DYNAMIC FIELD PROCESS SIMULATION WITHIN GIS: THE VORONOI APPROCHE

L. Hashemi Beni*, M. A. Mostafavi, J. Pouliot

Dept. of Geomatics, Laval University, Quebec, Canada
Leila.hashemi.1@ulaval.ca, (Mir-Abolfazl.Mostafavi, Jacynthe.Pouliot) @scg.ulaval.ca

Commission II, ICWG II/IV

KEY WORDS: Data Structure, Voronoi Diagram, 3D GIS, Simulation, Dynamic

ABSTRACT:

Simulation of a dynamic and continuous phenomenon (field) is a difficult task for GISs as their data structures are 2D and static and are not well-adapted to manage neither the dynamic behavior of the phenomenon nor its geometrical and topological information. In this paper we present the Voronoi diagram as an alternative data structure that, through its useful geometrical and topological properties, provides an adequate discretization of a field and can represent its temporal changes by providing numerical integration methods on either dynamic or kinetic mesh.

1. INTRODUCTION

Simulation of dynamic fields is usually needed for a wide variety of applications to better understand, analyze, and predict their behaviors. Simulations of fluid flows, meteorological and environmental application are examples of these applications. For example, questions such as: "where does ground water come from?", "how does it travel through a complex geological system?" and "how is water pollution behavior in an aquifer?" can be partially answered using simulation tools such as HYDROGEOSPHERE (Therrien, 2006) and MODFLOW. However, many spatial analysis capabilities are absent within such tools which limit their utilities in modelling and representation of those phenomena.

When dealing with simulation of a dynamic phenomenon, several factors must be taken into account such as the dynamic behavior of the phenomenon and its geometrical and topological representation (spatial modeling). This is where a Geographic Information System (GIS) is a valuable tool. Geographic Information System (GIS), through its powerful capabilities to process spatial data, provides modelers with strong computing platforms for data management, integration, visualization, querying, and analysis and thus greatly facilitate the implementation of various process models. However, the simulation of the real-world phenomena which are usually 3D and dynamic (they change with respect to space and time) is difficult within the current GISs as their data structures are 2D and static. Indeed, the current GIS data structures are not well-adapted to represent and manage neither geometry nor topology of 3D spatial data and are incapable of properly handing the dynamic behavior of a phenomenon (Ellul and Haklay, 2006; Mostafavi, 2002). This paper analyses the potentials and limitations of Voronoi diagram as an adaptive spatial data structure for simulating a dynamic and continuous phenomenon in a 2D and 3D space. In the remainder of this paper, we briefly explain how a dynamic and continuous phenomenon (field) can be represented followed by a review of the limitations of the current data structures for this purpose. Then, an alternative data structure, Voronoi diagram, is introduced. Next, we study the potential of the Voronoi data structure for different simulation methods in 2D and 3D. Finally, we present some results of the application of the kinetic Voronoi diagram for two case studies and discuss the results.

2. FIELD SIMULATION

Space can be represented either as a set of objects with spatial properties or as a set of locations with properties which is referred to as a field (Worboys and Duckham, 2004). Topographic elevation, air temperature and water pollution are examples of spatial fields. Mark (1999) defines a 2D field (F) as any single-valued function of location in 2D space: $F = f(x, y)$. This definition can be generalized to 3D as: $F = f(x, y, z)$. This value also becomes a function of time when dealing with a dynamic filed. Depending on how time is taken into account in the function, two different approaches can be introduced: We can investigate the changes of a field at a fixed location over time: $F(t) = f(x, y, z, t)$ or we can describe the changes of the field at a time-dependent location: $F(t) = f(x(t), y(t), z(t))$. These visions correspond to static and dynamic views of a process presented by Peuquet (1999) and Eulerian and Lagrangian view points in computational methods (Price, 2005) which are discussed later in this paper.

Dynamic fields have generally a very strong spatial dimension and researchers in geospatial domain have been increasingly interested in modeling and representing them within a GIS (Maggio, 1999). Traditionally, simulation tools have generally been developed outside of GIS. MODFLOW and HYDROSPHERE are two examples of simulation tools. However, these tools suffer from insufficient spatial analytical component, low performance of the spatial data visualization capability, and non-conveniences in spatial data integration such as digital maps, satellite image, aerial photos (Bivand and Lucas 1997; Densham 1993; Nyerges 1993). Therefore, the integration of simulation tools and GIS was considered by many researchers and the mutual benefits of this integration were confirmed by several papers published on this area (Chapman and Thornes 2003, Sui and Maggio 1999, Valavanis

2002). Depending on the level of interaction between a simulation tool and GIS, three integration approaches can be developed. A Loose coupling approach is defined as merely passing input and output between a GIS and a simulation tool. In this level of integration, the input into a simulation tool is from GIS and the output of the simulation tool is exported back to GIS for analysis and visualization. A Tight coupling approach involves processing data in same database for a GIS and simulation tool.

In this approach, although the process model is developed outside of GIS, the model is configured with the interactive tools of the GIS and the data is exchanged automatically. A Full coupling approach is defined as embedding a process model in a GIS. This approach involves implementing a process model within the GIS and taking full advantage of the built-in GIS functionalities. In addition, this allows a GIS to go beyond being a simple data management tool to offer more sophisticated analyses and simulations of natural phenomena (Maggio, 1999). Full coupling approaches that are referred to as GIS-based simulation approaches require a data structure to manage multidimensional and dynamic phenomena. Current GIS miss such data structure and then can not support such applications.

Dynamic fields are continuous and it is practically impossible to measure them anywhere and anytime. Hence, our knowledge of spatio-temporal fields is usually limited to a set of observations in given locations and time. Therefore, the continuous field must be represented using these data samples which are usually a set of unconnected points. Each point is defined by its position in 2D or 3D space and its field value at a given time. To represent a continuous phenomenon from a set of discrete samples, it is necessary to create a tessellation which rebuilds the continuity and connectivity between the discrete observations. Tessellation, referred to as meshing or grid generation, is a partition of the space by a set of elements such that the union of all elements completely fills the space (Worboys and Duckham, 2004). Meshes can be of two types, regular and irregular. The elements of a regular mesh have uniform shapes and sizes such as 2D rectangles and pixels or 3D cubes and voxels. In a regular spatial tessellation, the topology between these elements exists implicitly, which means that all neighbors of a given element can be accessed by a simple index, either (i, j) in 2D or (i, j, k) in 3D. However, a major limitation of regular meshes is related to the handling of field data (unconnected points) with an irregular distribution. Indeed, in this case, a large number of elements required for a fine resolution, which, especially in 3D, can be huge. To overcome the limitations of regular meshes, hierarchical meshes can be used, where a tree is created. Quadtree and Octree are two examples of this data structure in 2D and 3D spaces respectively. In fact, these methods subdivide the space into four squares (Quadtree, in 2D) or four cubes (Octree, in 3D) of equal size until either each element contains one homogeneous region or reaches a desired resolution. Although these methods reduce the required memory for a fine resolution mesh, the tree data structure can be unbalanced when the data distribution is irregular (de Berg *et al*., 2000). In addition, a small change in field data may result in a quiet different tree. Irregular meshes can also be used to model a continuous phenomenon. The elements of irregular meshes can be of 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. Thus, they can be adapted to the point distribution and provide conformity to complex phenomena. However, in irregular

meshes, the connectivity between the mesh elements or topology does not implicitly exist. Hence, topology needs to be computed and stored explicitly and updated after any change which can be a difficult task for current GISs data structures. Since, the data structures are static and following any change in the spatial information, all spatial relationships (topology) must be rebuilt. This problem becomes more complicated for 3D mesh. In addition to this problem, the GISs data structures are not able to represent both objects and fields at same time which is required when dealing with a continuous phenomenon observed with discrete points.

Gold and Condal (1995) suggested that a Voronoi and Delaunay spatial data structure can be a good candidate for the simulation of dynamic fields within GIS as they can properly model multidimensional spatial data over time and space. Some attempts have been made to apply these data structures for simulating a continuous phenomenon such as a fluid flow in two- and three-dimensional space (Mostafavi and Gold, 2004, Hashemi and Mostafavi, 2008; Blessent et al. 2008) which are discussed in the following sections and the advantages and limitations of the approach are discussed.

3. VORONOI DIAGRAM AS UNDERLYING MESH FOR A PROCESS SIMULATION

A Voronoi diagram (VD) for a set of points S in R^n is constructed by partitioning the space into regions with one region for each point, so that the positions in the region for point $p \in S$ are closer to p than any other point in S (fig.1a); so a Voronoi diagram in R^3 is (Edelsbrunner, 2001):

$$V_p = \left\{ x \in R^3 \mid \|x - p\| \leq \|x - q\|, \forall q \in S \right\}$$

And DT (S) in 3D is Delaunay Triangulation of S such that there are no points of S inside the circumsphere of any tetrahedron (fig.1b). This resulting mesh is unique for the point set S, except when five or more points are co-spherical in 3D (Edelsbrunner, 2001).
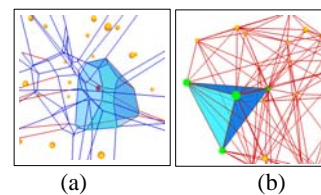

| (a) | (b) |

Figure 1. a) 3D Voronoi diagram, b) 3D Delaunay triangulation

To simulate a dynamic field such as a fluid flow, the first step consists on the discretization of the continuous phenomenon to mesh elements. The dynamic behavior of the flow, which is typically defined by partial differential equations (PDE), is approximated by the dynamic behavior of the mesh elements at a set of time snap-shots. This can be done using two different Eulerian and Lagrangian approaches (Price, 2005; Mostafavi, 2002).
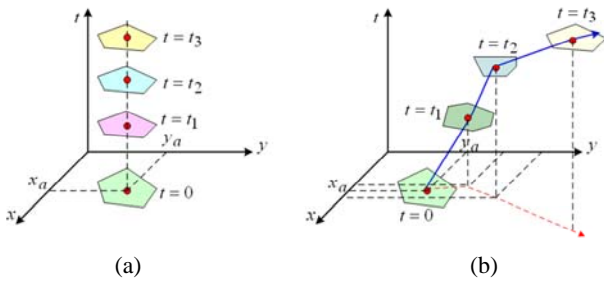
Figure 2. Two different simulation methods

a) Eulerian methods describe changes of field at a fixed location at a series of snap-shots; b) Lagrangian methods describe changes which 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.

In Eulerian methods, the equations are solved using a static mesh (fig. 2a). In Lagrangian methods, the mesh moves with the fluid flow hence, the positions of the mesh elements change constantly over time (fig. 2b). Voronoi diagram (VD) and its dual, Delaunay triangulation (DT), thorough their useful properties provide an adequate discretization of the space for both Eulerian and Lagrangian fluid flow simulation approaches, ensuring that physically realistic results are obtained form the numerical integration of the PDE.

4. EULERIAN METHODS AND VORONOI DIAGRAM

Eulerian methods take advantage of a fixed mesh that is simple to generate and maintain. These methods have been widely used for fluid flow simulations. In these methods, the size and shape of mesh elements as well as the value of time-interval have to be carefully selected to ensure a realistic representation of a fluid flow. A uniformly fine mesh is computationally costly and is generally not well suited to handle the dynamics of a fluid flow and tracking problems in a complex system (such as a hydrogeological system). A mesh based on a dynamic Voronoi diagram is an interesting alternative for these requirements. Voronoi cells can be defined by points with an arbitrary distribution, creating mesh elements of different sizes and shapes which can adapt to complex geometries. For instant, for regions with either high rates of flow or discontinuities, the Voronoi diagram can provide a fine resolution mesh. Each cell can have an arbitrary number of neighbors which their connectivity with the given cell is clearly defined and can be explicitly retrieved if needed. In addition, dynamic Voronoi diagram offers the local editing and manipulating possibility of the mesh which is usually necessary for the refining of the mesh without having to rebuild the whole mesh. Regarding these properties, several research works used VD and DT as underlying mesh in fluid flow simulation. Hale (2002) applied DT and VD to reservoir simulations using 3D seismic images and demonstrated the potential of both DT and VD for flow simulation during all steps of seismic interpretation, fault framework building, and reservoir modeling. Lardin (1999) and Blessent et al. (2008) applied this data structure to groundwater simulation in 3D space and showed that VDs are well-adapted to the Control Volume Finite Element (CVFE) method. The CVFE methods are based on the principle of mass conservation. Thus, a volume of influence is assigned to each point or element and equations are defined to describe the interaction of the element with its neighbors. This interaction is expressed by

mass balance, which states that the difference between inflow and outflow in each element must be equal to the variation in fluid stored in the same volume (Therrien et al. 2006). Fig.3 shows the examples of Voronoi elements in 2D and 3D.
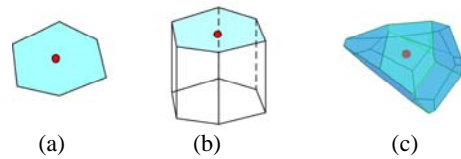

Figure 3. Examples of Voronoi elements in 2D(a) and 3D (b),(c).

5. LAGRANGIAN METHODS AND VORONOI DIYAGRAM

A Lagrangian method are often the most efficient way to simulate a fluid flow, as the mesh moves and conforms to the complexity of geometries (Price, 2005). However, when the points move with fluid velocities, the connectivity between mesh elements remains unchanged. This can result in a tangled (deformed) mesh. For this reason, these methods are very limited for highly turbulent fluid flow simulations. An alternative approach consists of using the Arbitrary Lagrangian-Eulerian methods where an Eulerian approach can be used to solve the equation at a given time and the mesh can then be moved to solve for the next time value. These methods reduce the mesh distortion by continuous 'remapping' or 'reconnecting' of the mesh. A mesh remapping approach can be considered as an Eulerian process, because fluid is transported across mesh cell boundary. Free-Lagrangian methods use this reconnecting concept with this difference that the fluid is not transported across mesh cell boundary. This reconnection process allows the representation of very complex fluid flows. However, a main problem of these methods is related to determining the optimal time interval. 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. Another problem with Free-Lagrangian methods lies on maintaining and processing of the connectivity relations between mesh elements at each time. To solve these problems, a kinetic data structure can be helpful which is based on the fact that "variation in space with time may be modeled not by snap-shots of the whole map at regular time intervals, but by local updates of spatial model at the time when they happen (event)" (Gold 1993). In a fluid flow simulation, these events can be the changes either on the field value or on the spatial relationship of the points which refer 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 topological events can properly be detected and updated by a kinetic Voronoi and Delaunay data structures as explained in follows.

Point movement may change the adjacency relationships of the point and its neighbors (fig4). Then, this displacement changes the configuration of the triangle/tetrahedra having the moving point as one of their vertexes. In a DT, a topological event occurs when a point (p) moves in or out of the

circumcircle/circumsphere of a triangle/tetrahedron in 2D/3D mesh. Therefore, to find the topological event of a moving point, only the spatial information of the triangles/tetrahedra having the moving point as one of their vertexes and their neighbors are used and the remaining triangles/tetrahedra in the mesh do not need to be tested. This can be computed using well-known predicted test (Guibas and Stolfi, 1985) to preserve the Delaunay empty circumcircle/circumsphere criterion. Since in a kinetic data structure, the position of points are time dependent, then, the value of the determinant will be time dependent as well. However, the cost of generating, computing and updating the predicate function is very expensive, especially when dealing with simultaneous moving of the points on complex trajectories as seen in a physical system. For example, a quadratic trajectory of a point in a 3D space results in a degree eight predicate function. As described in Guibas and Russel (2004), the computational cost can be reduced by minimizing the degree of the predicate function. To minimize the degree of the function, we assume that only one point is allowed to move at a time on a linear trajectory. Therefore, one row of the predicate determinant must be allowed to vary linearly. Equation 1 shows the predicted function for a moving point in 3D Delaunay triangulation. According to this equation, a topological event for point p occurs when p moves in or moves out of the circumsphere of the tetrahedron $abcd$, i.e. the value of the predicate function is 0.

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

Mostafavi and Gold (2003) have implemented a similar algorithm on the plane that minimizes the number of triangles which must be tested to detect the closest topological event of a moving point. To do so, the algorithm computes the intersection between the trajectory of p (a line segment) and the neighboring circumcircles that cut the trajectory between the origin and the destination of the moving point. In fact, the triangles that the orthogonal projection of their circumcenter on the trajectory of p are behind the point p, with respect to the moving direction, are not considered. Ledoux (2006) has extended this algorithm to deal with the moving of a single point in a 3D Delaunay triangulation. Although the algorithm is functional for the general cases, it needs to be improved for the degeneracies and complicated cases as well as the complexities of several moving points in 3D. For this purpose, we need to develop a strategy that should not only maintain the validity of the mesh but also give realistic results for the simulation process. Here, our strategy is based on the management of the topological events of the moving points based on a defined priority criterion. This priority is defined based on the value of the simulation time ($t_{simulation}$) for each moving point. The simulation time is the total time that takes for each point to reach from its origin to its new location on the trajectory. Therefore, first, we compute all the topological events of the moving points in the mesh. Next, the time taken for each point to reach its closest topological event t_{event} is computed. This time depends on the velocity (v) of the moving point and the distance (d) between its current position and the location of its

next closest topological event on its trajectory. We define the local time (t_{local}) as the time that it takes for each point to move from its origin to its current position. The relation between these times is:

$$t_{simulation} = t_{local} + t_{event}$$

To facilitate the management of the topological events, we have used a priority queues data structure by organizing the moving points based on the increasing value of $t_{simulation}$. Therefore, the first member of the queue which has the smallest simulation time is processed first i.e. the moving point is moved to its new location and a local update in the mesh is carried out in the mesh for the moving point and its neighbors.

(a)   (b)

Figure 4. Flips are dynamic operations and make appropriate local updates in mesh data structure, a) flip22 in 2D, b) flip23 and flip32 in 3D.

In 2D, the *flip22* is used that converts two neighboring triangles to two different neighboring triangles by changing the diagonal of the quadrilateral formed by them (fig. 4a). In 3D, several types of flips including the *flip23* or *flip32* are used (fig.4b). The *flip23* or the face-to-edge flip operator converts two neighbor tetrahedra to three tetrahedra. The *flip32* or the edge-to-face flip operator converts three neighbor tetrahedra to two in order to guarantee the Delaunay empty circumsphere criterion (Joe, 1991; Shewchuk, 2005).

In a fluid flow simulation context, following any topological changes in the mesh, we need to update the physical parameters of the affected points. The governing equations that define the nature of the dynamic field, allow to compute the new physical parameters such as the velocity for each moving point and its neighbors. This means that $t_{simulation}$ is updated after each topological event for the points involved in this operation. As a result, the priorities of some of the moving points may change. This occurs because, when a point moves, the related circumcircle/circumspheres and event times of the neighboring points change. The above process is reiterated until the end of the simulation process. Fig. 5 is an example of evolution of adjacency relationships for a single moving point in a 2D kinetic VD.
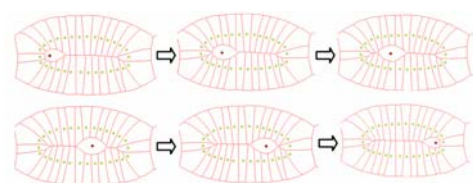

Figure 5. Evolution of adjacency relationships for a single moving point in a 2D kinetic VD (Mostafavi 2002)

5.1 Application examples in 2D and 3D

In this section, through two application examples, we show the potential of a kinetic Voronoi data structure for free-Lagrangian fluid flow simulation in two-and three-dimensional spaces. The first example which is described in details in Mostafavi and Gold (2004), presents the application of the 2D kinetic Voronoi data structure for a dame-braking problem. This example uses the discrete form of governing equations that describe the behavior of an incompressible fluid flow. As the first step, the fluid region is sampled by a discrete set of points and a Voronoi-based mesh is used to tessellate the space and define the elements and their topological relations. Next, to each element of the mesh, an initial velocity and height are assigned. According the law of the conservation of mass, mass and volume of each element must be constant during the simulation process. Hence, the height of each particle is computed by: $H_i = \dfrac{m_i}{A_i}$; where m_i and A_i are the mass and area of the ith particle. In this example, we assume that the middle of the surface is five time higher than other parts (fig. 6a). To start the simulation, first topological event is computed for each element and stored in a priority queue. The first point of the queue is processed by moving it to its new position and making the appropriate topological updates (flip22). Next, using the governing equation, the velocity of the moving point as well as the area and height of the moving point and its neighbors are updated. The motion of each element is the result of its interactions with its neighboring elements. Following these changes, the first topological event of the moving point and its neighbors are recomputed and finally, the priority queue is updated using these new values. The results of this example (fig.6) show the higher elements in the middle of the surface begin to move outward so their area becomes larger. Using the assumption that the mass of each element is constant during the simulation process, the height of these elements becomes lower and lower until the equilibrium is created in the simulation region. This example reveals that, although the fluid flow in this particular case has a significant motion, there has been no mesh distortion or overshoots and undershoots that occurred during the simulation process.
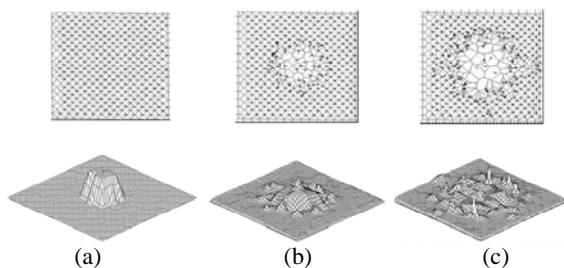

Figure 6. The result of dam breaking problem using a 2D kinetic VD (upper images) and the mesh perspective (below images). a) Initial mesh, b) mesh after 5000 topological events, c) mesh after 10000 topological events (Mostafavi and Gold, 2004)

The second example consists on a 3D hydrodynamics simulation example which uses a 3D kinetic Voronoi data structure. In this example, we investigate the potential of the 3D KVD for a gas dynamics simulation. The governing equations mainly describe the conservation of the mass, momentum and total energy for a compressible fluid flow (Campbell and Shashkov, 2003). For the sack of simplicity, we consider that the fluid domain is bounded by a 3D rectangular boundaries and the fluid can move horizontally. For numerical integration of the governing equations, first, the simulation domain is discretized using 3D Voronoi diagram where each Voronoi polyhedron represents a flow cell. Therefore, the relationship between mesh cells is explicitly stored and automatically maintained during the simulation process. To start the simulation, the initial velocity (v_x, v_y, v_z), pressure P , density ρ , internal energy e , and mass m values are assigned to each cell. 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 cell remains constant during the simulation process (fig.7a). Each polyhedral cell shares a face with each of its neighboring cells (fluid or boundary cell). The surface of those faces as well as the volume and the shape of the Voronoi polyhedron may change during the simulation process. Similar to 2D algorithm, in 3D space, for each cell, the first topological event for each moving point is computed and stored in a priority queue. Then, the first point on the priority queue is moved to its new position using a velocity vector. The velocity vector is the result of the total force acting on the cell by its neighbors. The neighboring cells forces are computed on the central polyhedron faces. Following the movement, the shape and the physical attributes (quantities) of the cell and its neighbors are updated. The shape and topological information are updated following a *flip23* and *flip32* and the physical attributes are computed using the governing equations. Consequently, for each polyhedron, the number of the faces and their respective area change in the new position. Following this step, next closest topological events for the points and its neighbors are computed and the priority queue is updated for these new values. This procedure is repeated until the equilibrium is created between the pressures of particles.

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 (fig.7). According to the assumption that the mass of the cells is constant during the simulation process, when the volume of a cell increases, the density as well as the pressure of the cell 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 expected i.e. The gas flow completely fills the simulation domain and equilibrium is achieved. The initial results obtained in this case study are very promising as the data structure is maintained during the simulation process representing correctly the geometrical and topological information of the underlying mesh and the results of the simulation are reasonably comparable to the one obtained from the analytical solution of the governing equations.
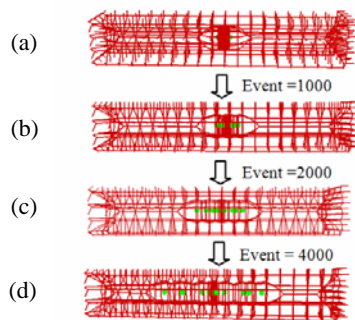
Figure 7. Simulation of gas dynamics using 3D kinetic VD a) initial mesh, b) mesh after 1000 topological events, c) mesh after 2000 topological events and d) mesh after 4000 topological events

6. CONCLUSIONS

In this paper we discussed simulation of a dynamic and continuous phenomenon (filed), a fluid flow in particular, that is a difficult task for GIS as its data structures are 2D and static. A 3D Voronoi data structure, as an alternative, can generate a mesh that accurately represents the geometrical, topological information of a fluid flow as well as its dynamic behavior in both static and dynamic manner. In the static or Eulerian methods, the structure assigns a volume of influence to each point and flow is assumed to be a transfer of fluid between these elements. Therefore, the change of fluid flow for each element is difference between inflow and outflow in it at a series of snapshots. In the dynamic or Lagrangian methods, data structure assigns a fixed mass of the fluid to each point. Therefore, mesh moves as fluid flow progress. The kinetic Voronoi diagram is also very well-adapted to free-Lagrangian mesh as it can properly update the topology, connectivity, and physical parameters of the mesh elements when they change.

This paper is a part of an ongoing research work that proposes a kinetic data structure for the simulation of 3D dynamic fields within GIS. In the research work, different issues regarding the development, implementation and application of such a data structure for the 3D simulation of fluid flow in hydrodynamics using Voronoi diagram have been studied. Although the preliminary results are very promising, there are several complexities related either to the data structure itself or to the nature of physical problem. Special attention is needed to address the problems of co-planarity of points and collision between moving points. These problems are more frequently present when the algorithm is implemented by floating-point arithmetic. Mesh resolution, boundary conditions and initial values are other important issues that have significant impact on the simulation results. All of these challenging problems are currently under investigation in the Geomatics research centre at Laval University.

REFERENCE

Blessent, D., Hashemi, L., Therrien, R., 2008. 3D modeling for hyrogeological simulations in fractured geological media. In: *Proceeding of Int. Conference on Modelling and Simulation*.

Bivand, R., and Lucas, A., 1997. Integrating models and geographical information systems. In: *Openshaw S,Abrahart R J Geocomputation*. Taylor & Francis, London, pp. 331-364.

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

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

De Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O., 2000. Computational Geometry: Algorithms and Applications, Second Edition, Springer-Verlag, Berlin.

Densham, P.J., 1993. Integrating GIS and spatial modelling: The role of visual interactive modelling in location selection. *Int. J Geographical Systems*, 1, pp. 203-220.

Edelsbrunner, H., 2001. Geometry and topology for mesh generation. Cambridge University Press, Cambridge, 190 pp.

Ellul, C., Haklay, M., 2006. Requirements for topology in 3D GIS. *Transactions in GIS*, Volume 10, Number 2, pp. 157-175.

Gavrilova, ML., and Rokne, J., 2003. Collision detection optimization in a multi-particle system. *Int. J Computational Geometry and Applications*, 13(4), pp. 279-302.

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

Gold, C.M., 1993. An outline of an event-driven spatial data structure for managing time varying maps. In: *Proceeding: Canadian conference on GIS*, pp.880-888.

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

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

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

Hashemi, L., and Mostafavi, M.A., 2008. A kinetic spatial data structure in support of a 3D Free Lagrangian hydrodynamics algorithm. In: *Proceeding of Int. Conference on Applied Modeling and Simulation*.

Joe, B., 1991. Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8(2), 123-142.

Lardin, P., 1999. Le diagramme Voronoi généralisé comme support a la simulation des écoulements d'eau souterraine par différence finis intégrées. MSc Thesis, Laval University.

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

Mark, D., 1999. Spatial Representation: A Cognitive View. In: Maguire, D. J., Goodchild, M. F., Rhind, D. W., and Longley, P. (editors) Geographical Information Systems: Principles and Applications, Second edition, v. 1, pp. 81-89.

Mostafavi, M.A., 2002. Development of a global dynamic data structure, PhD thesis, university of Laval, Canada.

Mostafavi, M.A., and Gold, C. M., 2004. A Global Spatial Data Structure for Marine Simulation. *Intl J Geographical Information Science*, 18, pp. 211-227.

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

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

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

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

Sui, D.Z., Maggio, R.C., 1999. Integrating GIS with hydrological modeling: practices, problems, and prospects. *J Computers, Environment and Urban Systems*, 23, pp.33–51.

Shewchuk, R. 2005. Star splaying: an algorithm for repairing Delaunay triangulation and convexhulls. In: *Proceeding of the Twenty-First Annual ACM symposium on Computational Geometry*, ACM Press New York, pp.237-246.

Therrien, R., McLaren, R.G., Sudicky, E.A., Panday, S.M., 2006. HydroSphere: A three-dimensional numerical model describing fully-integrated subsurface and surface Flow and solute Transport, User's manual.

Valavanis, V.D., 2002. Geographic information systems in oceanography and fisheries. Taylor &Francis.

Worboys, M. F., Duckham, M., 2004. GIS: A computing perspective (CRC Press, second edition).