure 1-5 (Kirkwood et al 2003):

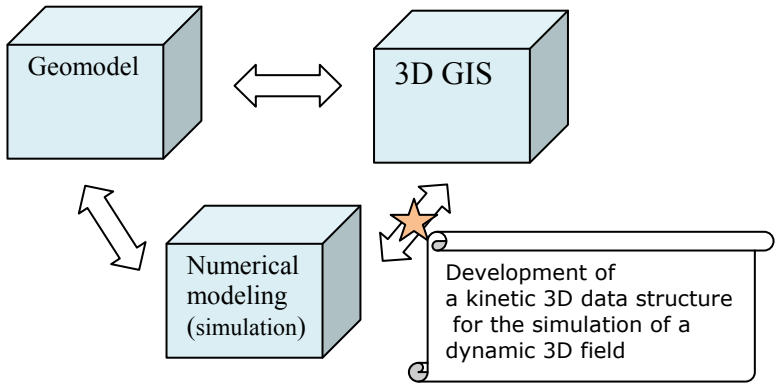


Figure 1-5 Situation of this research work within the GeoTopo3D Project.

- ***Geomodels*:** To develop a workflow for the construction of Geomodels by integrating 3D geosciences modeling approaches
- ***Numerical modeling*:** To develop a workflow for numerical modeling of flow and mass transport properties in fractured media
- ***3D GIS*:** To explore 3D GIS and develop a topological data structure for the construction, management and spatial query support for decision makers that plan and manage natural resources,

The present research work will contribute to link the numerical modeling and the development of 3D GIS components of the GeoTopo3D project. We develop a prototype which deals with a 3D kinetic data structure to simulate 3D dynamic

[2] GEOID (GEOmatics for Informed Decision) :http://www.geoide.ulaval.ca
[3] http://geotopo3d.scg.ulaval.ca

field. The prototype is implemented in the Borland Delphi environment (visual Pascal). Delphi is an object oriented programming language with the possibility to develop a visual interface. The implementation of the prototype is made on the top of a source code of 3D DT and VD construction was provided by Prof. Gold's team (Ledoux 2006) at the University of Glamorgen, England.

To achieve the overall objectives of the research work as well as the development, implement and test of the prototype, the methodology is based on four phases (fig. 1-6):

Phase 1. Literature review

This phase satisfies the theorical requirements of the research work by a review of the practices, problems, and prospects for a GIS-based simulation of dynamic continuous phenomena. This review included:

- Requirements of applications in geosciences dealing with 3D dynamic fields
- Capabilities of the current spatial data models and spatial data structure to manage the spatial information (geometry and topology) and the properties of 3D dynamic phenomena and their changes over 3D space and time
- Numerical simulation methods in geosciences (for example, for the simulation of hydrodynamic flow such as groundwater or gas)
- Voronoi diagram and its properties in context of simulation and related data structure to storage and manipulate the 3D field data
- Computational geometry for the data structure development

A literature review in these topics was required to show what has been studied for the spatial modeling and simulation of 3D dynamic fields in general and within GIS in particular. An intensive research work on these areas allowed us to better identify the weaknesses of the existing methods for this purpose and to define the research project problems, objectives and methodology.

Phase 2. Construction of an adaptive 3D tessellation based on dynamic 3D Voronoi diagram

1. *To choose a suitable tessellation construction algorithm:* Several algorithms have been introduced for constructing a 3D tessellation based on VD for simulation purposes. We should choose and develop an algorithm that satisfies our objectives e.g. it must be 3D and dynamic. Concerning the properties of existing algorithms, it seems that the 3D incremental method is a good candidate.

2. *To construct the 3D tessellation:* in an incremental method it is not necessary to know the whole data set prior to tessellation construction, and using many dynamic operations (delete, insert, flip) on each time a new object will be inserted or removed in the existing tessellation. On the other word, local modification is allowed after any change. These dynamic operations are discussed in several papers (Ledoux 2006; Miller and Pav 2002). I will use the source code for 3D Voronoi diagram provided by Prof. Gold's team at University of Glamorgen, England (Please see Ledoux 2006).

3. *To store and manipulate the 3D tessellation:* Spatial proximity information is critical for many applications. Maintaining and intensive exploitation of such relations are very important in dynamic simulation context where the interaction between the tessellation elements must be modeled. There are several data structures that can be used for the storage and the navigation through a 3D tessellation. Among these data structures Tetrahedron-based data structure is chosen due to its simple nature and efficiency.

4. *To implement an interactive and adaptive refinement method*: The 3D tessellation generation algorithm discussed above does not prevent the creation of elements that are not adequate for simulation. Several methods have been developed to refine these poor quality elements. However, most of these refinement methods deal exclusively with theoretical considerations

for geometry and rarely deal with physical properties of fields in geosciences. In this step, an algorithm for the refinement and the manipulation of 3D geosciences datasets (for example for the data of a hydrogeological system) is developed and implemented.

Phase 3. Development of a 3D kinetic data structure

As it was mentioned previously, the kinetic Voronoi data structure allows several elements to move simultaneously in the tessellation with the capacity for local topology updating.

1. *To manage the movement of an element within the 3D tessellation (one moving sampled point):* In order to extend a static Voronoi diagram to a kinetic Voronoi diagram, first, the moving of one element (or its generator) from its initial position toward a given destination is managed. For this purpose, a local and on-the-fly topological updating method will be developed for the management of the connectivity changes between the moving element and their neighbors within the 3D tessellation. In addition, the problem of fixed time steps will be addressed by providing a flexible mechanism to detect and manage different changes (events) in the spatial tessellation.

2. *To manage the simultaneous movement of several elements within the 3D tessellation (several moving sampled points)*: In the context of a dynamic field representation and simulation, it needs to manage a large number of topological changes in a deforming kinetic spatial tessellation in order to preserve the validity of the 3D tessellation. The sequence of the management of these changes has an important impact on the simulation results. This will be done using a priority queue data structure, where the moving elements are organized with respect to their priority.

3. *To verify and validate the data structure:* The algorithm will be executed with various conditions e.g. the size of sample, different velocity, etc**.** the experimentation will allow us to identify the possible problems and complexities in the proposed approach.

Phase4. Feasibility study of 3D Voronoi diagram in the context of simulation of dynamic continuous phenomena

In this phase, the kinetic 3D Voronoi diagram is applied for the simulation of a dynamic field to ensure that it can appropriately represent the dynamic behavior of the field in 3D.

1. *To select the governing equation (PDEs):* In this step the equations that describe the dynamic behavior of a field must be selected. The equations are continuous, thus, a discrete form of equations is required for the numerical simulation.

2. *To solve the governing equation numerically using the kinetic 3D Voronoi diagram:* in this step, the field is spatially discretized by a tessellation based on 3D Voronoi diagram. Therefore, the adjacency relationship (Topology) will be readily defined among the elements. Also we will consider an event driven method for discretizing time. To simulate fluid flow, the discrete governing equation assigns a value for the properties of the field (such as pressure, density and 3D velocity vector in a hydrodynamic simulation) to each point (generator in VD, vertex in DT) which will be updated after any topological event. In fact, these properties values may change because of the interaction between the points (particles), hence points will move in the resultant force direction.

3. *To validate the simulation results:* In order to test the results of this research work, I will consider the visual match or numerical comparison between the simulation results with the one obtained from the analytical and numerical

solutions of the same governing equations. Satisfactory results will indicate the objectives were realized.

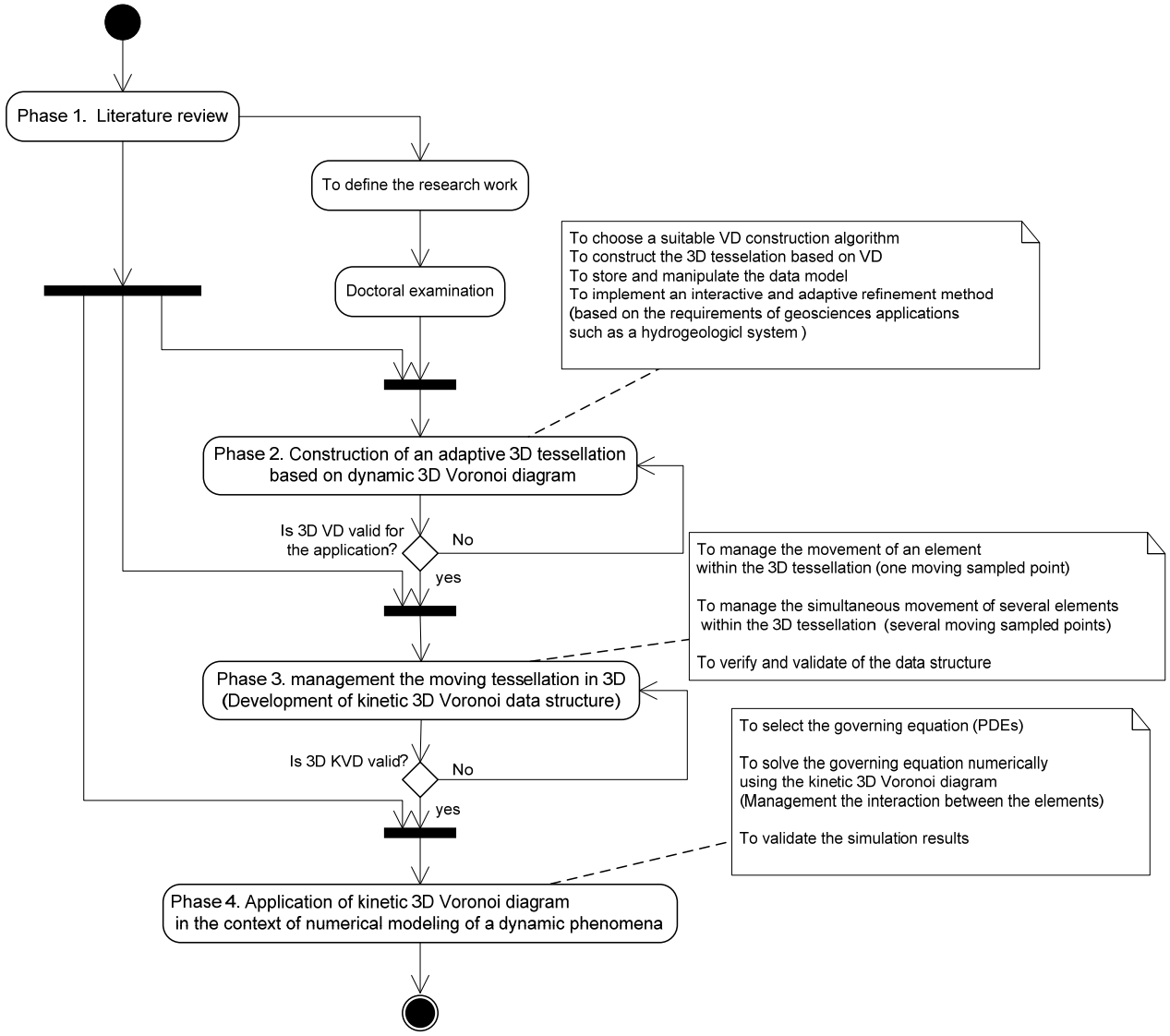


Figure 1-6 The methodology of the research work.

1.6 Thesis organization

This thesis consists of seven chapters and is organized as follows:

- **Chapter 1** discusses the motivation for the simulation and modeling of 3D dynamic fields in GIS. The problems, objectives and methodology of the research work are introduced.

- **Chapter 2** provides an overview of the modeling and simulation of dynamic phenomena in GIS. The different steps from conceptual models of the reality to its representation are explained and the existing methods are reviewed.

- **Chapter 3** introduces Voronoi and Delaunay tessellations. First some formal definitions and interesting properties of these data structures are presented, followed by their construction and storage in 3D.

- **Chapter 4** discusses a refinement method for the construction of a tessellation conforming to the complexity of fields. This method defines, detects and modifies the poor-quality elements of the tessellation that are not suitable for numerical modeling purposes.

- **Chapter 5** describes the development and implementation of 3D kinetic DT and VD data structures for the simulation purposes. It also explains how the data structure allows the establishment, analysis and local update of the topological relations between the moving elements in a 3D tessellation.

- **Chapter 6** deals with different steps to adapt the proposed data structure for simulation of 3D dynamic fields. The data structure is applied to the simulation of three case studies of 3D fluid flow and it is investigated how the data structures can represent the evolution of field-like properties over 3D space and time.

- **Chapter 7** presents the conclusions of the research work and recommendations for further research.

Chapter 2: Simulation of Spatial Dynamic Phenomena within GIS

2.1 Introduction

In this chapter, we discuss the simulation and modeling of real-world phenomena in GIS. Real-world phenomena are usually very complex and change continuously over space and time and it is impossible to adequately model and represent all aspects of such phenomena. Therefore, some simplifications or approximations are taken into account for modeling purpose. Peuquet (1988) has identified three levels of abstraction for the real world representation (figure 2-1): conceptual representation, functionally-oriented representation and implementational format that respectively refer to as conceptual models, data models and data structures in Burrough (1996) and Kemp (1997). Conceptual models evolve an understanding of the phenomenon being studied and extract its important characteristics for modeling purposes. Data models are the formalization of the conceptual models without any conventions or restrictions of implementation. In fact, data models are defined as a set of entities and their relationship used to represent the complexity of the reality

(Goodchild 1992). Finally, data structures provide the information that the computer requires to reconstruct the data model in digital form.

Figure 2-1 Three levels of abstraction for the real world representation.

We will discuss these three levels of modeling within a GIS where the spatial component of the phenomena is critical and then, briefly review the literature to understand why GIS are limited for simulation purposes.

2.2 Conceptual models

A conceptual model deals with the abstraction of reality into a representation scheme (Burrough 1996). With a conceptual model, modelers develop an understanding of the reality being studied and identify its salient features from the background of infinite complexity in nature (Albrecht 1996). GIS are used for

spatial modeling and representation of the real world, which continuously change over space and time (Peuquet 1984). Therefore, the spatial and dynamic aspects of a spatial phenomenon are among the most important specifications and must be considered in the spatial modeling and representation of phenomena.

2.2.1 Conceptual modeling of spatial phenomena

In the GIS literature, there are traditionally two distinct ways of conceptualizing spatial phenomena either as discrete objects or as continuous fields (Couclelis 1992, Goodchild 1992, Burrough and McDonnell 1998, Worboys and Duckham 2004). The former perceives the space as an empty domain except where it is littered with discrete, countable objects having well-defined boundary. The latter describes the space by a set of locations with properties, thus, the space is considered as an absolute framework within which attributes are measured. The object-based point of view provide answers to questions of the form "where is such-and-such an object?" while fields answer to such questions of the form "what is the phenomena like at a such-and-such a location?" (Galton 2004).

In addition to these fundamental points of view, defining the entities that are included in the models, some research works have demonstrated that fields and objects can coexist and thus tried to integrate them within a single framework (Worboys 1995). Camara et al. (2000) presented a unified conceptual frame work by a general formal definition of *geographical objects*, and showed that both types of field and object data can be expressed as particular cases of this definition. They defined a *geographical object* as a triple (*S, A, f*) where *S* and *A* are a subset of the space and attribute domain and *f* is a function of *S* and *A*. In this framework a discrete object is defined as a simple and homogeneous *geographical* object and a continuous field is defined as a simple and non-homogenous *geographical* object.

Goodchild et *al.* (2007) integrate the continuous field and discrete object concepts by introducing the concept of geo-atom. A geo-atom is defined as a tuple <*x, z, Z(x)*> where *x* and *z* define a point in space-time and a property, respectively and

Z(x) defines the particular value of the property at the point. Therefore, in this framework, the discrete object is represented as a set of atomic statements about the point itself *(0D)*, the points along the centerline *(1D)* or the points filing the polygon *(2D)* or volume *(3D)*. Continuous fields are defined as the variation of one or more properties *(Z(x))* over a domain of space-time. However, in this framework, there are an infinite number of geo-atoms that can be stored in aggregate form, based on exact or approximate uniformity or other simple rules. A *geo-field* is defined as an aggregation of geo-atoms over space by property *z*. The property *z* can be either in interval and ordinal scales such as temperature or elevation (in DTM) or in nominal scale such as owner, soil type, etc. A *geo-object* is defined as an aggregation of points in space-time whose geo-atoms meet certain requirement, such as having specified values for certain properties. Geo-objects represent geo-atoms aggregated by the values of properties using specified rules rather than by property alone as in the case of geo-fields. However, the aggregation process for objects identifying is not straightforward as in geo-fields where the geometry of geo-objects may be defined by arcs or curves rather than straight lines and even they may contain the holes and enclaves. Thus, most methods consider the location and associated geometries and then their attribute. In this conceptual frame work, geo-fields and geo-objects are interchangeable. For example by a density estimation method, it is possible to create a continuous field of density from a collection of objects and discrete objects can be extracted from continuous fields by object extraction methods.

Yuan (2001) and Goodchild et al. (2007) argued that some phenomena may have both object and field properties and defined the *field object* concept where phenomena are considered as discrete objects with internal heterogeneity conceptualized as fields. For instant, ground water may be bounded by water-front lines, but within these boundaries, there is the variation of field-like properties including pressure and density of the groundwater. Several researchers believe that the field view is more fundamental than the object view, which usually is high level of abstractions (Frank 2001). On this basis, Cova and Goodchild (2002) defined an object as a continuous field in which locations are mapped to spatial objects (*object*

field) and Liu et al. (2008) introduced the General Field (G-Field) model for *object fields* and conventional fields.

All these conceptual models, perceiving a spatial phenomenon as an *object*, *field*, *field object* or *object field*, provide different ways to deal with the spatial data and their operations to extract information.

2.2.2 Conceptual modeling of the dynamic aspect of spatial phenomena

Spatial phenomena are mostly dynamic and change continuously over time. Here again, the object/field distinction provides a convenient framework for the discussion of processes of dynamic phenomena. This is because some processes are conceptualized as interactions between objects (discrete process models), while others are conceptualized entirely within field domain (continuous process models) (Goodchild 2008). The dynamic behavior of objects is usually modeled through the object-to-object forces such as the behaviors that are presented by humans or animals as they interact over space. The dynamic behavior of fields such as atmospheric pressure, soil moisture and ground elevation is usually described by variables that are continuous functions of location and time.

To represent the dynamic aspect of a spatial phenomenon, Lagran (1992) and Goodchild et al. (2007) considered time as a single extra axis added to a Cartesian locational frame. For example, in 2D space, the concept of spatio-temporality is a three dimensional cube that represents the process of two-dimensional space along a third temporal dimension. Similarly, in 3D space, time is viewed as an additional axis in a 4-Dimensional Cartesian system, that is x, y, z and t, with t representing time in a hypercube coordinate volume space.

In these approaches, 3D fields and objects are thus defined in four dimensions. The dynamic behavior of fields can be captured by a mapping from temporal location to values as a series of snapshots. As Ganon (2004) state: “a snapshot may be thought of as composed of the values of a set of temporal fields at a given temporal

location". The dynamic behavior of an object is captured not by simply a change in its attributes values as exist in field, but by considering three different conditions that may vary for a dynamic object (figure2.2)(Goodchild et al. 2007):

- *Geometry condition:* some objects such as vehicles may retain their geometry (shape) over time and other objects such as clouds may change it.
- *Movement condition*: the objects may move (oil spills) or may be static (building).
- *Internal structure condition:* some objects (field object) have internal variation that may change over time (soil in a watershed) and some are homogenous (a street).

As figure 2-2 shows, these three conditions make eight combinations and table 2-1 provides examples for each combination.

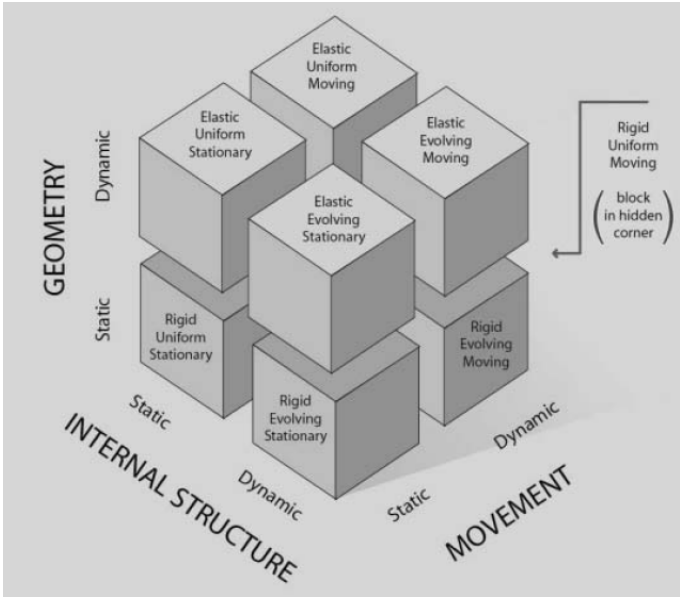


Figure 2-2 Three fundamental dimensions of object dynamics (Goodchild et al. 2007).

Regarding these examples (table 2.1), it can be seen that the dynamic behavior of spatial phenomena can be conceptualized as the change of their geometry, location and attributes over space and time. This helps to explain different ways to deal with spatial data that continuously changes over space and time.

Table 2-1 Examples of different types of objects regarding their dynamics (Goodchild et al. 2007).

Object type	***example***
Rigid-Stationary- Uniform	Buildings in a city
Elastic-Stationary- Uniform	The seasonal expansion/contraction of a lake (when the extend lake is considered)
Rigid-Moving- Uniform	Moving vehicles
Elastic-Moving- Uniform	A spreading wildfire (when the burn scar is considered)
Rigid-Stationary- Evolving	Soils in watershed
Elastic-Stationary- Evolving	Vegetation cover during desertification
Rigid-Moving- Evolving	Changing landscapes on moving , rigid tectonic plates
Elastic-Moving- Evolving	Oil spills

Up to now, we discussed the conceptual views of a dynamic spatial phenomenon such as dynamic continuous fields and discrete objects. However, these models are different ways in which we think about dynamic spatial phenomena and they are not designed to deal with the digital representation of computers. We now turn to data models which aim to formulate these conceptual models to solve of the digital representation problem.

2.3 Data models

Data models are defined as a set of entities and relationships that describe and represent the selected aspects of spatial phenomena (Goodchild 1992, Longley et al. 2005). Thus, data models deal with the representation of continuous fields or discrete objects, and their dynamic behavior within GIS. However, it should be noted that the dynamic behavior of a spatial phenomenon is usually formulated by process models. For example, the dynamic behavior of objects can be described by gravity or spatial interaction models such as agent-based models for travel behavior or migration modeling. While the dynamic behavior of fields is usually described by partial differential equations (PDEs) such as the Darcy flow equation defining the behavior of groundwater. Therefore, in addition to the spatial components,

spatial phenomena include always a strong temporal component that should be properly modeled and represented in a GIS.

Here, first, the existing "spatial data models" are studied for modeling and representing the continuous field and discrete objects without considering their dynamic behavior (independent of time), and then it is investigated how the spatial data models are currently associated with the temporal components, i.e. "spatio-temporal data models" and address the representation of dynamic behavior of phenomena.

2.3.1 Spatial data models

Raster and vector data models are two fundamental models for representing both conceptual models mentioned above, i.e. fields and objects (Longley et al. 2005). Raster data models divide a spatial phenomenon into arrays of cells and assign attributes to these cells. Vector data models represent the spatial phenomenon by different primitives including points, lines, polygons and polyhedra where the point is the basic building block and other geometric primitives (lines, polygons and polyhedra) are constructed by connecting a set of points. For example, points, lines, polygons and polyhedra can be used to represent individual cars, roads, lakes and buildings respectively. In practice, there is a strong association between raster data models and fields, and between vector data models and objects.

2.3.1.1 Field-based data models

A spatial continuous phenomenon can be represented as a set of locations with properties, which is referred to as a field in GIS (Worboys and Duckham, 2004). Formally, a field (F) is defined as any single-valued function of location in the space: $F= f(x,y)$ or $F=\mathrm{f}(x,y,z)$ in 2D and 3D space respectively. Topographic elevation, air temperature and water pollution are examples of fields. In a spatial field, we can associate a *scalar*, *vector* or *tensor* to every location in the space which refer to as a *scalar field*, vector *field* and *tensor field*. The temperature

distribution throughout space and the atmospheric pressure are examples of a scalar field. A vector field is used to indicate the magnitude and direction of fluid flow throughout space. Analysis of stress and strain requires a tensor field and is increasingly used in many applications such as fluid dynamics, meteorology and earth sciences. The values of a field can be measured in different scales namely nominal, ordinal, interval, ratio (see Longley et al. 2005, Worboys and Duckham 2004). Nominal measurement consists of assigning the locations to specific classes or categories such as name of countries, houses, cars, colors. Ordinal measurements are ordered in such a way that higher numbers represent higher values for example the classification of agriculture land based on its soil quality in which numbers 1 to 5 shows the poor to the best quality. Interval measurements are ordered in such a way that higher numbers represent higher values and differences between measurements make sense. A good example of an interval scale is the Fahrenheit scale for temperature. Equal differences on this scale represent equal differences in temperature, but a temperature of 30 degrees is not twice as warm as one of 15 degrees. Ratio scales are like interval scales except they have true zero points. Therefore the ratios between values make sense. A good example is the Kelvin scale of temperature, since a temperature of 300 Kelvin is twice as high as a temperature of 150 Kelvin.

Fields are continuous and it is almost impossible to measure them everywhere. Hence, in practice, fields are divided into a finite number of samples for which the properties can be measured and recorded. This means that our knowledge of fields is usually limited to a set of observations at given locations and time. However, according to Tobler first law of Geography that explains: "*All things are related, but nearby things are more related than distant things*" (Tobler, 1970), it is possible to extract unknown field values between the observations using some rules. Therefore, to represent a continuous field from its discrete samples, it is necessary to define a set of rules that can present a complete or incomplete representation of the continuous field (Goodchild 1992, Kemp1997, Goodchild et al. 2007).

Complete representation approaches assign the property of the field at any location by using a simple function to estimate the spatial variation. Complete representation approaches include:

a) Piecewise constant in regular cells: the field property is constant over a regular shaped area (figure 2-3a), such as a grayscale value for each pixel in an aerial image in 2D or a rock property for each 3D cell in a geological model.

b) Piecewise constant in irregular cells: the field property is constant over an irregular shaped area (figure 2-3b), such as vegetation cover class or name of a parcel's owner in 2D or 3D space.

c) Piecewise linear in irregular cells: as triangular irregular network (TIN) method where there is a linear varying over triangular elements (figure 2-3c), such as the elevation values in a TIN-based digital terrain model.

In the incomplete representation, there is no information about the values of the field between sample points and some "unspecified" form of interpolation is required. These representation approaches are:

d) Sampled at a set of regularly spaced sample points, such as the elevation values in a regular digital terrain model (figure 2-3d)

e) Sampled at a set of irregularly spaced sample points, such as weather temperature values in irregular stations (figure 2-3e)

f) Isolines of a surface in 2S or isosurface of a volume in 3D which a constant value of a contoured variable (figure 2-3f), such as representing elevation and pressure in 2D or 3D respectively.

Approaches of (a) and (d) are based on raster models, while approaches of (b), (c), (e) and (f) are based on vector data model where (b) and (c) use polygons, (e) uses points and (f) uses lines to represent the continuous variation of field. Although all approaches can be used to represent any field, the representation method is chosen

based on the sampling scheme used, the proposed analysis and the functionality of the software (Kemp 1997).

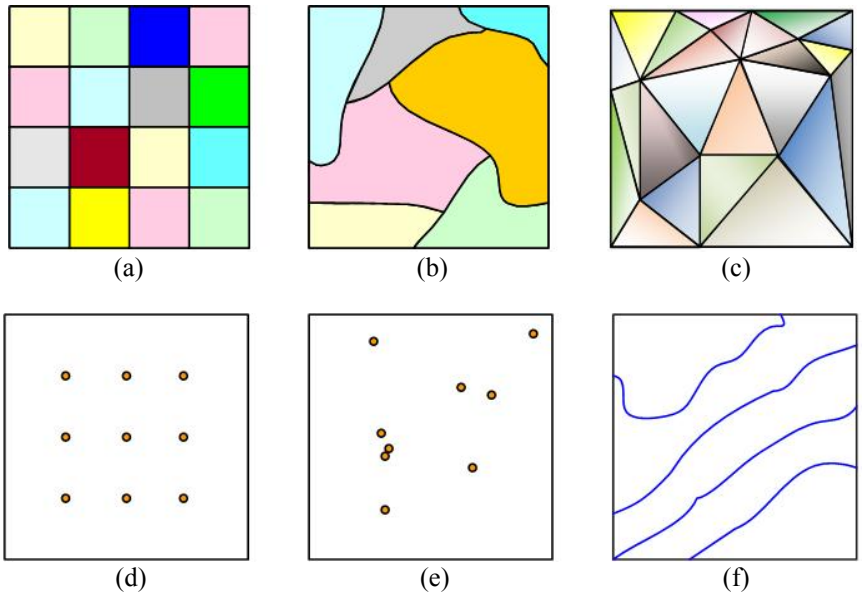


Figure 2-3 Six different approaches for the spatial representation of a field in 2D.

In geosciences, sample data may come from different techniques such as seismic profiling, remote sensing, weather balloons, probes, systematic survey and borehole drilling (Raper 2000). However, because of sampling methods limitations, the characterization of fields is uncertain, and getting complete spatial and temporal information is impossible (Fisher 1993). Regarding the measurement techniques, except for image-based technique, sample data is usually 3D points with irregular distribution. Each data point is defined by its geometric position in 3D space (x, y, z) and can have one or more attributes attached to it. However, since the field space is usually heterogeneous and anisotropic, it is important to know if the points are representative of their neighborhoods. This helps to choose a better tessellation approach to rebuild the continuity and connectivity between the discrete sample data. For example, using an inappropriate tessellation, boundaries or discontinuities (such as faults in a hydrogeological system) may fail to appear.

Regular and irregular methods (Boots 2005) are used as two general kinds of tessellations, where the variation of a field within each element is modeled as a

constant, linear or polynomial function of location. The elements of a regular tessellation have uniform shapes and sizes such as 2D rectangles and pixels (figure 2-4a) or 3D cubes and voxels (figure 2-4c). For example, DEM, rainfall or temperature maps are examples of 2D regular tessellations where the space is divided in 2D rectangles and an elevation, rain amount or temperature is assigned to each rectangle. The modeling of rock-property variations is another example in 3D where the 3D space is divided into Voxels having rock properties as their attribute (Bosch, 1998, Jessell 2001).

In a regular spatial tessellation, the spatial positions implicit in the ordering of the elements, then the adjacent elements are readily detected. However, a major limitation of regular tessellation is related to the handling of field data (unconnected points) with an irregular distribution. Indeed, in this case, a large number of elements is required for a fine resolution, which, especially in 3D, is computationally costly.

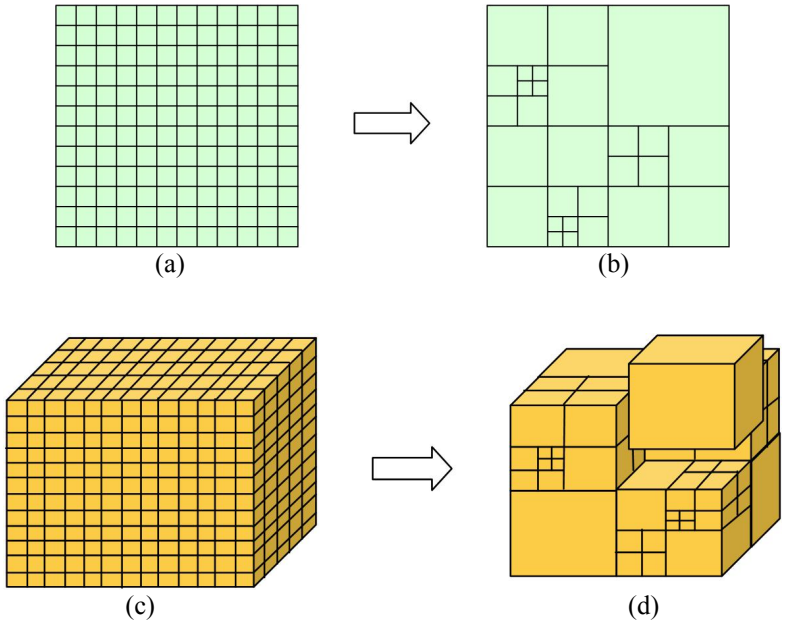
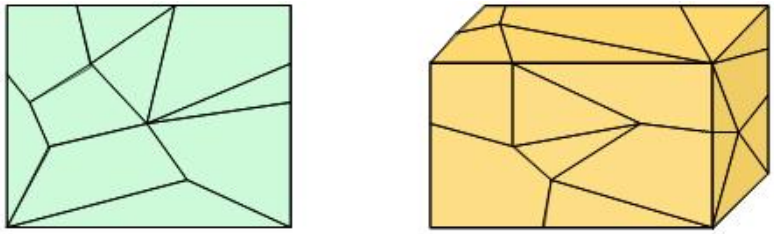


Figure 2-4 Regular tessellations in 2D and 3D.

To overcome the limitations of regular tessellations, hierarchical tessellations can be used, where a tree is created. Quadtree (figure 2-4b) and Octree (figure2-4d) are two examples of this data model in 2D and 3D spaces respectively (Samet 1984). In

fact, these methods subdivide the space into four squares (Quadtree, in 2D) or eight cubes (Octree, in 3D) of equal size until either each element contains one homogeneous region or reaches a desired resolution, such as Li's work (1996) for the subsurface geological modeling. Therefore, in these hierarchical models, a coarse resolution is used to represent large homogeneous areas, while a finer resolution is used for areas of high spatial variability. Although these methods reduce the required memory for a fine resolution mesh, a small change in field data may result in a quiet different tree (de Berg et al. 2000, Worboys and Duckham 2004).

Irregular tessellations can also be used to model a continuous phenomenon. The elements of irregular tessellations can have any size and shape (for example triangles and polygons in 2D or tetrahedrons and polyhedrons in 3D) and follow the outline of the data points (figure 2-5). Thus, they can be adapted to the point distribution and provide conformity to complex phenomena. For example, Lattuada and Raper (1995) demonstrate the potential of 3D triangular tessellation for modeling coastal landforms from borehole data. However, in irregular tessellations, the connectivity between the elements or topology does not implicitly exist. Hence, topology needs to be computed and stored explicitly and updated after changes, such as tessellation refinement or modification, which is usually occurred during the modeling and representation of a field. For example, in modeling of a hydrogeological system, it is sometimes required to remove or insert faults or modifying the location of wells without having to rebuild the whole tessellation. However, this is a difficult task for current GISs data structures as they are static. Therefore, following any change in the spatial information, all spatial relationships (topology) must be rebuilt. This problem becomes more complicated for 3D tessellations.



(a) (b)

Figure 2-5 Examples of irregular tessellations a) a 2D tessellation and b) a 3D tessellation.

2.3.1.2 Object-based data models

As previously mentioned, object-based spatial data models perceive the space as empty domain consisting of discrete spatial objects. These objects are identified by their dimensionality, in the sense of how they occupy the space (Longley et al. 2005). The objects may be more like points (0D) such as an individual car, lines (1D) such as a road, polygons or surfaces (2D) such as water bodies or polyhedra or volumes (3D) such as a building. Therefore, the spatial objects may be geometrically represented by points, lines, polygons and polyhedra.

In 2D data models, the position of points is defined by 2-dimensional Cartesian coordinates (X, Y) and lines, polygons are defined by connecting sequences of (X, Y) coordinate pairs. The data model can be either topological where the spatial relationship between these primitives is explicitly stored (such as the arc/node model) or non-topological where each primitive is recorded separately with no records of intersections or connections with other primitives (such as the spaghetti model) (Aronoff 1989).

In 3D data models, each primitive (a point, line, polygon or polyhedron) is defined in a three-dimensional Cartesian coordinates system. Since the data models are limited to represent 3D objects, 3D objects (solid) are sometimes represented as a quasi-3D representation through the use of surfaces. In this model, only a single elevation (z) can be assigned to any surface at any given location, such as a digital terrain model (DTM). However, this model is inadequate for representing some objects that require a true 3D spatial model (with three independent coordinates x,

y, and z) that can accept several z values for any given location (Raper and Kelk 1991, Bonhamm-Carter 1996, Abdul-Rahman and Pilouk 2008, Pouliot et al. 2008). For example in a hydrogeological system, several important geological objects, such as faults causing repetition of a single horizon at given locations, cannot be represented by this 2.D model. Moreover, the space between the surfaces (volume) cannot be described (Houlding 1994).

In GIS, most work in 3D has focused on geometrical modeling of regular objects (Cambray 1993, Bric 1993). Much of these works have used boundary representation models (B-rep), which involve representing an object as a set of bounding surfaces (Abdul-Rahman and Pilouk 2008) (figure 2-6). The major problem of this model is that surfaces are constructed independently of each other and thus, information about the connectivity of these separated surfaces is required, which involves many intersection calculations between all of the surfaces.

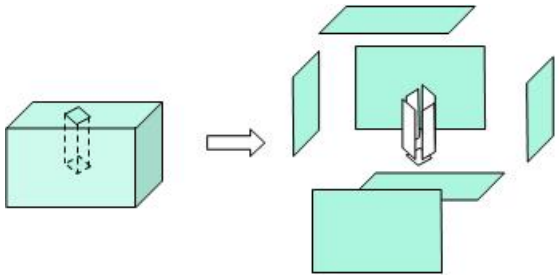


Figure 2-6 Boundary Representation: a cube with a hole (see Breunig 1996).

The Constructive Solid Geometry model (CSG) is used widely in computer aided design (CAD) for modeling in architecture and engineering application (Lee 1999). Several works such as Cambray and Yeh (1994), Bric et al. (1994) and Zeitouni and de Cambray (1995) used CSG for modeling a man-made solid object. As illustrated in figure 2-7, CSG represents an object by combining simple primitive sets (such as spheres, cones, cylinders) using Boolean operations (such as tree structure). Therefore, CSG models deal only with simple and regular objects, because complex solids may result in a very deep tree structure for real world modeling (Stoter, 2004). Another problem is that the model has no topological structure at all.

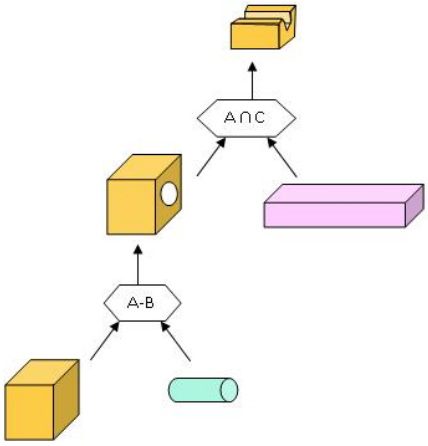


Figure 2-7 Constructive Solid Geometry (see Breunig 1996).

Regular and irregular 3D decomposition, like as field-based data model, are other data model that can model objects by a union of discrete cells having regular or irregular size and shape respectively. Cells are adjacent, connected and do not intersect. Examples of regular decomposition are given in Jones (1989) and Pilouk uses a 3D TIN model for regular objects on the terrain. Figure 2-8 shows an example of 3D regular and irregular decomposition.

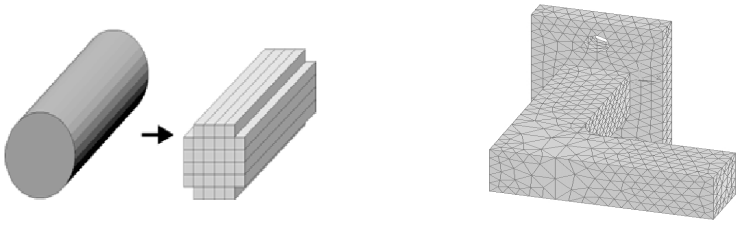


Figure 2-8 a) Regular decomposition (e.g. voxel), b) Irregular decomposition (e.g. tetrahedron) (see Breunig, 1996).

We investigated spatial field- and object-based data models. Here, we will explain how these spatial data models (section 2.3.1) are currently associated with the temporal components, i.e. spatio-temporal data models.

2.3.2 Spatio-temporal data models

To model and represent the dynamic phenomena in GIS, it is essential to extend the existing spatial data models and structures to represent the information about the temporal component of the phenomena. An extension of "spatial models" to "spatio-temporal data models" may be designed to answer two types of questions: those concerning the world state at a specific time (the static point view) and those concerning with changes and processes in the world state during a specific time interval (the dynamic point of view). Understanding of a dynamic phenomenon such as a flood event or pollutant transport requires a dynamic view of the phenomena. Based on a literature review, the most important models that incorporate temporal aspects of phenomena into spatial data models are: *snapshot model, amendment vector model, spatiotemporal model and event-based model.*

a) Snapshot model

Most of the approaches of representing changes are based on location-based concept or the snapshot model (Armstrong, 1988). In this model, a new complete snapshot (spatial model) is constructed for different time-slices within a specific time interval and the occurrences time is stamped (figure 2-9). The process of retrieving the changes is based on cell-by-cell (in a raster model) or vector-vector (in a vector model) comparison of the adjacent snapshots, which can be very time consuming.

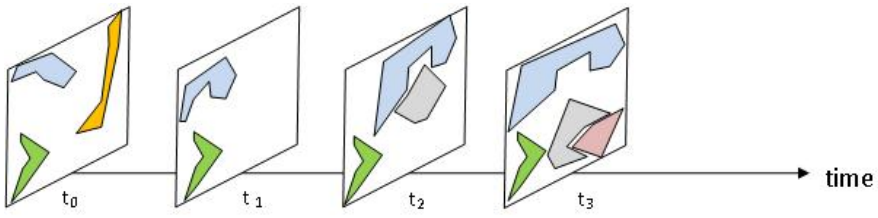


Figure 2-9 An example of snapshot model.

Another major shortcoming of this model is breaking up the continuity of the dynamic phenomena. In fact, we have no information about the changes between

the time slices. Moreover, the model stores all of the data, changed or not changed, at each time-slice which duplicates all the unchanged data.

b) Amendment vector model

Langran (1989) proposed an object-based model to describe the change of a spatial object through a period of time. This model relies on amendment vectors where the spatial objects are defined by a set of points, lines and polygons, and their connectivity (topology) is explicitly maintained. Also, a database is used to store the attribute values of the objects in which each object is represented by a unique identifier and a sequence of attribute values. In this model, as Peuquet (2001) states, any changes subsequent to some initial point in time in the configuration of linear or polygonal objects are incrementally recorded. Figure 2-10 shows an example for the changes sequence in road in time t_1, t_2 and t_3. Hazelton et al. (1990) extended this model to 4D by using the hierarchy of points, lines, polygons, polytopes and a similar model proposed by Kelmelis (1991) using points, lines, surfaces and volumes as the primitives within a 4D Cartesian space-time system (Peuquet 2001). However, in these models, following any change in geometry, connectivity or attribute of vectors, a global updating of model is required; i.e. re-organization of whole spatial data and databases. This is a time consuming task, especially when the number of amendment vectors increase. To solve this problem, spatial-temporal objects model were introduced.

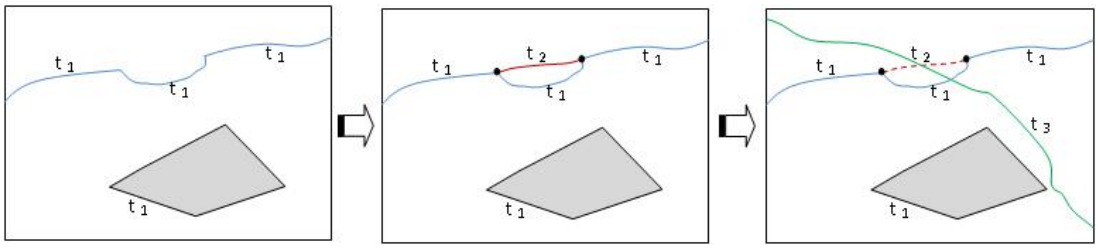


Figure 2-10 An example of an amendment vector model (see Peuquet 2001).

c) Spatiotemporal object model

Worboys (1992) proposes a spatiotemporal object model that represents the world as a set of discrete 3D objects (2D space and 1D time). Each object containing a unique identifier is an aggregate of homogenous disjoint prisms (ST-atoms) whose heights and bases represent their time extent (lifetime) and spatial extents. For example, figure2-11 shows two areal objects at t_0 (O and O'). The spatial extent of objects O and O' are respectively reduced and enlarged at time t_1. With reducing the spatial extend O' at time t_2, a new object O" is created. Therefore, the spatio-temporal objects O, O' and O" can be described as composed of a finite Spatio-temporal atoms :{<S_1,[t_0, t_1]>, <S_2,[t_1, t_∞]>}, {<T_1,[t_0, t_1]>, <T_2,[t_1, t_2]>,<T_3,[t_2, t_∞]>}and <U,[t_2, t_∞]>.

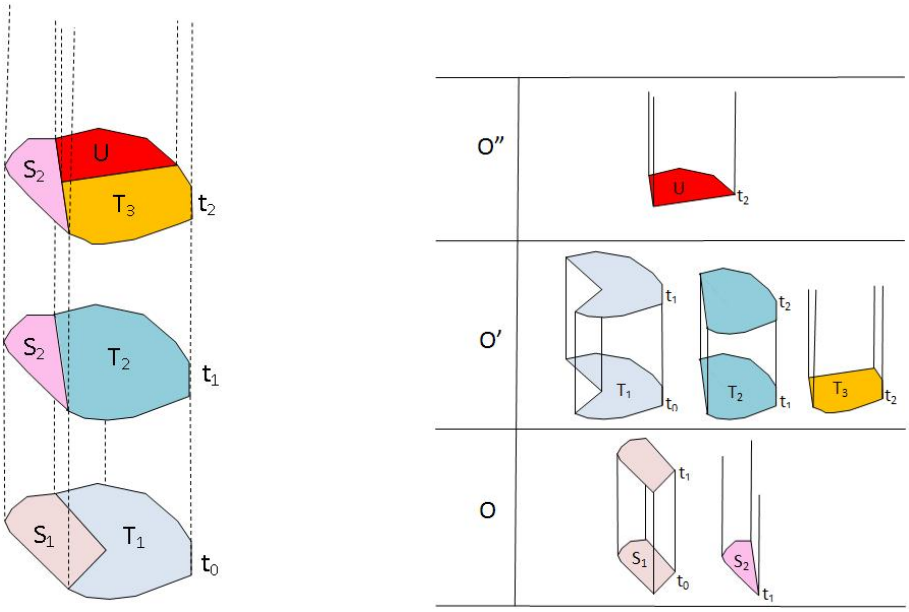


Figure 2-11 An example of spatiotemporal object model (Worboys 1992).

In this model the behavior of an object is expressed as a set of operations that the object can perform under appropriate conditions (Worboys 1992). These operations beside the inheritance property of the object can represent the object dynamics such as merge and split of objects. However, these spatio-temporal atoms have a discrete structure, modeling of gradual changes and process over space and time with this model is impossible (Yuan 2001).

d) Event based model

The spatiotemporal raster-based data model (Peuquet and Duan, 1995) is an example of such model that is a modified approach of snapshot representation for a raster model. This model stores 'changes' or ‘events’ in relation to a previous state rather than a snapshot of a state. Every event is time-stamped and associated with a list of event components to indicate where changes have occurred. Therefore, each list can have a variable length, which represents the change history for the cell (figure 2-12). Although this approach efficiency supports both spatial and temporal queries for a raster model, it is not compatible with a vector-based model where we need to keep track of the individual objects and their topology. In addition this model stores only changes at pre-defined locations and it is not able to support the representation of dynamic phenomena which move, split or merge.

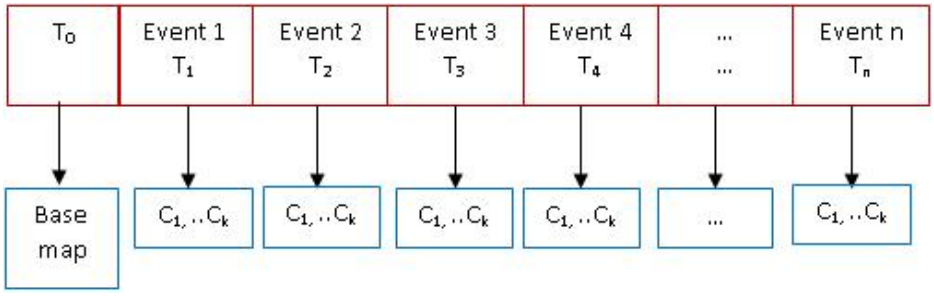


Figure 2-12 An example of an event-based model (Peuquet and Duan, 1995).

To solve this problem, some advanced data models go beyond the snap-shot approach and propose the events or processes approaches. An example of this model is oogeomorph proposed by Raper and Livingstone (1995), which is an object-oriented spatiotemporal model for geomorphology. Similar to spatio-temporal object model, oogeomorph model represents the geographical entities as collections of spatiotemporal atoms. Nevertheless, in this model each entity refers to a 3D position and the time dimension. In addition, an object (dynamic phenomenon) is modeled by composing it to instances of “forms”, “processes” and “materials”. This model is designed to model linear features (shore lines) and has difficulty in manipulating area data and topological relationships. Another example

of event-based approach is Yuan's work (2001). She proposes an event-based model for representing dynamic phenomena that have both field and object characteristics such as flood process.

We have discussed the conceptual and data models for the representation and simulation of phenomena. In conceptual models, the spatial and dynamic aspects of phenomena were identified as the most important aspects which must be considered in GIS-based simulations. Then, we reviewed the spatial data models that aimed to define a set of entities and their relations used to model and represent a phenomenon at a specific time and then, the spatiotemporal models were discussed which designed to represent the dynamic behavior of the phenomena describing by the process models. These data models are implemented by data structures. Here, we turn to GIS data structures, which, as Kemp (1997) stated, describe details of the specific implementations of the spatial data models.

2.4 Spatial Data structures

Spatial data structures provide the information that the computer requires to reconstruct the data model in digital form. Data structures must:

- Represent the selected data models completely and unambiguously,
- Efficiently support the spatial analyses operations
- Use storage space economically,
- Be robust.

Data structures can be classified according to whether they are defined based on raster or vector data models.

2.4.1 Vector data structures

Points, lines, polygons and solids are the spatial primitives in a vector data structure. The position of points is defined by 3-dimensional Cartesian coordinates

(X, Y, Z) and lines, polygons and solids are defined by connecting between sequence of (X, Y,Z) coordinate pairs. Vector data structures can be classified as *non-topological* or *topological* data structures. In non-topological data structures, the spatial relationship between adjacent entities is not explicitly stored (such as a Spaghetti data structure) while in a topological data structure the inherent spatial connectivity and adjacency relationships of features are explicitly stored. This helps insure that information is not unnecessarily repeated (Zlatanova et al. 2002). Ellul and Haklay (2006) describe three different approaches to model the results of topological relationship determination including:

- Matrix-based approaches where the relationships between objects are stored in a matrix such as the 2D formal framework of Egenhofer and Franzosa (1993) and it extension to 3D (Wei et al. 1998),
- Graph based approaches where the topological information is represented by a node-edge model where the nodes and edges represent the objects in question and their connections, respectively (Gibbons, 1989; Gross & Yellen, 1999; Temperley, 1981; Wilson, 1996),
- Topological primitive approaches where objects are composed of topological primitive (nodes, edges, faces and volumes). For example, 3D FDS (Molenaar 1990), 3D GT (Zeitouni and de Cambray 1995), SSM (Zlatanova 2000), and SOMAS (Pfund 2001) are proposed as storage data structures for boundary-representation models. Pilouk (1996), Tse and Gold (2003) and Lachance et al. (2006) use a 'triangle/tetrahedron' simplex approach to primitive creation for irregular 3D decomposition models.

The vector-based representations have some advantage such as:

- High accuracy for object representation
- Facilitates topological analysis
- Simple execution of coordinate transformations (efficient in terms of computing)
- Simple realization of distance operations like the Euclidean distance

- Efficient data storage

A significant shortcoming of vector based models is the difficult computation of intersections and neighborhoods. In addition, the ability of this data structure to handle truly a 3D object is very limited. In addition overlay operations are difficult to implement. With regard to dynamics, the vector data structure is ill-suited to define the region affected by a single change in geometry and connectivity of entities such as in amendment vector method. Hence, a global reconstruction of the topology is required after any change. Another important problem of vector models when simulating a continuous field is that the variation of the field properties is undefined across the features of this model (Goodchild 2008).

2.4.2 Raster data structures

In a raster-based data structure, the information is represented as a non-overlapping grid of "cells" where each cell is addressed by its position in the array, and only the value of the cell is stored in the computer. Cells are adjacent, connected and do not intersect. This creates a continuous surface for the data, where topology between the cells of any particular object (not between the objects) is implicit in the data structure. Raster data is best suited for continuous data representation (Breunig 1996; Li 1994). This data structure can efficiently be used for integrating with arterial or satellite images for analysis purposes. Overlay operations are easily implemented. Some software packages support Voxel structures (Neteier and Mitasova, 2002). GRASS 5.0 (Ciolli et al. 2004), GOCAD (Mallet 1992) and 3D-visualization software (OpenDX 2004) are examples of theses softwares. The advantage of this type of representation is that most of the topological properties are implicitly given. However, its disadvantages include:

- The execution of coordinate transformation such as scaling and rotation is difficult.
- Objects must be generated by costly methods (e.g. component labelling)

- Spatial relationships, such as connectivity between two objects, are difficult to assess
- Higher data storage requirements to represent the variations of the heterogeneous field over the space especially in 3D. Some spatial data models have been developed to address this problem such as Quad-tree and Octree, comparing to vector data structures.

With regard to dynamics, a raster data structure is unable to represent and manage any changes in position of objects or field (motion). The raster grid is static by nature and cannot move or be deformed to represent the behavior of a moving field or object.

2.5 GIS Operations for spatial analysis on fields and objects

The GIS operations allow manipulating and analyzing the spatial data. There exist many possibilities to group GIS operations for different purposes. Tomlin (1990) provides a frame work, called map algebra for the manipulation of field data. As explained in Worboys and Duckhum (2004), these operations can be organized in three groups, namely local, focal and zonal operations.

- Local operations act on one or more fields and compute a new value that is function of the value of the fields at the same location. For example, to add or multiple two fields *a* and *b*, the values of fields *a* and *b* are added at the same location. An analysis example is “classifying the areas without vegetation in which their slope is greater than 15%”.

- Focal operations act on a location and compute a new value that is function of the value of the location and its neighbors; for example, computing the

slope or aspect of a field of topographical altitudes over an area. An analysis example is “classifying the areas with slope greater than 15%”.

- Zonal operations act on a location and compute a new value that is function of the value of the location and all locations in a zone; for example, computing the average temperature of a zone. An analysis example is “classifying the areas with the temperature average greater than 50”.

The operations may be unary, binary or n-ary in the term of acting on one single field, two or more fields respectively. Binary or n-ary operations must sometimes act on fields data in various data models. In these cases, it requires the conversion of all data models to spatially equivalent ones (Kemp 1996). Kemp (1996) explained that this process is done by “resampling”, which involves deriving the continuous field from a spatial data model by a set of rules called interpolation (Goodchild 1992) and then sampling the continuous field to produce the target model.

The Map Algebra operations are implemented in many GIS for scalar field. Several works extended these operations. Li and Hodgson (2004) presented these operations for vector fields. Scott (1997) and Ledoux (2006) extend the operations for 3D Voxel data and 3D Voronoi tessellation respectively. Mennis et al. (2005) present the operations for the combination of time and 2D and 3D field.

Worboys and Duckhume (2004) categorize the operations acting on spatial objects in three groups namely set-oriented, topological and Euclidean operation. These operations act on one or more objects as inputs and produce a result. “Equals”, “union”, “is empty” are examples of the set-orientations operation. For example, “equal” acts on two spatial objects and determines if these two objects are same. “Meet”, “is inside”, “cover” are examples of the topological operations. For example, “meet” acts on two objects and determines if they touch externally in a common portion of their boundary. “Distance”, “area” and “centroid” are examples of Euclidean operations which determine the Metric relations between object.

Albrecht (1996) created a classification of 2D and 2.5D GIS operations as follows:

- Search (interpolation, search-by-region, search-by-attribute, classification)
- Locational analysis (buffer, corridor, overlay, Voronoi/Thiessen),
- Terrain analysis (slope/aspect, catchment/basins, drainage/network, viewshed),
- Distribution and neighborhood (cost/diffusion/spread, proximity nearest-neighbor)
- Spatial analysis multivariate analysis, pattern/dispersion, centrality/connectedness, shape) and measurements.

Raper (2000) classifies 3D spatial query and analysis functions as follows:

- Visualization (Translate, Rotate, Scale, Reflect)
- Transformation (Shear), Selection (AND, OR, XOR, NOT)
- Inter-relationships (Metric, Topological)
- Characterization (Volume, Surface area, Centre of mass, Orientation) and Modeling (Build).

Kim et al. (1998) introduce another classification based on metric analysis, buffering and near operations. Please see Göbel and Zipf (2008) for an overview of the existing taxonomies of GIS-operations.

2.6 The need for spatial dynamic process modeling within GIS

Based on the literature review, to represent and simulate a dynamic field or the evolution of discrete objects, it is necessary to consider the changes during a specific time interval. Although, the spatiotemporal models (see 2.3.2) explicitly

include time, they are not designed for explicit representation of the process of dynamic phenomena. These data models have been developed to facilitate visualization and analysis of dynamic phenomena that are important issues in databases, and are unable to conceptualize ramification of the process, which is critical in some analysis (Brown et al.2005). For example, we may desire to interpolate phenomena (object/or field) over time, or by a given hypotheses we would like to predict the state of phenomena in future. These types of analysis involve the simulation of process, which requires representing the dynamic behavior of spatial features (field or object), which is described by process models. Here, we review the process models and explain the linking between spatial data models and process models for simulation purposes.

2.6.1 Process models

There are two different categories of process models which describe the dynamic behavior of continuous fields and discrete objects.

2.6.1.1 Continuous models

The dynamic behavior of fields is usually described by mathematical models in the form of partial differential equations (PDEs) including independent variables that are continuous functions of location such as atmospheric pressure, soil moisture and ground elevation. These equations which implicitly recognize the continuity of space and the constantly changing values of the fields over time are solved by numerical methods (Kemp 1997). The numerical methods discretize time and space into small units and estimate the variation of fields values over time at a fixed location: $\mathrm{F}(t) = f(x, y, z, t)$ or at a time-dependent location: $\mathrm{F}(t) = f(x(t), y(t), z(t))$ and a final solution is achieved by 'integrating' the results across the entire study area and time period. These visions correspond to *Eulerian* and *Lagrangian* points of view in computational methods (Price 2005, Brown et al. 2005). In the former, processes are described either as change at a fixed location. Therefore, these

approaches take advantage of a fixed tessellation, which is simple to generate and maintain. However, the size and shape of the tessellation elements as well as the value of time-interval between snapshots have to be carefully selected to ensure the field continuity and a realistic representation of the dynamic field. In the latter, processes are modeled by tracking the changing location, shape and values of particles over space and require a local update of particles when the changes occur instead of updating all at once.

2.6.1.2 Discrete models

Discrete models describe the relationships and interaction between discrete objects in space and govern their changes or motion (Goodchild, 2005). An example of discrete models is a spatial interaction model that estimates the flow of people, material or information between locations in space and is typically used to predict the number of migrants, shopping trips between a discrete origin and a discrete destination. Another example is agent-based models where the dynamic behavior of a system is described through rules governing the actions of a number of autonomous agents. For example the behavior of a crowd can be modeled through a series of rules about each discrete object's behavior. Discrete models are usually best in describing and representing objects dynamic such as buildings or vehicles. To simulate dynamic processes, the objects may change their spatial location, their movement speed and direction, their shape, and the state of their other internal properties over time. Therefore, it is necessary to locally update objects when the changes occur instead of updating all at once.

Since this thesis is concerned with the modeling of dynamic fields, in the next section, the link between a continuous process model and a field-based data model is investigated.

2.6.2 GIS-based simulation of a dynamic field: Linking process models with spatial data models

Dynamic field simulation may be based on the numerical methods, which involves the discretization of PDEs and their numerical solution using a spatial tessellation (which must be represented by a data model) covering the domain of interest. The efficient solution of the equations frequently requires methods that are adaptive in both space and time. PDEs are generally solved in space-time using finite difference methods (FDM) or finite element methods (FEM) in either a static manner from the Eulerian point of view or in a dynamic manner from the Lagrangian point of view (figure 2-13). These visions correspond to the well-known Location-based and object-based representation methods of changes by Peuquet (1999) with this difference that an object, here, refers to an element of the tessellation and has no meaning in reality.

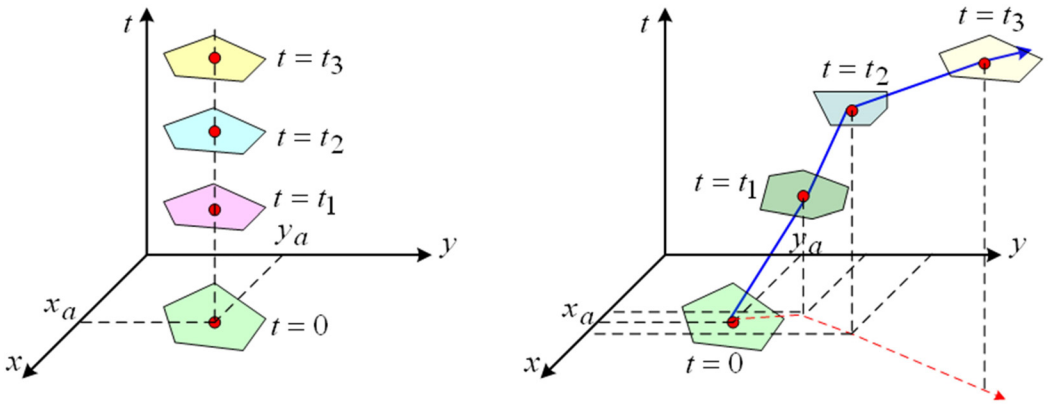


Figure 2-13 Two different simulation methods, a) Eulerian methods describe changes of field at a fixed location at a series of snap-shots; b) Lagrangian methods describe changes that occur as you follow a fluid element along its trajectory (blue line) at a series of snap-shots, the shape and size of the element may change.

FDM approximates the PDEs with a set of difference equations in the tessellation. Slope calculation from a raster DEM in GIS is a simple example of FDM where the derivative of the elevation is estimated with respect to the horizontal dimensions. In fact, FDM solutions for regular tessellation approximate the derivative using

difference in a simple raster and are readily supported by raster functions in GIS (Karssenberg and De Jong 2005). As an example, PCRaster using a command language (van Deursen 1995) has been designed for dynamic process simulation (such as environmental modeling). It maintains a separated raster layer for each time-step. However, this package only adopts a very simple approach to data management (Goodchild 2008).

FEM approximate the variation of fields using curvilinear functions within elements. This requires continuity of value, gradient and curvature across element edges (Carey 1995). Although continuity of value can be met in some GIS data models such as TIN, the gradients are undefined across edges. This make the solution of PDEs very difficult using GIS tessellations. For this reason, the role of GISs for FEM remains primarily in the area of data processing, analysis, visualization and management of simulation results that must often be sliced into several 2D datasets.

The problems related to FDM and FEM solutions in GIS become more complicated when dealing with simulation in a dynamic manner (Lagrangian point of view) where the variation of field is investigated at a moving coordinate frame; i.e. the tessellation moves. Following any movement in the tessellation, the shape and position of elements may change and it can result in a tangled (deformed) tessellation if the connectivity between elements (topology) remains unchanged. This is the major problem of the traditional Lagrangian methods. To solve this problem, the free-Lagrangian methods were proposed (Fritts et al. 1985, Mostafavi and Gold 2004). In these methods connectivity relations can be changed and hence there is no deformation occurring within the tessellation. This allows the representation of very complex fields where elements move with very different velocities. However, a main problem of these methods is related to determining the optimal time interval between snapshots. For example, a large time-step causes problems such as overshoots and undetected collisions as the connectivity updates is done only within a predefined time steps. As a result, we may observe some abnormal behavior in the simulation results. For a small time-step, an extensive

computation effort will be required to check for changes at time when none occurred. Another problem with Free-Lagrangian methods lies on maintaining and processing of the connectivity relations between tessellation elements at each time. To solve these problems, a spatiotemporal data model is required to model the variation in space with time by locally updating the spatial relationship at the time when they happen (event) (Gold 1996). These events can be the changes either on the field value or on the spatial relationship of the objects which is referred to as trajectory event and topological event respectively (Roos, 1997; Gavrilova and Rokne, 2003). Trajectory events are related to the physical problem description and defined by the governing equations, while the detection of topological events as well as local updating of the spatial tessellation should be done by a spatial data model.

2.7 Discussion: Why is simulation of 3D dynamic field difficult to realize within GIS?

The literature review on modeling and simulation in GIS shows that the main limitation for modeling and simulation of a dynamic phenomenon within GIS is due to the fact that current GIS suffer from the lack of proper spatial data structures to represent the three-dimensionality and the dynamic nature of fields. As previously mentioned, raster and vector data models and structures are 2D and static and thus not adequate for the representation of 3D dynamic fields. In a raster data structure, space is partitioned into a static regular tessellation which is not computationally efficient to represent the variations of a heterogeneous field over the 3D space. Another problem with raster data structure is related to the representation and management of moving tessellation elements. The raster grid is static by nature and cannot move or be deformed to represent the behavior of a dynamic field. This is because of the fact that the spatial position and topology of the elements is implicit in the ordering of the elements. Thus, it is impossible to update of topology after any change in the elements spatial relationship and the

tessellation must be rebuilt. A vector data structure, such as TIN, is more flexible to represent the complexity of a dynamic field. However, current GIS do not support a continuously moving and deformable irregular tessellation. In addition, topological relations need to be updated continuously in order to maintain the validity of the structure, but, vector data structures are ill-suited to defining the region affected by a single change and any movement of elements requires a global reconstruction of the topology.

To address these problems, a data structure is required with the capability for local update of topology after any change in the tessellation relationships. Gold (1991), Gold and Condal (1995) and Okabe (2000) argued that a spatial data structure based on Voronoi diagram is a good candidate for the tessellation of continuous fields within GIS. Voronoi cells can be defined by points with an arbitrary distribution, creating elements of different sizes and shapes that can adapt to complexity of field. Each element can have an arbitrary number of neighbors which their connectivity with the given element is clearly defined and can be explicitly retrieved if needed. In addition this data structure can be *"dynamic"* and *"kinetic"*. A dynamic Voronoi data structure allows the addition or the removal of generators without any global reconstruction of topology and a kinetic Voronoi structure allows the generators move along given continuous trajectories and interact with each other. In this basic, several works on *dynamic* VD and its dual DT show the potential of those structures for providing an adequate tessellation for field representation in 2D (Mostafavi 2002) and 3D space (Ledoux and Gold 2008). There have also been works that used DT and VD in different geosciences applications that require modeling of fields such as Marine GIS (Gold and Condal, 1995; Gold, 1999) or reservoir simulation in a hydrogeological system (Hale 2002). In addition, Mostafavi and Gold (2004) developed a 2D *kinetic* data structure based on Delaunay triangulation and Voronoi diagram and showed the data structure keep the connectivity relationships of the elements consistently during the simulation process. In addition, in their work the problem of fixed time steps is addressed by providing very flexible mechanisms to detect and manage different changes (events) in the spatial tessellation. Hence, this thesis extends their work to 3D by

developing a 3D kinetic data structure based Delaunay tetrahedralization and 3D Voronoi diagram, which bring the some benefits for simulation of a 3D dynamic continuous phenomenon.

2.8 Conclusion

In this chapter, we have presented an overview of the conceptual models, data models and data structures for the representation and simulation of a dynamic phenomenon within GIS. In the conceptual models, the spatial and dynamic aspects of phenomena were identified as the most important aspects to consider in GIS-based simulations. We explained the two fundamental categories of spatial data models namely field and object based data models. Next, the spatiotemporal models were discussed to represent the dynamic behavior of the phenomenon. These data models can be implemented by the vector or raster data structures. These data structures have several shortcomings to address the three-dimensionality and the dynamic nature of the phenomenon and even more to handle the spatial process.

Several works suggest that VD&DT have many interesting potentials for the representation of a dynamic field in 2D and 3D spaces. In the next chapter, we briefly review the Delaunay and Voronoi tessellation and their properties in the context of spatial modeling and simulation.

Chapter 3: Voronoi and Delaunay Tessellations, Their Properties and Construction

3.1 Introduction

This chapter describes the Voronoi and Delaunay tessellations with a brief review of their definitions, structures and relevant spatial properties in the context of this thesis i.e. modeling and simulation of 3D dynamic fields. Then, the methods for their construction and storage are introduced, followed by a discussion on some complicated cases during their construction processes.

3.2 Definitions of Voronoi and Delaunay Tessellation

A Voronoi tessellation, also called a Voronoi diagram (VD), is a special kind of subdivision of a metric space with respect to the distances between a specified discrete set of objects in space (Voronoi 1907). Among the metric spaces, Euclidean space is the most widely used in GIS, because as Couclelis (2005) stated

"the Euclidean space is an abstraction of people's experience with the spatial properties of the local to medium-scale environments". However, it should be noted that Voronoi tessellation can be defined for metrics other than Euclidean space such as the Manhattan distances (Okabe et al. 2001). In addition, since a spatial field is usually sampled to a set of discrete points, hence, the Voronoi tessellation for points is discussed in details. However, Voronoi tessellation can also be defined by measuring distances to objects that are not points, such as lines, polygons or volumes. The Voronoi tessellation with these cells is also called the medial axis, which is beyond of the scope of this thesis.

3.2.1 Euclidean space

The Euclidean space in d-dimensional space is the *d*-dimensional real Cartesian space, R^d, where a point, *p*, is defined by a *d*-tuples of real numbers $(p_1, p_2, ..., p_d)$. In this space, the distance between two points, *p* and *q*, is defined as (Okabe et al. 2001):

$$\|p - q\| = \sqrt{\sum_{i-1}^{d} (p_i - q_i)} \qquad (3\text{-}1)$$

Therefore, a 2D Euclidean space is a 2D Cartesian coordinate system in two dimensions containing two axes (*x, y*), at right angles to each other, forming a plane (an *xy*-plane). The distance between two points, *p* and *q*, is defined as: $\|p - q\| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$. In a 3D coordinate system, another axis, *z*, is added, providing a third dimension of space measurement. Similarity, the distance between two points, $p(p_x, p_y, p_z)$ and $q(q_x, q_y, q_z)$, is defined as: $\|p - q\| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2}$.

3.2.2 Voronoi tessellation

Voronoi tessellation for a finite set of points in Euclidean space is a division of the space based on nearest neighbor rule where every location in the space is assigned to the closet member in the point set. Formally, let S be a set of points in the *d*-dimensional Euclidean space, the Voronoi tessellation of S, associates to each point $p \in S$ a Voronoi cell (element) $V(p)$ such that (Okabe et al. 2001, Edelsbrunner 2001):

$$V(p) = \left\{ x \in R^d \middle\| x - p \right\| \le \left\| x - q \right\|, \forall q \in S \right\} \qquad (3\text{-}2)$$

Where $\|x - p\|$ denotes the Euclidean distance between points x and p. Therefore, a Voronoi element is defined as the set of points $x \in R^d$ that are at least as close to p as to any other point in S. Therefore, a Voronoi tessellation of S is the collection of all Voronoi elements:

$$v = \{\mathrm{V(p)}, ..., \mathrm{V(q)}\} \qquad (3\text{-}3)$$

Voronoi tessellation v covers the space due to the fact that each point $x \in R^d$ has at least one nearest point in S and it, thus, lies in at least one Voronoi cell. Each Voronoi element is defined by the bisecting hyperplane between a given point and all the other points in the set. The bisector separating two points $p, q \in S$ is:

$$B(p,q) = \left\{ x \middle\| x - p \right\| = \left\| x - q \right\| \right\} \qquad (3\text{-}4)$$

A hyperplane is perpendicular to the line segment $\overline{pq}$ that divides the space into two halfspaces. The halfspace containing point p over q is defined as:

$$H(p,q) = \left\{ x \middle\| x - p \right\| \le \left\| x - q \right\| \right\} \qquad (3\text{-}5)$$

Therefore, the Voronoi cell associated with $p\,(V(p))$ over $(n-1)$ points in S is obtained by intersecting the (n-1) halfspaces:

$$V(p) = \bigcap_{q \in s\{p\}} H(p,q) \qquad (3\text{-}6)$$

This results in a convex polytope. The boundary of a Voronoi polytope consists of maximum (n-1) facets where each Voronoi facet is a (d-1) dimensional face (Okab et al. 2001). The boundary of a Voronoi face consists of (d-2) dimensional faces and so on. For example, a two-dimensional face is represented by a Voronoi polygon which is equidistant from (d − 1) points. The boundaries of a two-dimensional face consist of line segments (one-dimensional faces). 1D boundaries are called Voronoi edges which are equidistant from d points of S . The boundaries of a Voronoi edge consist of points, called Voronoi vertices, in which a Voronoi vertex is equidistant from (d + 1) points. Figure 3-1 shows an example of a Voronoi tessellation in 2D which is a polygonal subdivision of a plane consisting of vertices, edges, polygonal faces.

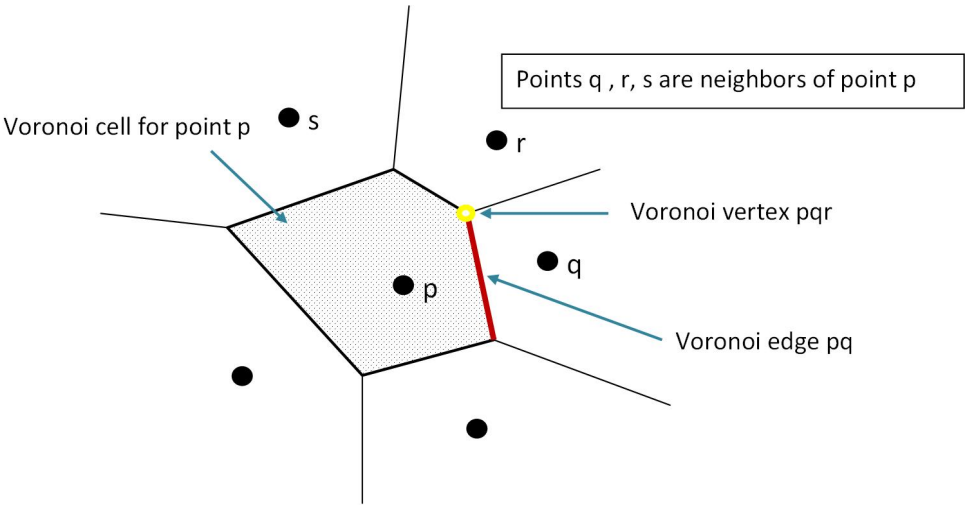


Figure 3-1 An example of a Voronoi tessellation in 2D and its components.

Figure 3-2 shows an example of a cell of Voronoi tessellation in 3D which is a polyhedral subdivision of a 3D space consisting of vertices, edges, polygonal faces and a polyhedral cell.

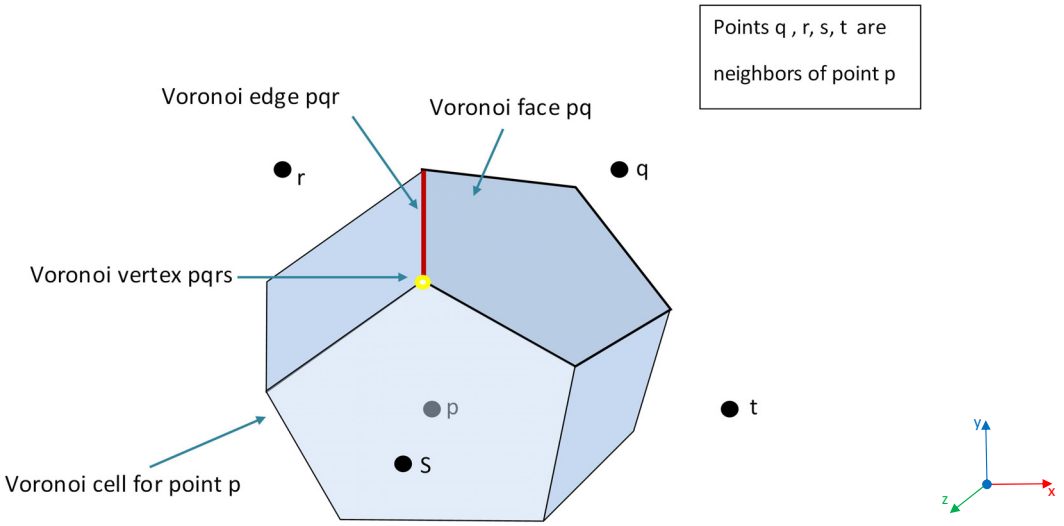


Figure 3-2 An example of a Voronoi cell in 3D and its components.

3.2.3 Delaunay Tessellation

Let S be a set of points in the *d*-dimensional Euclidean space, a Delaunay tessellation $DT(S)$ is a collection of d-dimensional simplexes where a simplex is the convex hull of a set of (d + 1) points and no point in S is inside the circum-ball of any other simplex in $DT(S)$ (okabe et. al 2001, Edelsbrunner 2001). In this definition the simplex, Convexhull and circumball are defined as follows:

Simplex: a simplex is the convex hull of a set of (d + 1) points. For example, a 0D simplex is a point, a 1D simplex is a line segment, a 2D simplex is a triangle, and a 3D simplex is a tetrahedron (figure 3-3).

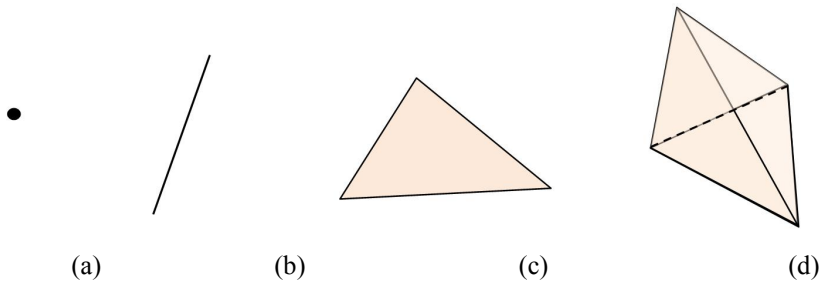


Figure 3-3 Simplex in a) 0D, b) 1D, c) 2D and d) 3D.

Convexhull: The convex hull of the set of points S is the smallest convex set containing the points S in which the line segment joining any pair of points lies entirely in the Convexhull (Preparata and Shamos 1985).

Circumball: in mathematics, a ball is defined as a set containing all points within a specified distance of a given point. "The circumball of a line segment is the ball admitting the segment as diameter. The circumball of a triangle is the ball admitting the circumscribing circle of the triangle as great circle" (Cohen-Steiner et al. 2004).

$DT(S)$ is unique, if S is a set of points in general position. This means no (d + 1) points are on the same hyperplane and no (d + 2) points are on the same ball. Therefore, a 2D Delaunay tessellation or *Delaunay triangulation* is a non-overlapping triangular subdivision of the space where each triangle has an empty circumcircle. The triangulation is unique, if no three (or more) points are collinear and no four (or more) points are on the same circumcircle. Figure 3-4 illustrates a 2D Delaunay triangulation and its components such as vertices, edges, triangles (elements).

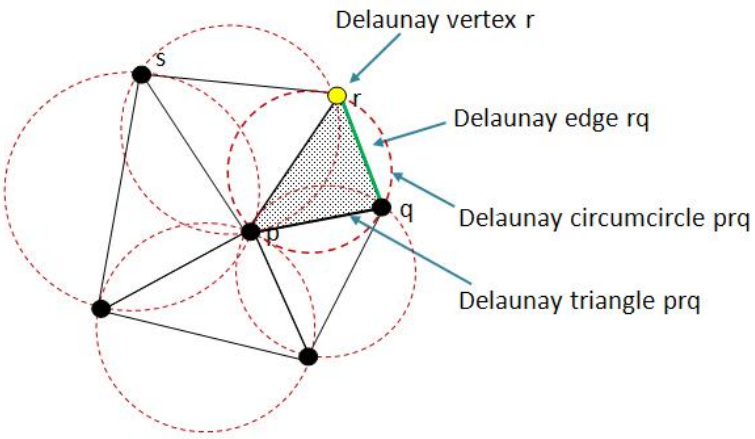


Figure 3-4 2D Delaunay triangulation and its components.

Similarity, a 3D Delaunay tessellation or *Delaunay tetrahedralization* is a non-overlapping tetrahedral subdivision of the space where each tetrahedron has an empty circumsphere. The tetrahedralization is unique, if no four (or more) points

are coplanar and no five (or more) points are on the same circumsphere. Figure 3-5 illustrates a 3D Delaunay tetrahedron and its components such as vertexes, edges, faces and elements.

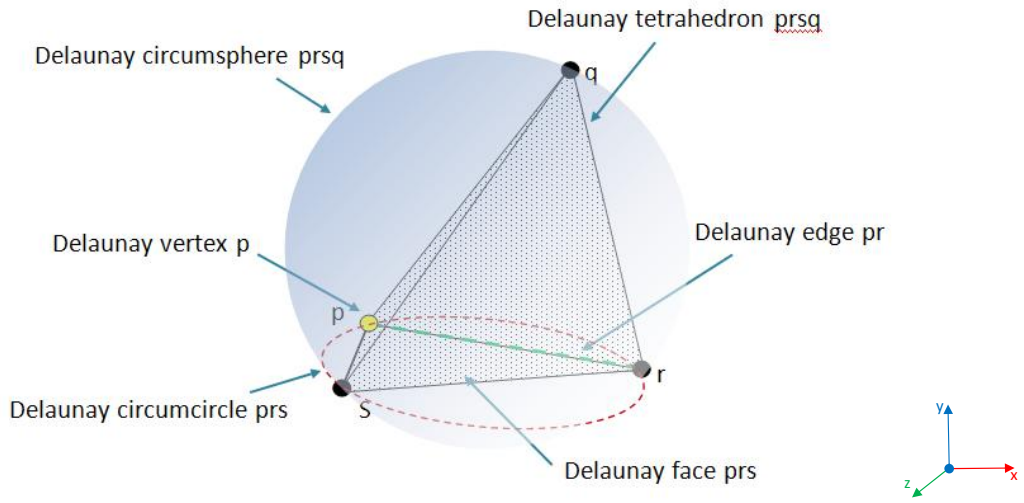


Figure 3-5 A 3D Delaunay tetrahedron and its components.

There are other important concepts in a Delaunay tessellation such as *star*, *link* and *ear*. In a Delaunay tessellation, a *star* of a Delaunay vertex consists of all the simplices in the tessellation that contain the vertex. A *link* consists of all faces of simplices in the *star* that do not intersect the vertex (Edelsbrunner 2001). Figures 3-6a and 3-6b show the *star* and *link* of vertex *p* in 2D and figures 3-6c and 3-6d illustrate these concepts in 3D respectively.

An *ear* in a 2D DT consists of three neighboring vertices which can make a new triangle ensuring that no another point is inside their circumcircle. For example in figure 3-6a, if point *p* is deleted or moved out from the circumcircle of three points *s,q,r,* it would makes new Delaunay triangle *sqr*. Similarly, an ear in a 3D Delaunay tessellation is four neighboring vertices that can make a new tetrahedron ensuring that no another point is inside their circumsphere. In 3D, depending on the points configuration, the four vertices can define an ear consisting of either three faces all sharing a vertex (figure 3-6e), or two adjacent faces (figure 3-6f).

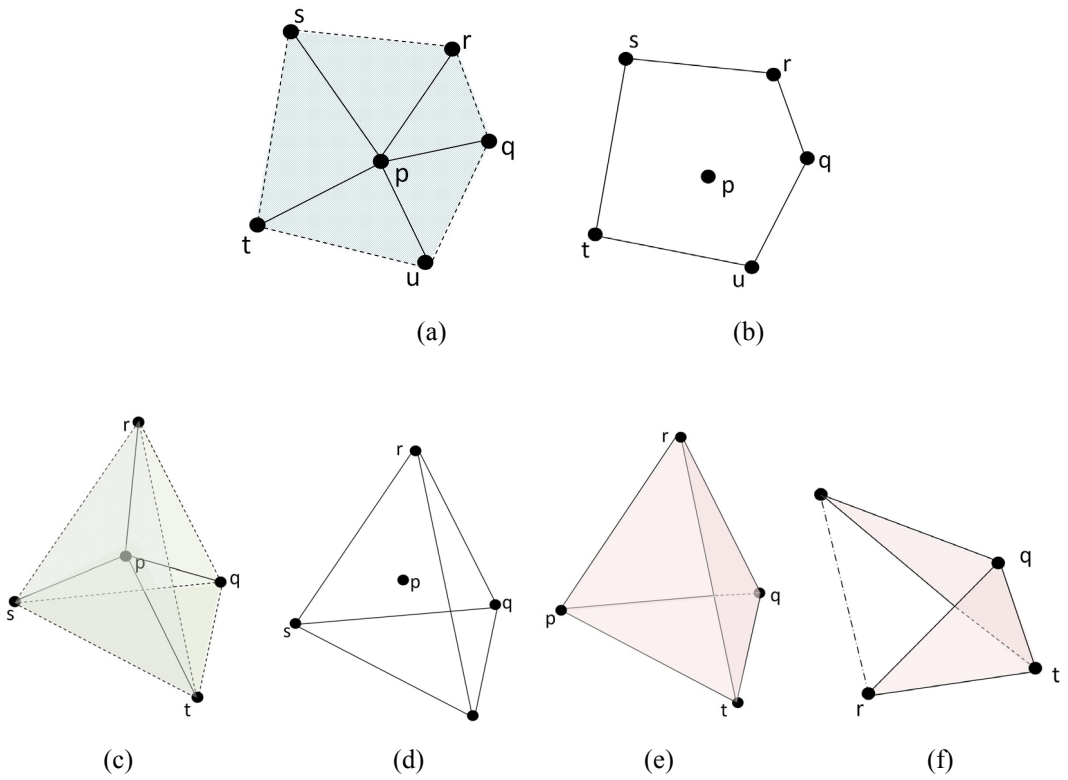


Figure 3-6 a) Star of vertex *p* in 2D, b) link of vertex *p* in 2D, c) star of vertex *p* in 3D, d) link of vertex *p* in 3D, e) an ear, which spans three adjacent faces all sharing vertex *q*, f) an ear, which spans two adjacent faces.

3.3 Properties of Voronoi and Delaunay tessellations

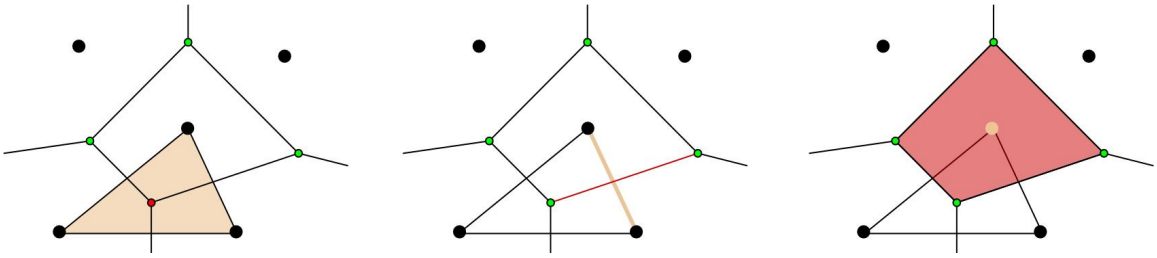
Voronoi and Delaunay tessellations have several interesting properties that make them attractive for many applications. Here, a summary of relevant properties in the context of this thesis, i.e. modeling and simulation of 3D dynamic fields is presented. Other properties of VDs and DTs can be found in Okabe et al. (2001).

a) Duality between DT and VD

There is a connection between the Delaunay and the Voronoi tessellations called duality. The duality means that VD and DT are closely related graphs. The duality between VD and DT is based on some specific correspondences between geometric elements of the two data structures. This allows extracting the Voronoi tessellation

from the Delaunay tessellation and vice versa. For a set of S points in a d-dimensional space, Delaunay tessellation can be obtained by joining all pairs of points in S whose Voronoi cells share a common (d-1) Voronoi facet. In addition, a k-dimensional face of d-dimensional Voronoi tessellation corresponds to a $(d - k)$ dimensional face in the Delaunay tessellation.

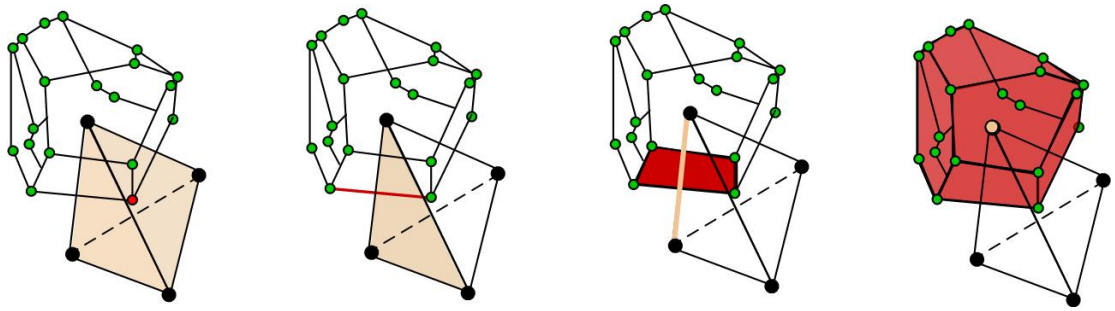
In 2D, each Delaunay triangle corresponds to a Voronoi vertex (figure 3-7a), each Delaunay edge corresponds to a Voronoi edge (figure 3-7b), and a Delaunay vertex corresponds to a Voronoi cell (figure 3-7c) and vice versa.



(a) (b) (c)

Figure 3-7 Duality between VD and DT in 2D.

Similarity, in a 3D space, each Delaunay tetrahedron corresponds to a Voronoi vertex (figure 3-8a), each Delaunay triangular face becomes an edge (figure 3-8b), a Delaunay edge corresponds to a Voronoi face (figure 3-8c), and finally, each Delaunay vertex corresponds Voronoi polyhedron (figure 3-8d) and vice versa.



(a) (b) (c) (d)

Figure 3-8 Duality between VD and DT in 3D.

It is technically easier to tessellate a continuous phenomenon using tetrahedra than arbitrary polyhedral (VD) where each tetrahedron has a constant number of vertices and adjacent elements (Icking et al. 2003). Therefore, using the duality between two data structures, the Voronoi tessellation can be easily obtained by connecting the circumsphere centers from the Delaunay tetrahedra. This property is very important in the simulation of a spatial field where the representation of the continuous field from the discrete samples (point objects) is required. Therefore, DT can adequately represent the discrete samples and their relationship while, VD can be used for representing the variation of the field properties across this data i.e. numerical integrating of PDEs.

b) Maximum-Minimum angle in DT

Another interesting property of Delaunay tessellation in 2D is that it maximizes the minimum angle among all triangulations of a given set of points (Lawson 1977). This property is important, because it implies that Delaunay triangulation tends to avoid skinny triangles. This is useful for many applications where triangles are used for the purposes of interpolation of field values. However, this property is valid in only 2D space and it does not generalize to three dimensions where tetrahedra with four almost coplanar vertices can be found. These tetrahedra are usually referred to as slivers and have a very small volume (almost zero).

c) Empty sphere in DT

A sphere circumscribing any simplex in the Delaunay tessellation does not contain any other points in S in its interior. For example, for 3D simplex *pqrs* in figure 3-5, there is an empty circumsphere passing through three points p, q,r and s. The center of this sphere is a vertex of the Voronoi tessellation between these points, because this vertex is equidistant from each of these sites and there is no other closer point. This property offers a more adequate discretization among all arbitrary triangular tessellation of the space. In addition, it is frequently used to choose the best place to

insert new points within the poor quality elements in refinement methods as discussed in the next chapter.

d) Nearest-neighbor property

If point *q* is the nearest-neighbor of point *p* in *S*, their associated Voronoi cells are adjacent cells in the tessellation and share a face (Boissonnat and Yvinec 1998). Since, according to the empty sphere criterion, the circle having these two points as its diameter cannot contain any other points. This property satisfies an important requirement in numerical simulation of a field: topology. Because, in the numerical modeling methods, motion equations (PDEs) are solved using a given tessellation elements and their neighbors. Therefore, the relationship between the elements of the tessellation (topological information) must be defined. Based on the nearest-neighbor property the adjacency relationship between tessellation elements (topological relations) is readily defined among the Voronoi cells.

e) Local optimality in DT

As mentioned previously, a Delaunay tessellation means a collection of d-dimensional simplexes that subdivide the convex hull of S in such a way that the union of all the simplexes covers the convex hull and every (d-1)-facet of the simplexes is locally Delaunay. For example in 2D, a triangulation is Delaunay if and only if its edges are locally Delaunay i.e. there is an empty circle passing through the endpoints of each edge (empty circle property). Therefore, local editing of all of non-Delaunay (d-1)-facet (edges in 2D and faces in 3D) in a tessellation (local optimality) results in a Delaunay tessellation (global optimality) (Devillers 2002a). Based on this interesting property, some local topological operations, *bistellar flips*, were developed (Joe 1991, Mucke 1998, Edelsbrunner and Shah 1996, Shewchuk 2005). These operations modify the configuration of adjacent elements to satisfy the Delaunay criterion (i.e. empty circumcircle test in 2D or circumsphere test in 3D). For example, in 2D, a *flip22* convert two neighboring

triangles *pqr* and *rtp* to two triangles *pqt* and *qrt* by changing the diagonal of quadrilateral formed by four points p, q, r, t (figure 3-9).

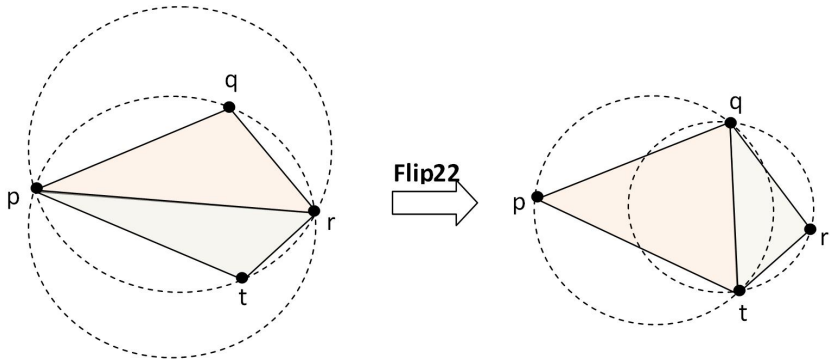


Figure 3-9 A *flip22* converts two neighboring triangles to other two neighboring triangles.

Flip14, *flip41*, *flip32, flip23* and *flip44* are examples of local topological operations in 3D (Shewchuk 1997). As figure 3-10 illustrates, a *flip14* replaces tetrahedron *pqrt* with four tetrahedra *pqst*, *qrst*, *srpt*, *pqrs* by connecting the point *s* to the vertices of the tetrahedron (*p,q,r,t*) and a *flip 41* converts inverse problem. The *flip41* and *flip14* have the effect of inserting or deleting a new point (*s*) in a tetrahedron, respectively (Shewchuk 1997).

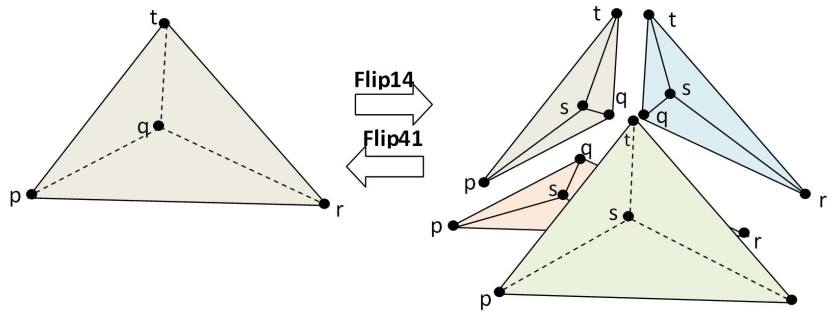


Figure 3-10 *Flip14* converts one tetrahedron to four neighboring tetrahedra and *flip41* converts four neighboring tetrahedra to one tetrahedron.

A *flip23* or face-to-edge flip operator converts two neighboring tetrahedra (tetrahedra *pqrs*, *qprt* in figure 3-11) to three tetrahedra (tetrahedra *pqts*, *ptrs*, *rqts*) and a *flip32* or edge-to-face *flip* operator converts three neighboring tetrahedra to two with respect to the Delaunay criterion.

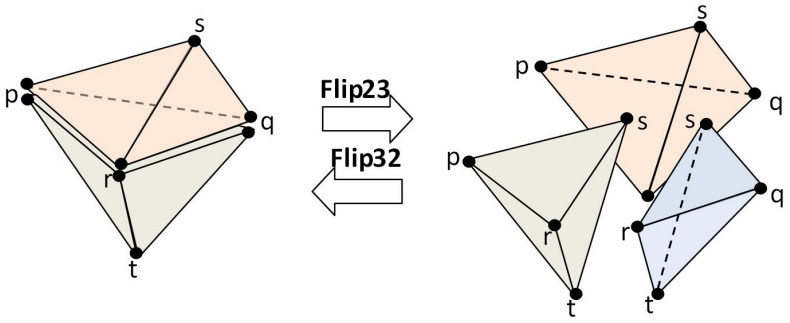


Figure 3-11 A *flip23* converts two neighboring tetrahedra to other three neighboring tetrahedra and a *flip32* converts three neighboring tetrahedra to other two neighboring tetrahedra.

A *flip44* replaces four tetrahedra *rtps*, *qtrs*, *utrq*, *rtup* with four tetrahedra *pqts*, *prqs*, *ptuq*, *pqur* by changing edge *tr* to *pq*.

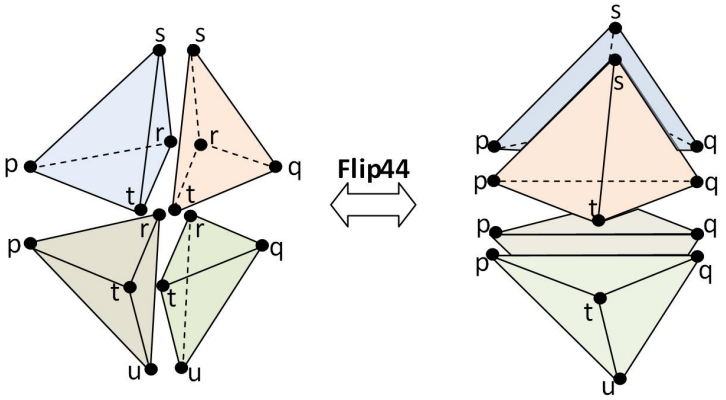


Figure 3-12 A *flip44* converts four neighboring tetrahedra to other four neighboring tetrahedra.

This property of DT and the dynamic local operations allow dynamically editing and manipulating the tessellation, such as removing or inserting faults or modifying the location of wells in a hydrogeological system without having to rebuild the whole tessellation.

f) Convexity property of VD

Each Voronoi cell is a non-empty convex polyhedron in d-dimensional space (Boissonnat and Yvinec 1998). The convexity property of Voronoi cells facilities

some computations during the simulation of continuous fields. For example, to compute the forces from the neighboring *s* acting on element *p*, the area of an irregular convex polygon separating *p* from its nearest neighbor *q* is required which can be found by dividing it into triangles and summing the areas of the triangles. The volume of a convex polyhedron can be found by dividing it into tetrahedra and summing the tetrahedra's volumes.

g) Delaunay triangulation and convex hulls

As mentioned previously, the convex hull of a set of points is the smallest convex set containing the points. A Delaunay tessellation is mathematically related to convex hulls. The boundary of the exterior faces of the Delaunay tessellation is the boundary of the convex hull of the point set *S*. In fact, the Delaunay tessellation of a point set S in R^d can be viewed as the projection of a convex hull of the projections of the points onto a (d+1)-dimensional paraboloid (Farin et al. 2002).

For example, in 2D, if we "lift" the points S onto the paraboloid of revolution, $z = x^2 + y^2$, we get a set S' of points in 3D (figure 3-13). Therefore, for each point in the plane, $p(x, y)$, the vertical projection of this point onto this paraboloid is $p'(x, y, x^2 + y^2)$ in 3D space. With this transformation, three points p', q', r' defines a *face* of the convex hull of S' if and only if pqr is a triangle of the Delaunay triangulation of S. This condition is verified via a well-known *Incircle* test (Guibas and Stolfi 1985) that determines if point t is inside, outside or on the circumcircle of three points p, q, r. Point t lies within the circumcircle of *p, q, r* if and only if t' lies on the lower side of the plane passing through *p', q', r'*. This condition can be tested via a determinant computation:

$$H = \begin{vmatrix} t_x & t_y & t_x^2 + t_y^2 & 1 \\ p_x & p_y & p_x^2 + p_y^2 & 1 \\ q_x & q_y & q_x^2 + q_y^2 & 1 \\ r_x & r_y & r_x^2 + r_y^2 & 1 \end{vmatrix} \qquad (3\text{-}7)$$

A positive H shows that t is in exterior of the circumcircle of *pqr* and a negative H shows that the circumcircle of *pqr* is not empty and contains point t.

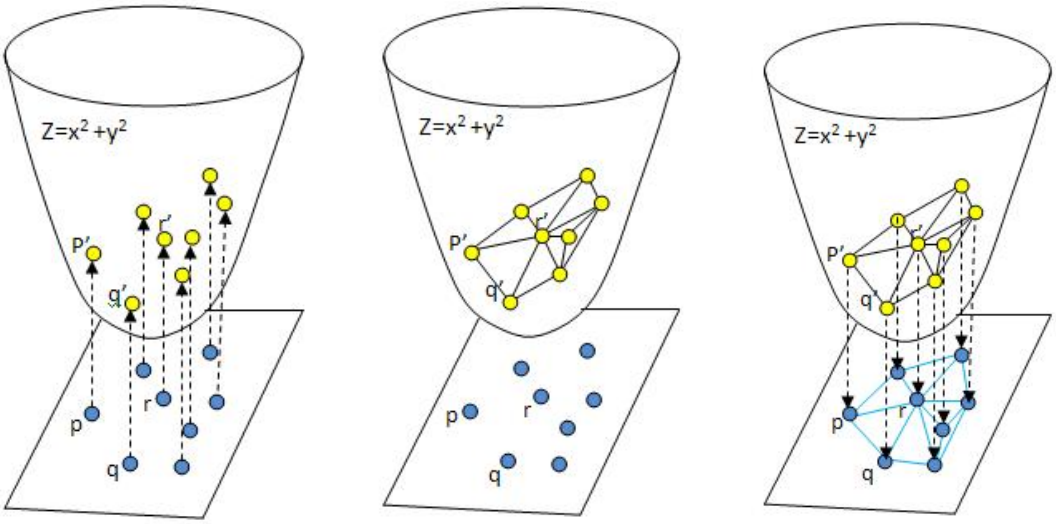


Figure 3-13 Delaunay tessellation of point set in a plane can be viewed as the projection of a convex hull of the projections of the points onto a 3D dimensional paraboloid[4].

Similarity, in a 3D space, point t lies inside, on, or outside of the circumsphere of four non-coplanar points p, q, r and s depending on whether the sign of *Insphere* test is negative, zero or positive. The sign of the *Insphere* test is determined by the sign of the determinate H defined as follows.

$$H = \begin{vmatrix} t_x & t_y & t_z & t_x^2 + t_y^2 + t_z^2 & 1 \\ p_x & p_y & p_z & p_x^2 + p_y^2 + p_z^2 & 1 \\ q_x & q_y & q_z & q_x^2 + q_y^2 + q_z^2 & 1 \\ r_x & r_y & r_z & r_x^2 + r_y^2 + r_z^2 & 1 \\ s_x & s_y & s_z & s_x^2 + s_y^2 + s_z^2 & 1 \end{vmatrix} \qquad (3\text{-}8)$$

[4] The figure from http://www.cs.wustl.edu/~pless/546/lectures/L13.html.

3.4 Construction of the Delaunay and Voronoi Tessellations

The methods for the construction of Delaunay and Voronoi tessellation of a set of points includes, those that use the divide and conquer paradigm (Avis and E1Gindy 1987, Edelsbrunner et al. 1986), basic sweepline (Fortune 1986) and the incremental approach (Guibas et al. 1992). Although these methods are largely used to construct DT and VD, however, only the incremental approach is considered to be dynamic in computer sciences. This means that with using an incremental method, it is not necessary to know the whole data set prior to the construction of a tessellation. New points can be inserted in the tessellation or deleted from it at any time, and then local modification is allowed after any change in the tessellation. This offers to explore interactively a data set to extract information while the static algorithms lack the ability to do it. For example, for a hydrogeological system, it is possible to refine the tessellation in areas where the hydraulic gradient is high, (e.g. in the vicinity of a pumping or injection wells) or to remove/insert faults or to modify the location of wells without having to rebuild the whole tessellation.

Here, we discuss the incremental construction of a Delaunay and Voronoi tessellation in 3D. As previously mentioned, since the construction of a Delaunay tessellation is technically easier than a Voronoi tessellation, we explain the construction of Delaunay tessellation and then, by knowledge of the duality between two data structures, the Voronoi tessellation will be extracted.

3.4.1 Incremental construction method of a 3D Delaunay tessellation

The incremental method for the construction of a Delaunay tessellation, consists of three important stages including initialization, point insertion and the tessellation optimization. In the initialization step, an initial tetrahedron large enough to enclose all the given points is constructed. In the point insertion step, the points are added

into the tessellation one at a time and using the *Walk* operation, the tetrahedron containing the new point is identified. To find out which triangle the new point *a* lies in, *Walk* operator starts to search from a given tetrahedron T. The search then identifies the neighbor of T such that the query point *a* and T are on each side of the face shared by T and its neighbor (for example, face *pqr* in figure 3-14). This condition is verified via the *orient* determinant defined as follows:

$$orient(pqra) = \begin{vmatrix} p_x & p_y & p_z & 1 \\ q_x & q_y & q_z & 1 \\ r_x & r_y & r_z & 1 \\ a_x & a_y & a_z & 1 \end{vmatrix} \qquad (3\text{-}9)$$

If there is a positive sign of this *orient* determinant, the tetrahedron containing *a* is T. Otherwise, the procedure is repeated for the adjacent tetrahedron whose face has negative sign for its *orient* determinant.

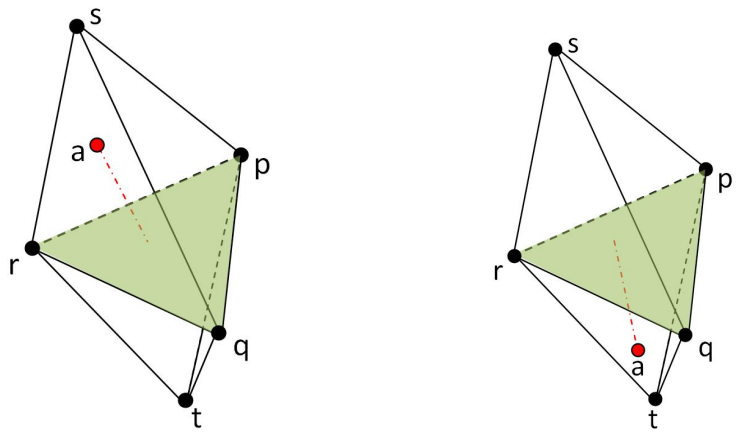


Figure 3-14 a) Tetrahedron *pqrs* contains point *a*, since the Orient determinant for *a* and all of its faces are positive, b) tetrahedron *pqrs* dose not contains point *a*, since the Orient determinant for *a* and face *pqr* is negative.

For example, in figure 3-14a, the O*rient* determinant for point *a* and faces *pqr*, *psq*, *rqs*, *prs* are positive then, tetrahedron *pqrs* contains point *a.* In figure 3-14b, the *Orient* determinant for point *a* and faces *psq*, *rqs*, *prs* are positive, but it is negative for face *pqr*, then, tetrahedron *pqrs* does not contain point *a*. The procedure is repeated for tetrahedron *trqp* whose face *pqr* has a negative sign for its orient

determinant. The walk function continues the search operation until a tetrahedron, having a positive sign for all its *Orient* determinants, is found.

Finally the tessellation is updated locally in the tessellation optimization step. A straightforward method is Bowyer-Waston method where all tetrahedra whose circumspheres contain new point *a* are removed (Joe 1999). For this purpose a *neighbor* search is carried out from tetrahedron T which contains *a*. *Neighbor* search examines the four adjacent tetrahedra of the tetrahedron. Then, those tetrahedra whose *Insphere* determinant (equation 3-8) would be negative by new point *a* are removed. The resulting hole is star-shaped around point *a* which is filled by connecting *a* to all points on its link with new edges. Although the method is simple, it is not robust against floating point round off errors (Shewchuk, 1997).

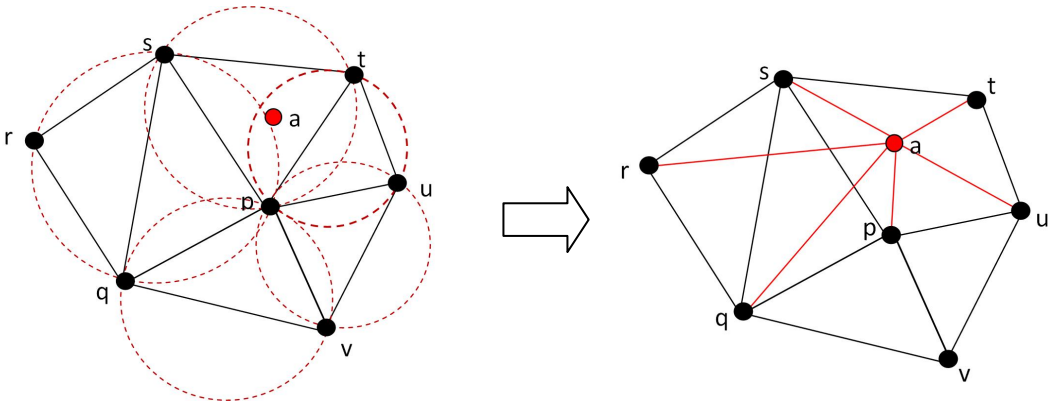


Figure 3-15 Bowyer-Waston method is not robust against floating point round off errors.

To avoid this problem, the *flip*-based methods such as Lawson's algorithm can be helpful. Therefore, to construct or modify a DT, several dynamic operations are required (Edelsbrunner and Shah, 1996; Miller and Pav 2002). For example, when new point *a* is added into the tessellation, the *insert* operation using a *flip14* splits the tetrahedron containing *a* into four new tetrahedra, each having *a* as their vertex. Then, using optimization operator such as *flip32*, *flip23* and *flip44*, the configuration of adjacent tetrahedra is modified to satisfy the empty circumsphere test. This condition is verified via an *Insphere* test (equation 3-8). Figure 3-16 illustrates the flowchart of the flip-based Delaunay construction.

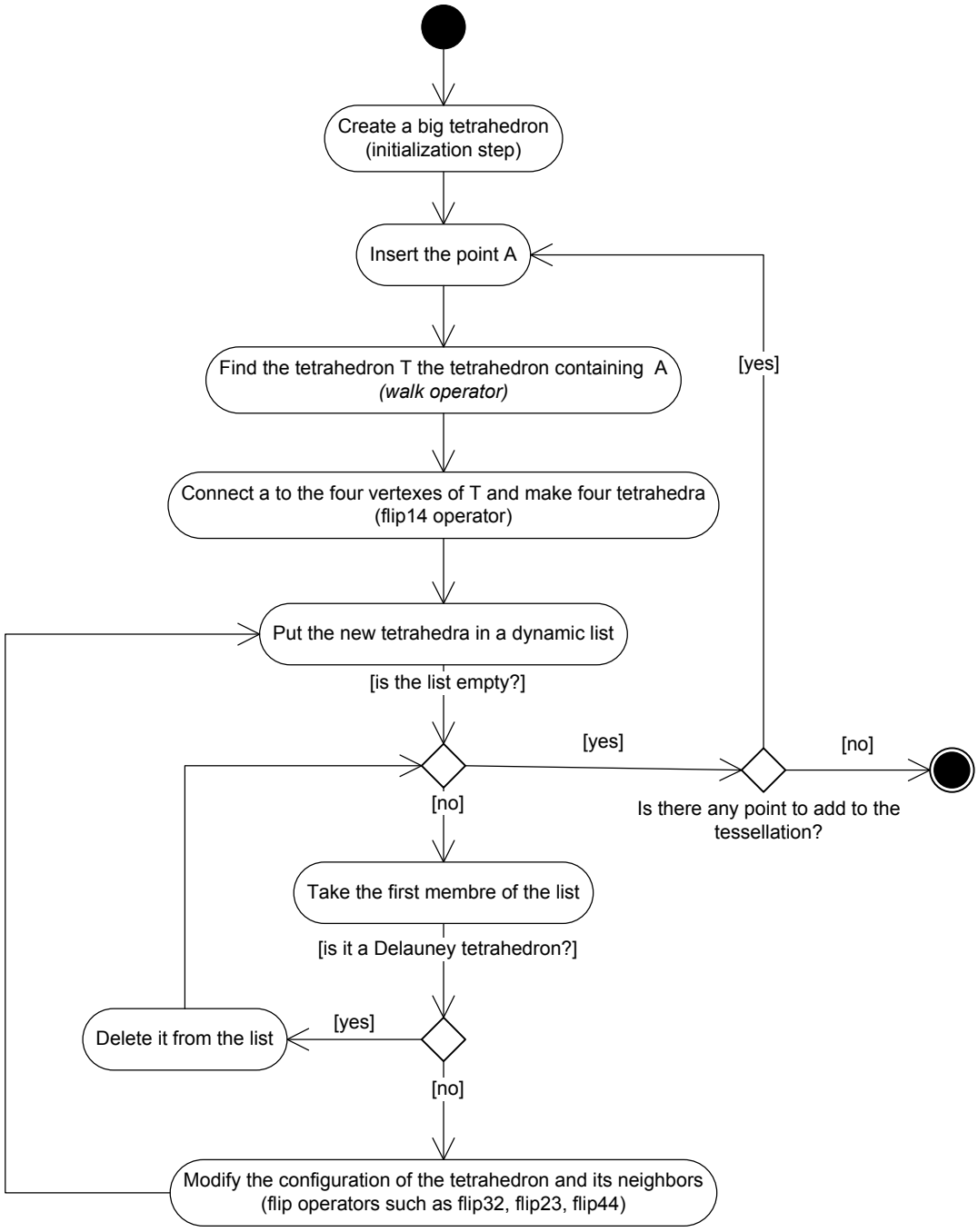


Figure 3-16 The flowchart of an incremental flip-based Delaunay construction.

As mentioned previously the data structure is dynamic, and it is possible to delete a point from the structure using a local operation. Studies on this topic have been carried out by a number of researchers that are mostly for 2D (Devillers 1999, Heller 1990, Mostafavi 2002), and very little can be found for 3D (Devillers and

Teillaud 2002, Ledoux 2006). In the literature, the deletion of a vertex p in a DT(S) is mentioned as the 'inverse' problem of the incremental insertion algorithm, i.e. DT(S \{p}). In these methods, to delete a point *p*, the simplices incident to p are removed and merged which make a hole. In the second step, the point p is deleted and the hole is triangulated. For More details please see Mostafavi (2003) and Ledoux (2006).

3.4.2 Extracting Voronoi tessellation from Delaunay tessellation

Based on the duality between DT&VD, a Voronoi tessellation can easily be extracted from its dual i.e. Delaunay tessellation:

- A *Voronoi vertex* is dual to a Delaunay tetrahedron in which it is equidistant to the four vertices of this Delaunay tetrahedron. Therefore, a Voronoi vertex is the center of the empty sphere of its dual.
- A *Voronoi edge* is dual to a Delaunay tetrahedron face connecting two Voronoi vertices. Therefore, a Voronoi edge is obtained by connecting the circumsphere centers of two tetrahedra sharing this face.
- A *Voronoi face* is dual to a Delaunay tetrahedron edge connecting arbitrary Voronoi vertices. Therefore, a Voronoi face is obtained by connecting the circumsphere centers of the tetrahedra sharing this edge.
- A *Voronoi polyhedron* is dual to a Delaunay tetrahedron vertex connecting arbitrary Voronoi vertices. Therefore, a Voronoi polyhedron is obtained by connecting the circumsphere centers of the tetrahedra sharing this Delaunay vertex.

3.4.3 Data storage structures for DT

There are several data structures that can be used for storage and navigation through a 3D triangular tessellation. These data structures store the geometrical and

topological information of a Delaunay tessellation using either 1) a record to represent each tetrahedron (tetrahedron-based structure), 2) a record to represent each face, or 3) a record to represent each pairing of a face and an edge, requiring three records per triangular face (Shewchuk, 1997). Among these data structures, tetrahedron-based data structures have been frequently used as they can associate attributes to elements and vertices and have also the lowest memory requirements (Shewchuk, 1997).

a) Tetrahedron-based data structure

In this data structure, for every tetrahedron, data storage consists of four pointers for its vertexes and four pointers to its four adjacent tetrahedra. It should be noted that to navigate and manipulate the tessellation, the vertices and faces of each tetrahedron must be considered in a consistent way. For example, the data structure can be consistent with the left-hand rule, when the ordering of *p*, *q* and *r* follows the direction of rotation of the curled fingers of the left hand, then the thumb points towards *s* (figure3-17). This can be expressed as the positive sign of a determinant depending on the locations of the points and is referred to as Orient determinant (Guibas and Stolfi 1985):

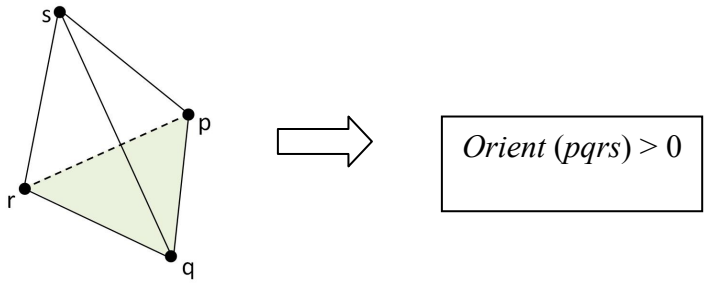


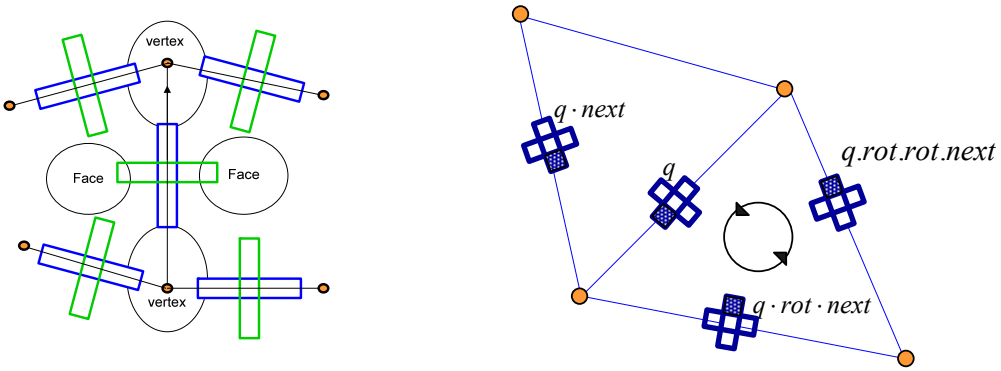
Figure 3-17 Orient test: the vertices of each tetrahedron are considered in a consistent way.

The tetrahedron-based data structure is simple and fast to implement. In this model, when boundary or volume analysis is needed (for example, in fluid flow simulations), faces, edges and Voronoi cells are extracted afterwards, which reduces the amount of required storage memory. However, the Voronoi diagram

cannot be explicitly stored using this data structure and hence, we need to recompute the Voronoi diagram.

b) Augmented Quad-edge

To manage 3D DT and VD simultaneously an extension of the Quad-Edge data structure of Guibas and Stolfi (1985) to 3D can be used. The Quad-Edge data structure can manage DT and VD because it stores both the primal and the dual data structures simultaneously and offers a simple and elegant way to navigate within a tessellation in 2D space. For each edge of the tessellation, the Quad-Edge data structure stores four directed edges (primal edges), two in the primal subdivision (DT), and two in the dual (VD). The directed edges point to two vertices of an edge (figure 3-18a). Every quad has three pointers (*next, org, rot*) that allow navigation from edge-to-edge through the tessellation by means of its algebraic operations. *Org* points to the vertex at the origin (q.org); *rot* points to the dual edge or the next quad counterclockwise in the same quad-edge (q.rot); and *next* points to the next edge counterclockwise around the origin in the tessellation (q.next) (figure3-18b).



(a) (b)

Figure 3-18 a) Each quad-edge stores four directed edges: 2 in the primal subdivision and 2 in the dual, b) Navigation operators of Quad-Edge in 2D.

The Augmented Quad-Edge (AQE) data structure (Gold et al. 2005, Ledoux and Gold 2007) is a theoretical extension of the Quad-Edge data structure to 3D. In the AQE, each element, either a tetrahedron or a Voronoi polyhedron, is stored

individually and the elements are then connected by the dual edge to the face shared by two elements. In this approach, for each AQE, there exist two types of quads: a quad pointing to a vertex (q_v) and a quad pointing to a face (q_f), because the dual of an edge is a face (figure3-19). Therefore, one *rot* operation applied to a q_v returns a q_f and vice-versa. Hence, with the aid of the AQE, any point, edge, face, and region of DT and VD can be retrieved.

To implement an incremental algorithm based on the AQE, some important operators for creation, modification and assemblage are required. They are topological operators that can be used to create a cell (for example with make-edge and splice-edge), modify an existing cell when using a topological operator (flips) and link two cells, respectively. Please see the Ledoux (2006) for details on construction a tessellation using an augmented quad-edge and navigation though it.

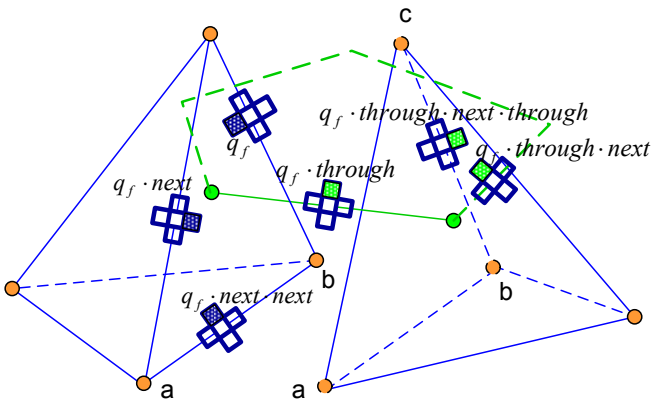


Figure 3-19 Navigation operators of the augmented Quad-Edge data structure (Gold et al. 2005).

Theoretically, the AQE offers a good trade-off between storage costs and the number of topological relationships between different elements of a subdivision (Gold et al. 2005). The main disadvantage is that the data structure is very costly in terms of the required storage memory. Because it is required to store 114 pointers for each tessellation element: 72 for a tetrahedron and 72 for its dual, since each tetrahedron has 6 edges and for each edge there are 4 quads containing three pointers (*next*, *org,* and *rot*) are stored. In comparison, the tetrahedron-based

structure uses only eight pointers per tetrahedron (Ledoux, 2006). However, this structure does not store any information concerning the dual tessellation.

3.4.4 Degeneracy cases

Up to here, we have assumed that the data points are in general position. If the points are not in general position, in the sense that (d+2) or more are co-ball, then the Delaunay tessellation is not unique and there are different tessellations that respect the Delaunay criterion. For example, for four co-circular points in 2D space, there are two different triangulations that respect the Delaunay criterion (figure 3-20). Similarity, for a 3D DT, this ambiguity occurs when five or more points lie on the same sphere, called co-spherical points, consequently the DT is not unique. These cases are called degeneracy cases.

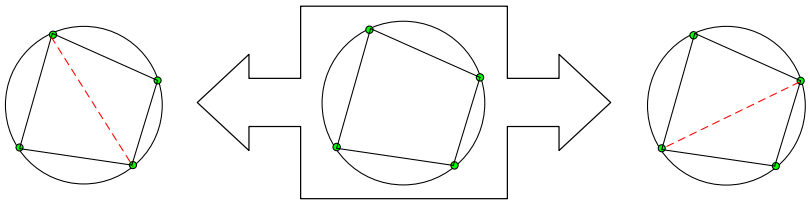


Figure 3-20 Two different Delaunay triangulations for four co-circular points in 2D space.

It is common to assume that the points are in a general position. Otherwise, the general position may be created using perturbation methods. A perturbation is usually defined as small deviation (ε) of the input data. Although a perturbation can almost remove degeneracy cases (Edelsbrunner 2001), it has some serious drawbacks. An important limitation with perturbation methods lies on needing an exact arithmetic to preserve the topological consistency (Sugihara 2007). As Sugihara mentioned, for the points configuration near to degeneracy, it is difficult to correctly judge the topological structure in presence of floating point round off errors, and hence a high precision computation is required to avoid the inconsistency. For instance, because of floating point round off errors, the points in

a configuration very close to co-spherical points could be judged as non-co spherical points and vice versa. Moreover, a perturbation method may take more time for computation or create some undesirable results. For example, in the problem of point intersection of n line segments, a perturbation may change the number of intersections from one to $O(n^2)$.

Another important example of degeneracies in a Delaunay tessellation is the case of coplanar points. A perturbation can cause flat tetrahedra or slivers in the tessellation. For this purpose, several works propose the perturbation of the points in the lifted space which is corresponding to the perturbation of the radii of the spheres (Devillers and Teillaud 2003, Facello 1995, Mucke 1998). However, using this perturbation, called *"hyper-perturbation" (*Sugihara, 2007), the problem change and we construct a DT for a given set of spheres instead of the given data points where point *t* (see equation 3.8) is mapped to a paraboloid in a four dimensional space by $(t_x, t_y, t_z, t_x^2 + t_y^2 + t_z^2 + \varepsilon, 1)$ instead of $(t_x, t_y, t_z, t_x^2 + t_y^2 + t_z^2, 1)$.

In addition to perturbation methods, there are attempts to handle generacies using *bistellar flips* such as a *flip32* and *flip23* for *cospherical* points and a *flip44* for *coplaner* points (Joe 1991, Mucke 1995, Edelsbrunner and Shah 1996, Shewchuk 2005, Ledoux 2006). We will discuss the degeneracy cases which arise during the development of the kinetic data structure in this research work.

3.5 Conclusion

This chapter discussed 2D and 3D Voronoi tessellation for a set of points in Euclidean pace, and its dual, 2D and 3D Delaunay tessellation. We have presented their definitions and some of their interesting spatial properties that will be used in this research work. In addition, we briefly explained the construction of a 3D dynamic Delaunay tessellation where local modifications are allowed after any change in the tessellation (insert and delete point objects). The DT can be stored by tetrahedron-based data structure which is simple and fast to implementation.

However, it requires extracting the 3D VD from DT afterward using the duality property. To manage 3D DT and VD simultaneously, Augmented Quad-Edge which is a theoretical extension of the Quad-Edge data structure to 3D can be used. However, this data structure is very costly in terms of the required storage memory. In the next chapter, we construct an adaptive tessellation for geosciences applications by developing an adaptive refinement method.

Chapter 4: Construction and Refinement of an Adaptive Tessellation for Representation of 3D Continuous Fields

4.1 Introduction

As it was mentioned in the previous chapters, the simulation of a 3D spatial dynamic field usually requires a 3D spatial tessellation to approximate the dynamic behavior of the field, which is described by a set of partial deferential equations (PDEs) over time and space. The tessellation must adequately represent the geometrical and physical complexity of the field. For this purpose, the elements of the tessellation must be 3-dimensional and irregular to provide conformity to the outline of the field data. The data are usually 3D points with irregular distribution where each data point is defined by its geometric position in 3D space (x, y, z) and can have one or more attributes attached to it such as soil type or temperature (Davis and Davis 1988, Gold et al. 2004).

The size and the shape of the tessellation elements have an important impact on the accuracy of the results of the simulation of a field as well as the computational

costs (Mansell et al., 2002). This is because, the solution of PDEs are obtained by approximating the dynamic behavior of the field for each tessellation element. Therefore, for specific regions with high variation of the field properties, a tessellation with fine resolution is needed. However, because the time required to solve the PDEs is proportional to the number of tessellation elements, a uniformly fine tessellation is computationally costly. Consequently, the specifications of a tessellation, such as its resolution and shape, are critical issues that remain among the most challenging problems in the simulation of 3D dynamic fields. However, the optimum resolution of a tessellation cannot usually be determined a priori (Shewchuk, 2002). Hence, automatic generation of an optimal tessellation with well-shaped elements and suitable resolution is sometimes more challenging than the numerical modeling process itself (Bower et al. 2005, Lepage 2003, Shewchuk 2002, Verma and Aziz 1997).

Many methods have been proposed for automatic tessellation generation and refinement for interpolation, rendering, and numerical modeling and simulation purposes (Shewchuk 2002). Most of these methods have been developed in other disciplines than the geosciences (ex. Computational Geometry) and are mostly based on theoretical considerations and are rarely application-oriented where the physical nature of fields (properties) plays an important role in defining the shape and resolution of the tessellation elements. Each application has its own requirements that must be considered by the selected tessellation generation algorithm (Kosik et al. 2000, Cordes and Putti 2001). This chapter explains a new method for automatic generation and refinement of an adaptive tessellation for geosciences applications such as fluid flow simulation in a hydrogeological system that takes into account the geometrical, topological, and physical properties of a field.

4.2 The need for refining the 3D Delaunay and Voronoi tessellations

In chapters 2 and 3, our literature review demonstrated that 3D VD and DT provide an adequate tessellation for representation of a 3D continuous space (see Aurenhammer 1991, Shewchuk 1997, Okabe 2001, Mostafavi 2002 and Ledoux and Gold 2008, Hashemi et al 2008, Hashemi and Mostafavi 2008). Delaunay tetrahedra and Voronoi polyhedra can be defined by data points with an arbitrary distribution, creating elements of different sizes and shapes that can fill any arbitrary surface or volume. In addition, each Voronoi polyhedron can have an arbitrary number of neighbors which their connectivity with the given element is clearly defined and can be explicitly retrieved if needed. Regarding these properties, several research works used 3D VD and DT as underlying tessellation in field simulation. Hale (2002) applied DT and VD to reservoir simulation using 3D seismic images and demonstrated the potentials of both DT and VD for flow simulation during all steps of seismic interpretation, fault framework building, and reservoir modeling. Carette et al. (2008) used 3D DT&VD for marine datasets. They used a 3D DT in cooperation of a α-shape clustering algorithm to acoustic fisheries data and represented and analyzed fish aggregations.

In spite of these mentioned advantages, the use of the VD and DT to model three-dimensional geoscientific datasets that are not randomly positioned is very challenging (Lattuada and Raper 1995, Raper 2000). For example, for simulation of ground water flow in a hydrogeological system, the system may be sampled by data collected either directly from prospective drills, well logs, or boreholes, or indirectly from seismic, magnetic and gravity surveys. This data form highly irregular set of points aligned in vertical direction and very sparse horizontally, which must represent the continuous field (groundwater) within a hydrogeological system composing of geological objects such as fractures, faults, and layers (fig 4-1).

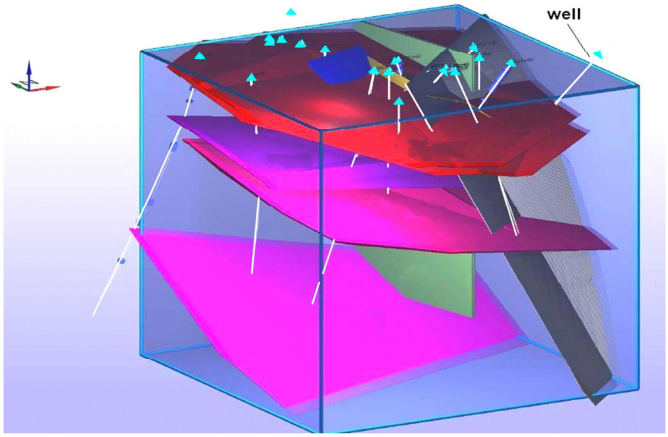


Figure 4-1 Example of a hydrogeological system (Ahokas and Koskinen, 2005). The system may be sampled by data collected directly from boreholes which are shown as white lines.

A 3D Voronoi tessellation of this data does not deal with the discontinuities within a field. This is because in a VD, a volume of influence is assigned to each point and the field value is assumed to be constant within the volume. Some works have used a DT to consider the discontinuity (Lattuada and Raper 1995, Schaap and deWeygaert 2000). However, the DT may not also represent the physical discontinuities very well where the vertices of the tetrahedra lie on the sampled points and a linear interpolation is done for computing the field value within each tetrahedron.

In addition, the DT of the data may not satisfy shape and size criteria (Shewchuk, 1997). For example, the tetrahedra may be smaller than required or larger than a desired size, then, the elements are not "well-spaced" or the angles within tetrahedra may be too small or too large then the elements are not "well-shaped" (figure4-2). These tetrahedra are not adequate for numerical integration or simulation, where the tetrahedra are used to form a piecewise linear approximation of a function based on physical variables (permeability, porosity), and their sizes and shapes thus effect the accuracy of the approximation (Shewchuk, 1997). These are why a refinement processing is usually applied following the initial tessellation construction or a low accuracy simulation (figure 4-3). In fact, the tessellation

refinement aims to obtain good quality (well-shaped and well-spaced) tessellation elements for a given application.

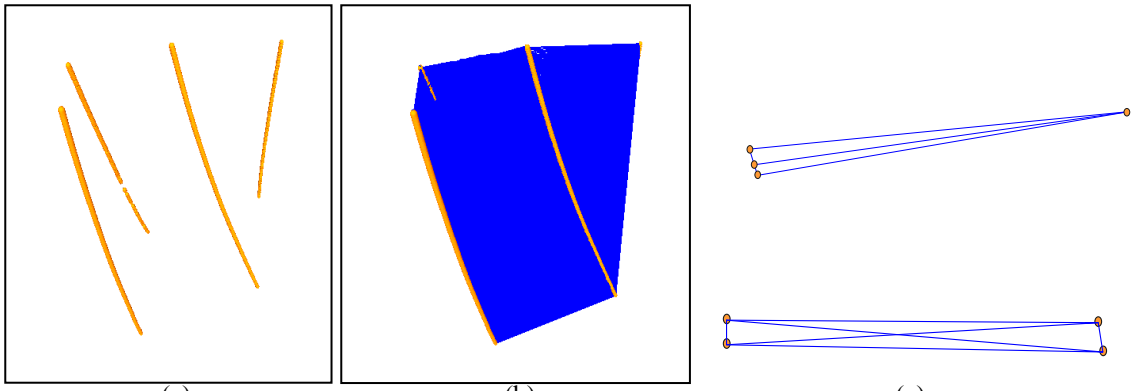


Figure 4-2 a) sampled points that do not have normal distribution, abundant vertically but very sparse horizontally, b) Delaunay tetrahedralization of these data, c) the element with poor quality is frequently existed in the tessellation.

In the existing Delaunay-based refinement methods, the first step is the construction of a Delaunay tessellation conforming to the data. Next, the poor quality tetrahedra are detected and refined by adding additional points in the tessellation. According to the local optimality property of DT (see section 3.2e), the refined tessellation remains Delaunay if and only if the circumball of any simplex (edges, faces and tetrahedra) in the DT is empty.

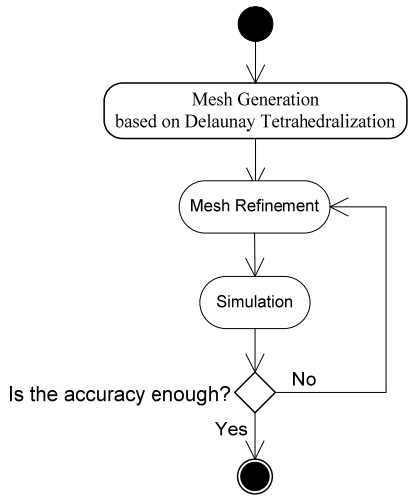


Figure 4-3 Tessellation refinement is usually applied following the initial tessellation construction or a low accuracy simulation.

In some refinement methods, here called ‘static’ methods, when an additional point is inserted into the tessellation, the tetrahedron containing the point is split in four tetrahedra, each having the new point as their vertex, and no local Delaunay updating is allowed. Therefore, static refinement methods produce a non-Delaunay tessellation. This problem is solved in ‘dynamic’ methods where the tessellation is locally updated after each point insertion by *bistellar flips* which is discussed in section 3. Here, we will discuss a dynamic refinement tessellation for geoscientific application.

4.3 Adaptive tessellation refinement

We develop a method for generation and refinement of an adaptive tessellation. The proposed tessellation will be conformed to represent the complexity of fields, considering the discontinuities and the shape and size criteria. The method consists of three stages namely refinement preprocessing, processing and post-processing (figure 4-4).

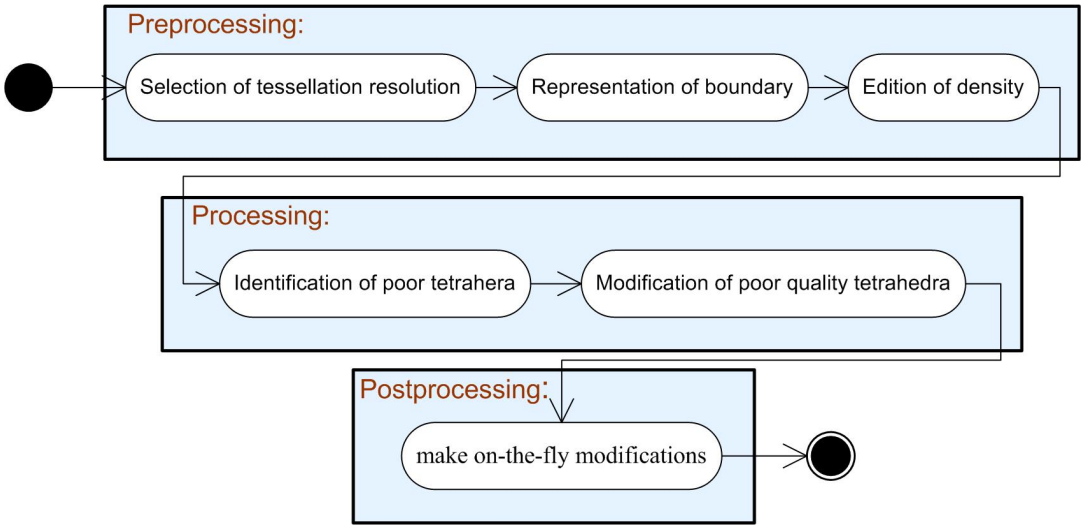


Figure 4-4 The steps of the proposed adaptive tessellation refinement.

4.3.1 Refinement preprocessing

In preprocessing stage, a desired resolution for the tessellation is selected, the boundaries are defined and the density of the data is edited. These actions are described in more details in the following sub-sections:

4.3.1.1 Resolution selection

The tessellation resolution is a criterion to reach a given amount of accuracy of simulation depends upon the behavior of the physical phenomena (fields) being modeled and may vary throughout the spatial framework. For example, a high-resolution tessellation is required in the regions of the simulation domain where a significant change of the variables (hydraulic head, solute concentration or temperature) is observed. These regions may be located, for example, near pumping wells or fractures. Also, a tessellation with lower resolution for the smooth regions is sufficient. The resolution required in a particular region depends on many factors such as the specific physics of the problem, the geometrical complexity of the spatial framework, the variation in field properties, and the boundary conditions (Bower et al., 2005). Using these criteria, a shortest and largest length for the tetrahedra edge ($L_{\min}, L_{\max}$) are defined. In addition, the minimum and maximum variation of the field properties along a tetrahedra edge ($\Delta f_{\min}, \Delta f_{\max}$) is selected.

4.3.1.2 Boundaries representation

As mentioned, it is important to represent the discontinuities within a given field and the field boundaries with its surrounding environment. For example, for a hydrogeological fluid flow representation we need to consider the faults and folds and other discontinuities in a given region. As another example, in a marine environment, we need to consider the air-water boundary layer, coasts, islands and other oceanographic boundaries. The boundaries are important features that separate the regions with different properties. In geosciences, the boundaries are usually defined by various methods such as a set of points with enough density

obtained via the methods of surface representation including non-uniform rational B-Splines or by a regular grid of points. Many approaches triangulate the boundaries prior to tetrahedralization in a separate step (Shewchuk 1997) (figure 4-5).

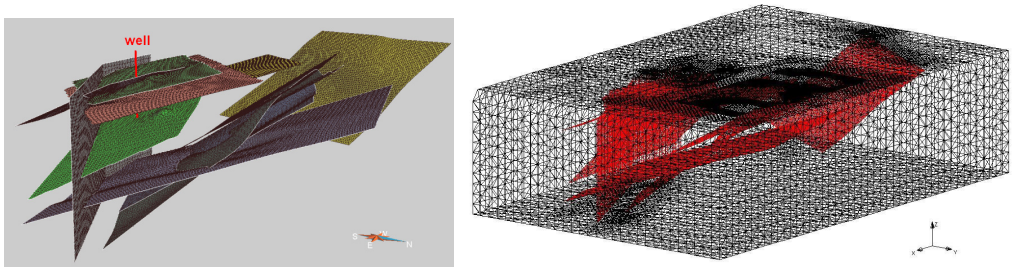


Figure 4-5 An example of discontinuities and boundary representation in a hydrogeological system (fracture network, see Blessent et al.2008) and its representation in the tessellation.

To minimize the number of the required points representing the boundaries on one hand and to respect the given boundaries during construction and refinement the tessellation on the other hand, it is possible to consider the boundaries with a minimum number of points (coarse triangulation) and refine them during the refinement process. This task is done by constrained Delaunay methods. In these methods the refinement process is governed based on rules such as *encroached* linear boundary and e*ncroached* facet (Ruppert 1995, Shewchuk 1997, Si 2005):

- *Encroached linear boundary*: A linear boundary is *encroached* if a point lies within its diametric sphere. The diametric sphere of a linear boundary is the smallest sphere containing the line. The encroached linear boundary is immediately split by inserting a point at its midpoint which followed by a local updating of the tessellation. The two resulting lines have smaller diametral spheres, and may or may not be encroached themselves (figure4-6).

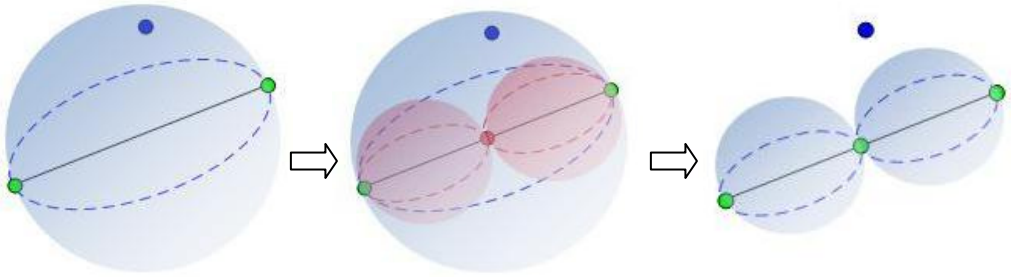


Figure 4-6 Any encroached linear boundary is split by inserting a point at its midpoint.

- *Encroached face boundary*: A facet is *encroached* if a point lies within its *diametric sphere*. The *diametric sphere* of a facet is the smallest sphere that contains the facet. The encroached is split by inserting a point at its circumcenter provided the new point does not encroach upon any linear boundary. Otherwise, the point is not inserted and the linear boundary is encroached upon its split (figure4.7).

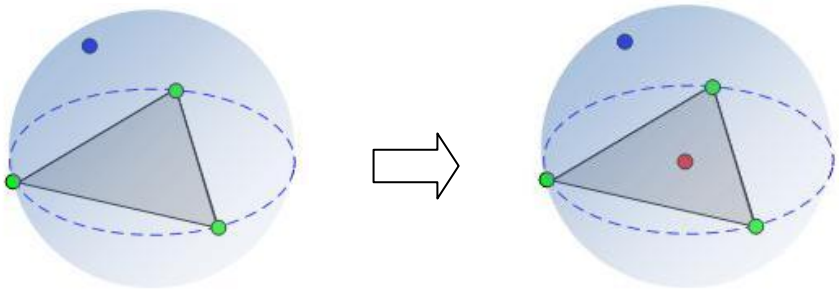


Figure 4-7 Any encroached facet boundary is split by inserting a point at its circumcenter.

However, in some other approaches, refinement process may be constrained as the boundary elements (facets) must remain unchanged during the tetrahedralization (non-Delaunay). In addition to the constrained Delaunay tessellation (CDT), there are several other methods that represent discontinuities and boundaries of the field by considering a set of points on each side of them (Mostafavi and Gold 2004, Courrioux et al. 2001, Okabe et al. 2000). In these methods, the density of the points representing the boundaries has to be high enough to ensure that tetrahedra will not extend beyond the boundaries or discontinuities. In our method, the points

density for the boundaries is selected as the length between two neighbor boundary points is equal to the shortest edge in the tessellation ($L_{\min}/2$), then, the boundary facets remains unchanged during the tetrahedralization (figure 4-8).

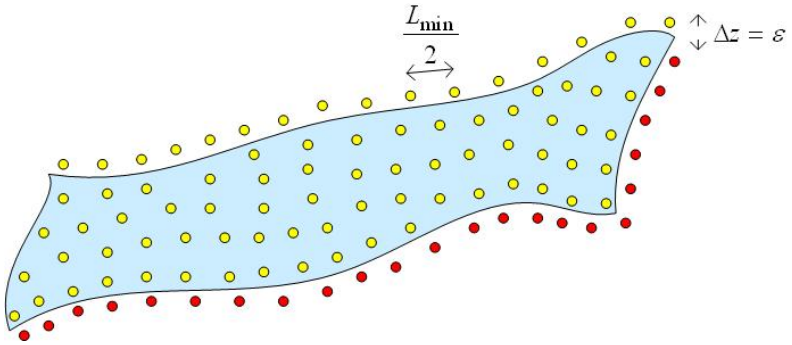


Figure 4-8 A boundary is represented by a set of points on its both sides.

4.3.1.3 Density edition

As previously mentioned, due to the measurement techniques in geosciences, field data usually composed of series of unconnected points with very irregular distribution. For example, in hydrological applications, the data collected in a borehole are generally closely spaced vertically, but very sparse horizontally. Similar distribution is observed for marine data where instruments such as CDT (Connectivity-Temperature-Depth) are designed for vertical data recording.

In the preprocessing step, the density of the point set is edited based on the suitable shortest interval in the tessellation. This task can be done with a dynamic "Deletion" operation. For this purpose, our algorithm considers a list of all tetrahedra present in the existing tessellation and each tetrahedron is tested. If there is at least one edge with a length smaller than the minimum length ($L < L_{\min}$), therefore, one of the vertices of the edge which is selected based on the nearest neighbor analysis and a defined tolerance, is deleted. Hence, the point with the smaller nearest neighbor distance is deleted (figure4-9).

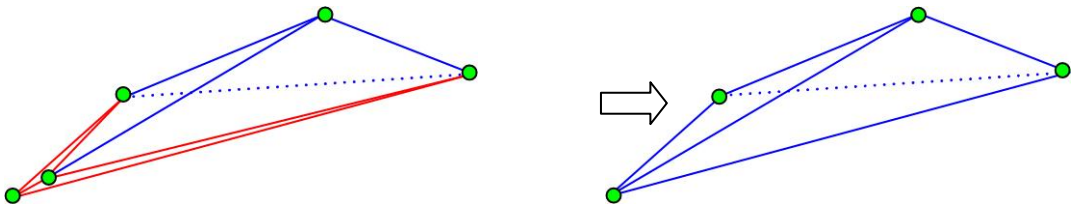


Figure 4-9 Tessellation refinement based on a tessellation element edge length criterion.

To remove points from the initial tessellation, deletion consists of locating the point to be deleted from the tessellation and reconfiguring the tetrahedra incident to the point with a sequence of *flip* operations. Each modified tetrahedron following the deletion operation must be inserted or updated into the dynamic list to density test.

Input: Γ : *initial tessellation,* D : dynamic list, $L_{\min}$*;*

$D \leftarrow \Gamma$
Repeat:
Get the first member of D : T *;*
If at least in one of its edge $L < L_{\min}$ return *vertex* v_1 and v_2 *;*
Choose v (v_1*or* v_2*) to be deleted based on its neighbor analysis;*
Delete v;
Update the tessellation;
Update D;
endif
Until D is empty*;*

4.3.2 Refinement processing

The processing stage identifies the poor quality tetrahedra and refines them as discussed in the following subsections.

4.3.2.1 Poor quality tetrahedra identification

In general, a tetrahedron quality is determined based on the error analysis of the PDEs solution and local errors show the elements that are poor quality and must be refined (Bern and Plassmann, 2000). As Shewchuk (1997) states, poor quality tetrahedra are typically flat or skinny. In a skinny tetrahedron, the vertices are close

to a line, and in a flat tetrahedron one of the vertices is close to the plane formed by other three vertices. A skinny tetrahedron has a circumradius much larger than its shortest edge. Therefore, the circumradius-to-shortest edge ratio of a tetrahedron or the ratio of the circumradius to the inscribed sphere radius is the most important criterion for detecting a skinny tetrahedron (Cavendish et al., 1985). These tetrahedra can be appeared in five types, as seen in figure 4-10 (Edelsbrunner, 2001).

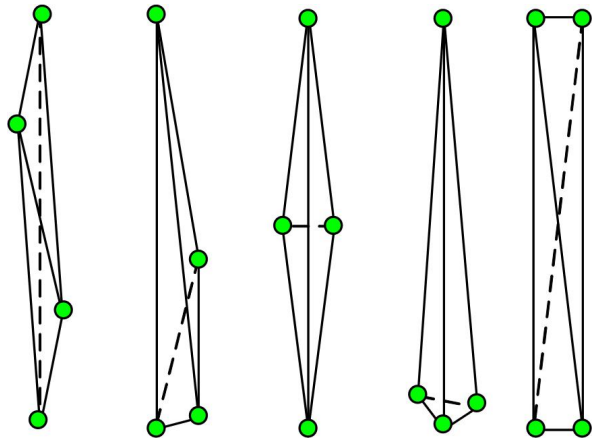


Figure 4-10 Five types of skinny tetrahedra (Edelsbrunner, 2001).

Flat tetrahedra can have a good circumradius to shortest edge ratios. However, their volume is very small (approaching zero) because one of their vertices is close to the plane formed by other three vertices of the tetrahedron. A tetrahedron with almost zero volume is called a sliver. As shown in figure 4-11, Flat tetrahedra have four types (Edelsbrunner, 2001).

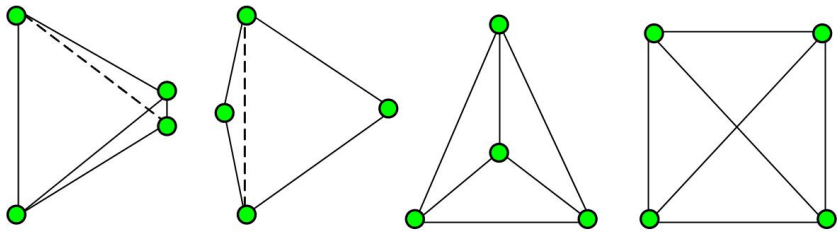


Figure 4-11 Four types of skinny tetrahedra, the last tetrahedron from left to right is a sliver.

A combination of measures, such as the ratios of maximum edge length to inscribed sphere radius and the ratio of the maximum edge length to the minimum edge length, can be used to distinguish a skinny tetrahedron from a flat tetrahedron (Parthasarathy et *al.* 1993). In some methods, the relation between the volume and the faces area or edges length is important. For example, Cougny et al. (1999) define a normalized aspect ratio based on the tetrahedron volume and its four facet areas. Dannenlogue and Tanguy (1991) use a ratio involving the tetrahedron volume and the average edge length of the tetrahedron edges to determine a flat tetrahedron.

As mentioned, all these criteria are purely geometrical and do not consider the physical nature of the system. For example, a tetrahedron meeting all these constraints will be poor quality for the simulation purpose if its size is larger than the given resolution. However, it will not be problematic if a tetrahedron, located in give area with a very smooth variation of field values, do not perfectly meet the mentioned criteria. In addition, using these geometrical criteria, the refinement for tessellation conforming to geoscientific data can be very time consuming due to the special configuration of the sampled points in one hand and geometry complexities of the fields on the other hand. For this reason, we propose to define the criteria which pay a special attention to both spatial and physical aspects of the problem. Therefore, some geometric and physical field variable criteria can be defined where the geometric criteria are related to the *shape and size* of elements while, the physical criteria are based on a *physical field variable criterion.* Some examples of geometric and physical criteria are:

- <u>*Edge criterion*</u>: The tetrahedra edge length must be less than a predefined tolerances for maximum length ($L_{\max}$) and larger than a predefined tolerances for minimum length ($L_{\min}$).

- <u>*Aspect-ratio criterion:*</u> let r and r' be the circumsphere and the inscribed sphere radiuses of a tetrahedron. The ratio of these values ($\beta = \frac{r}{r'}$) of the tetrahedra must be more than a user supplied tolerances ($\beta_{\min}$).

- *Volume criterion:* the volume of element must not be more or less than $V_{\max}$ or $V_{\min}$; the condition $V < V_{\min}$ is utilized to identify flat tetrahedra.

- *Physical field variable criterion:* The gradient of the field values along any edge of the tetrahedron must be less than a predefined tolerance for maximum gradient ($\Delta f_{\max}$).

Therefore, the tetrahedra that do not satisfy such these constraints are identified as poor quality tetrahedra.

4.3.2.2 Poor quality tetrahedra handling

Poor quality tetrahedra are unsuitable for a dynamic field simulation and must be modified or removed from the tessellation. In order to treat the poor quality tetrahedra, the new points are carefully inserted into the tessellation until it meets constraints on element quality. However, the central question of any refinement algorithm is, "Where should the new points be inserted within a poor quality element?" A reasonable answer is "As far as possible from the element vertices" (Shewchuk, 1997). If a new point is inserted too close to another point, the resulting small edge will create another poor tetrahedron. According to the empty sphere property of a Delaunay tessellation (see section 3.2c), the answer is the center of its circumcircle which is as far as possible from the vertices of the element. This idea is used frequently in 2D tessellation refinement algorithms. For example, Frey (1987) removes poor-quality elements from a triangulation by inserting new vertices at the centers of their circumcircles while satisfying the Delaunay criterion in the triangulation.

In 3D, several algorithms for the tetrahedralization refinement consider the insertion of additional vertices into the tessellation (Shewchuk, 1997, Si 2005). In some of these algorithms such as the algorithm used in LaGriT[5], the tessellation elements may become distorted or may become non-Delaunay. The main reason for this problem is that the data structure of tessellation is static; i.e., once constructed,

[5] http://lagrit.lanl.gov

it cannot usually be modified on-the-fly, so following any changes (e.g., inserting or deleting a point), it is necessary to reconstruct the tessellation globally. Therefore, a dynamic method i.e. an incremental method is required to insert the point. However, many non-Delaunay tetrahedra would be created using a dynamic constrained Delaunay method that constrains refinement process as the boundary triangulation (facets) remains unchanged. This problem is solved in the constraint Delaunay tessellation methods which allow the boundary elements to be refined during the refinement process. In these methods, point insertion in a Delaunay tessellation refinement is governed by considering the *encroached* linear boundary and e*ncroached* facet:

- A poor quality tetrahedron is split to four tetrahedra by inserting a point at its *circumcenter* (the center of its circumsphere); the Delaunay property guarantees that the tetrahedron is eliminated. If the new point encroaches upon any linear boundary (or facet), the point is not inserted and the linear boundary (or facet) it encroached upon is split (figure4-12). After each point insertion, the tessellation is locally modified with the help of *flip* that are described in chapter 3.

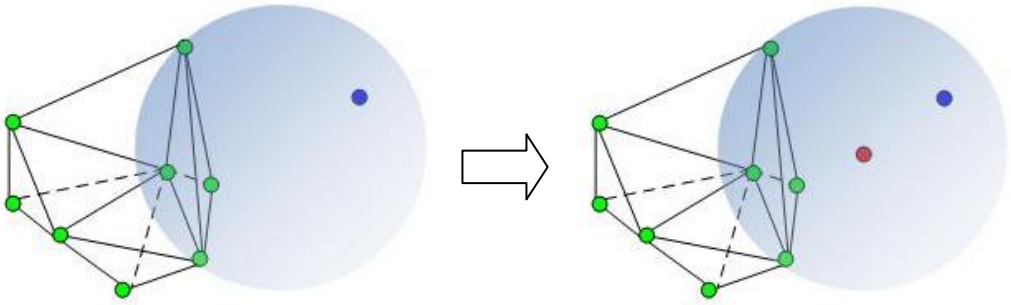


Figure 4-12 Poor quality tetrahedra are split by inserting a point at its circumcenter.

This method produces well-shaped tessellation, provided there is no small angle between the boundaries (Shewchuk 1997, Si 2005). This constrain is not practical for a complex region such a hydrogeological system where boundaries can make small angles. There are methods that can handle small input angles formed by

boundaries and constraints. These methods consider the protecting spheres around boundaries with small angles and then a tessellation refinement is done in its interior region (Pav and Walkington 2004). Although Good tessellation quality is provided inside the tessellation, there are bad quality and non-adaptive tetrahedra close to the boundaries with small angles.

To have a well-spaced tessellation conforming to the boundaries, some methods take arbitrary sizing functions into account. For example, a local feature size is defined in 2D by Ruppert (1993) and extended to higher dimension by Miller et al. (1996). The local feature size at a point is defined as the radius of smallest sphere centered at the point in such a way that the sphere touches two nonadjacent facets of the boundary. Therefore, they generate the candidate points that can be inserted to tessellation using the sizing function and then, insert the points that meet constrained Delaunay criterion. In brief, the algorithm goes as follows:

Input: *initial tessellation, radius-edge ratios, sizing function;*
Repeat:
Generate a new point (v) by the point generating rules (three rules for local optimality);
If the point v passes the point accepting rules then
Insert v and update the tessellation;
endif
Until no new point can be inserted;

However, it should be noted that some generated points do not pass the point accepting rules, for example when a point lies outside the tessellation, and then there may remain poor quality tetrahedra. In addition to this type of poor quality tetrahedra, there exist other slivers in the tessellation that may not have a small radius-edge ratio but nearly zero volume. To remove slivers, the tessellation smoothing is increasingly used via small perturbations of points, for example moving a point to the centroid of the points to which it is connected (Cheng et al. 1999, Edelsbrunner et al. 2000).

In our method, since the boundaries are defined with a high density in the preprocessing step, they remain unchanged during the refinement process.

Therefore, the tessellation is refined by adding points within all poor-quality elements using the “insert” operation. For tessellation refinement, each tetrahedron is tested separately for refinement. This test is done based on the defined geometric and physical criteria. Thus, the poor quality tetrahedron is refined by inserting an additional point in tessellation. Following to a point insertion, the configuration of the adjacent tetrahedra is changed by *flips* to satisfy the empty sphere criterion of DT. Therefore, each modified tetrahedron following the optimization operation must be tested for its quality. This process is repeated until all tetrahedra respect the mentioned conditions.

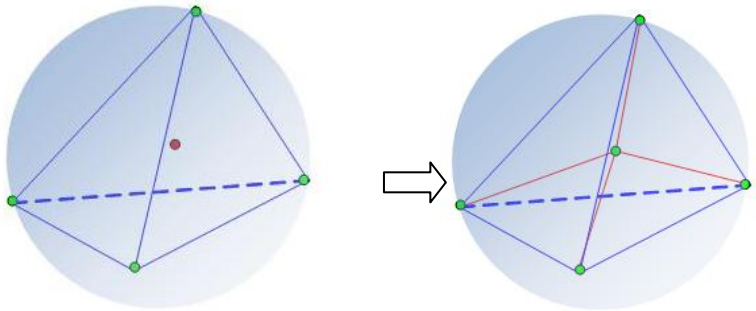


Figure 4-13 A poor quality tetrahedron is split to four tetrahedra by inserting a point at its circumcenter.

For each newly created point, we need to determine its physical attributes. Regarding the assumption for the variation of a field between two boundaries, the physical attribute variation can modeled as a constant, linear or polynomial function of location. For example, if the space is considered to be homogeneous, the attribute values of the corresponding volumetric object is assigned to the point located inside the object. Otherwise, it is necessary to use interpolation methods to obtain the value of the physical attribute, or parameters, at these locations p(x, y, z) which discussed in the next section.

4.3.2.3 Spatial interpolation

Spatial interpolation methods can be used to determine the value of a field at a given location where there is no observed data. Goodchild (1992) states that a

spatial interpolation method computes a complete continuous space from a set of sample points. Considering the nature of a field data set, an interpolation method must have several properties (Watson 1992). First, it must be exact and interpolators pass through all points data whose values are known. Second, the interpolated value at each point must be unique to ensure the continuity of the interpolated field. Third, the interpolators must be local and use a small portion of the total set of points to estimate the field values for the new points. Since, in global interpolation methods where the whole data are used to estimate a value at unknown points, a change in one data value affects the entire interpolated values. In addition, the errors in the field data may propagate though the tessellation. Fourth, the interpolated values for the unknown points between the data points must be relative to the data configuration. For example, for a smooth region, the interpolation must represent the gradual changes without any abrupt changes. Finally, the interpolator should account for the anisotropy of field data and data with different sampling patterns.

Several methods for spatial interpolation exist. The nearest neighbor, inverse-distance weighted averaging (IDWA), kriging and the natural neighbor interpolation methods are among the most frequently used methods. In these methods, if we consider that each data point has an attribute z_j, the interpolated value at the location p, z_p, is computed by the following equation:

$$z_p = \sum_{j}^{n} w_j z_j \qquad (4\text{-}1)$$

where w_j is the weight functions assigned to each data point. Therefore, the value of the attribute at location p is estimated by the weighted sum of the attributes of data points involved in the interpolation at location p.

The nearest neighbor interpolation method simply assigns a value to each point which is equal to the value of the spatially nearest data point, and does not consider the values of other neighboring points at all. Therefore, the weight of the nearest neighbor point is one. Although the method is simple, the interpolation is based on

piecewise-constant function and thus the interpolated values are not smooth and continuous.

IDWA methods assign a value to each point which is a function of the distance to its spatially neighboring data points and their values. This method is based on the assumption that the interpolating value must be influenced most by the nearby data points and less by the more distant points. Therefore, an inverse distance weight is used to determine the influence of the neighboring data points. In these methods, the neighbors can be selected in different ways. For example, the neighborhood can be defined as every point within a user-defined sphere or cube centered on the point location (figure4-14) or every point whose Voronoi cell has a common face with the cell of the interpolated point. A larger number of neighboring points results in a smoother result. Also, IDWA methods are unable to account for anisotropy in the data set when interpolating data.

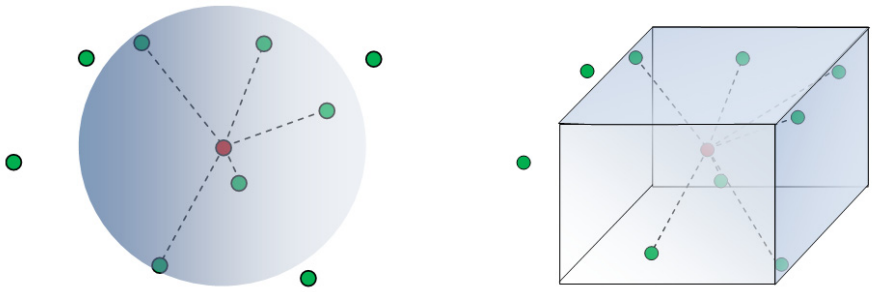


Figure 4-14 Two examples for selecting the neighboring data points for a query point by user-defined sphere or cube centered on the point location.

Kriging methods similar to the IDW methods apply weights to the neighbors based on distance, with this difference that they take into account the anisotropy of field data and data with different sampling patterns. Kriging methods use a semivariogram, which measures the spatial dependence between samples. On this basis, “optimal” weights are assigned to data points values that produce the minimum error variance at each location. Because of this property, kriging methods are among the best known in the geosciences (Meyers 1994). However, these interpolation methods are very time-consuming tasks in dynamic refinement

methods where with inserting a point, it neighbors must be searched and the cost of the calculation of the unknown value is huge.

Natural neighbor methods use the closest subset of data points to a query point and give volume-weighting to them for interpolation (Sibson 1981). Voronoi diagram can clearly and uniquely define the neighbors of any point in the space. In natural neighbor interpolation methods, the neighbors of a point p are the Voronoi polytopes that share a common face with the polytope of the point p. In order to proceed with the interpolation at a location p, first the point p is inserted in the tessellation, and then the volumes of the neighboring polytopes are computed. Next, the point p is removed from the tessellation and the volumes of the polytopes are re-computed. The weight w_j of each neighbor corresponds to the ratio of the difference of these volumes for each neighbor and the volume of the polytope of the point p. Among all of mentioned methods, natural neighbor interpolation methods meet all requirements conditions (Gold, 1989; Watson and Phillip, 1987).

4.3.3 Refinement post-processing

In the post-processing stage, the users can modify and adjust the density of the tessellation element locally, since the data structure of the tessellation is dynamic. It is possible to make on-the-fly modifications and explore interactively a data set to extract information. For example, we can remove or insert faults or modify the location of wells in a hydrogeological system without having to rebuild the whole tessellation. This will help to perform different `what if?' scenarios during the simulation process to better understand and analysis the field behavior. It is also possible to analyze and make some spatial queries such as k-neighbors queries on tessellation elements and to modify the tessellation over time.

Briefly, flowchart of the algorithm for an adaptive refinement of a Delaunay tessellation is shown in figure 4-15.

:

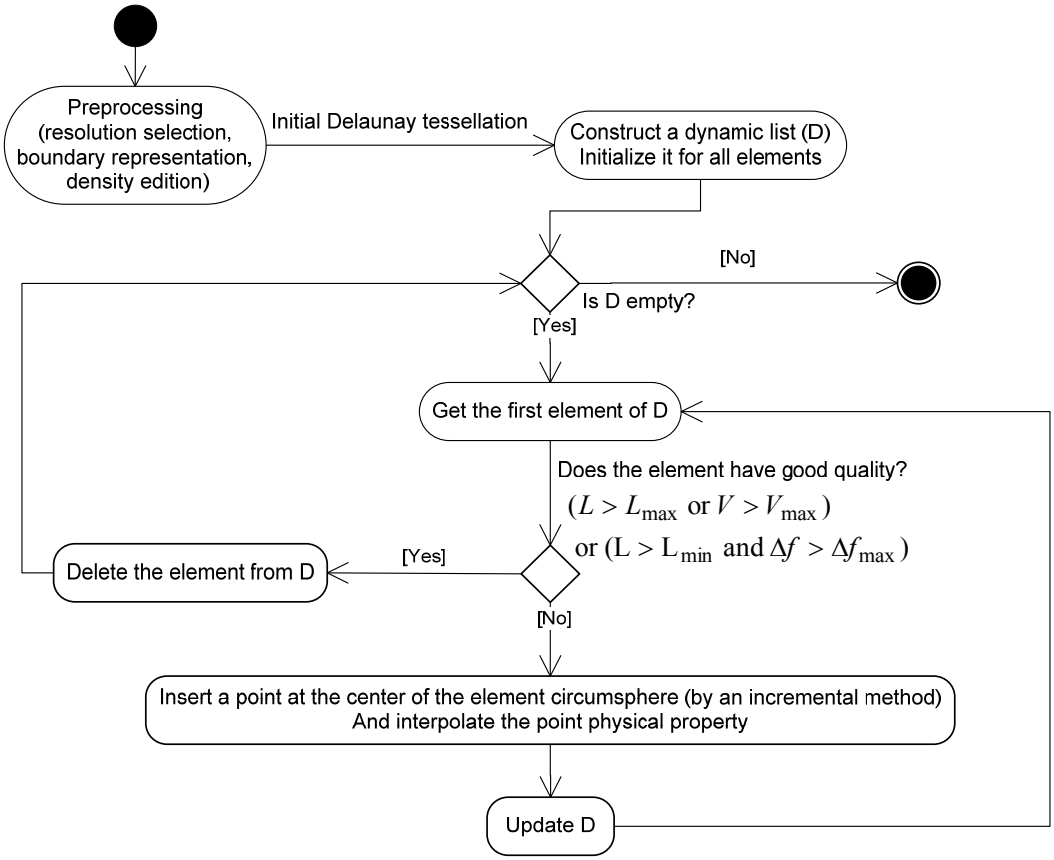


Figure 4-15 Flowchart of the algorithm for an adaptive refinement of a Delaunay tessellation

4.4 Test and discussion

The proposed algorithm for an adaptive refinement of a Delaunay tessellation was implemented in Borland Delphi programming language and runs under Microsoft Windows XP. For evaluation purposes, we have applied the algorithm to a set of 1292 points that were collected in four boreholes of a hydrogeological system in the Olkiluoto site [Ahokas and Koskinen 2005]. Olkilutoto Island is located on the west coast of Finland. The bedrock in the site is mainly characterized by a low

permeability gneiss rock intersected by several fracture zones. The covered area was about 740 m long, 1000 m width and about 1000 m height (figure 4-16).

For the construction and refinement of a tessellation using the data set, we defined several criteria for the tessellation elements size and physical properties. These criteria were selected based on the experts' knowledge of the field for this specific application. Based on these criteria, the shortest and largest length for a tetrahedra edges should be 20m and 80m respectively and the minimum and maximum variation of the hydraulic conductivity along a tetrahedra edge should be 10^{-12} m/s and 10^{-8} m/s respectively.

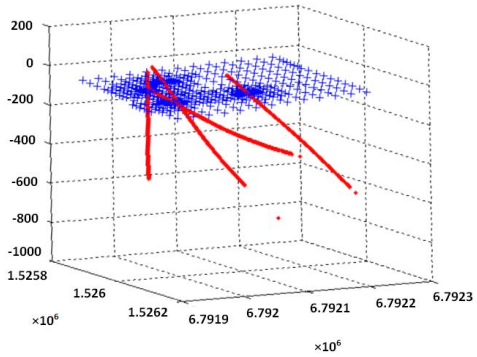
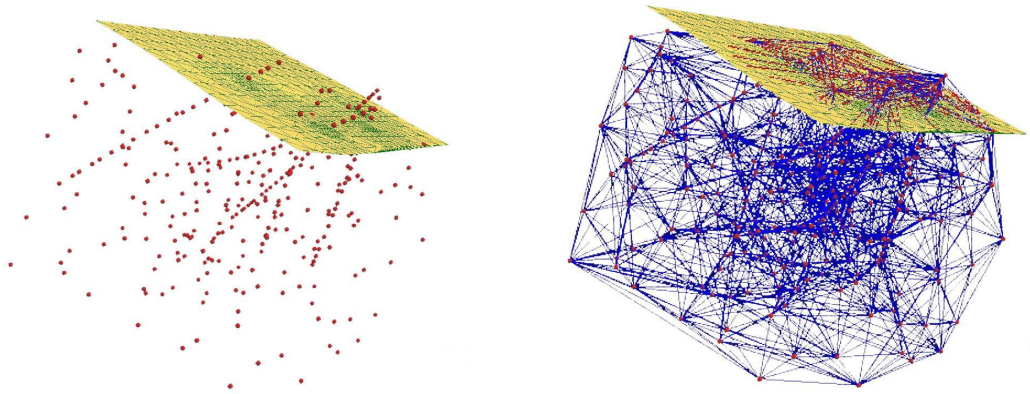


Figure 4-16 Data points collected in four boreholes (red points) and a planer fracture represented by blue points.

The site is characterized by the presence of a discontinuity in the region which is a fault. The fault is modeled using a set of 465 points with a resolution of 20 m (shortest length for the tetrahedra) on each side of it. The points collected from the boreholes were very dense vertically and sparse horizontally. In average the distance between two adjacent points along a borehole axis was 2-3 m, while the boreholes were 100-200 m apart from each other. Therefore, using a nearest neighbor analysis and the dynamic "Deletion" operation, the density of the point set

is edited based on the shortest edge criterion in the tessellation (20 m). In this step, the number of data points was decrease to 633 points.

Next, a 3D Delaunay tessellation is constructed using the data set and a number of 3766 tetrahedra were created. When the refinement algorithm is used, about 12.1% of skinny tetrahedra and about 5.4% flat (slivers) tetrahedra were detected in the tessellation with respect to the given size and physical property thresholds (72000 m3). Following the application of the refinement algorithm, 153 new points were inserted in the 3D tessellation and the total number of the points was increased to 786 points resulting in a total number of 4657 tetrahedra.



(a) (b)

Figure 4-17 a) The data set from boreholes after the refinement process, b) the resulted tessellation on the data set

As we can see from the results, the proposed algorithm allowed us to properly detect and eliminate poorly shaped elements from the tessellation. The proposed algorithm is not only comparable with other existing algorithms developed in the field of Computational Geometry and provides possibility of dynamic and local edition of the Delaunay tessellation, but also offers the advantage of considering

physical criteria for the refinement. The mentioned algorithms are purely based on geometrical criteria and do not consider the physical criteria for the refinement process. In addition, these algorithms are in some cases, not efficient enough to refine a tessellation with high irregular distribution of points and with a very smooth variation of physical values.

Table 4-1 the qualitative comparison between the proposed method and the existing methods.

Methods	**Preprocessing step**	**Processing step**			**Postprocessing step**
		Poor quality tetrahedra identification	**Poor quality tetrahedra handling**	**Interpolation**	
Methods developed in computation geometry	Automatically (possible)	Based on pure geometric criteria	Local updating (dynamic (incremental) methods)	No	yes
Developed for geosciences (such as LaGrit)	manually	Based on both geometric and physical criteria	Global updating (static methods)	Yes	non
The proposed method	Automatically	Based on both geometric and physical criteria	Local updating (dynamic (incremental) methods)	Yes	yes

In terms of physical criteria, algorithms such as in LaGrit, are static and are not able to satisfy the Delaunay criterion following any change in the tessellation (insertion or deletion of a point). In addition, the refinement procedure is based only on the insertion of new points in the center of each poor quality element without considering the adjacent elements. This may result in some other poor quality elements and the users have to rebuild the tessellation using the new data set obtained from the refinement process. Table 4-1 summarizes the qualitative comparison between the proposed method and the existing methods.

In order to have a more practical comparison, we have compared our results with the ones obtained from LaGrit in terms of the number of poor quality elements and the resulted tessellation elements after the refinement process. Here we should mention that since the preprocessing of points in LaGrit is manual, we have not used that procedure and instead the data set resulted from our preprocessing stage was used for refinement in LaGrit. Using the same geometrical and physical criteria, the same number of poor quality elements was detected (12.1% and 5.4% skinny and flat tetrahedra). The total number of the tetrahedra created using the software was 4827 following the refinement of the tessellation. However, the resulted tessellation from LaGrit does respect neither the Delaunay Criterion nor the minimum geometric and physical thresholds (minimum length or physical variation of the property). Therefore, our method presents a great advantage over the refinement method employed by LaGriT for geosciences applications. This is because, in most of the geoscientific applications, the estimation of the given physical properties is usually based on the proximity criterion and is obtained by an interpolation operation. The Delaunay criterion ensures that we have an optimal tessellation of the space meaning that the adjacency between the points is more adequately defined and they better respect the proximity criterion.

The time complexity of our on-the-fly dynamic Delaunay refinement algorithm is determined as follows:

Refinement preprocessing: The 3D DT of a set of n points in R^3 is constructed using the incremental algorithm. The performance of this operation is $O(n^2)$ in the worst case. In the case of point deletion, the time complexity is $O(es)$ where e is the number of *ears* of the point star and s is the point degree. The number of tests

to determine poor-quality tetrahedra corresponds to the number of tetrahedra in the structure.

Refinement processing: The processing stage includes updating the data structure by point insertion in the poor-quality tetrahedra. In the case of point insertion, if the new point conflicts with k simplices, then $O(k)$ time is needed to insert it and compute its attribute based on natural neighbor interpolation. For any insert or delete operation the dynamic list should be updated which takes $O(\log n)$.

Therefore, in the worst case scenario, where there are both very large and very small tetrahedra, $O((m+l)^2)$ time is needed, where m is the size of the final tessellation and l is the number of deleted points.

4.5 Conclusion

This chapter discussed the construction of a 3D adaptive tessellation for the representation of 3D spatial fields. Based on the literature review, 3D VD and DT can provide an adequate tessellation for the representation of a 3D continuous space. However, because of the nature of the geoscientific datasets, such in hydrogeological applications, which is usually highly irregular, poor-quality mesh, elements may appear in the tessellation. These poor-quality elements such as slivers, caps and thin tetrahedra, are not suitable for simulation purposes. We developed a 3D refinement method based on the dynamic Delaunay tetrahedralization. This method provides a tessellation conformed to the complexity of fields concerning shape and size criteria. Fortunately, the proposed algorithm through its dynamic operations like insert, delete and flips, offers an efficient refinement method that allows us to modify and adjust the density of the tessellation elements locally, i.e. on-the-fly delete and insert point within the tessellation. In the next chapter, we will investigate how such an adaptive tessellation can be used to approximate the dynamic behavior of a field which

usually described by a set of partial deferential equations (PDEs) over time and space.

Chapter 5: Development of a 3D Kinetic Data Structure for Representation and Simulation of 3D Dynamic Fields

5.1 Introduction

As mentioned previously, the dynamic behavior of a field is usually described by a set of partial differential equations (PDEs). Representing field evolution over time involves a numerical solution of the PDEs using the underlying tessellation. This means that the dynamic behavior of the field is approximated by the sum of the dynamic behavior of the tessellation elements. This can be done in either a static manner from the Eulerian point of view, where the equations are solved on fixed tessellation during a simulation, or in a dynamic manner from the Lagrangian point of view, where the tessellation moves (Price 2005, Mostafavi 2002). In addition, we explained that in Lagrangian approaches, when the elements move, the connectivity between the elements (topology) remains unchanged, which may cause difficulties such as tessellation tangling and deformation. The tangling problem becomes especially acute in non-uniform media with multiple sources and complex boundary conditions (Neuman 1984). Therefore, in these cases, the tessellation

used for Lagrangian methods may become distorted over time and complete re-tessellating is frequently required (Malcevic 2002). Development of mixed Eulerian-Lagrangian methods has led to the class of arbitrary Lagrangian-Eulerian (ALE) methods (Crowley 1970). These methods reduce tessellation distortion by "remapping" or "reconnecting" of the mesh. Tessellation remapping can be regarded as an Eulerian process, because mass is transported across tessellation elements boundaries. The principle of continual remapping has led to the Free-Lagrange method (Fritts et al. 1985, Mostafavi and Gold 2004). The difference between the Free-Lagrangian and the classical Lagrangian methods is that the latter attempt to maintain the initial tessellation connectivity during the simulation process. The Free-Lagrangian method allows updating the tessellation connectivity as part of the problem to be solved (Whitehurst 1995). However, maintaining and processing of the connectivity relations between mesh elements is a very difficult task especially in 3D. In this chapter, we first investigate how the 3D spatial relationships of the constructed tessellation in chapter 4 can be updated to address the representation of dynamic behavior of a field in a Free-Lagrangian manner, then we propose and develop a new 3D kinetic data structure that effectively maintains the relations between the moving mesh elements and update them locally when they change.

5.2 Free-Lagrangian representation of a dynamic field using 3D DT and VD

Using Voronoi diagram as the underlying tessellation for a Free-Lagrangian method implies that all the Voronoi polyhedra within the tessellation are free to move and interact with each other; as a result, the tessellation must be updated during the continuous motion of the elements. Based on our literature review, it can be distinguished three main methods to represent the moving tessellation over time based on static, dynamic, and kinetic VD and DT:

a) Methods based on *Static Voronoi & Delaunay data structures* take each moving point p (Voronoi polyhedron generator or Delaunay vertex) in turn, compute its

new position p' based on its velocity and a defined time-step and rebuild the whole spatial tessellation. These methods are poor and time consuming as they are unable to locally update the spatial tessellation and the whole Voronoi tessellation, even the unchanged areas, is rebuilt. In addition, the value of time-interval has to be carefully selected to ensure a realistic representation of dynamic behavior of a field. For example, a large time-step causes problems such as overshoots and undetected collisions and, as a result, we may observe some abnormal behavior in the simulation results. For a small time-step, an extensive computation effort will be required to check for changes at time when none occurred. However, these methods are very straightforward and can be implemented by any VD and DT construction methods including the Sweep line and Divide and conquer algorithms (Okabe *et al.* 2000). These methods are generally used for 2D and 3D simulation purpose (Campbell and Shashkov 2003).

b) Methods based on *Dynamic Voronoi & Delaunay data structures* take each moving point p in turn, find the tetrahedron containing its new position p', insert a new point in the new position and remove p from the tessellation and update the spatial tessellation. These methods use the insert and delete operations (Gold 1995, Devillers 2002, Mostafavi *et al.* 2003, Ledoux 2006) to represent the continuous motion of a data point in the space. Local modifications are allowed following any change in the data structure. Among the construction methods for the VD and DT, the incremental method is the only dynamic method capable of supporting these methods (Mostafavi 2002). Such methods are not very suitable for a simulation purpose as the mentioned dynamic operations, particularly the deletion operation, are computationally very expensive. Consequently, updating the spatial tessellation is typically very slower than rebuilding it (Russel 2007). This problem becomes even more important when the simulation process is based on time-steps and the data structure does not remain valid between two time-steps. Another problem is related to complexities related to some complicated point insert and delete cases within a Delaunay tetrahedralization (ex. unflippable cases) (Russel 2007). In

practice, because of the efficiency problems, these methods are used only when the number of the moving points is very limited (Guibas 1998).

c) Methods based on the *Kinetic Voronoi & Delaunay data structure* is an alternative solution to eliminate problems related to the deletion and insertion of a point in a spatial tessellation (VD and DT) problems related to the unflippable cases in a dynamic data structure as well as the problem of global rebuilding of the spatial tessellation in a static data structure (Albers *et al.* 1998, Mostafavi and Gold 2004, Ledoux 2006, Russel 2007). A kinetic data structure aims to handle the elements that move smoothly in geographical space without jumping from one part of the space to another part (Guibas 1998). In such methods, the points are moved and their relations are updated by bistellar flip-based method (Joe 1991, Shewchuk, 2005), thus a local modification in the data structure is allowed after any change without deleting and inserting a point. Moreover, these methods use adaptive time-steps and update the spatial tessellation when an event occurs within it.

Studies of 2D kinetic Voronoi and Delaunay data structure have been carried out by a number of researchers. The problems addressed include the estimation of upper and lower bounds on the number of combinatorial changes to the tessellation (e.g. Roos 1993), query problems on both Delaunay triangulation and Voronoi diagram of moving points (e.g. Devillers *et al.* 1996), and several implementation issues of the kinetic Delaunay data structure and its application for Free-Lagrangian tidal simulation on the globe (e.g. Mostafavi and Gold 2004). Despite the improvements in 2D kinetic Voronoi and Delaunay data structure construction, its extension to 3D is still in its infancy and the related works are mostly at a theoretical level. Albers *et al.* (1998) has conceptually extended Roos's 2D work to 3D space. A similar method has also been presented by Gavrilova and Rokne (2003) to extend the methods for Voronoi diagram of moving points to n-dimensional space. The method deals with the maximum number of combinatorial changes possible in a 3D Voronoi diagram of moving points with defined trajectories and the problem of collision detection optimization. Russel *et al.* (2007) present a theoretical development of a kinetic Delaunay triangulation in three dimensions for CGAL. Ledoux (2006) has extended Mostafavi and Gold's 2D method (2002) to 3D space and developed a kinetic data structure

that deals with a single moving point in a 3D Voronoi tessellation. Although the algorithm is functional for general cases, it needs to be improved for degeneracies and special cases as well as problems related to the simultaneous motion of a set of points in a 3D Delaunay triangulation and Voronoi diagram. Dealing with these problems at theoretical and implementation levels is necessary for a successful dynamic simulation. Here, we explain the fundamental concepts for the development of the proposed 3D kinetic DT and VD.

5.3 3D Kinetic Delaunay tessellation

Point movement in a Delaunay tetrahedralization (DT) may change the configuration of tetrahedra that have the moving point as one of their vertexes. A topological event is defined as any primitive change in topological relations of a moving point (Roos 1993, Gold 1996). In a tetrahedralization, a topological event occurs when a point moves in or out of a circumsphere of a tetrahedron on its trajectory. In this case, a local topological change is necessary to preserve the Delaunay empty circumsphere criterion. However, if the location of the point is changed without any topological event, the spatial relation of the point does not need to be updated (figure5-1).

To represent the continuous motion of a point, it is possible to detect all topological events on the trajectory of the moving point, move the point to its new positions, one by one, on its trajectory, and update the tetrahedral data structure, so the continuous motion is replaced by a set of discrete events (Roos 1993, Gold 1996, Gavrilova and Rokne 2003a, Mostafavi and Gold 2004, Ledoux 2006). In addition to topological events, there is another type of events called trajectory events (Gavrilova and Rokne 2003). Points are allowed to change their trajectories at any time. These changes in trajectory correspond to the physical nature of the problem, but in a general case, it can be defined arbitrarily, as long as the trajectory remains continuous. When a trajectory update occurs, all topological events affected by those changes must be redetected.

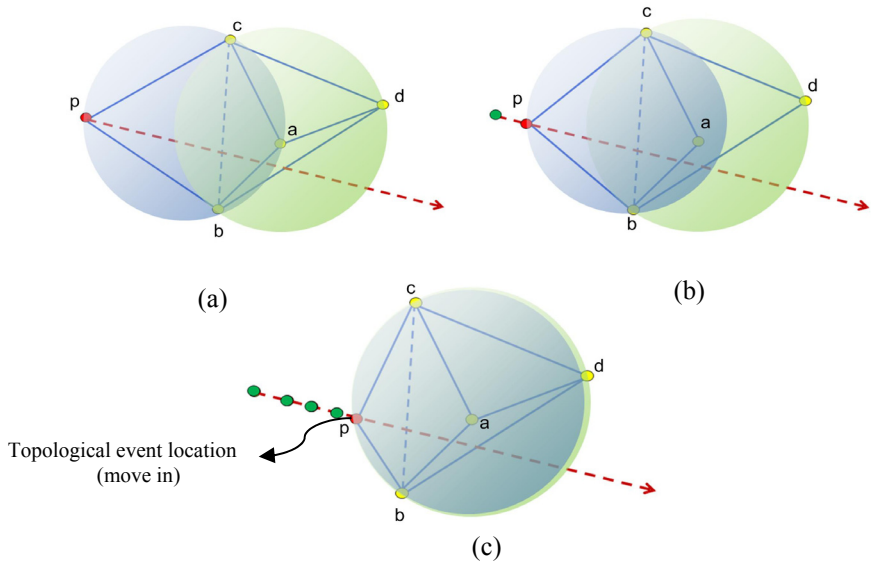


Figure 5-1 a) Point p moves within the tessellation on a linear trajectory, b) the location of point p changes, but the spatial relation of the point does not change, c) a topological event occurs when a point moves in or out of the circumsphere of a tetrahedron in 3D.

To detect a topological event, the existing algorithms use a well-known predicate test (Guibas and Stolfi 1985) that determines if a point p is inside, outside or on a circumsphere of a tetrahedron defined by four points a, b, c, d. For each moving point p, the test is done via a determinant computation on the coordinates of the points in a lifted space (Aurenhammer 1987) where the 3D point set is mapped to a paraboloid in a four dimensional space as follows:

$$(x \quad y \quad z) \rightarrow \left(x \quad y \quad z \quad x^2 + y^2 + z^2\right) \qquad (5\text{-}1)$$

In a spatial kinetic data structure, the positions of points are time dependent; therefore, the value of the determinant is time dependent and is given by a function, called predicate function (Guibas and Russel 2004). A topological event occurs when the predicate function has a root determining the next position of the moving point (p) on its trajectory.

$$\begin{vmatrix} p_x(t) & p_y(t) & p_z(t) & p_x^2(t)+p_y^2(t)+p_z^2(t) & 1 \\ a_x(t) & a_y(t) & a_z(t) & a_x^2(t)+a_y^2(t)+a_z^2(t) & 1 \\ b_x(t) & b_y(t) & b_z(t) & b_x^2(t)+b_y^2(t)+b_z^2(t) & 1 \\ c_x(t) & c_y(t) & c_z(t) & c_x^2(t)+c_y^2(t)+c_z^2(t) & 1 \\ d_x(t) & d_y(t) & d_z(t) & d_x^2(t)+d_y^2(t)+d_z^2(t) & 1 \end{vmatrix} = 0 \qquad (5\text{-}2)$$

In the predicate function, t represents time (or simulation-time in the context of simulation) implying that the trajectories of a moving point must be defined a priori (Guibas and Russel 2004). The cost of generating, computing and updating the predicate function is very expensive, especially when dealing with several simultaneous moving points on very complex trajectories. For example a quadratic trajectory of a moving point results in a degree eight predicate function. Guibas and Russel (2004) argue that the computational cost can be reduced by minimizing the degree of the above predicate function. This can be done by allowing only one row of the predicate determinant to change linearly, which corresponds to moving one point at a time on a linear trajectory.

$$\begin{vmatrix} p_x(t) & p_y(t) & p_z(t) & p_x^2(t)+p_y^2(t)+p_z^2(t) & 1 \\ a_x & a_y & a_z & a_x^2+a_y^2+a_z^2 & 1 \\ b_x & b_y & b_z & b_x^2+b_y^2+b_z^2 & 1 \\ c_x & c_y & c_z & c_x^2+c_y^2+c_z^2 & 1 \\ d_x & d_y & d_z & d_x^2+d_y^2+d_z^2 & 1 \end{vmatrix} = 0 \qquad (5\text{-}3)$$

Mostafavi and Gold (2003 and 2004) have developed and implemented a different kinetic data structure based on Delaunay triangulation and Voronoi diagram in 2D space. Their algorithm, allows the points to move within the tessellation, one by one, on piecewise linear trajectories in the tessellation. In fact, the trajectory of each moving point is defined by a polyline, in such a way that each line segment represents the trajectory between the point and its closest topological event location. This allows having a complex trajectory without increasing the computational cost. In addition, the algorithm minimizes the number of triangles that must be tested to detect the closest topological event using a simple geometrical test as discussed in the next section.

5.4 Management of one moving point within a 3D Delaunay tessellation

To manage the movement of a single point in a 2D Delaunay tessellation, Mostafavi and Gold's algorithm (2002) computes a predicate function for only the neighboring triangles whose circumcircle cut the trajectory of the moving point between its current position and its destination; not all the neighboring triangles. To detect topological events, all *imaginary* and *real* triangles of a moving point in the DT are considered (figure5-2). As described in Mostafavi (2002), a *real* triangle is defined as a triangle that is incident to the edge of each triangle containing p (opposite adjacent triangles to the point p). In figure 5-2, ΔAEB, ΔBFC, ΔCGD and ΔDHA are examples of *real* triangles. An *imaginary* triangle is a triangle formed by the combination of each three consecutive neighbors of p and would exist if p was moved out of its circumcircle. In figure 5-2, ΔABC is an example of an *imaginary* triangle.

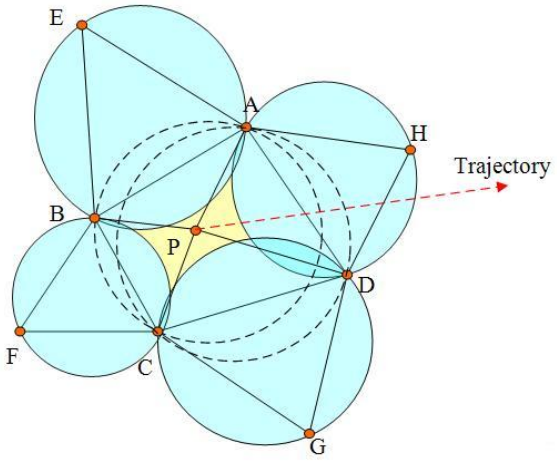


Figure 5-2 Spatial relationship does not need to be updated as long as the point p does not move in or out of one of the circumcircles (dotted and blue circles).

These definitions can be easily generalized to 3D space for a Delaunay tetrahedralization (Ledoux 2006, Mostafavi et *al.* 2009). Thus, in 3D, a *real* tetrahedron is a tetrahedron that is incident to the face of the tetrahedra containing p but outside of them. An *imaginary*

tetrahedron is a tetrahedron formed by the combination of each four neighbors of p and would exist if p was moved out of its circumsphere.

Referring to figure 5-2, we can see that only some of the *real* triangles around p must be tested for topological event detection. The *real* triangles behind p do not change following the moving of the point p (ex. ΔAEB, ΔBFC). These triangles are referred to as behind triangles. This means that the orthogonal projection of their circumcenter on the trajectory of the point is behind p with respect to the direction of movement. This fact allows us to compute the topological events for a fewer number of triangles among the neighboring triangles. Similarly, in 3D, the criterion for a real tetrahedron to be a behind tetrahedron is that the orthogonal projection of the centre of its circumsphere (h), onto the trajectory falls behind p (figure 5-3a).

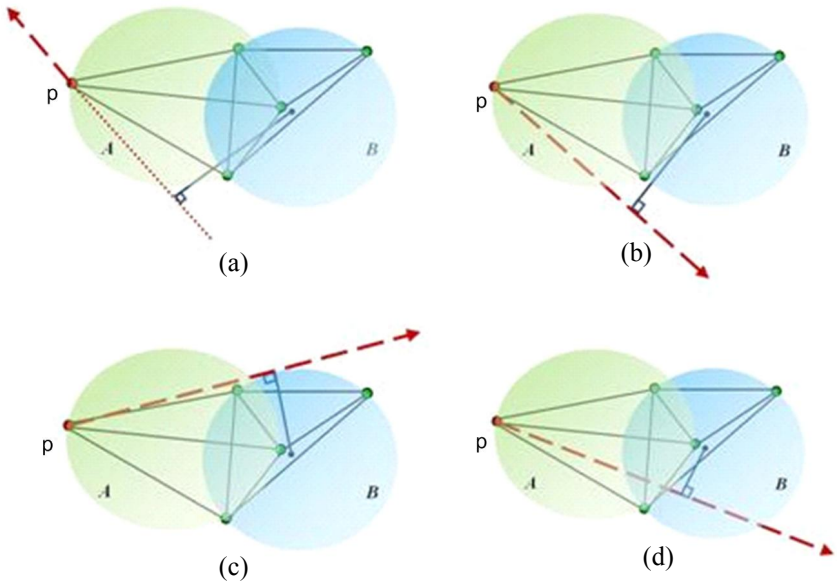


Figure 5-3 Intersection between the trajectory of a moving point and one of the circumspheres of imaginary or real, tetrahedra (here the real circumsphere B): a) behind tetrahedron: the orthogonal projection of the centre of circumsphere B, onto the trajectory falls behind p, as a result, it will not be tested, b) the trajectory does not intersect the circumsphere B, c) The trajectory is tangent to the circumsphere B, d) The trajectory intersects the circumsphere B.

To move a point from its current position towards a given destination, it is essential to move the point step by step to the closest topological event position and make appropriate

local updates to the data structure. The closest topological event location (t_{event}) is the location of the intersection between the circumsphere of a *real* or an *imaginary* tetrahedron and the trajectory of the moving point between its current position and its destination. For this purpose, a simple test that computes the intersection between a line and a sphere is used. Let p be the coordinates of the point before the topological event and p' be its new coordinates. Then we have:

$$p'(t) = p + t_{event} \cdot D \qquad (5\text{-}4)$$

Where D is the distance of the moving point p to its destination and $t \cdot D$ is the closet topological event distance i.e. the distance required for the moving point to cut the first real or imaginary circumsphere on its trajectory (p'). Substituting the equation (5-4) into the equation of a sphere results in a quadratic equation as follows:

$$at_{event}^2 + bt_{event} + c = 0 \qquad (5\text{-}5)$$

Regarding this equation, there are three possible trajectory-sphere intersections:

a) No intersection when the trajectory does not intersect the sphere (figure 5-3b),
b) Point intersection when the trajectory is tangent to the sphere (figure 5-3c), and
c) Two intersection points when the trajectory intersects the sphere (figure 5-3d).

We can see that the intersection may occur at the origin, between the origin and the destination, at the destination, or beyond the destination of a moving point for t_{event}=0, 0<t_{event} <1, t_{event}=1 and t_{event}>1, respectively. As mentioned, we ignore the intersections occurring before the origin where t<0.

5.4.1 Optimization of Delaunay Tetrahedralization

Let us consider that a Delaunay tetrahedralization of a point set s is denoted by $DT(s)$, when a point p ($p \in s$) is moved to its next closest topological event location p'. Therefore, the overall problem is to update $DT(s)$ for a given p'. Three cases may occur:

a) A point (p) moves in a Real tetrahedron circumsphere, consequently we have two locations for the topological events with respect to this circumsphere,
b) The point moves out from an *imaginary* tetrahedron circumsphere; in this case we have only one location for the topological event with respect to the circumsphere,
c) The point trajectory is tangential with the circumsphere of the given tetrahedron; then we have two equal locations for a move-in and a move-out event,
d) The point meets another point of the tessellation on its trajectory.

As seen, the point movement based on topological event makes an ambiguity in the data structure. For DT in R^3, this ambiguity occurs when five or more points lie on the same sphere, which called Co-spherical points, consequently the DT($S(t)$) is not unique. Depending on the configuration of the moving point over other points, different solutions are required, as discussed as follows:

5.4.1.1 The point moves in a real tetrahedron circumsphere

There are two possible configurations in which five points are co-spherical with considering co-planarity situation.

Configuration (I): *There are five points cospherical in which no four points are coplanar*
When the moving point lies on circumsphere of a real tetrahedron, depending on the points configuration the tetrahedralization is done by two or three tetrahedra as can be seen in figure 4 and one or more flips are required to replace one tetrahedralization by another. According to Joe (1991), a *flip23* is always possible when the points are in configuration (5-4a). Therefore, the first tetrahedralization containing tetrahedra *abcd* and *acbp* is replaced with the second containing *capd*, *cpbd*, *bapd.* Therefore, connectivity changes, for example points *a* and *b* are not neighbors anymore. The tetrahedra in configuration (5-4b) are flippable if there is the third tetrahedron (tetrahedron *apbd*) to fill the convex hull of the five points. In this case, a *flip32* is possible and thus, tetrahedra *abcd*, *acbp* and *apbd* are replaced with tetrahedra *abdc* and *adbp*. However, if tetrahedron *apbd* does not exist, no flip is performed. As observed, since every tetrahedron cannot be flipped, a flip-ability test must be used.

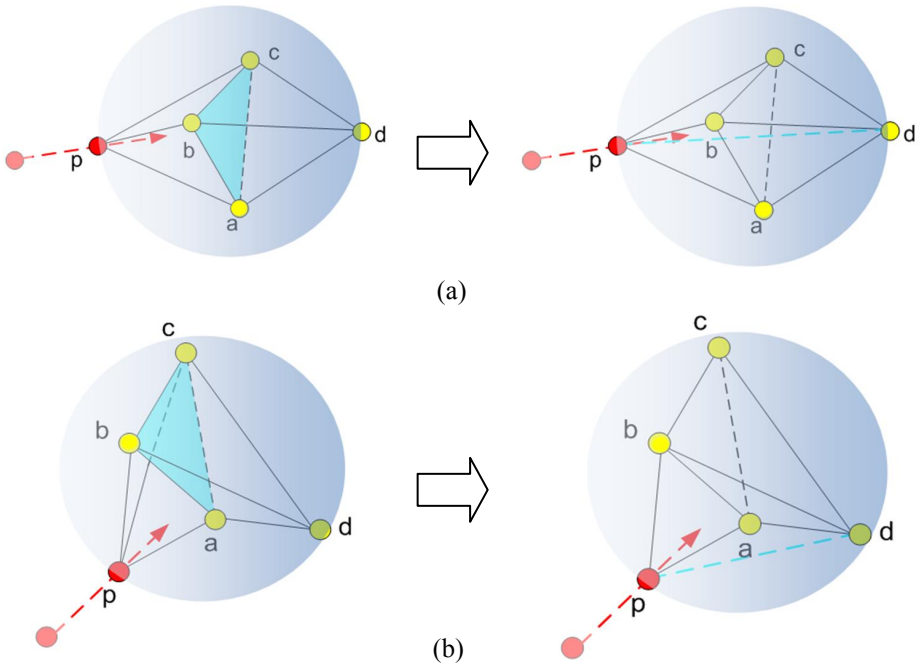


Figure 5-4 Optimizing the tetrahedra when point *p* moves in the circumsphere of tetrahedron *abcd* in which no four points are coplanar.

Configuration (II): *There are five points co-spherical in which four points are coplanar*

In this configuration, the moving point lies not only on the same sphere but also on the same plane of other three points. For example, figure 5-5 shows that point *p* moves in the circumsphere of tetrahedron *acbd* and is coplanar with points *a*, *b* and *c*. in this case, although a *flip23* is possible, it creates a flat tetrahedron *abcp*. To avoid this problem, when the points are in this configuration, a *flip 44* is done if there is a vertex on the opposite side of the coplanar points (Joe 1991), otherwise no flip is performed.

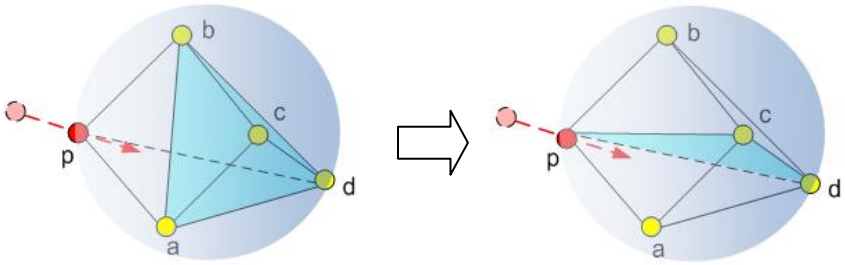


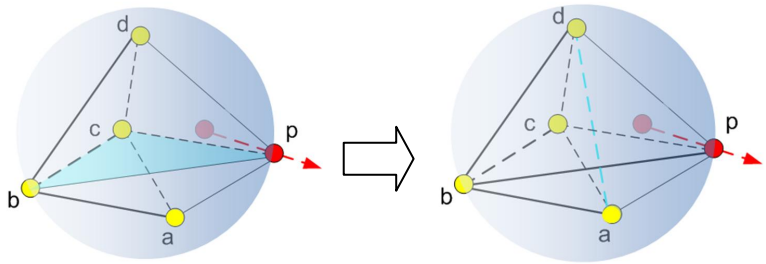
Figure 5-5 Optimizing the tetrahedra when point *p* moves in the circumsphere of tetrahedron *abcd* in which it lies on the same plane of other three points.

5.4.1.2 The point moves out from an imaginary tetrahedron circumsphere

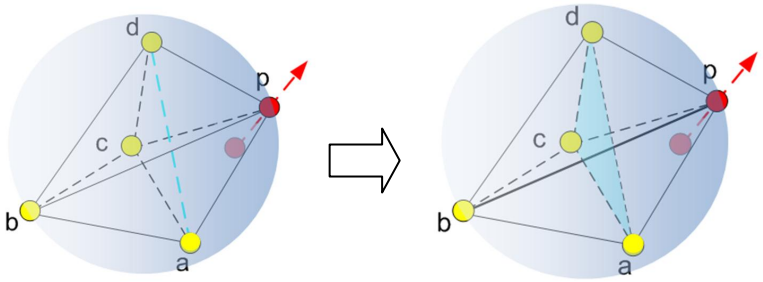
Here, the two possible configurations in which five points are co-spherical with considering co-planarity situation are discussed.

Configuration (I): *There are five points cospherical in which no four points are coplanar*
When the moving point lies on circumsphere of an imaginary tetrahedron, depending on the type of the ear, different tetrahedralizations can be done. As illustrated in figure 5-6, point *p* is moving out from the circumsphere of imaginary tetrahedron formed by the ear *abcd*:

- If the ear consists of two adjacent faces *abc* and *cbd* sharing line cb, a *flip23* is always possible (Joe 1991) (figure 5-6a). Therefore, the *flip32* replaces tetrahedra *cbdp* and *cabp* by three tetrahedra *abcd*, *pabd* and *pacd*.
- If the ear consists of three faces *abc*, *cbd* and *abd,* all sharing point b, a flip32 is preformed and three tetrahedra *abcp*, *cbdp* and *abdp* are replaced with two tetrahedra *abcd* and *acdp* (figure 5-6b).



(a)



(b)

Figure 5-6 Optimizing the tetrahedra when point *p* moves out from the circumsphere of the imaginary tetrahedron *abcd* in which no four points are coplanar, a) flipping of the ear consisting of two adjacent faces *abc* and *cbd*, b) flipping of the ear consisting of three faces *abc*, *cbd* and *abd*.

Configuration (II): *There are five points cospherical in which four points are coplanar*
In this configuration, the moving point is coplanar with a planer ear. Depending on the points configuration, one or more flips are performed. As can be seen in figure 5-7, points *p*, *a*, *b*, *c* and *d* are co-spherical and points *a*, *b*, *c* and *d* are on the same plane and make a planer ear. The ear consists of two adjacent faces *abc* and *acd* sharing line *ca*. A *flip23* results in the flat tetrahedron *abcd*. The problem becomes more serious when three points are also collinear (fig 5-7b) and a *flip23* may create two flat tetrahedra *abcd* and *adcp*. Therefore, in this case, a *flip44* is performed if there is a vertex on the opposite side of the coplanar points (Joe 1991), otherwise no flip is performed.

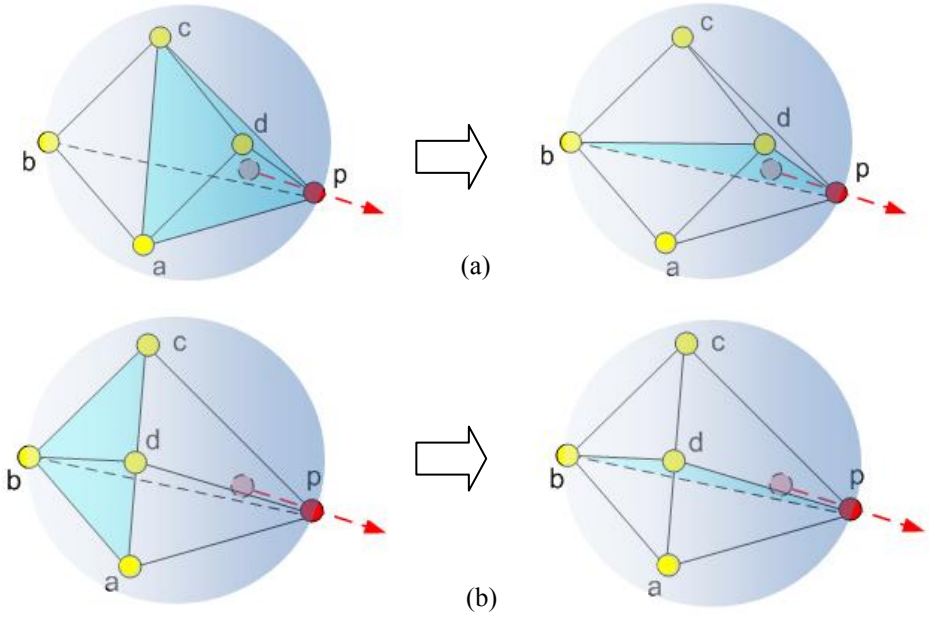


Figure 5-7 Optimizing the tetrahedra when point p moves in the circumsphere of tetrahedron *abcd* in which it lies on the same plane of other three points, a) flipping of the ear consists of two adjacent faces *abc* and *acd*, b) flipping of the ear consists of two adjacent faces *abd* and *bcd*, in which points *a*, *d*, *c* are colinear.

5.4.1.3 The trajectory of the moving point is tangential with the circumsphere of the given tetrahedron

In order to update the data structure when a point moves in a real tetrahedron circumsphere or moves out from an imaginary tetrahedron circumsphere or the point trajectory is

tangential with a circumsphere, one *flip32* or *flip23* or no flip is required. However, in the tangential trajectory case, it is easier to ignore the first topological event and proceed with the second one. As seen, there is a solution for this type of cospherical points and this configuration is not a problematic case for kinetic data structure.

5.4.1.4 The point moving in or out of a circumsphere, collides another point

Collision between two points happens when a moving point (p) meets another point of the tessellation on its trajectory. In other words, a collision happens when there exists a point at the exact location where p was supposed to be moved. The collision detection methods are commonly studied in the fields of computational geometry, physical-based modeling, geometric modeling, robotics, computer graphics and computer simulation (see Okabe et. al 92; Held et. al. 96; Medvedev et. al. 2006). A comprehensive review of collision detection optimization methods in dynamic simulation can be found in Gavrilova and Rokne (2003). These methods typically have two stages. The first stage identifies the pairs of points that can potentially be in contact at a given time by maintaining for every point a list of neighbors. The second stage determines, for every pair defined by the neighborhood, whether or not there is actually a contact. This is done by testing the distance between two points x and $y \in s$, against a tolerance. If the distance is smaller than the tolerance, there is a collision (eq. 5-6):

$$|x-y| = \left(\sum_{i=1}^{3} |x_i - y_i|^2 \right)^{1/2} \leq Tolerance, \quad \forall x, y \in S \qquad (5\text{-}6)$$

For collision detection, several neighborhood techniques have been used. In a straightforward approach each pair of points is considered to be neighbors, i.e. there are $\binom{n}{2}$ neighbor pairs. Therefore, this gives an $O(n^2)$ algorithm that is computationally very expensive. In our algorithm, 3D Delaunay and Voronoi data structures store the topological information. Thus collision checks are $O(n)$ in the worst case.

After collision detection stage, the central question is: “What should happen on collisions between the points?” In a spatial dynamic process modeling application, the trajectories of points are described by PDEs and the points represent either the fluid flow like buckets of water (in a hydrogeological system), buckets of gas (for a hydrodynamics application) or the fixed or moving boundaries; therefore the type of decision on the colliding points is defined according to the physical model. As a result, the trajectory, velocity and other attributes of each point change. Form a geometrical point of view, the collisions can be handled by (figure 5-8):

- Deleting one point and reinserts it after second object along its new trajectory (figure 5-8a),
- Merge one point and updating the physical attribute and trajectory of second one (figure 5-8b),

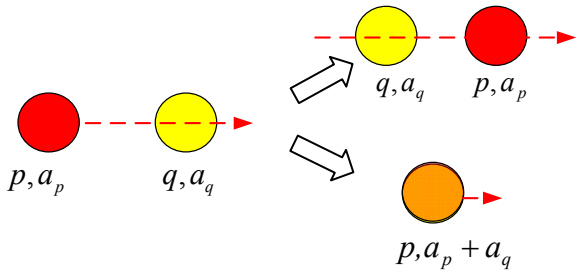


Figure 5-8 Collision case; a) Deleting one point and reinserting it after second object along the trajectory, b) Deleting one point and updating the physical attribute and trajectory of second one.

In brief, the method for updating the spatial relationship of a moving point in a 3D Delaunay tessellation is presented in figure 5-9.

It is obvious that following a flip, the tetrahedra configuration change in the tessellation and the process to find the next closest topological event for the moving point must be repeated as described earlier in this section.

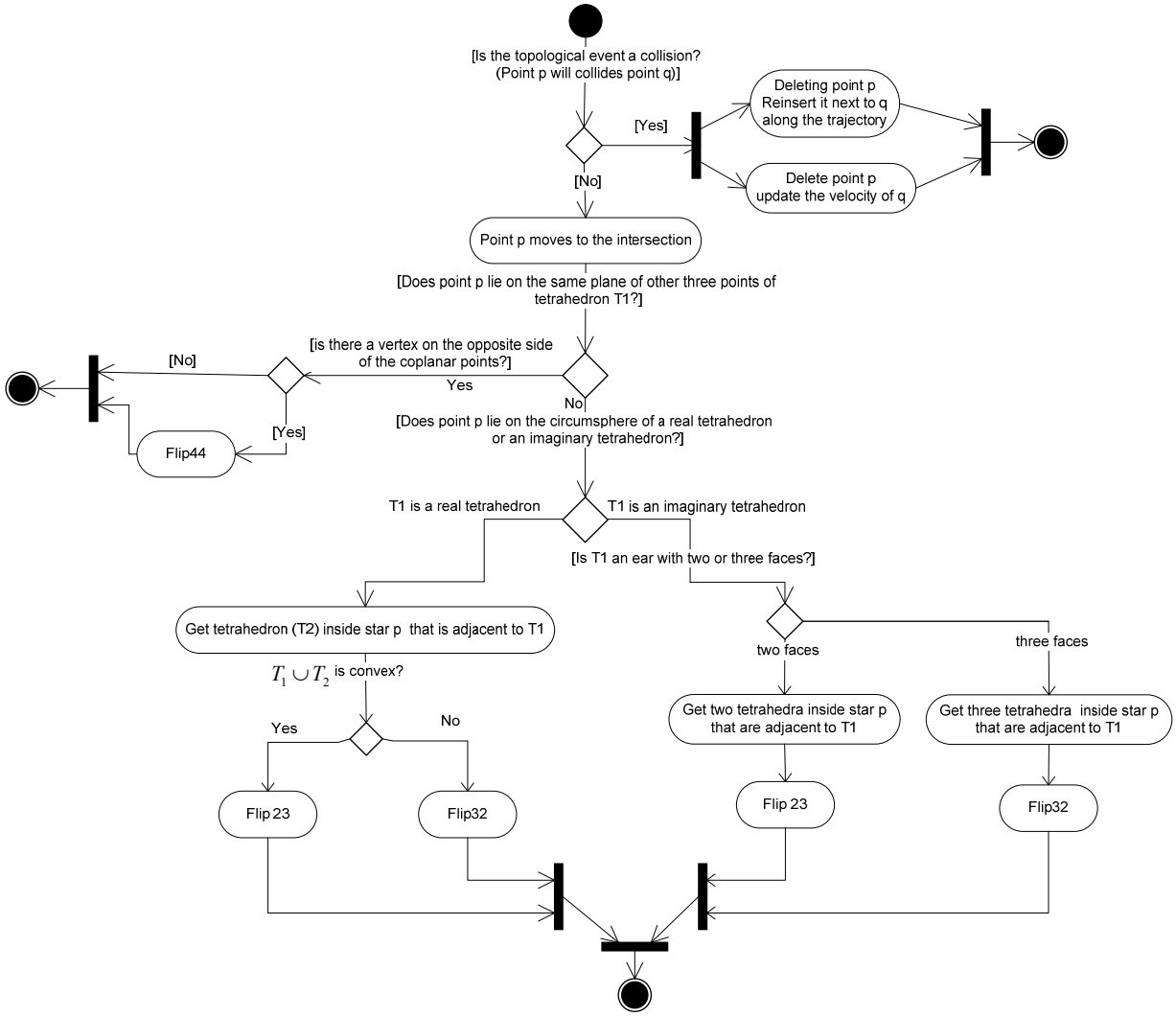


Figure 5-9 Flowchart of the updating the spatial relationship of a moving point in a 3D Delaunay tessellation.

Summary

The discussion on the managing of a moving point in a 3D Delaunay tessellation is summarized in the flowchart presented in Figure 5-10.

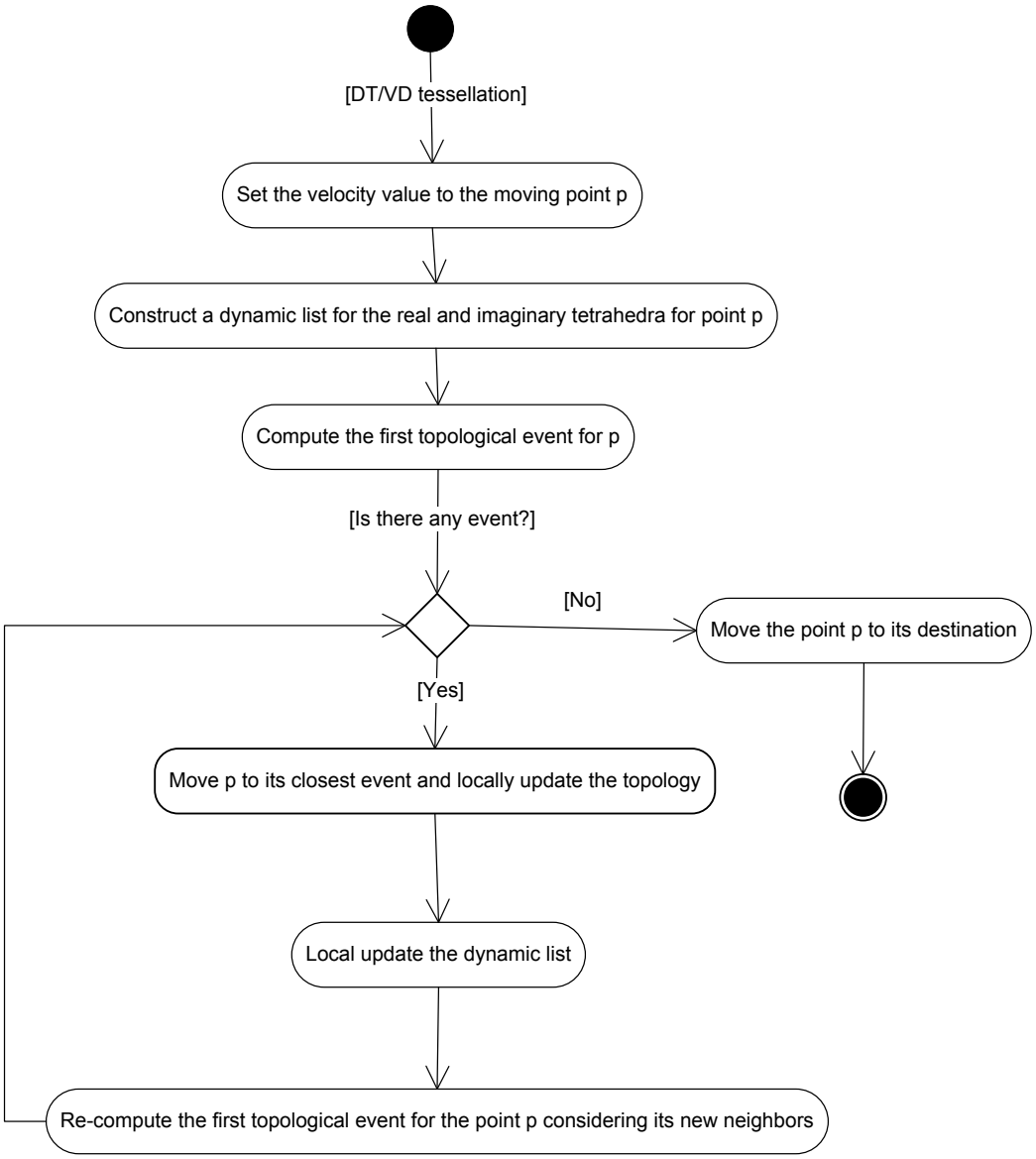
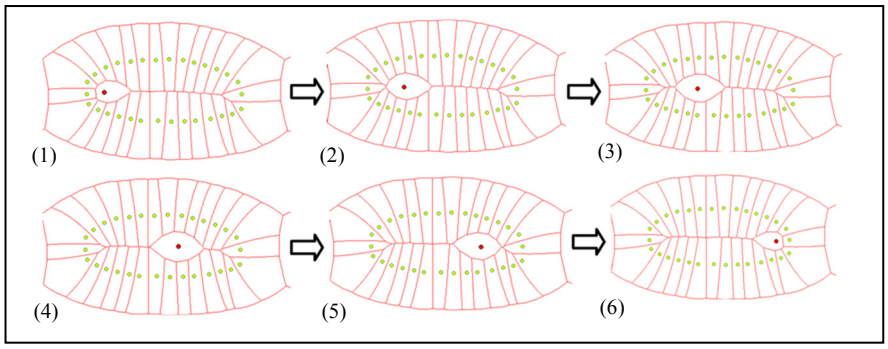
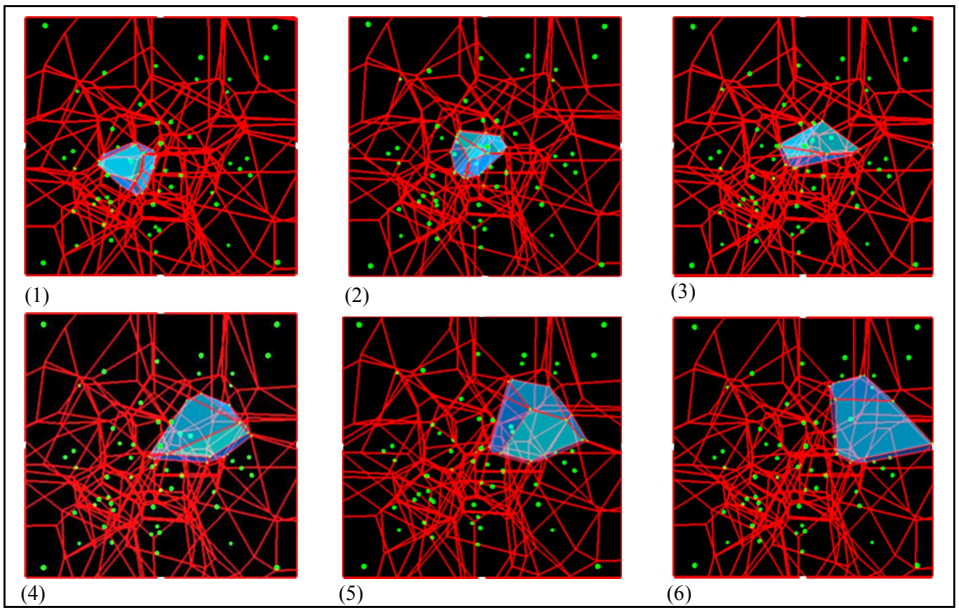


Figure 5-10 Flowchart of the management of a single moving point in a 3D DT and Voronoi tessellation.

During the motion, the moving point can change its set of nearest neighbors by *flip* operators. This modifies the shape of the Voronoi polyhedron surrounding the point. The neighbor changing process is smooth and continuous because of the use of the topological event (event-driven method). Due to the ability to change the connectivity, this method prevents the creation of deformed elements. Figure 5-11 demonstrates the movement of a single point in a 2D and 3D Voronoi tessellations.



(a)



(b)

Figure 5-11 The moving point can change its set of nearest neighbors by modifying the shape of the Voronoi polygon and polyhedron surrounding the point in 2D (Mostafavi 2002) (a) and 3D (b) respectively.

5.5 Management of several moving points within a 3D Delaunay tessellation

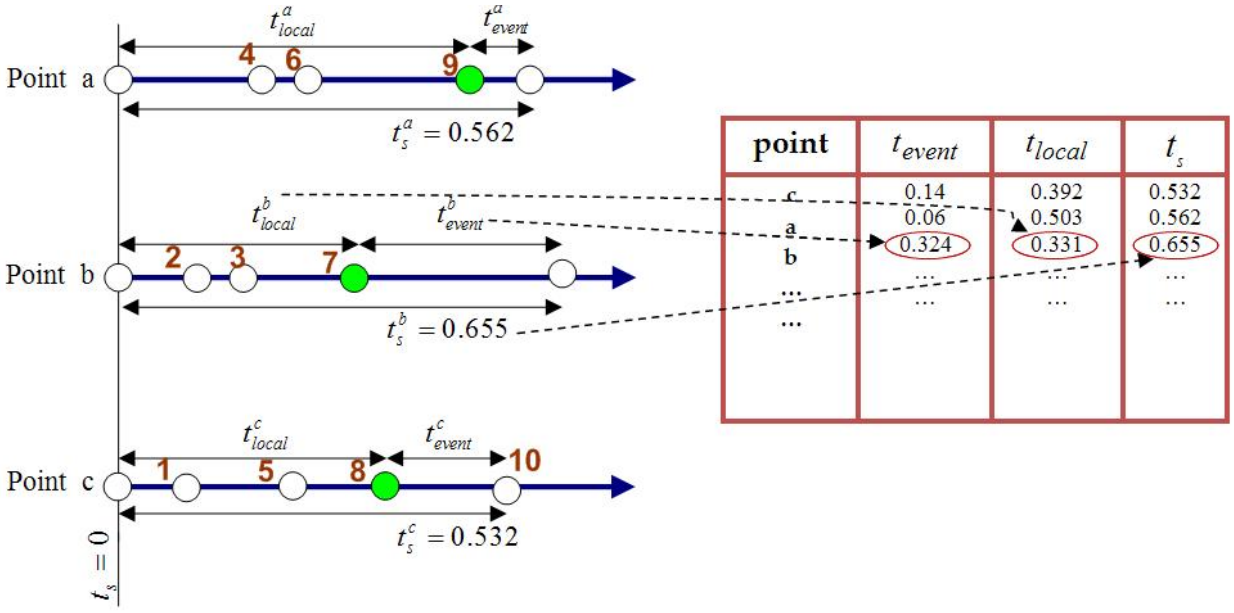
In the context of a dynamic field representation and simulation, we need to manage a large number of events in a deforming kinetic spatial tessellation in order to preserve the validity of the 3D tessellation. The sequence of the management of these events has an important

impact on the simulation results. We implemented an algorithm to manage several moving points in 3D which is an extension of Mostafavi and Gold's work in 2D (2004).

In order to manage the motion of several points in 3D tessellation, we need to manage the topological events of all the moving points in the tessellation simultaneously. This can be done using a priority queue data structure, where the moving points are organized with respect to their priority. The priority of each point is defined based on the value of its *simulation time* (t_S). The simulation time is the total time that takes for each point to move from its origin to its next closest topological event location on the trajectory. Thus, as a first step, the proposed algorithm computes the time that it takes for each moving point to move from its current position to its next topological event location. This time is referred to as event time (t_{event}) and is defined as: $t_{event} = d / v$, where d is the distance between the current position of the point and the location of its next closest topological event on its trajectory and $v = (v_x, v_y, v_z)$ is the velocity vector of the moving point. To link the simulation time and the topological event time, we define the local time (t_{local}) which is the time that it takes for each point to move from its origin to its current position. The relation between t_{event}, t_s and t_{local} is given by the following equation (figure12):

$$t_s = t_{event} + t_{local} \qquad (5\text{-}7)$$

We organize the points in the priority queue based on the value of their simulation times (figure5-12b). Therefore, the first member of the queue with the smallest simulation time is processed i.e. the first point on the list is moved to its new position and the necessary updates in the topological relationships of the point and its neighbors are made as seen in the previous section (figure 5-9). Following any motion, the priority queue is updated. This means t_s is updated after each topological event for the points involved in this motion in the priority queue. As a result, the priorities of some of the moving points may change. This occurs because, when a point moves, the related circumspheres to this point as well as the event time for the neighboring points change.



(a) (b)

Figure 5-12 The priority of the moving points is defined based on the value of their simulation time: a) an example of three moving points, their local, event, and simulation times and the order of movements (red numbers). In this example, for event #10, the point c is the next point to be moved, because $(t^c_s < t^a_s, t^c_s)$, (Mostafavi and Gold, 2004); b) the organization of the moving points within the priority queue.

In some applications such as in a fluid flow simulation, in addition to the data structure and the priority queue, the physical parameters and attributes associated with each moving point (particle in fluid flow simulation) and its neighbors must be updated after each topological event. The above process is reiterated until the end of the simulation process. The flowchart of the method for the management of several moving points in a 3D tessellation is demonstrated in figure 5-13.

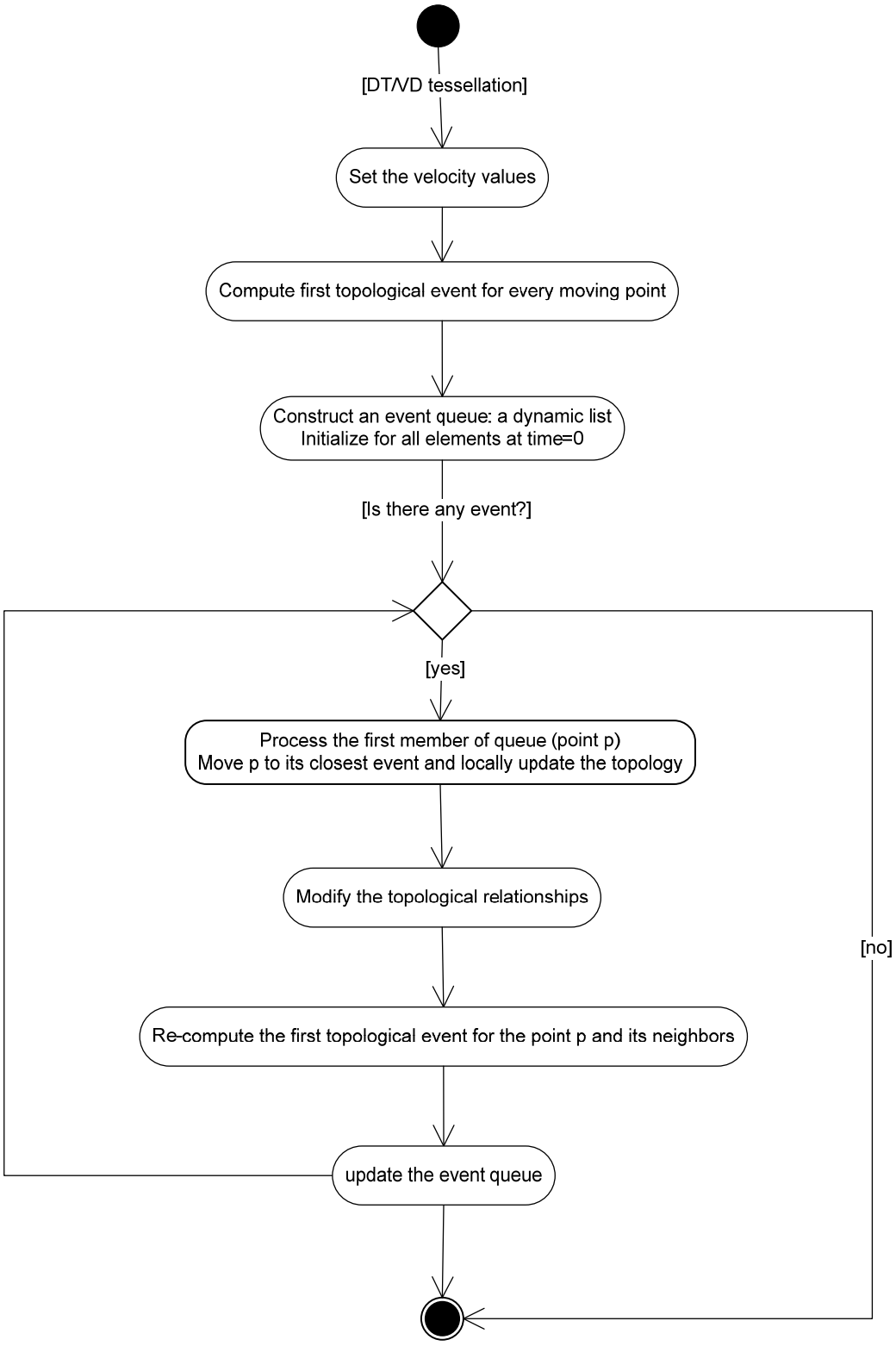


Figure 5-13 The flowchart of the movement of several moving points in a 3D DT and Voronoi tessellation.

Figure 5-14 demonstrates the initial and final Voronoi tessellation after the movement of several points in a 2D and 3D space. Figure 5-14a illustrates a 2D Voronoi tessellation, each point is free to move with a velocity. The priority of the moving points is defined based on the value of their simulation time. The order of movement and the location of topological event for each moving point are shown by green numbers and hallow point in figure 5-14b. During the motion, the moving points can change their set of nearest neighbors by *flip* operators (figure5-14c). For example, after topological event #11, the Voronoi cells for the points *a,b* are not neighbor anymore. This modifies the shape of the Voronoi polyhedron surrounding each moving point. Figure 5-14d and 5-14e similarity shows the neighborhood of the Voronoi cells for point *a* and *b*, also *c* and *d* change after motion in 3D.

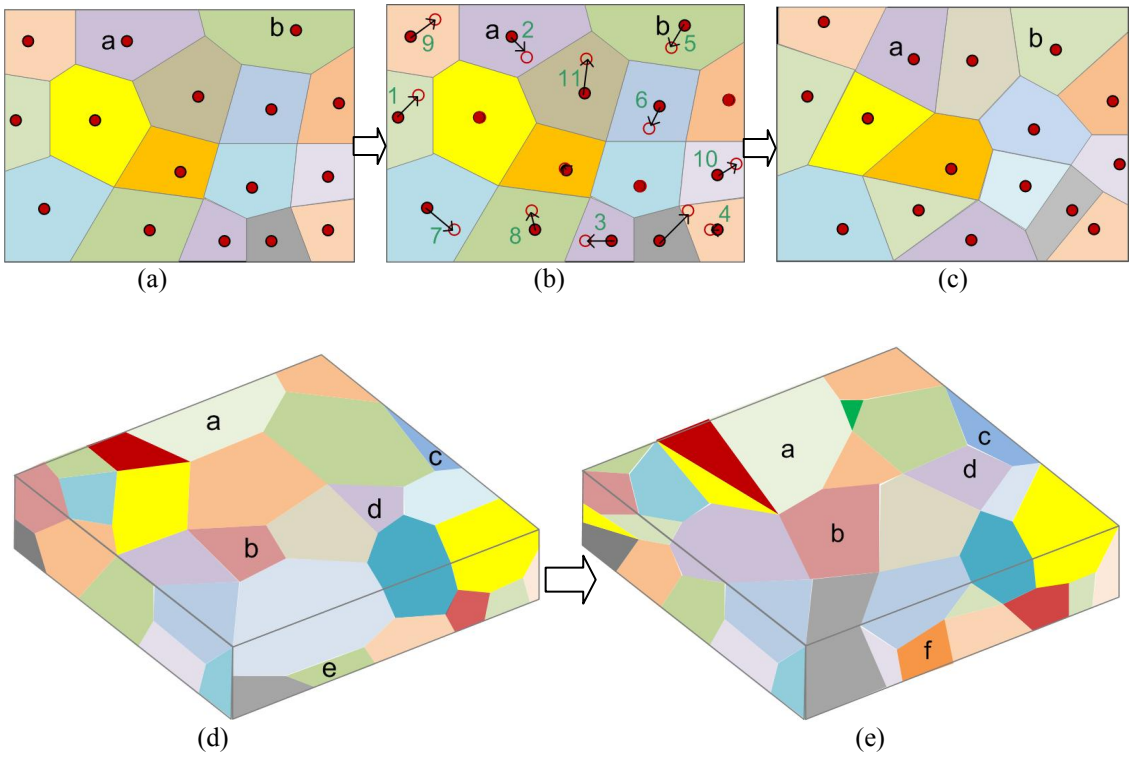


Figure 5-14 The initial and final Voronoi tessellation after the movement of several points in 2D and 3D spaces.

5.6 Complicated cases

During the management of the topological events of several moving points, some degeneracies and complicated cases may arise. Degeneracy cases substantiality exist due to the fact that we bring deliberately the points on the circumsphere of the adjacent tetrahedra which were discussed earlier. Therefore, here the complicated cases are explained.

5.6.1 Event ambiguity

During the management of the topological events of several moving points, some complicated cases, we called "event ambiguity", may arise that requires special care. These complexities occur when:

1) There are the simultaneous events, thus it becomes difficult to decide which event must be processed first. However, there are some simultaneous events that involve the areas that are far apart from each other and thus, there is no interference between them. These events can be processed in any order.

2) There are permanently zero events, this is due to the fact that, in some degenerate cases, the processing of a zero event can generate one of several new zero events. As a result, the algorithm enters an infinite loop.

Both cases may occur when the configuration of the point set is high regular. To treat these complicated cases, we use a perturbation method applying a small deviation (ε) in the position of the points on a paraboloid defined by projecting of all the points on a 4D space $(p_x, p_y, p_z, p_x^2 + p_y^2 + p_z^2 + \varepsilon, 1)$ (please see section 3.3.4).

5.6.2 Floating-point arithmetic

Another important problem is related to the computational precision of the geometrical algorithms. In this work, the algorithm for simulation has been implemented by floating-point arithmetic. Such computations are "inexact" due to round off errors and can create

problems for the geometric algorithm. For instance, in the case of computing the intersections between the trajectory of a moving point and the circumsphere of a tetrahedron in Euclidean geometry, when a trajectory is very close to a sphere, a non-intersecting case could be calculated as an intersection and vice versa. Figure 5-15 shows the possible locations of topological events of a moving point along its trajectory. In the figure, the yellow points show the events resulting from real triangles and the events related to imaginary triangles have not been shown for the sake of simplicity. These events were first obtained by applying the exact computing approach. Next, the computation of events was also performed using the double precision method. According to the results obtained from this experiment, only about 20% of the event locations computed were exactly the same using the “exact” versus the “double precision” computation methods.

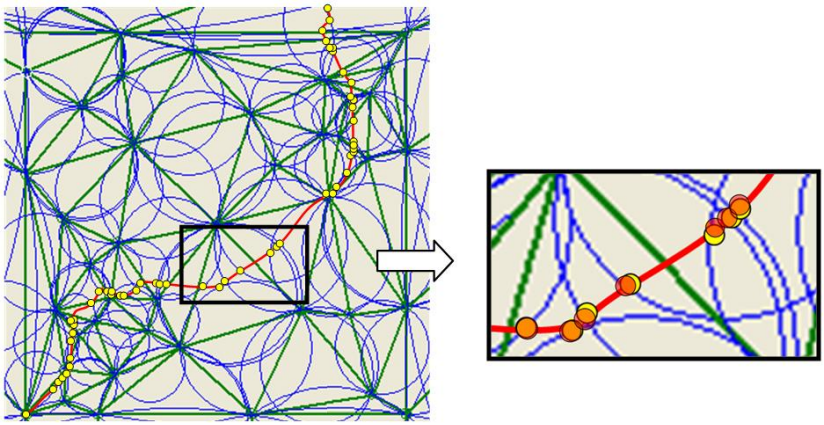


Figure 5-15 Events locations computed by two different computational methods; the yellow points are the locations of the topological events using an ``exact computation” and the red points represent the locations of the topological events computed using a ``double precision`` method.

Because of this round off errors, the complexity becomes more serious when discussing more than five co-spherical or coplanar points, or several near superposed spheres (figure 5-16). In these situations which are close to degeneracy positions, the floating-point arithmetic makes it very difficult to decide on the correct topological relations.

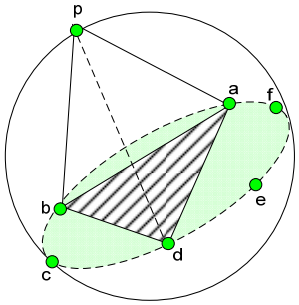


Figure 5-16 Degenerate case due to cospherical and coplanar points at the same time.

To solve this problem, an exact computation method must be used to find the topological event. In this method, a high precision computation is carried out which is computationally expensive. To reduce the cost of exact computation, Russel (2007) proposes using the exact computation approach only where there is some doubt on the accuracy of the rounded result and otherwise an inexact computation would be more efficient for general computation purposes.

The computation method used in this research work is based on the inexact method and because of the round off errors we frequently use a test based on the perturbation method (see section 3.3.4). This test is helpful to verify the correctness of topological events computation. This test should only be performed when two or more topological events (intersections) are very close to each other (using a defined tolerance). Although this solution ensures the data structure is almost valid during the simulation, further work is still required to improve the efficiency of the algorithm that is not defined as the scope of the thesis.

5.7 Conclusion

In this chapter, the development and the implementation of a 3D kinetic topological data structure based on Delaunay tetrahedralization was discussed. The proposed data structure allows the establishment and analysis of the topological relations between the moving elements of a tessellation and their updates. The time complexity of the proposed algorithm is $O(n^2)$. The development of the data structure allowed us to identify several challenging

issues such as collisions and coplanarity configurations or the complexities due to inexact arithmetic approach which was used in the implementation of the algorithm.

In the next chapter we explain the fundamental steps for adapting this data structure for the simulating of 3D dynamic fields and explain how this data structure keeps update the spatial tessellation describing the continuous field in a moving coordinate frame context (Free-Lagrangian methods).

Chapter 6: Application of Kinetic DT & VD for a Dynamic Field Simulation

6.1 Introduction

In this chapter, we study the feasibility of the application of the proposed data structure for a 3D dynamic field simulation in a moving coordinate frame (Free-Lagrangian method). The motivation of this chapter is due to the need to a 3D Free-Lagrangian method for the simulation and representation of the fields evolution over 3D space and time, especially for highly sheared flows, where mesh distortion and mesh tangling may occur when only the Lagrangian approaches are applied. Simulation of saturation fronts encountered in an oil-reservoir or simulation of material interfaces in gas-liquid-solid flow systems are good examples of the cases which tangling mesh may be observed. On the other hand, several works have confirmed the advantages of free-lagrangian simulations using KVD in 2D space. For example, Mostafavi and Gold (2004) has applied a 2D KVD for the simulation of global tide and investigated the advantages of the 2D kinetic data structure for modeling the interaction between the moving fluid elements and also between the moving fluid elements and fixed objects such as coastlines. Another example is Pons and Boissonnat's works (2007). They demonstrated the applicability and the efficiency of 2D kinetic

Delaunay and Voronoi data structures for modeling of dynamic interfaces between different materials undergoing large deformation and topology changes.

We, here, discuss the different steps to adapt the proposed kinetic data structure for simulation of a 3D dynamic field from the spatial discretization of the 3D field to its numerical integration based on an event driven method. Next, we apply the data structure for the simulation of a fluid flow as a good example for dynamic field and investigate how it represents the behavior of the dynamic phenomenon over 3D space and time. Finally we show and discuss the results of our approach with a series of numerical experiments.

6.1 Representation of a 3D spatial field: Underlying mesh generation

For simulation purposes, the fluid domain must first be discretized using an underlying mesh (spatial tessellation). As discussed in chapter 4, the mesh must provide a valid representation of the topological and geometrical information of the fluid domain. Therefore, we use the 3D Voronoi diagram to discretize the fluid region where each polyhedron represents a fixed mass of fluid; for example gas or water packets in a forest fire or groundwater simulation. The Voronoi diagram offers several important advantages (please see chapter 4):

- The mesh is composed of Voronoi polyhedra defined by points with an arbitrary distribution (figure6-1). Every polyhedron is unique and does not overlap with other polyhedra.
- The mesh elements are well-shaped and can represent the geometric complexity of the physical system more accurately than regularly defined mesh elements.
- The mesh is dynamic due to use an incremental construction method. Thus, it is possible to locally edit and manipulate the mesh (ex. mesh refinement), without having to rebuild the whole data structure.

- The mesh has a topological data structure. The adjacency relations between the mesh elements (topology) are readily defined among the polyhedral elements and there is no need to compute the connectivity matrixes at each time-step.

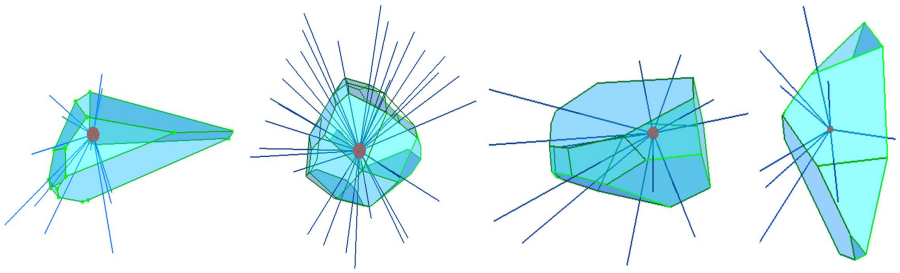


Figure 6-1 Examples of Voronoi elements which are polyhedral shaped elements. The straight blue lines represent the connections between the generating point of each Voronoi polyhedron and its nearest neighbors.

It is also important to represent the discontinuities and boundaries of the field region with its surrounding environment such as the air-water boundary layer, coast, islands, or other oceanographic boundaries in marine GIS or faults and folds in a hydrogeological fluid flow representation. These boundaries have a significant impact on simulation results. For example, the dynamic behavior of groundwater in a hydrogeological system with discontinuities or complex boundaries (such as impermeable layers or layers with different materials) is quite different to that of a homogenous space. In geosciences, boundaries are usually defined by various methods such as a set of points obtained via surface representation including non-uniform rational B-Splines or a regular grid of points. To represent these important features using a VD approach, as explained in chapter 4, some studies consider a set of points on each side of the boundaries (Okabe *et al.* 2000, Courrioux *et al.* 2001, Mostafavi and Gold 2004).

6.2 Representation of the dynamic behavior of the a 3D spatial field

The second step is the representation of the discrete field evolution by the numerical solution of a set of governing equations (PDEs) on the 3D mesh over time.

6.2.1 Governing equations

In this step the governing equations that describe the dynamic behavior of the field must be selected. For example, the law of the conservation of mass (eq. 6-1), conservation of momentum (eq. 6-2), and total energy (eq. 6-3) are the governing equations that describe the dynamic behavior of a compressible fluid flow (Campbell and Shashkov 2003):

$$\frac{1}{\rho}\frac{d\rho}{dt} = -\nabla \cdot u \qquad (6\text{-}1)$$

$$\rho\frac{du}{dt} = -\nabla p - \nabla.\tau \qquad (6\text{-}2)$$

$$\rho\frac{de}{dt} = -p\nabla \cdot u - \nabla.(\tau.u) \qquad (6\text{-}3)$$

where ρ, p and e are the density, pressure, and the total specific energy of the fluid respectively. $u = u(x, y, z)$ is the fluid velocity, τ is stress tensor, and (∇) and $(\nabla\cdot)$ are gradient and divergence operators.

The governing equations are continuous, thus, a discrete form of equations is required for numerical simulation. Mandal and Trease (1989) have developed a discrete form of the above governing equations in 3D space for fluid dynamics with a fixed time-step as:

$$\rho_i^{n+1} = \frac{m_i}{V_i^{n+1}} \qquad (6\text{-}4)$$

$$m_i \frac{{u_i}^{n+1} - {u_i}^{n}}{\Delta t} = \sum_{j=1}^{j}\left(p_{ij} - p_i\right)\cdot \hat{n}_{ij} A_{ij} - \sum_{j=1}^{j}\tau \cdot \hat{n}_{ij} A_{ij} \qquad (6\text{-}5)$$

$$m_i \frac{I^{n+1} - I^{n}}{\Delta t} = -\sum_{j=1}^{J}\left((pu)_j^{n+1/2} - (pu)_i^{n+1/2} \right)\cdot \hat{n}_{ij} A_{ij} - \sum_{j=1}^{J}(\tau \cdot u)_j \cdot \hat{n}_{ij} A_{ij} \qquad (6\text{-}6)$$

Where:

i	Element i
j	jth nearest neighbor of the element i
J	Total number of nearest neighbors associated with the element i
$n+1$	Next time step
$n+1/2$	Half time step level
V_i	Volume of the element i
u_{ij}	Fluid velocity on the face associated with the element i and the nearest neighbor j
A_{ij}	Area of a face separating the element i from its nearest neighbor j
M_i	Mass of the fluid associated with the element i
p_{ij}	Fluid pressure at the face associated with the element i and the nearest neighbor j
n_{ij}	Normal vector to the face associated with the element i and the nearest neighbor j
Δt	a fixed time-step

To solve the above three equations, an additional equation is required which defines the relation between the pressure, density and the internal energy such that (Campbell and Shashkov 2003):

$$p = (\frac{2}{3})I\rho \qquad (6\text{-}7)$$

Here, we explain the numerical solution of the governing equations using the proposed 3D kinetic Voronoi data structure.

6.2.2 Numerical integration of the governing equations by the KDT

These equations must be solved by considering a set of specific initial and boundary conditions. The initial condition explains the state of the field at the beginning of simulation process ($t=0$) and the field evolution will start from this initial condition. The initial values should be able to reflect the field state at $t=0$ or, at least, should be an acceptable approximation of it. In addition, boundary conditions must be introduced that

define the external forces acting on the field region. Boundary conditions are very important as different boundary conditions may introduce quite different simulation results.

In the proposed method, initial values such as initial velocity $u_0(x,y,z)$, pressure p_0, density ρ_0, internal energy e_0, and mass m are assigned to each element of the tessellation (figure 6-2). These quantities are element centered and, due to the interaction between each element and its neighboring elements, they change over time. For example, the velocity vector of each element at each time corresponds to the resultant force (vector F_i) of the local forces acting on the element (red vectors f_j in figure 2). The local forces are the forces from the neighboring elements or the forces defined by boundary conditions acting on the element:

$$F_i = \sum_{j=1}^{j} f_j = m_i \frac{du_i}{dt} \qquad (6\text{-}8)$$

The local forces (f_j) are computed on the polyhedron faces which are perpendicular bisecting planes between the generator of the element and each of its neighbors.

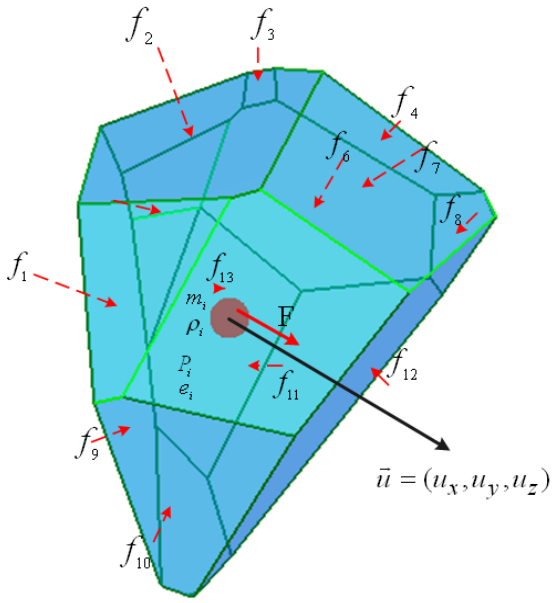


Figure 6-2 Forces acting on a mesh element (Voronoi polyhedron); each mesh element moves with a velocity vector proportional to the resultant force of the neighboring element forces acting on the element.

When an element moves with $u_0(x, y, z)$, the shape and number of the neighboring elements changes. This results in a new force, velocity vector, and other new attributes for the element. The new values of field variables are obtained from the discrete form of the governing equations in 3D space. One of the interesting advantages of the simulation based on the KDV is that time is managed by an event-driven method instead of a snap-shot model. Therefore, the fixed time-step (Δt) in the equations (6-5) and (6-6) is replaced by the topological event time in our approach which is variable.

To start the simulation process, the first topological event time (t_{event}) is computed for each element (i) and stored in increasing order in a priority queue. Then, the first element of the queue with the smallest topological event time is processed by moving it to its new position using its velocity vector and making the appropriate topological updates (see chapter 5). As mentioned, the velocity vector is resulted from of the total neighboring elements forces which are computed on the polyhedral element faces. Following the movement, for each element, the number of faces and their respective areas change in the new position. Next, using the governing equations (6-4), (6-5), and (6-6), the velocity of the moving element, as well as other variables of the moving element and its neighbors is updated. In our method, the updating process is relatively simple because the adjacency relationships are readily defined among the polyhedral elements using the underlying 3D topological data structure.

Following these changes, the first topological event time of the moving point and that of its neighbors are recomputed and finally, the priority queue is locally updated using these new values. This procedure is repeated until a user defined simulation time is reached. Figure 6-3 shows the fundamental steps for the simulation of a dynamic field such as fluid flow using the proposed 3D kinetic data structure.

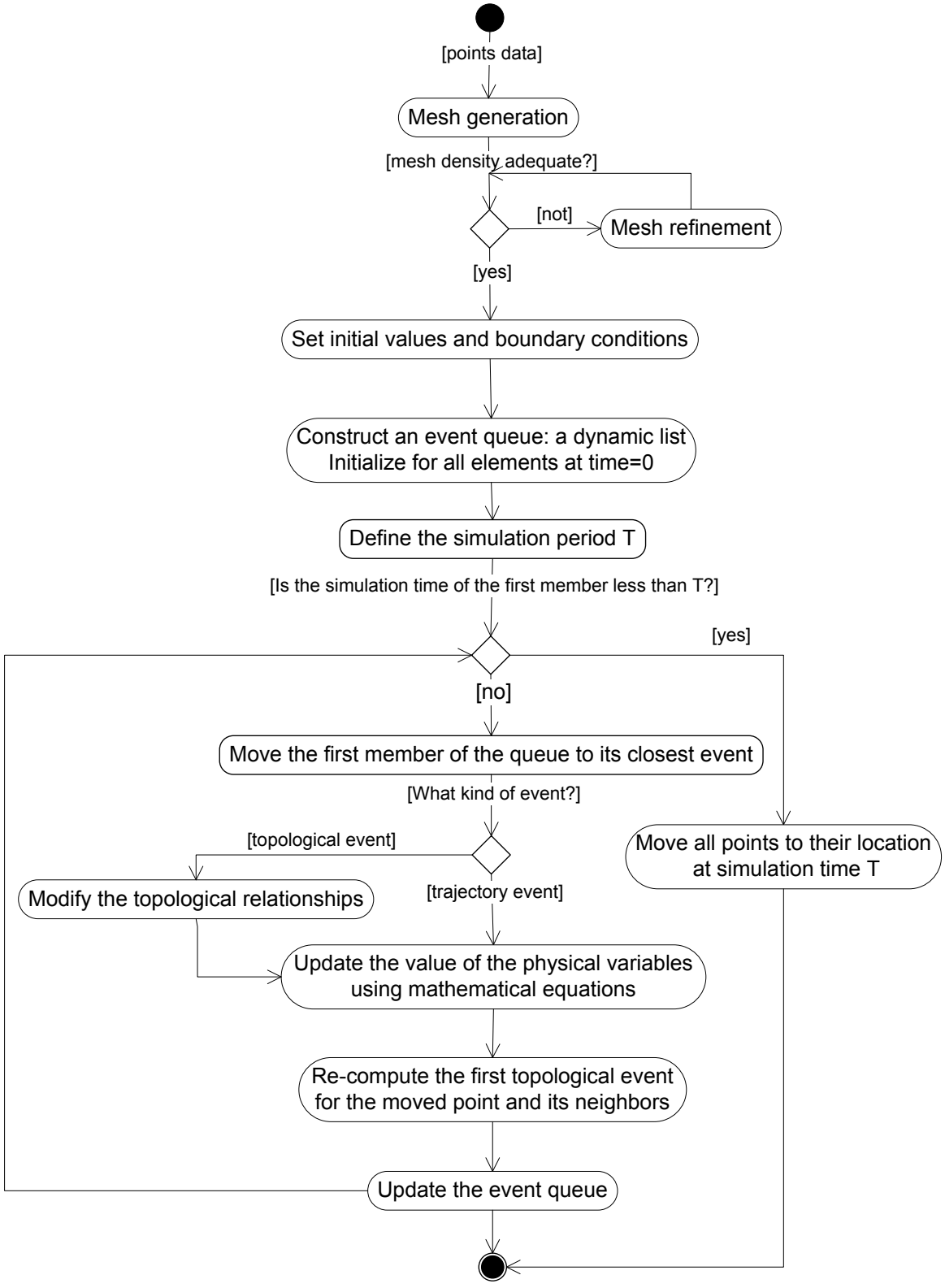


Figure 6-3 The fundamental steps for the simulation of a dynamic field such as fluid flow using the proposed 3D kinetic data structure.

6.3 Validation and results

To validate the proposed kinetic data structure and also, to study the feasibility of its application to the simulation of a 3D dynamic field, we applied the 3D kinetic DT &VD to three case studies for fluid flow simulation in 3D space. In all cases, the partial differential equations describing the field dynamics were the same (equations (6-5), (6-6) and (6-7)). For two first simulation examples, we have no literature to compare our experiment, thus the validations are based on a qualitative evaluation of the simulation results which represent the behavior of gases flow describing with the governing equations. The last case study is used in the literature, then, this provides the possibility of comparison and validation of the obtained results for our experiment. Hence, this case study provides a convenient way to evaluate the functionality of the 3D kinetic data structure for simulation purposes. In the following sections, we present and discuss the results obtained from each of these experiments.

6.3.1 First case study: Reflection

In the first case study, the simulation region is considered as a cube [(0, 1), (0, 1), (0, 1)] in such a way that the compressible fluid enters from the face x=0 with a given velocity and moves through the cube towards a rigid wall at x=1(figure6-4). Because of the interaction with the boundary, the fluid close to the boundary (x=1) is pushed back inside and its velocity direction is reversed which here called reflection (blue vectors in figure6-4).

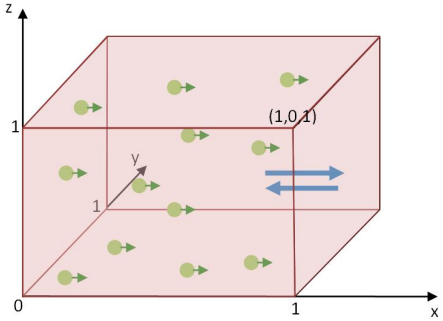


Figure 6-4 The simulation domain for the first example.

The simulation domain is represented by a set of 117 mass points with their respective Voronoi polyhedra distributed over the simulation region. The boundaries of the region are represented by two parallel sets of static points and the initial conditions have been obtained from a preliminary simulation of the compressible fluid flow (figure 6-5a).

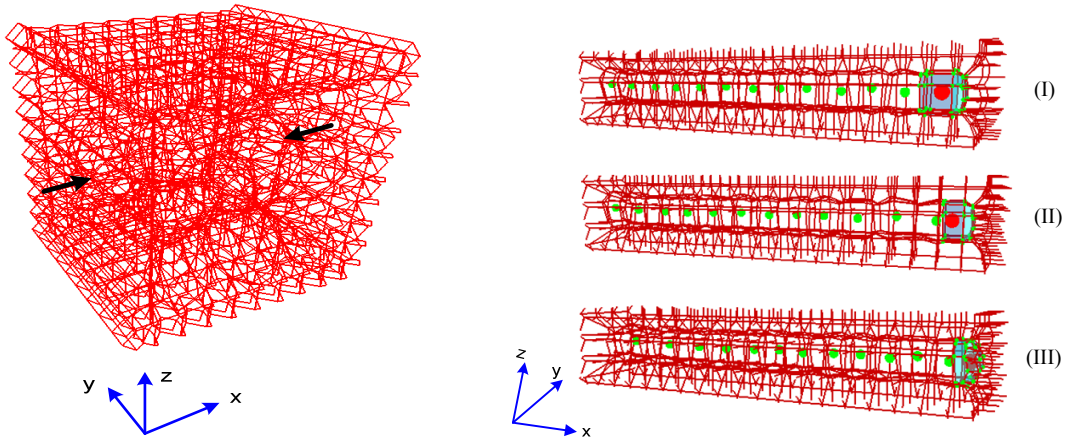


Figure 6-5 a) Boundaries and initial conditions: the fluid enters from the face x=0 with a given velocity and moves through the cube toward face x=1, b) Simulation results: a part of the simulation region close to the boundary, x=1, after 200(6-5b-I), 700(6-5b-II) and 1100(6-5b-III) topological events.

When the simulation process starts, the fluid elements (volume) begin to move toward the face x=1. Figure 6-5b shows the simulation results of a part of the simulation region close to the boundary after 200, 700 and 1100 topological events. The analysis of the physical properties of the elements shows that the velocity of the points and their polyhedra volumes decreases near the boundary, which is considered as a rigid wall. However, the pressure, density, and temperature of the elements increase. This process continues and the velocity of the highlighted element becomes almost zero with very high pressure, density, and temperature values. When updating the physical parameters of this element and its adjacent elements, a new velocity is computed. This velocity is more than zero but is in opposite direction (reflection). These results are in accordance with the behavior of gases flow which describes with the governing equations. In addition, the proposed KDS was able to manage the simultaneous motion of the 3D mesh elements and their continuously changing spatial relations.

6.3.2 Second case study: Gas expansion in one direction

For the second case study, we applied the 3D KV and DT to simulate the gas expansion. For the sake of simplicity, we consider that the fluid domain is bounded by a 3D rectangular boundaries [(0, 1), (0, 0.25), (0, 0.25)] and the fluid can move in one dimension (for example, horizontally). In this example, we assume that the middle of fluid domain has a higher density and pressure and according to the law of the conservation of mass, the mass of each element remains constant during the simulation process (figure6-6).

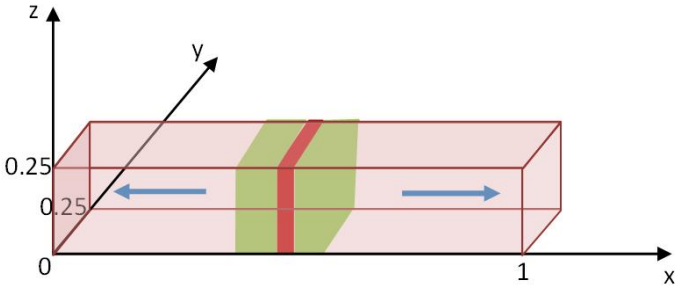


Figure 6-6 The simulation domain for the second example.

The simulation domain is represented by a set of 25 mass points with their respective Voronoi polyhedra distributed over the simulation region and similar to the previous case study, the boundaries of the region are represented by two parallel sets of static points.

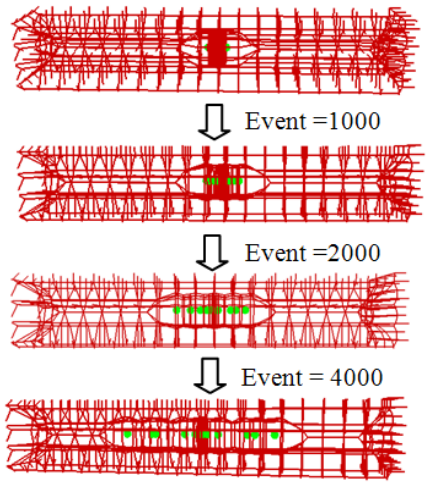


Figure 6-7 The simulation results after 1000, 2000 and 4000 topological events.

The results of this example show that the particles with higher pressure in the middle part of the simulation domain begin to move outward symmetrically and their volumes become larger (figure6-7). According to the assumption that the mass of the elements is constant during the simulation process, when the volume of a element increases, the density as well as the pressure of the element decrease. This process continues as long as the equilibrium does not exist in the simulation domain. It can be noted from the results that the proposed algorithm manages properly the movement of the mesh in 3D space and the dynamic behavior of the gas flow corresponds to our expectations i.e. the gas flow completely fills the simulation domain and equilibrium is achieved.

6.3.3 Third case study: Gas expansion in three directions

The third case study is the extension of the 1D fluid flow expansion to 3D (figure6-8). This is a very good example of a 3D dynamic field, where all mass points are free to move in 3D space and the spatial relations of the tessellation (mesh) elements representing the field change rapidly during the simulation process.

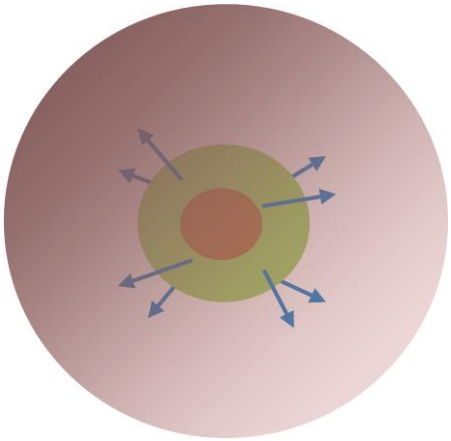
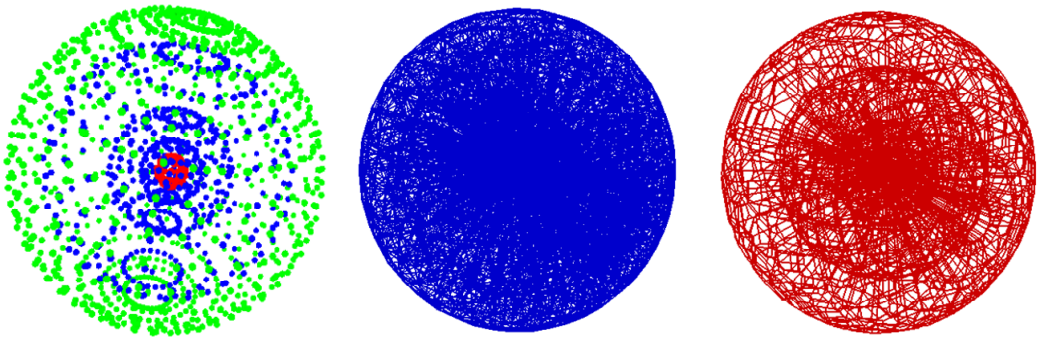


Figure 6-8 The simulation domain for the gas expansion in 3D.

Simulation of such a field using a Free-Lagrangian method necessitates the management of a very fast evolution of the state of the field as well as the spatial relations between the mesh elements. For this test, we used a set of points to represent the simulation domain. The boundary of the simulation region is represented by a set of stationary green points. Figure6-9b and 6-9c show the initial Delaunay tetrahedralization and Voronoi diagram for

the set of points respectively. For the definition of the initial state of the field, the initial values of Noh problem (Ghemyr et al. 1997) are used. The reason is that this problem is commonly used in literature for the validation of numerical integration methods for dynamic field simulation as there are analytical solutions for the Noh problem (Campbell and Shashkov 2003). Therefore, this provides the possibility of comparison and validation of the obtained results for our case study. Hence, the case study is an excellent example for the validation of the proposed data structure and the feasibility of its application for a dynamic field simulation.



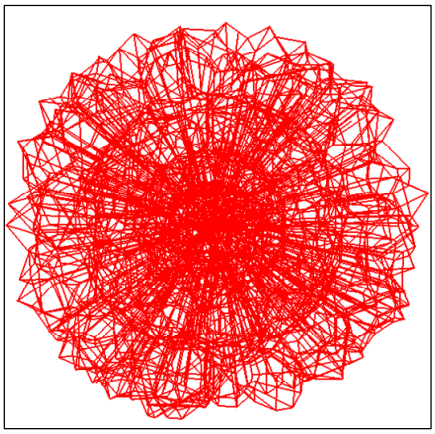
(a) (b) (c)

Figure 6-9 a) The simulation domain which is represented by a set of points, the points in the middle of the simulation domain have a very high density and pressure (red points) and other elements have a very low density and pressure (blue points), the boundary of the simulation domain is represented by stationary green points, b) Delaunay tessellation of the set of points, and c) 3D Voronoi diagram.

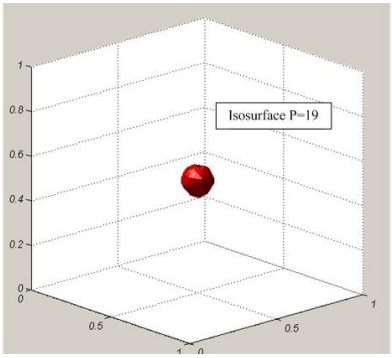
Therefore, we assume that the elements in the middle of the simulation domain (red points) have a very high density and pressure ($\rho_0 = 64, p_0 = 21$) moves in radial directions outward with speed ($v_0 = 1/3$). Other elements have a very low density and pressure which move inward at unit speed ($v_0 = 1$) (blue points) (figure6-9a). In addition, similar to other mentioned examples, we use the mass conservation law which means that the mass of each element remains constant during the simulation process. According to the governing equations, the mesh elements will move due to the pressure difference between the elements in the simulation region and their attributes changes accordingly.

When the simulation process begins, the elements move and interact with each other. Due to the interaction between the elements with higher pressures placing in the middle part of the simulation domain with their neighbors, the shape and volume of the elements with the low pressure are changed rapidly. According to the assumption that the mass of the element is constant during the simulation process, when the volume of an element decreases, its density as well as its pressure increases and vice versa.

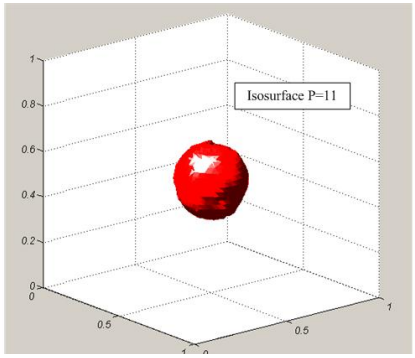
During the simulation process, we can observe that the interaction between the tessellation elements is managed properly and their relations are updated adequately when needed. The neighbor changing process is smooth and continuous because of the integral nature of the governing equations. The proposed 3D KDS have again allowed a continuous evolution of the 3D mesh without any mesh tangling problems by allowing local updates in the spatial relations between the mesh elements.



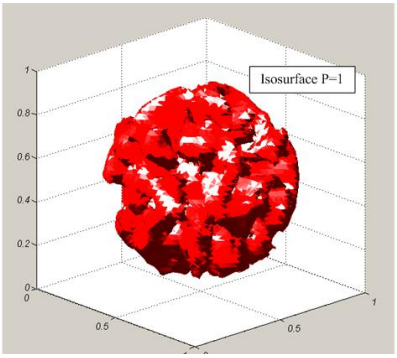
(a)



(b)



(c)



(d)

Figure 6-10 a) The Voronoi tessellation of simulation region after t_s=0.6 and iso-surfaces for (a) $p = 19$, (b) $p = 11$ and (c) $p = 1$ at t_s=0.6.

To validate the obtained results from this experiment, we compared them to results obtained from the analytical and other numerical solutions of the Noh problem in Ghemyr *et al.* (1997) and Mandal and Trease (1989). For this purpose, we have made several snapshots of the field during the simulation process for visualization and analysis purposes. Since our method uses an event driven approach, the location of each point depends on its own simulation time. However, for each snapshot, the points must be in their correct location for a given time t_s. Figure 6-10 shows the result of the fluid flow simulation at $t_s = 0.6$. In this figure, the Voronoi tessellation shows the state of the field at $t_s = 0.6$ (figure6-10a) and figure 6-10b, 6-10c, and 6-10d show the state of the pressure in the simulation domain using iso-surfaces for $p = 19, p = 11$ and $p = 1$ respectively at this time.

To better analyze the spatiotemporal evolution of the field, we investigate the simulation results by creating three slices on an x- y- z- plane. Our expectation is that the spatial distribution of the pressure will be symmetrical in all directions in 3D space. Figure 6-11 demonstrates the color map of the pressure evolution in the simulation domain confirming our expectations. This means that the topology and physical properties are properly updated during the simulation process and the dynamic behavior of the fluid flow is adequately represented. These results are comparable to the ones obtained from the analytical and numerical solution of the Noh problem as well, which can be found in Mandal and Trease (1989), Ghemyr *et al.* (1997), and Campbell and Shashkov (2003). However, it is important to mention that, in those studies, a fixed time-step method is used during the simulation process where all topological relations are reconstructed after each time-step. This means that, for each time step, the new position of all moving points is computed, the Voronoi tessellation is rebuilt, and the physical properties of all the mesh elements are updated. This process is computationally very costly. In addition, problems such as overshoots and undetected collisions may occur during the process which is avoided in the proposed method, because it uses an adaptive time step based on topological events in the mesh.

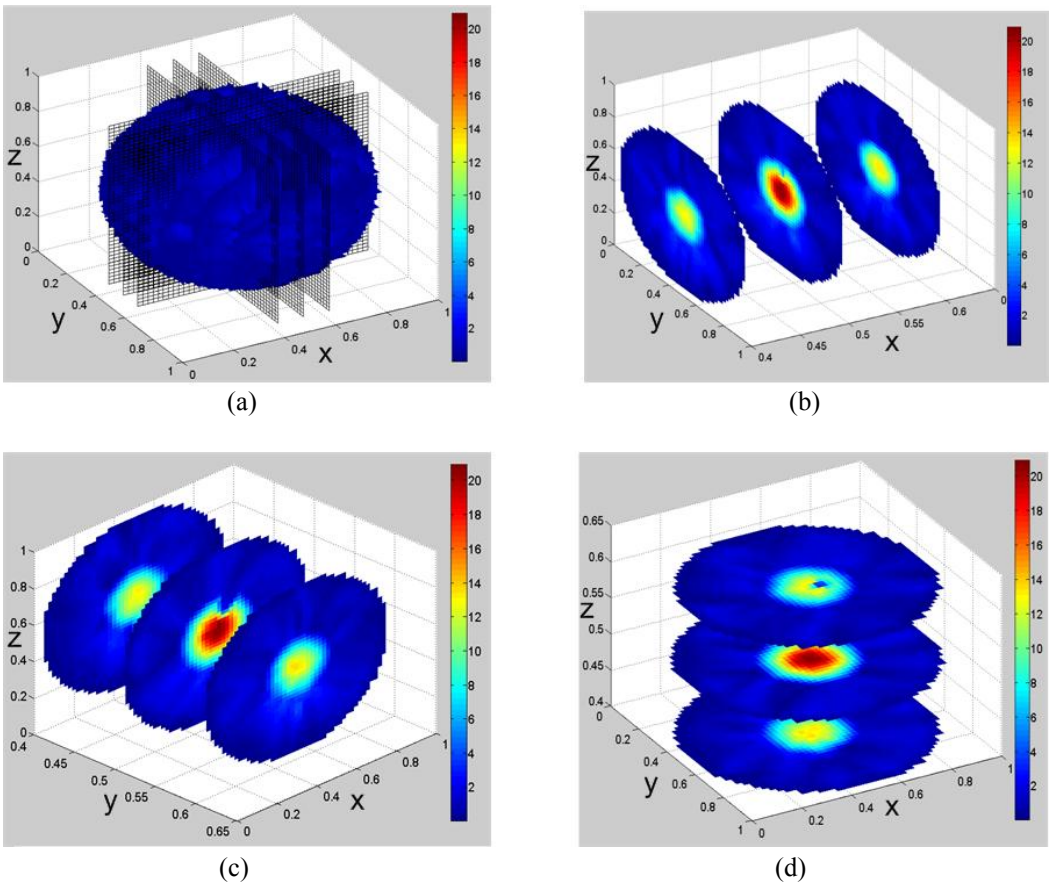


Figure 6-11 The simulation results at t_s=0.6: the colormap results show the pressure evolves symmetrically that is a reasonable representation of the physical symmetric problem.

The results also indicate a fair approximation of the density variation in the simulation domain. Figure 6-12 compares density values obtained from the simulation process (Red profile) and the results of the analytical solution of the Noh problem (Blue profile) for the same profile across the simulation domain. However, we can also see some differences between the two solutions across the simulation domain. These differences confirm the statements by other works such as Ghemyr *et al.* (1997) indicating the close relationship between the quality of the simulation results and the spatial resolution of tessellation. This is due to the fact that, depending on the complexity of the field, choosing an appropriate spatial and temporal resolution for the tessellation and time is essential to better approximate the field and its dynamic behavior. In the proposed method, time-steps are replaced by topological events which depend on the spatial resolution of the mesh. If the physical domain is discretized to an insufficient number of elements, the topological

events (t_{event}) may be large and the elements may move a large distance without properly updating their physical parameters. This may result in incorrect trajectories for the moving elements and consequently, unreliable simulation results. To solve this problem, we define a maximum tolerated value for the distance ($D_{\max}$) which each point is allowed to move on its trajectory without updating its physical properties. Then, based on $D_{\max}$ and the velocity, a maximum tolerated value for the topological event time ($t_{event-\max}$) is computed for each point. If the topological event time for a given point is smaller than $t_{event-\max}$, the process is carried out as described in the previous sections. However, when the topological event time is larger than $t_{event-\max}$, the point is moved to its new position for $t_{event-\max}$ without causing any change in the data structure and only the priority of the point and its neighbors as well as their physical properties are updated. Based on this method, we obtain a new density graph that shows a significant improvement in the estimation of density values (Green profile) which are closer to the results obtained from the exact solution of the Noh problem.

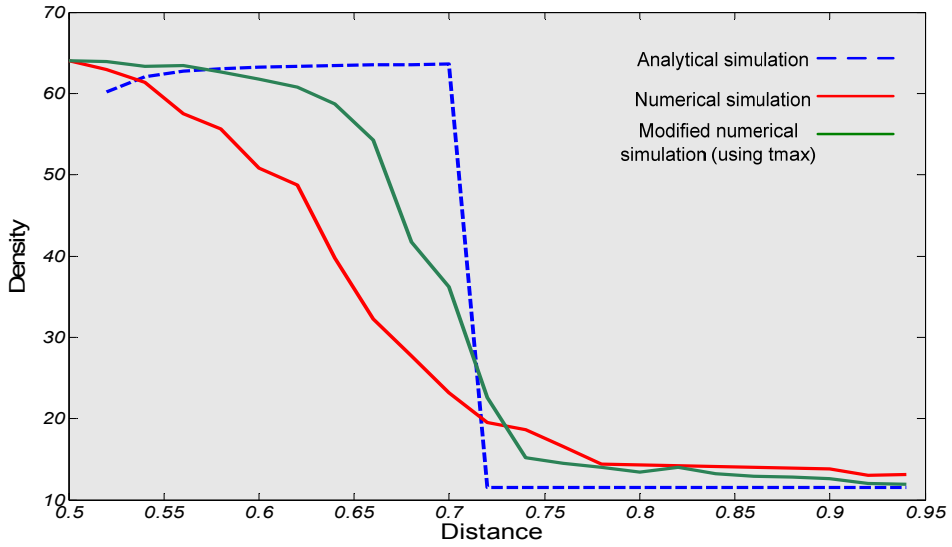


Figure 6-12 Comparison of density profiles obtained from numerical solution of the PDEs (green and blue graph) with its analytical solution at $t_s = 0.6$.

6.4 Discussion

The experiment allowed us to identify several other important issues for the simulation of 3D dynamic fields which are briefly described in the following section.

6.4.1 Complicated issues

The remaining question is: what is the optimal spatial resolution of a tessellation that provides the most accurate representation of a field? In a tessellation with a coarse spatial resolution, we may observe abnormal behavior in the simulation results because, within the kinetic spatial data structure, coarse spatial resolution results in large topological event times (t_{event}) and the elements move a large distance through the simulation domain without properly updating their physical parameters. To solve this problem, we define a maximum topological event time $(t_{event-\max})$. Although this solution generates improved results (figure 6-12), it is not a realistic representation of the physical properties of a field.

On the other hand, although a mesh with a very fine resolution provides a closer approximation of the field compared to reality, an extensive computation effort is required to check for collisions at times when none have occurred. Moreover, in a fine resolution case, there are a large number of topological events with bad transpositions in the queue due to round-off errors. This may cause a mesh crash. To solve this problem, an "exact" computation method must be used to find topological events and manage the queue. Therefore, choosing an optimal spatial resolution for the tessellation is important for a realistic representation of a dynamic field. Beower *et al.* (2005) argue that the resolution required for a tessellation to represent a dynamic field depends on several factors including the complexity and variability of the field, the specific physics of the problem being solved, and the boundary conditions.

Initial values and boundary conditions are important factors which have significant impacts on simulation results. With a set of poor initial values and boundary conditions, the simulation process may not be convergent and the results will be unreliable. These factors

are dependent on space and time. This means that, for example, the selection of the boundary value needs to consider the effect of spatial and physical information of the neighborhood which may change over time.

GIS can play an important role in preprocessing and preparing the initial and boundary conditions for the simulation process (Maguaire et al. 2005). In addition, GIS can play an important role in integration of the simulation results with other information. It can also be helpful for data organization, visualization, spatial querying, analysis, and prediction (particularly to support decision making based on multiple factors of spatial information).

6.4.2 Efficiency issues of the proposed method

Another important issue in the dynamic simulation process is the large amount of data to be managed which leads to increased storage needs and computing time. We estimate the time complexity of the proposed algorithm in three stages: pre-processing, initializing, and processing. During the pre-processing stage, the DT of a set of n points in R^3 is constructed in $O(n^2)$ using the incremental algorithm (in the worst case). During the initialization stage, $O(n)$ time is required to assign initial parameters (e.g. velocity, mass, pressure) to each point and placing the initial events into the event queue takes $O(n)$ time. Similar to the classification suggested in Gavrilova and Rokne (2003), the computation of the complexity of the first potential event for each moving point is as follows:

- Topological event: the upper bound on the number of predicted topological events at any time is $O(n^2)$ in 3D.
- Collision tests: The number of collision tests is equal to the number of neighbors a point has. This number is $O(n)$ in the worst case for each point.
- Trajectory event check: The trajectory event check is related to the point location problem which uses a tree data structure (a quad-tree). Therefore the complexity is $O(\log n)$.

The processing stage includes updating the data structure in response to topological and trajectory events, a collision detection test, and updating the event queue. Updating and

processing the data structure is based on the flip algorithm that is $O(n^2)$ in the worst case. Updating trajectory events is $O(1)$ and updating the collision event is $O(n^2)$. For all events, the event queue should be updated, which takes $O(log\ n)$.

6.4.3 Potential applications of the proposed kinetic data structure

In this research work, we have developed and implemented the proposed kinetic data structure for the representation and simulation of 3D dynamic fields which is required in some applications such as simulation of fluid flows, meteorological and environmental applications. Although the focus is on the representation and simulation of 3D dynamic fields, this data structure has capabilities for wide variety applications that need to represent and manage moving objects in 3D space such as aircraft control, ship navigation, and inter-satellite communication, etc. The data structure allows the establishment and analysis of topological relations between the moving objects which can be represented by points. Some analysis examples are as follows. In these examples, by objects we refer to either the objects which have an exact definition in real world such as aircrafts for aircraft control, or discrete elements (particles) that represent continuous fields:

a) Tracking of moving objects and analysis of their trajectories

The trajectory of a moving object can be represented by a set of line segments in three-dimensional space. In fact, a trajectory represents the geometric and temporal information that can be of key importance for some analyses. Analysis of the object trajectories offers the possibility of extracting information on the behavior of the moving objects and their interactions with the environment. These interactions are closely related to proximity problems. The proposed data structure based on the Voronoi diagram provides a clear definition of the proximity of the moving objects within a dynamic environment throughout the monitoring time. This can help to track oil plume, people or mobile devices in the spatial models of the dynamic environment and more importantly to help to recognize what activities they are carrying out.

b) Querying on the moving particles and objects and their neighbors in 3D space

Queries are frequently used in GIS. There are several types of queries including location queries, range queries, nearest neighbor queries and k-nearest neighbor queries. The Voronoi diagram defines adjacency relations between objects. Two objects in a Voronoi diagram are adjacent if their respective Voronoi cells share a common face. Hence, it is easy to collect k-nearest neighbors for the particle or object of interest using its adjacent cells in the Voronoi diagram. These queries become more complex when dealing with a 3D dynamic space. Since, in such environment, objects move and their relations change continuously. Therefore, the queries must respect the time dimension. These queries aim at determining the state of moving objects in the past, present and future and their relative position with respect to each other and other static objects within the dynamic environment. A typical example of such queries is 'report the closest ship in the sea that is moving towards another ship in difficulties'. Another type of query aims at determining the moving particles and objects that will intersect a query window referred to as a predicative spatio-temporal window query (Kollios et al., 1999). We may also be interested in answering queries on the moving objects and their neighbors in 3D environment such as:

"What were the k-nearest neighbors of an object of interest in a specific moment in the past?" "What are the closest neighbors of the object at the present time?" "Which objects will be closer to the object of interest at a given time in the future?"

Querying on the past position of particles or objects can be done using the history of the trajectory of each particle or object. This requires keeping track of the trajectories of the moving objects and the evolution of the topological relations between objects. Considering the number of the topological events, this may not be feasible. An alternative solution would be to return all objects to their position in a required time in the past and rebuild the part of the topology in a given region. There is still room for further investigation on this topic.

Querying nearest neighbors at present time can directly be done by the topological model. We assume that the update operation in the topological model is efficient enough and the query operation at present time includes latest changes in the

neighborhood of the object of interest. In the case of querying nearest neighbors of a moving point at future time, we need information on the trajectories of moving objects and their velocity vectors. This knowledge is necessary for the prediction of the position of the moving objects in the future. One solution for this kind of query is to allow the topological data structure to change until the moving objects arrive to their predicted position and then proceed with the computation of nearest neighbors as suggested for queries at present time. The second solution will be a reconstruction of the topology for the predicted position of the objects in the future and making the nearest neighbor queries on the model.

c) Event detection and management

During the monitoring of a dynamic phenomenon, several kinds of events may be of interest for the users of the system. For example, the rapprochement of a particular object to a point of interest, possibility of collision between two or more moving objects, their agglomeration or dispersal and visibility of a particular object could be considered as an important event within a dynamic scene. Hence, in order to support the user in his task, these events should be detected within the dynamic space automatically based on some spatial and temporal criteria. The proposed kinetic data structure is an event based approach for the representation of a dynamic phenomenon. The events can not only be defined based on as the topological changes, but also we can consider the so-called trajectory events that consist of direction or property change of a moving object. These events can be combined with other spatial and spatiotemporal query operators as criteria for the kinetic data structure. Therefore, this could help to detect or predict those kinds of event in the dynamic environment and respond to it adequately and timely.

6.5 Conclusion

In this chapter, we investigated the feasibility of the application of the 3D kinetic Voronoi data structure to simulate 3D dynamic fields and demonstrated that this method is

inherently adaptable to the complexity of the field. The kinetic Voronoi data structure generates a mesh adaptable to arbitrary distribution of the points and allows the mesh moves and the topology, and physical parameters (e.g. pressure, density, velocity) of the mesh elements are properly updated at each time they change. In addition, to solving the problems resulting from using a fixed time step, such as overshoots and undetected collisions, an event-driven algorithm has been designed to detect events by 3D Delaunay data structure. The simulation results were satisfactory and reasonably comparable to the ones obtained from the analytical and numerical solutions of the same governing equations. In the end of the chapter, we discussed the important issues which are mainly related to the characteristics of the dynamic phenomena to be represented and simulated using the data structure namely spatial resolution of the tessellation, initial and boundaries values, and the large amount of data to be managed. The discussion was followed by explaining some of the potential applications of the proposed kinetic data structure for the spatial analysis of both dynamic fields and moving objects.

Chapter 7: Conclusions

The value of GIS in spatial modeling and simulation efforts is widely known for a large variety of applications. GIS can provide modelers with strong computing platforms for spatial data management, visualization, querying, and analysis. In particular, GISs are an effective tool to integrate various heterogeneous sources of spatial and non-spatial information. However, GIS have very limited capability for the simulation of 3D dynamic fields. The role of GIS remains primarily in the area of data processing, analysis, visualization and management of simulation results that must often be sliced into several 2D datasets and it is impossible to completely embed a simulation process directly in GISs as one of their analytical function. As argued in the thesis, the problem originates from the fact that current GIS raster and vector data structures are mostly 2D and static and are not designed to address the dynamic and three-dimensional aspects of fields.

This brought us to propose a new 3D kinetic data structure for the modeling and simulation of 3D dynamic field as explained in chapter one. The data structure is based on 3D Voronoi diagram (VD) and its dual Delaunay tetrahedralization (DT) which is a good candidate to deal with 3D and dynamics behavior of fields. On this basis, three specific objectives were defined:

I. *Constructing an adaptive tessellation for fields modeling in a three dimensional space by developing a refinement method based on 3D dynamic Voronoi data structure*

II. *Handling a moving tessellation in a 3D dynamic environment by developing a 3D kinetic Voronoi data structure*

III. *Evaluating the feasibility of the application of the three dimensional kinetic Voronoi diagram for a dynamic field simulation*

7.1 Contribution of the research work

Our literature review, chapter 2 and 3, demonstrated that VD and DT have elegant properties for representation of a field. They provide a natural discretization of space based on the samples which are collected from the field. DT and VD can be defined by points with an arbitrary distribution, creating elements of different sizes and shapes which can fill any arbitrary surface or volume. In addition, each Voronoi polyhedron can have an arbitrary number of neighbors whose connectivity are clearly defined and can be explicitly retrieved if needed. However, as explained in chapter four, the use of the VD to model three-dimensional geoscientific datasets which are not randomly positioned is very challenging. A 3D VD tessellation of those data does not deal with discontinuities within a field such as faults or folds in hydrogeological systems. This is because in a VD, a volume of influence is assigned to each point and the field value is assumed to be constant within the volume. In several works DT were used to preserve the discontinuities, since the vertices of the tetrahedra lie on the sampled points. However, because of the nature of the dataset which is usually highly irregular, poor-quality elements may appear in the tessellation. These poor-quality elements are not suitable for simulation purposes.

Therefore, the first contribution of this research work is the development of a new refinement method for providing a well-shaped and well-spaced tessellation for representing of a field as discussed in chapter 4. The proposed method is dynamic in the sense of begin capable to add or remove of a point within a tessellation without any global reconstruction of topology and generates a tessellation conforming to the complexity of a field in three stages: preprocessing, processing and post processing. In the preprocessing stage, the discontinuities within a given field and the field boundaries with its surrounding

environment are represented by a set of points on each side of them in which the length between two neighboring boundary points is equal to the shortest edge in the tessellation ($L_{\min}$). In addition, the density of the data point representing the field is edited based on $L_{\min}$ using the "deletion" operation. The processing stage identifies and improves the poor quality tetrahedra considering both geometrical and physical characteristic of the field. The geometrical criteria are related to the *shape and size* of elements such as shortest or longest length of edge ($L_{\min}, L_{\max}$), angles in tetrahedra (β) or the volume of each element ($V_{\min}, V_{\max}$), while the physical criteria are based on a *physical field variable criterion* such as minimum and maximum gradient of the field values. In addition, the method through its dynamic operations like insert, delete and flips, offers an efficient refinement method that has some significant advantages over the existing algorithms. This allows the users, through a preprocessing stage, to modify and adjust the density of the tessellation elements locally, i.e. on-the-fly delete and insert point within the tessellation and explore interactively a data set to extract information while the static algorithms lack the ability to do it. The time complexity of the algorithm is estimated as $O((m+l)^2)$, where m is the points number of the final tessellation and l is the number of deleted points the preprocessing stage.

The second contribution of the research work is the handling of dynamic behavior of 3D fields by extending of 3D dynamic DT and VD to "kinetic" one. By kinetic, we mean the data structure is capable of keeping update the 3D spatial tessellation during a dynamic simulation process which is described in chapter 5. Therefore, the tessellation elements are allowed to move with different speed and interact with each other in 3D. The 3D adjacency relationship between elements (topology) is readily defined and maintained among the Voronoi cells. However, the points movement may change the adjacency relationship. In Delaunay tetrahedralization, the dual of VD, the 3D spatial relation changes when a point moves in the circumsphere of a *real* tetrahedron, moves out form the circumsphere of an *imaginary* tetrahedron, moves on a trajectory being tangential with the circumsphere of a real tetrahedron or meets another point of the tessellation on its trajectory. In these cases, which are called topological events, the connectivity of the tessellation must be updated.

Therefore, the first topological event (t_{event}) for each moving element is detected and stored in the priority queue. All elements move, where the priority of the moving points is defined based on the value of their simulation time ($t_{simulation}$), and the update of the 3D topology is local and on-the-fly. In addition to topological events, it is possible to consider the trajectory events identifying the specific time that points may change their trajectories. the proposed method shows that the use of an event-driven method allows solving some of the inherent problems to the application of the fixed time step in the existing methods such as overshoots, undershoots and undetected collisions in the 3D tessellation. The structure also provides the necessary operations to easily track the moving elements in 3D environment and make necessary queries on their location and configuration at any time.

The third contribution of the research work is the adapting of this 3D kinetic data structure for the representation and simulation of a 3D spatial field and its dynamic behavior as discussed in chapter 6. This chapter explains the fundamental steps of this adaption which leads us to propose a new 3D Free-Lagrangian method, where the underlying mesh is defined by the 3D kinetic VD. In the proposed method, initial values (such as initial velocity, pressure, density) are assigned to each element of the tessellation. These values may change over time due to the interaction between each element and its neighboring elements. For example, the velocity vector of an element may continuously change, because it is the result of the total neighboring elements forces which changes as its neighbors change. Using the proposed kinetic data structure, the 3D topology, connectivity, and physical parameters of the mesh elements can properly be updated when a topological event happens. Therefore, the proposed method properly addresses two important shortcomings of the existing Free-Lagrangian methods. The first weakness is related to maintaining and processing of the connectivity relations between tessellation elements during the simulation which are not efficient enough. The second problem lies in determining the optimal time-step to avoid overshoots or undershoots in the underlying mesh during the simulation process. However, it was observed that a very coarse mesh results in incorrect trajectories for the moving elements and consequently, unreliable simulation results. This is because the topological events (t_{event}) may be large and the elements may move a large distance in 3D space without properly updating their physical

parameters. To solve this problem, we constrained topological events to not exceed from a maximum tolerated value for the topological event time $(t_{event-\max})$. This constraint significantly improved the simulation results. The proposed method was validated for several experiments. The overall qualitative and quantities evaluation of the simulation results reveals that the 3D field dynamic behavior for the given examples was comparable to the analytical solutions of the same governing equations.

Although the focus of the research work was on the representation and simulation of a 3D dynamic field, this data structure has capabilities for the representation and management of moving objects in 3D space. It allows the establishment and analysis of topological relations between the moving objects. Hence, the proposed data structure has a significant potential for a wide variety of applications that need to represent and manage moving objects or dynamic fields.

7.2 Future works

The research work proposed a data structure which can effectively deal with the simulation of 3D dynamic fields in a moving coordinate frame context (Free Lagrangian methods). However, some important issues still need further work for a more realistic simulation within GIS using this data structure. These challenging issues can be divided into two areas: the computational aspects of the data structure and the complexities related to application domain.

- ***Studying the computational aspects of the data structure***

During the development of the data structure, we faced some degenerate cases which complicated the movement of one or several points in the 3D tessellation. These cases were related to the computational aspects of the data structure such the complexities due to an inexact arithmetic approach, collision and coplanar configuration of points. Although these complexities were considered in the research work and the solutions were presented, more research work for making the method more efficient would be worthy.

In addition, in this research work, the data structure has been developed for sets of data points. Therefore, the simulation domain must modeled by a series of points. However, there exists some linear, planer or volumetric features in real world such as boundaries and discontinuities which must be considered. For example, faults in groundwater simulation or buildings in weather pollution modeling in urban area are more like planer or volumetric features. Therefore, it would be particularly interesting to model these important features using line segments, polygons or volumes instead of using only points. For this purpose, there is a need for developing the data structure for line segment and polygons.

- ***Studying the challenging issues related to application domain***

One of the important issue concerns the characteristics of the dynamic phenomena to be represented and simulated using the data structure such as spatial resolution of the tessellation and initial and boundary values, and the large amount of data being managed. The research work used the initial and boundary conditions which have been obtained from a preliminary simulation of the compressible fluid flow. In addition, it was assumed that the data is stored in a GIS database. Therefore, it would be interesting to investigate on practically using a GIS database for data management (data storage and retrieval), manipulation and analysis. This will be helpful in preprocessing and preparing the initial and boundary conditions for the simulation process. In addition, it will bring other advantages such integration of the simulation results with other information, data visualization, spatial querying, analysis and prediction (particularly to support decision making based on multiple factors of spatial information). Investigating on these important issues and their impact on the efficiency of the algorithm and the simulation results are essential to completely embed a simulation process in GIS.

The research work validated the 3D kinetic data structure itself and its application for simulation of a 3D dynamic field by a simple case sturdy. It would be interesting to apply the data structure for integrating of complex process models with GIS using real data and consider some complex geometrical and physical information such as the geometry of boundaries and discontinuities in the 3D space. Finally, comparing the simulation results

with other methods based on irregular or regular tessellations based on criteria such as efficiency, accuracy, functionality and stability, flexibility would be worthy research work.

References

Abdul-Rahman, A., and Pilouk, M., *Spatial Data Modelling for 3D GIS*, Berlin: Springer, 2008.

Ahokas H., Koskinen L. Task 7: Modelling the KR24 long-term pumping test at Olkiluoto. Taskforce; April 18, 2005.

Alber, G., Guibas, L.J., Mitchell, J.S.B., and Roos, T., Voronoi diagrams of moving points, *International Journal of Computational Geometry and Applications*, 8, 365–380, 2007.

Albrecht, J.H., Universal GIS operations, Ph.D. thesis, ISPA—University of Vechta, Germany, 1996.

Aronoff, S., *Geographic Information Systems: A Management Perspective,* WDL Publications, Ottawa, 1989.

Armstrong, M.P., Temporality in spatial databases, in *Proceedings GIS/LIS '88*, San Antonio, USA, 2, 880–889, 1988.

Aurenhammer, F., Power Diagrams: Properties, algorithms, and applications, *SIAM Journal on Computing*, 16, 78-96, 1987.

Avis, D., and E1Gindy, H., Triangulating point sets in space, *Discrete and Computational Geometry*, 2, 99-111, 1987.

Bekey, J.A., Karplus, W.J., and Kogan, B.Y., *Modeling and Simulation: Theory and Practice,* Springer, 2003.

Bennet, D.A., A Framework for the integration of geographical information systems and model base management, *International Journal of Geographical Information Science*, 11(4), 337-357, 1997.

Bern, M.W., Flaherty, J.E., and Luskin, M., *Grid generation and adaptive algorithms*, The Institute for Mathematics and its Applications, Vol. 113. Springer-Verlag, New York, NY, 1999.

Bivand, R. and Lucas, A., Integrating models and geographical information systems, in Openshaw, S., and Abrahardt, R.J. (eds.)*, Geocomputation,* Taylor & Francis, London*,* 331-363, 2000.

Boissonnat, J.D., Yvinec, M., *Algorithmic Geometry*, Cambridge University Press, New York, 1998.

Blessent, D., Hashemi Beni, L., Therrien, R., 3D modeling for hydrogeological simulation in fractured geological media, in: *Proceeding of the 19th IASTED International Conference on Modeling and Simulation*, 2008.

Bonham-Carter, G.F., Geographic Information Systems for Geoscientists, Modeling with GIS (Computer Methods in the Geosciences), Vol. 13: United Kingdom, Pergamon, 1996.

Boots, B., Spatial tessellations, in: Longley P.A., Goodchild, M. F., Maguire, D.J., and Rhind, D.W., *Geographic Information Systems and Science,* Second Edition, John Wiley, Chichester, 2005.

Bosch, M., Lithologic tomography: method and application to the Côte díArmor region (France). Conference Abstracts: 3D Modelling of Natural Objects: a Challenge for the 2000, Vol. 2. Nancy, France, 4–5th June, 1998.

Bower, K.M., Gable, C., and Zyvoloski, G., Grid resolution study of ground water flow and transport, Yucca Mountain, *Ground Water*, 43(1), 122–132, 2005.

Breunig, M., *Integration of Spatial Information for Geo-Information Systems*, LNES No. 61, Springer Verlag, 1996.

Brimicombe, A., *GIS, Environmental Modelling and Engineering and Simulation*, CRC Press, 2003.

Bric, V., 3D Vector data structure and modelling of simple objects in GIS, MSc thesis, ITC, 1994.

Bric, V., Pilouk, M., and Tempfli, K., Towards 3D GIS: experimenting with a vector data structure. In *proceedings of the symposium mapping and Geographic Information Systems (ISPRS)*, 30, 634-640, 1994.

Brown, D.G., Riolo, R., Robinson, D., North, M., and Rand, W., Spatial process and data models: Toward integration of agent-based models and GIS. *Journal of Geographical Systems,*7(1), 1-23, 2005.

Burrough, P.A., Environmental modeling with geographical information systems, in Stein, A., et al. (ed.), *Models in action,* 59–69, 1996.

Burrough, P.A., and Mcdonnell, R.A., *Principles of Geographical Information Systems*, Oxford: Oxford University Press, 1998.

Burrough, P.A., Vendeursen, W. and Heuveklink, G., Linking spatial process models and GIS: A marriage of convenience or a blossoming partnership, In *Proceedings GIS/LIS*, San Antonio, Texas, USA, 88(2), 598-607, 1988.

Camara, G., Monteiro, A.M.V., Paiva, J.A., Gomes, J., and Velho, L., Towards a unified framework for spatial data models, *Journal of the Brazilian Computing Society*, 7, 17–25, 2000.

Cambray, B., Three-Dimensional (3D) Modelling in a geographical database, in: *11th International Symposium on Computer-Assisted Cartography*, 338-347, 1993.

Cambray, B., and Yeh, T.S., A Multidimensional (2D, 2.5D and 3D) geographical data model, in: *6th International Conference on Management of Data*, 1994.

Champan, L., and Thornes, J.E., The use of geographical information systems in climatology and meteorology, *Journal of Progress in Physical Geography*, 27(3), 313–330, 2003.

Campbell, C.J. and Shashkov, M., A compatible Lagrangian hydrodynamics algorithm for unstructured grids, *International Selcuk Journal of Applied Mathematics*, 2003, 4, 53-70.

Carey, G. F., *Finite Element Modelling of Environmental Problems: Surface and Subsurface Flow and Transport*, Wiley & Sons, New York, 1995.

Carette, V., Mostafavi, M A., Devillers, R., Rose, G., Hashemi, L., Extending marine GIS capabilities: 3D representation of fish aggregations using Delaunay tetrahedralisation and Alpha shapes, *Geomatica*, 62(4), 361-374, 2008.

Cheng, S.W., Dey, T.K., Edelsbrunner, H., Facello, M.A., and Teng, S.H., Silver exudation, in: *Proceeding of 15th ACM Symposium on Computational Geometry*, 1-13, 1999.

Ciolli, M., de Franceschi, M., Rea, R., Zardi, D.and Zatelli, P., , Modeling of evaporation processes over tilted slopes by means of 3D GRASS Raster, in: *Proceedings Open Source Free Software GIS – GRASS users conference 2002*, 2002a.

Cohen-Steiner, D., de Verdière, E.C., and Yvinec, M, Conforming Delaunay Triangulations in 3D, in: *Proceedings of the Eighteenth Annual Symposium on Computational Geometry*, 199–208, 2002.

Cordes, C., Putti, M., Accuracy of Galerkin finite elements for the groundwater flow equation in two and three dimensional triangulations, *International Numerical Methods in Engineering*, 52, 371–387, 2001.

Couclelis, H., People manipulate objects (but cultivate fields): Beyond the raster-vector debate in GIS. In Frank AU, Campari I, and Formentini U, editors, *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, volume 639 of Lecture Notes in Computer Science, Springer-Verlag, 65–77, 1992.

Couclelis, H., Space, time, geography, in: Longley PA, Goodchild MF, Maguire DJ, and Rhind DW, editors, *Geographical Information Systems,* John Wiley & Sons, second edition, 29–38, 2005.

Courrioux, G., Nullans, S., Guillen, A., Boissonat, J.D., Repusseau, P., Renaud, X., and Thibaut, M., 3D volumetric modelling of Cadomian terranes (Northern Britanny, France): An automatic method using Voronoi diagrams, *Tectonophysics*, 331, 181-196, 2001.

Cova, T.J., and Goodchild, M.F., Extending geographical representation to include fields of spatial objects, *International Journal of Geographical Information Science*, 16(6), 509-532, 2002.

Crowley, W.P., A Free-Lagrange method for numerically simulating hydrodynamic flows in two dimensions, in: *Proceeding of the Second International Conference on Numerical Methods in Fluid Dynamics,* 1970.

Dannenlogue, H.H. and Tanguy, P.A., Three dimensional adaptive finite element computations and applications to non-Newtonian flows, *International Journal for Numerical Methods in Fluids,* 13, 145-165, 1991.

Davis, B.E., and Davis, P.E., Marine GIS: concepts and considerations, in: *Proceedings GIS/LIS '88*, 1988.

de Berg, M., Van Kreveld, M., Overmars, M., and Schwarzkopf, O., *Computational Geometry: Algorithms and Applications,* Springer-Verlag, Berlin, second edition, 2000.

de Cougny, H.L., and Shephard, M.S., Parallel refinement and coarsening of tetrahedral meshes, *International Journal of Numerical Methods in Engineering*, 46, 1101–1125, 1999.

Devillers, O., The Delaunay hierarchy, *International Journal of Foundations of Computer Science*, 13(2), 163–180, 2002a.

Devillers, O., On deletion in Delaunay triangulations, *International Journal of Computational Geometry and Applications*, 12, 193-205, 2002b.

Devillers, O., Teillaud, M., Perturbations and vertex removal in a 3D Delaunay triangulation, in: *Proceedings 14th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 313–319, 2003.

Devillers, O. and Pion, S., Efficient exact geometric predicates for Delaunay triangulations, in: *Workshop Algorithm Engineering Experiments*, 37-44, 2003.

Di Luzio, M., Srinivasan R., and Arnold, J.G., A GIS-coupled hydrological model system for the watershed assessment of agricultural nonpoint and point sources of pollution, *Transactions in GIS*, 8(1), 113-136, 2004.

Edelsbrunner, H., *Geometry and Topology for Mesh Generation*, Cambridge University Press, Cambridge, 2001.

Edelsbrunner, H., and Seidel, R., Voronoi diagrams and arrangements, *Discrete and Computational Geometry*, 1, 25–44, 1986.

Edelsbrunner, H., and Shah, N.R., Incremental topological flipping works for regular triangulations, *Algorithmica*, 15, 223-241, 1996.

Egenhofer, M., and Franzosa, R., Point-set topological spatial relations, *International Journal for Geographical Information Systems,* 5, 161–74, 1991.

Egenhofer., M., Clementini, E., and Di Felice, P., Topological relations between regions with holes, *International Journal of Geographical Information Systems,* 8, 129–42, 1994.

Ellul, C. and Haklay, M., Requirements for Topology in 3D GIS, *Transactions in GIS*, 10(2), 157-175, 2006.

Farin, G., Hoschek, J., Kim, M., *Handbook of Computer Aided Geometric Design*, Elsevier, 2002.

Fedra, K., Embedded GIS in environmental management, *GIS Development*, 10(2), 16-22, 2006.

Fisher, P., Visualizing uncertainty in soil maps by animation, *Cartographica*, 30(2-3), 20-27, 1993.

Facello, M.A., Implementation of a randomized algorithm for Delaunay and regular triangulations in three dimensions, *Computer Aided Geometric Design*, 12, 349–370, 1995.

Fisher, T. R., Integrated three-dimensional geoscientific information system (GSIS) technologies for groundwater and contaminant modeling, in: *HydroGIS 93: Application of Geographic Information Systems in Hydrology and Water Resources* (ed. by K. Kovar & H. P. Nachtenebel), 211, 235-242, 1993.

Flowerdew, R., Spatial data integration, in: *Geographic Information Systems: Principles and Application*, 1, 375-387, 1991.

Fotheinghman, A.S., and Wegener, M., *Spatial Models and GIS: New Potential and New Models*, Taylor and Francis, London, 2000.

Fortune, S., A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2, 153–174, 1987.

Frank, A.U., Tiers of ontology and consistency constraints in geographical information systems, *International Journal of Geographical Information Science*, 15, 667–678, 2001.

Ferrez, J.A., Dynamic triangulations for efficient 3D simulation of granular materials. PhD thesis, EPFL University, 2001.

Fery W.H., Selective refinement: a new strategy for automatic node placement in graded triangular meshes, *International Journal for Numerical Methods in Engineering*, 24(11), 2183–2200, 1987.

Frittes, M.J., Crowley, W.P., and Trease, H.E., *The Free-Langrange Method*, volume 238. Springler-Verlag, Berlin, 1985.

Gable, C.W., Trease, H., and Cherry, T., Automated grid generation from models of complex geologic structure and stratigraphy, Santa Barbara: National Center for Geographic Information and Analysis, *proceedings paper LA-UR-96*, 1996.

Galton, A., Fields and objects in space, time, and space-time, *Spatial Cognition and Computation*, 4(1), 39–68, 2004.

Gavrilova, M., Proximity and Applications in General Metrics, PhD Thesis, University of Calgary, 1998.

Gavrilova, M., Rokne, J. and Gavrilov, D., Dynamic collision detection algorithms in computational geometry, in: *12th European workshop on Computational Geometry,* Munster, Germany, 103–106, 1996.

Gavrilova, M., Rokne, J., Collision detection optimization in a multi-particle system, *International Journal of Computational Geometry and Applications*, 13, 279–302, 2003a.

Gavrilova, M. and Rokne, J., Updating the topology of the dynamic Voronoi diagram for spheres in Euclidean d-dimensional space. *Computer Aided Geometric Design*, 20, 231–242, 2003b.

Gibbons, A., *Algorithmic Graph Theory,* Cambridge University Press, 1985.

Gimblett, H.R., *Integrating geographic information systems and agent-based modeling techniques for simulating social and ecological processes*, Oxford University Press, London, 2002.

Ghemyr, M., Cheng, B., and Mihalas, D., Noh's constant-velocity shock problem revisited, *Shock Waves*, 7(5), 255-274, 1997.

Göbel, R. and Zipf, Al., How to define 3D Geoprocessing Operations for the OGC Web Processing Service (WPS)? Towards a Classification of 3D Operations, in:

Proceeding of International Workshop on Computational Geoinformatics, ICCSA (1), 708-723, 2008.

Gold, C.M., Problems with handling spatial data the Voronoi approach, *CISM Journal*, 45(1), 65-80, 1991.

Gold, C.M., An event-driven approach to spatio-temporal mapping, *Geomatica*, 50(4), 415–424, 1996.

Gold, C.M., What is GIS and what is not?, *Transactions in GIS*, 10(4), 505–520, 2006.

Gold, C.M., Surface interpolation, spatial adjacency and GIS, in: *Three Dimensional Applications in Geographic Information Systems*, ed. J. Raper. London: Taylor and Francis Ltd, 21-35, 1989.

Gold, C. M., Chau, M., Dzieszko, M. and Goralski, R., 3D Geographic Visualization: The Marine GIS, in: *11th International Symposium on Spatial Data Handling*, (Ed.: Fisher, P. F.), Springer, pp. 17-28, 2004.

Gold, C.M., and Condal, A.R., A spatial data structure integrating GIS and simulation in a marine environment, *Marine Geodesy*, 18, 213-228, 1995.

Gold, C.M., Ledoux, J., and Dzieszko, M., A data structure for the construction and navigation of 3D Voronoi and Delaunay cell complexes, in: *13th International Conference in Central Europe on Computer Graphics*, *Visualization and Computer Vision*, Plzen, Czech Republic, 21–22, 2005.

Goodchild, M.F, Geographical data modeling, *Computers & Geosciences*, 18(4), 401–408, 1992a.

Goodchild, M., Geographical data modeling, *Computers & Geosciences*, 18, 401-408, 1992.

Goodchild, M., GIS, spatial analysis, and modelling: overview, In Maguire, D.J., Batty, M. and Goodchild, M.F., *GIS, Spatial Analysis and Modelling*, ESRI Press, Redlands, California, 2005.

Goodchild, M., Yuan, M. and Cova, T.J., Towards a general theory of geographic representation in GIS, *International Journal of Geographical Information Science*, 21, 239-260, 2007.

Goodchild, M.F., and Glennon, A., Representation and computation of geographic dynamics, In: Hornsby, K.S., and Yuan, M., editors, *Understanding Dynamics of Geographic Domains*, Boca Raton: CRC Press, 13–30, 2008.

Gross, J. T., and Yellen, J., *Graph Theory and Its Applications*, Boca Raton, FL: CRC Press, 1999.

Guibas, L., Knuth, D.E., and Sharir, M., Randomized incremental construction of Delaunay and Voronoi diagrams, *Algorithmica*, 7(4), 381-413, 1992.

Guibas, L.J., Kinetic data structures: A state of the art report, in: *3rd Workshop on Algorithmic Foundations of Robotics*, 1998.

Guibas, L.J., and Russel, D., An empirical comparison of techniques for updating Delaunay triangulations, in: *Symposium on Computational Geometry*, 170-179, 2004.

Guibas, L.J., and Stolfi, J., Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, *ACM Transactions on Graphics*, 4, 74–123, 1985.

Hale, D., Atomic meshes: from seismic imaging to reservoir simulation, in: *Proceeding of 8th European Conference on the Mathematics of Oil Recovery*, Freiberg, Germany, 2002.

Hazelton, N.W.J., Leahy, F.J., and Williamson, I.P., On the Design of Temporal Referenced 3-D Geographic Information Systems: Development of Four-Dimensional GIS, in: *Proceeding of GIS/LIS '90*, 357-372, 1990.

Hashemi, L., Blessent, D., Mostafavi, M.A. and Terrien, R., A 3D Free-Lagrangian method to simulate three-dimensional groundwater flow and mass transport, in: *19th IASTED International Conference on Modeling and Simulation,* Quebec, CA, 2008.

Hashemi, L., Mostafavi, M.A., and Pouliot, J., Dynamic field process simulation within GIS: the VORONOI approach, in: *Proceeding of ISPRS 2008*, China, 891-898, 2008.

Held, M., Klosewski, J., and Mitchell, J., Collision detection for fly-through in virtual environments, in: *12th Annual ACM Symposium on Computational Geometry*, V13-V14, 1996.

Houlding, S.W., *3D Geoscience Modeling: Computer Techniques for Geological Characterization*, Springer-Verlag, Berlin, 1994.

Icking, C., Klein, R., Köllner, P., and Ma, L., Java applets for the dynamic visualization of Voronoi diagrams, in: *Lecture Notes in Computer Science*, 191 – 205, 2003.

Jessell, M., Three-dimensional geological modelling of potential-field data, *Computers & Geosciences*, 27(4), 455-465, 2001.

Joe, B., Construction of three-dimensional Delaunay triangulations using local transformations, *Computer Aided Geometric Design*, 8, 123–142, 1991.

Jones, C.B., Data structures for three-dimensional spatial information systems in geology, *International Journal of Geographical Information Systems*, 3, 15–31, 1989.

Karssenberg, D., and de Jong, K., Dynamic environmental modelling in GIS: 1. Modelling in three spatial dimensions, *International Journal of Geographical Information Science*, 19, 559–579, 2005.

Kelmelis, J., Time and Space in Geographic Information: Toward a Fourth-Dimensional Spatio-Temporal Data Model, Ph.D. Thesis, Pennsylvania State University, 1991.

Kemp, K.K., Fields as a Framework for Integrating GIS and Environmental Process Models: Part I and II, *Transactions in GIS*, 1, 219–246, 1997.

Kim, KH., Lee, K., and Lee, JH., 3D geographical analysis within JAVA/VRML-based GIS: lantern operation, in: *Proceedings of the 3rd International Conference on GeoComputation*, 1998.

Kirkwood, D., Pouliot, J., Therrien, R., Macquarrie, K., and Li, S., GeoTopo3D development of a 3D predictive modeling platform for exploration, assessment and efficient management of mineral, petroleum and groundwater resources, Proposal for a GEOIDE Project, 2003.

Kose, T., Yamamoto, T., Anegawa, A., Mohri, S., and Ono, Y., Source analysis for polycyclic aromatic hydrocarbon in road dust and urban runoff using marker compounds, *Desalination*, 226, 151–159, 2008.

Kosik, R., Fleischmann, P., Haindl, B., Pietra, P., and Selberherr, S., On the interplay between meshing and discretization in three-dimensional diffusion simulation, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(11), 1233-40, 2000.

Lee, K., *Principles of CAD/CAM/CAE Systems*, Addison Wesley Longman, Inc, 1999.

Langran, G., A review of temporal database research and its use in GIS applications, *International Journal of Geographical Information Systems,* 3, 215-32, 1989.

Langran, G., *Time in Geographic Information Systems*, Taylor & Francis, London, 1992.

Lattuada, R., and Raper, J., Application of 3D Delaunay triangulation algorithms in geoscientific modeling, in: *Proceeding GIS Research UK 95*, 150 – 153, 1995.

Lawson, C.L., Software for C^1 Surface Interpolation, in: Rive, J. (Ed.) *Mathematical Software* III, Academic Press, New York, 161-194, 1977.

Ledoux, H., Modelling Three-dimensional Fields in Geosciences with the Voronoi Diagram and its Dual, PhD thesis, University of Glamorgan, 2006.

Ledoux, H., and Gold, C.M., Simultaneous storage of primal and dual three-dimensional subdivisions, Computers, Environment and Urban Systems, 31, 393-408, 2007.

Ledoux, H. and Gold, C.M., Modelling Three-dimensional Geoscientific Fields with the Voronoi Diagram and its Dual, *International Journal of Geographical Information Science*, 22(5), 547-574, 2008.

Lepage, F., Génération de maillages tridimensionnels pour la simulation des phénomènes physiques en géosciences, PhD thesis, University of Nancy, 2003.

Li, R., Data structures and application issues in 3-D geographic information systems, Geomatica, 3, 209-224, 1994.

Li, R., Chen, Y., Dong, F., and Qian, L., 3D Data Structures and Applications in Geological Subsurface Modeling, *Intl. Archives of PRS*, Vol. XXXI, Part B4, 508-513, 1996.

Li, X., Hodgson, M.E.: Vector Fields Data Model and Operations, Journal of *GIScience and Remote Sensing*, 41(1), 1–24, 2004.

Liu, Y., Goodchild, M., Guo, Q., Tian, Y., and Wu, L., Towards a General Field model and its order in GIS, *International Journal of Geographical Information Science,* 6, 623-643, 2008.

Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W., *Geographical Information Systems: Principles, Techniques, Management and Applications*, John Wiley and Sons, USA, 2005.

Maantay, J.A., Tu,J., Maroko, A.W., Loose-coupling an air dispersion model and a geographic information system (GIS) for studying air pollution and asthma in the Bronx, New York City, *International Journal of Environmental Health Research*, 19(1), 59 – 79, 2009.

Maguire, D.J., Batty, D.J., and Goodchild, M., *GIS, Spatial Analysis and Modelling*, ESRI Press, 2005.

Malcevic, O.G., Dynamic-mesh finite element method for Lagrangian computational fluid dynamics, *Advances in Water Resources*, 38, 965-982, 2002.

Mallet, J.L., GOCAD: a computer aided design program for geological applications, In: Keith Turner (ed.) *Three-Dimensional Modeling with Geoscientific Information Systems*, NATO ASI Series, 123-142, 1992.

Mallet, J.L., *Geomodeling*, Oxford University Press, 2002.

Mandal, D.A., and Trease, H.E., Parallel processing a three-dimensional Free-Lagrange code: a case history, *International Journal of High Performance Computing Applications*, 3(2), 92-99, 1989.

Mansell, R.S., Liwang, M., Ahujab, L.R., and Blooma, S.A., Reviews and analyses adaptive grid refinement in numerical models for water flow and chemical transport in Soil, *Vadose Zone Journal*, 1:222–238, 2002.

Mennis, J., Leong, J., and Khanna, R., Multidimensional map algebra, in: *Proceedings of the 8th International Conference on GeoComputation*, 2005.

Medvedev, N., Voloshin, V.P., Luchnikov, V.A. and Gavrilova, M., The algorithm for three-dimensional Voronoi S-network, *Journal of Computational Chemistry*, 27, 1676–1692, 2006.

Meyers, D. E., Spatial interpolation: An overview, *Geoderma*, 62, 17–28, 1994.

Miller, G.L., Talmor, D., Teng, S.H., Walkington, N., and Wang, H., Control volume meshes using sphere packing: generation, refinement and coarsening, in: *Fifth International Meshing Roundtable (Pittsburgh, Pennsylvania),* 47–61, 1996.

Miller, G., and Pav, S.E., Fully incremental 3D Delaunay refinement mesh generation, in: *Proceedings of the 11th International Meshing Roundtable*, IMR 2002, September 15-18, Ithaca, New York, USA, 75-86, 2002.

Mitas, L., and Mitasova, H., Distributed erosion modeling for effective erosion prevention, Water Resources Research, 34(3), 505-516, 1998.

Molenaar, M., A formal data structure for three-dimensional vector maps, in: *Proceedings of the Fourth International Symposium on Spatial Data Handling*, 1990.

Mostafavi, M.A., Development of a Global Dynamic Data Structure, PhD Thesis, Laval University, 2002.

Mostafavi, M.A., Gold, C.M. and Dakowicz, M., Delete and insert operations in Voronoi/Delaunay methods and applications, *Journal of Computers & Geosciences*, 29(4), 523-530, 2003.

Mostafavi, M.A. and Gold, C.M., A global kinetic spatial data structure for a marine simulation, *International Journal of Geographical Information Science*, 18, 211-228, 2004.

Mostafavi, M.A., Hashemi Beni, L., and Gavrilova, M., 3D Dynamic Scene Surveillance and Management Using a 3D Kinetic Spatial Data Structure, in:

Suzana Dragicevic Dumitru Roman, Vlad Tanasescu (eds) *The International Conference on Advanced Geographic Information Systems & Web Services* (GEOWS 2009), IEEE computer society, 45-53, 2009.

Mucke, E.P., A robust implementation for three-dimensional Delaunay triangulations, *International Journal of Computational Geometry and Applications*, 8(2), 255-276, 1998.

Neteier, M., and Mitasova, H., *Open Source GIS: a GRASS GIS Approach*, The Kluwer international series in Engineering and Computer Science, 2002.

Neuman, Adaptive Eulerian-Lagrangian finite element method for advection-dispersion, *International Journal for Numerical Methods in Engineering*, 20, 321-337, 1984.

Nyerges, T.L., Understanding the scope of GIS: Its relationship to environmental modeling, In: Goodchild, M.F, Parks, B.O., and Steyaert, T., (eds.), *Environmental Modeling with GIS*, New York: Oxford University Press, 75-93, 1993.

Obenour, D., Developing an Arc Hydro Dataset, 2002. http://civilu.ce.utexas.edu/stu/obenoudr/gisreport.htm

Okabe, A., Boots, B., and Sugihara, K., *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, Chichester, West Sussex, England, 1992.

Okabe, A., Boots, B., Sugihara, K., and Chiu, S.N., Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, John Wiley and Sons, second edition, 2000.

OpenDX 2004, http://www.opendx.org/ (access date: March 2009).

Parthasarathy, V.N., Graichen, C.M., and Hathaway, A.F., A comparison of tetrahedron quality measures, *Finite Elements in Analysis and Design*, 15, 255-261, 1993.

Pav, S.E., Walkington, N., Robust Three Dimensional Delaunay Refinement, in: *Proceeding of IMR 2004*, 145-156, 2004.

Pfund, M., Topologic data structure for a 3D GIS, in: *Proceedings of ISPRS*, 34, 233-237, 2001.

Pilouk, M., Integrated modelling for 3D GIS, PhD thesis, ITC, The Netherlands, 1996.

Pons, J.P., and Boissonnat, J.D., A Lagrangian Approach to Dynamic Interfaces through Kinetic Triangulation of the Ambient Space, *Computer Graphics Forum*, 26(2), 227-239, 2007.

Pouliot, J., Lachance, B, Kirkwood, D., L'importance de la modélisation géométrique 3D lors de l'élaboration d'un SIG 3D: Exemple du développement d'une structure topologique pour une application géologique, La revue internationale de géomatique: Information géographique tridimensionnelle, 16(1), 29-49, 2006.

Preparata, F., and Shamos, M., *Computational Geometry*, Monographs in Computer Science, Springer-Verlag, 1985.

Price, J.F., Lagrangian and Eulerian representations of fluid flow: Part I, kinematics and the equations of Motion, 2005.

Peuquet, D.J., A conceptual framework and comparison of spatial data models, *Cartographica*, 21(4), 66–113, 1984.

Peuquet, D.J., Representations of geographic space: Toward a conceptual synthesis, *Annals of the Association of American Geographers*, 78, 375-394, 1988.

Peuquet, D.J., Time in GIS and Geographical Databases, In: Maguire, D.J., Goodchild, M.F., Rhind, D.W., and Longley, P. (eds.) *Geographical Information Systems, Principles and Applications*, Second edition, 1, 1999.

Peuquet, D.J., Making space for time: Issues in space-time data representation. *GeoInformatica*, 5(1), 11–32, 2001.

Peuquet, D.J., and Duan, N., An Event-Based Spatiotemporal Data Model (ESTDM) for temporal analysis of geographical data, *International Journal of Geographical Information Systems,* 9 (1), 7–24, 1995.

Pouliot, J., Bédard, K.,Kirkwood, D., Lachance, B., Reasoning about geological space: Coupling 3D Geomodels and topological queries as an aid to spatial data selection, *Computers and Geosciences*, 34(5), 529-541, 2008.

Raper, J., *Three dimensional applications in Geographic Information Systems,* London: Taylor & Francis, 1989.

Rapar, J., *Multidimensional Geographical Information Science*, Taylor & Francis, London, 2000.

Raper, J.F., and Kelk, B., Three-dimensionality GIS, Geograpicl Information Systems, Maguire, D., Goodchild, M., Rhind, D., eds., *Geographic Information Systems*: *Principles and Applications*, Harlow: Longman, 299-317, 1991.

Raper, J., and Livingstone, D., Development of a geomorphological spatial model using object-oriented design, *International Journal of Geographical Information Science*, 9(4), 359–383, 1995.

Rivington, M., Matthews, K.B., Bellocchi, G., Buchan, K., Stockle, C.O., and Donatelli, M., An integrated modelling approach to conduct multifactorial analyses on the impacts of climate change on whole-farm systems, *Environmental Modelling & Software,* 22(2), 202-210, 2007.

Roos, T., New upper bounds on Voronoi diagrams of moving points, *Nordic Journal of Computing*, 4, 167–171, 1997.

Ruppert, J., A New and Simple Algorithm for Quality 2-Dimensional Mesh Generation, in: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms, 83-92, 1993.

Ruppert, J., A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation, *Algorithms,* 18(3), 548–585, 1995.

Russel, D., Kinetic data structure in practice, PhD thesis, Stanford University, 2007.

Russel, D., Karavelasb, M.I., and Guibas, L.J., A package for exact kinetic data structures and sweepline algorithms, *Computational Geometry*, 38, 111-127, 2007.

Samet, H., The quadtree and related hierarchical data structures, in: *Proceeding of ACM Computing Surveys*, 16(2), 187–260, 1984.

Schaap, W.E., and de Weygaert, R.V., Continuous fields and discrete samples: reconstruction through Delaunay tessellations, *Astrophys*., 363, L29–L32, 2000.

Schaller, G. and Meyer-Hermann, M., Kinetic and dynamic Delaunay tetrahedralizations in three dimensions, *Computer Physics Communications*, 162, 9-23, 2004.

Scott, M.S., Extending map algebra concepts for volumetric geographic analysis, In: Proceeding of GIS/LIS 1997 Annual Conference & Exposition,1997.

Shewchuk, R., Delaunay refinement mesh generation, PhD Thesis, Carnegie Mellon University, 1997.

Shewchuk, R., What Is a Good Linear Element? Interpolation, Conditioning, and Quality Measures, in: *Proceeding of the eleventh International Meshing Roundtable* (Ithaca, New York), 115-126, 2002.

Shewchuk, R., Star splaying: an algorithm for repairing Delaunay triangulations and convex hulls, in: *21th Annual ACM Symposium on Computational Geometry*, ACM Press, New York, 237–246, 2005.

Si, H, On Refinement of Constrained Delaunay Tetrahedralizations, in: *Proceedings of the 15th International Meshing Roundtable*, 509-528, 2006.

Sibson R., A brief description of natural neighbour interpolation, in: Barnett V, editor, *Interpreting Multivariate Data*, 21–36, 1981.

Soni, B.K., Grid generation: Past, present, and future, *Applied Numerical Mathematics*, Elsevier, 32, 361-369, 2000.

Stallings, C., Huffman, R.L., Khorram, S., and Guo, Z., Linking GLEAMS and GIS, paper no. 92-3613, 1992.

Stoter, J.E., 3D Cadastre, PhD thesis, NCG, The Netherlands, 2004.

Sugihara, K., Sliver-free perturbation for the Delaunay tetrahedrization, *Computer-Aided Design*, 39(2), 87-94, 2007.

Sui, D.Z. and Maggio, R.C., Integrating GIS with hydrological modeling: practices, problems, and prospects, *Journal of Computers, Environment and Urban Systems*, 23, 33–51, 1999.

Temperley, H., *Graph Theory and Applications*, Ellis Horwood Ltd., Chichester (UK), 1981.

Therrien, R., McLaren, R.G., Sudicky, E.A., and Panday, S.M., HydroSphere: A three-dimensional numerical model describing fully-integrated subsurface and surface Flow and solute transport, Groundwater Simulations Group, draft, 2006.

Tobler, W.R., A computer movie simulating urban growth in the Detroit Region, *Economic Geography*, 46, 234–240, 1970.

Tomlin, C.D., A map algebra, in: *Proceedings of the 1983 Harvard Computer Graphics Conference*, 127–150, 1983.

Tse, R., and Gold, C.M., TIN meets CAD—extending the TIN concept in GIS, *Future Generation Computer Systems*, 20(7), 1171–1184, 2004.

Valavanis, V.D., *Geographic information systems in oceanography and fisheries*, Taylor &Francis, 2002.

Van Deursen, W.P.A., Geographical Information Systems and Dynamic Models: development and application of a prototype spatial modelling language, PhD Thesis, Utrecht University, 1995.

Van der Knijff, J., and De Roo, A., LISFLOOD – distributed water balance and flood simulation model, Revised User Manual, 2008.

Voronoi, G., Nouvelles applications des paramètres continus à la théorie des formes quadratiques, *Journal für die Reine und Angewandte Mathematik*, 133, 97-178, 1907.

USACE, Geospatial Hydrologic Modelling Extension HEC-GeoHMS, User's Manual, 2003.

Verma, S., Aziz, K., A control volume scheme for flexible grids in reservoir simulation, in: *14th SPE Reservoir Simulation, Symposium*, 215- 227, 1997.

Watson, D.F., *Contouring: A Guide to the Analysis and Display of Spatial Data*. Pergamon Press, Oxford, UK, 1992.

Watson D.F., and Phillip G.M., Neighborhood-based interpolation, *Geobyte*, 2(2), 12-16, 1987.

Whitehurst, R., A Free-lagrangian method for gas dynamics, Monthly Notices of the Royal Astronomical Soc., 277, 655-680, 1995.

Wilson, R., *Introduction to Graph Theory* (4th ed.), Longman, Harlow (UK), 1996.

Wei, G., Ping, Z., and Jun, C., Topological data modelling for 3D GIS, in: Fritsch D, Englich M, and Sester M (eds.), *Proceedings of the ISPRS Commission IV Symposium on GIS: Between Visions and Applications*, 657–61, 1998.

Worboys, M.F., A model for spatio-temporal information, in: *Proceedings 5th International Symposium on Spatial Data Handling*, 2, 602–611, 1992.

Worboys, M. F., Event-oriented approaches to geographic phenomena, *International Journal of Geographic Information Science*, 19(1):1-28, 2005.

Worboys, M.F. and Duckham, M., *GIS: A Computing Perspective*, CRC Press, second edition, 2004.

Worboys, M. F. and Hornsby, K., From objects to events, GEM, the geospatial event model, *Third International Conference on GIScience 2004*, M. Egenhofer, C. Freksa, H. Miller (eds.), Lecture Notes in Computer Science 3234, Berlin: SpringerVerlag, 327-344, 2004.

Worboys, M.F. and Duckham, M., Monitoring qualitative spatial change for geosensor networks, *International Journal of Geographic Information Science*, in press.

Yuan, M., Representing complex geographic phenomena with both object- and field-like properties, *Cartography and Geographic Information Science*, 28, 83-96, 2001.

Yuan, M., Adding Time, *Handbook of Geographic Information Science*, J. Wilson and S. Fotheringham (eds.), published by Blackwell Publishing, 169-184, 2007.

Zeitouni, K., and de Cambray, B., Topological modelling for 3D GIS, in: *Proceeding of Fourth International Conference on Computers in Urban Planning and Urban Management*, 479-494, 1995.

Zhang, Z., Wang, S., Sun, G., McNulty, S., Zhang, H., Li, J., Zhang, M., Klaghofer, E., and Strauss, P., Evaluation of the MIKE SHE model for application in the Loess Plateau, China, *Journal of The American Water Resources Association*, 44(5), 1108-1120, 2008.

Zlatanova, S., 3D GIS for Urban Development, PhD Thesis, ITC, 2000.

Zlatanova, S., Abdul Rahman, A., and Pilouk, M., Trends in 3D GIS development, *Journal of Geospatial Engineering*, 4 (2), 1-10, 2002.

Zlatanova, S., Abdul Rahman, A., and Shi, W., Topological models and frameworks for 3D spatial objects, *Computers & Geosciences*, 30, 419–428, 2004.