Database Support for Unstructured Meshes

Alireza Rezaei Mahdiraji
supervised by: Peter Baumann
Jacobs-University
Bremen, Germany
a.rezaeim@jacobs-university.de

ABSTRACT

Despite ubiquitous usage of unstructured mesh in many application domains (e.g., computer aided design, scientific simulation, climate modeling, etc.), there is no specialized mesh database which supports storing and querying such data structures. Existing mesh libraries use file-based APIs which do not support declarative querying and are difficult to maintain. A mesh database can benefit these domains in several ways such as: declarative query language, ease of maintenance, query optimization, etc. In this thesis work, the core idea is to have a very general model which can represent objects from different domains and specialize it to smaller object classes using combina- torial constraints. We propose the Incidence multi-Graph Complex (ImG-Complex) data model for storing combina- torial aspect of meshes in a database. We extend incidence graph (IG) representation with multi-incidence information (ImG) to represent a class of objects which we call ImG- Complexes. ImG-Complex can support a wide range of ap- plication domains. We introduce optional and application- specific constraints to restrain the general ImG model to specific object classes or specific geometric representations. The constraints check validity of meshes based on the properties of the modeled object class. Finally, we show how graph databases can be utilized and reused to query some combinatorial mesh queries based on the (possibly constrained) ImG model. In particular, we show the strengths and limitations of a graph-only query language in expressing combinatorial mesh queries.

1. INTRODUCTION

Meshes appear in many application domains such as computer graphics, computer aided design, solid modeling, scientific simulation (e.g., finite element analysis), oceanography, climate modeling, Geographic Information Systems (GIS), etc. In these domains, meshes are used as (often approximate) representations of physical objects. By subdividing a domain into smaller and simpler geometric elements (typical examples are triangles, tetrahedrons, or other simple polyhedrons [14]), meshes provide the means to computationally manipulate complicated physical structures. In principle, more accuracy of representation and approximation can be achieved by using a finer subdivision, i.e., more cells. Informally, a mesh is composed of a set of (usually simple) cells, which are connected to each other according to some incidence relationships.

A mesh has three main components: 1) A combinatoric structure which describes how cells are connected to form the mesh, 2) A geometry which describes geometric embeddings of cells and the mesh, and 3) A set of fields which assigns data values to cells (e.g., temperature to vertices).

Although meshes are used in many application domains, there is no specialized mesh database which allows storing and querying such data structures. Researchers aimed at identifying requirements for mesh algorithms and definition of algebraic framework to manipulate meshes [9][5], but to the best of our knowledge there is no research in the database community which investigates support of unstructured meshes in databases. Utilizing relational databases for meshes is not efficient and it does not offer sufficient abstraction mechanisms to deal with meshes [10]. Existing mesh applications are usually written in languages such as C++ and often low-level file-based APIs are used for I/O. While it is possible to design data-structure neutral interfaces like in GrAL [6], the development of queries based on such APIs requires deep programming knowledge and is limited to a single language. More often, even the possible abstraction mechanisms (e.g., by a C++ API) are not used, and such implementations rely highly on input file structure and internal mesh representation, i.e., any changes there require changes in the implementation. Hence, they are less reusable and their maintenance cost is high.

A mesh database offers several advantages, namely: 1) users easily describe their information need using a declarative language (declarative access), 2) ease of maintenance, 3) hiding mesh storage structure from applications (physical data independence), and 4) transparent query optimization.

In this paper, we propose the ImG-Complex model (see Section 5) to store meshes in a database system. Our model has very few inherent limitations and thus its object domain encompasses the requirements of the wide range of mesh applications. We instrument ImG with sets of optional and application-specific constraints which can be used to check validity of meshes for a specific class of object such as manifold meshes, pseudo-manifold, simplicial manifold, etc. Currently, our model is only concerned with combinatorial aspect of meshes; we are working to add support for other aspects of meshes (i.e., geometric realization and data association) to the model in the future.

This paper is organized as follows: Section 2 presents two motivating application examples, Section 3 introduces some basic concepts, and Section 4 discusses related work. In Section 5 we present the ImG-Complex data model in detail, and discuss some mesh constraints in Section 6. Section 7 presents important combinatorial mesh queries. In Section 8 we investigate potentials and limitations of existing graph databases for representing combinatorial aspects of meshes based on the (constrained) ImG Model, and present some mesh queries built on top of graph queries. Section 9 concludes the paper.

2. MOTIVATING MESH APPLICATIONS

The need for a specialized mesh database is motivated by many real-world applications. Two motivating application domains are briefly described here. These domains have similar queries and operational requirements such as finding neighbors of a cell, subsetting (i.e., to find cells contained or overlapping with a query bounding box), mapping data from one mesh to another, etc.

Oceanography: Unstructured meshes are used in several branches of oceanography such as tidal and coastal modeling, global ocean model, etc. Advantages of unstructured meshes for ocean modeling are: 1) a more accurate and efficient representation of the domain, 2) variable resolution which is very important for focused regional modeling, and 3) continuous representation of coastlines. Recently, ocean or coastal models based on unstructured meshes are growing, but large-scale ocean circulation modeling based on unstructured mesh are computationally expensive and hence not common [16]. Figure 1 shows surface of North Atlantic modeled as unstructured mesh [16].

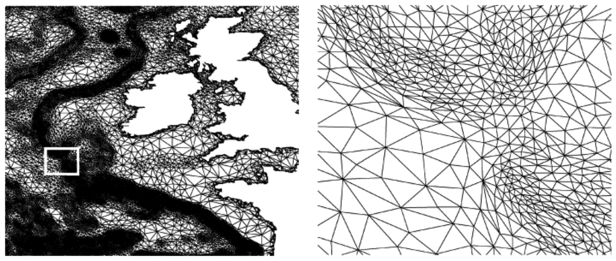

Figure 1: See floor surface of North Atlantic modeled as 2D simplicial mesh [16].

Seismology: In earthquake simulation, the earthquake propagation is computed for a given piece of earth and its initial conditions. The goal of the simulation is to enable modeling and prediction of strong earth movement during earthquakes. The simulation needs a discrete model of the earth as input and computes the velocity of each mesh point. The mesh model needs terabytes of storage [8].

3. CONCEPTS AND NOTATIONS

In this section, we introduce some of the concepts and notation which will be used in later sections.

Boundary of a subset S of a topological space (represented by ∂S) is the set of points p of S such that every neighborhood of p contains points from S and its complement. An n-dimensional **Closed n-ball** D_n with radius r is defined as: $D_n = \left\{(x_1, x_2, ..., x_n) \mid \sum x_i^2 \leq r^2\right\}$.

DEFINITION 1 (**n-Manifold with Boundary**). *n-Manifold with boundary is a topological space in which each interior point has a neighborhood topologically equivalent (i.e., one can be transformed to the other without cutting or creating holes) to D_n and each boundary point has a neighborhood equivalent to the "half n-ball" $D_n \cap \{x \mid x_1 \geq 0\}$ [11].*

DEFINITION 2 (**CW-Complex**). *A (finite) CW-complex is a partition of a space S into pairwise disjoint open sets called (open) cells such that for each cell c of dimension n (an n-cell) there is a continuous bijective (one-to-one) map from D_n to c mapping the boundary of the ball to the union of lower dimensional cells in S.*

If the map in definition of CW-complex is bijective on the closure of D_n (i.e., on its interior and boundary), then the complex is called *regular*. The maximal dimension d of its cells is called the dimension of the complex.
A **d-mesh** is a finite CW-complex of dimension d.

DEFINITION 3 (**Side-Of Relationship**). *For two cells c_1 and c_2, if $c_1 \subset \partial c_2$ holds, then c_1 is* side-of *c_2 and we write $c_1 \prec c_2$. We also say that c_1 and c_2 are incident if one is the side of the other.*

Two cells are **adjacent (neighbor)** if both of them are side-of another cell or there is a cell which is side-of both.

DEFINITION 4 (**n-dimensional Simplex**). *The convex hull of the $(n+1)$ points defines a n-dimensional simplex.*

Simplex is a generalization of the triangle and tetrahedron to higher dimension ($d > 3$).

DEFINITION 5 (**Simplicial Complex**). *A simplicial complex C is a set of simplices which satisfies two conditions: 1) any face of a simplex in C is a simplex in C, and 2) the intersection of any two simplices of C is a face of the both simplices [13].*

The set of cells of a mesh and the side-of relationship form a finite **strict poset** (partially ordered set) $(M, \prec)$ which is irreflexive, transitive, and antisymmetric.

A **chain** is a subset of a poset which is totally ordered, it is *maximal* if it cannot be extended. A poset is *graded* if every maximal chain has the same length, i.e., d. The poset of a homogeneous mesh is graded. The *rank* of a poset is the length of its maximal chain.

If $a \prec b$, then the **interval** $[a, b]$ is the set of all cells which "a" is incident to and they are incident to "b": $[a, b] = \{c \in S | a \prec c \prec b\}$.

DEFINITION 6 (**Incidence Graph**). *An incidence graph (IG) of a mesh M is a graph (N, E) where N is the finite set of cells of the mesh labeled by dimension, and E is the finite set of edges linking the cells using the side-of relationship.*

Figure 2 depicts a tetrahedron and its corresponding IG.

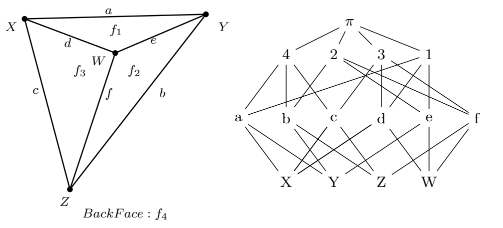

Figure 2: Tetrahedron (left) and its corresponding IG (right).

4. RELATED WORK

Cw-complexes, a concept from topology, are too general for many practical applications (in particular w.r.t. geometry). Several mesh representation models presented in the literature aim to restrict the cw-complexes to specific object domains, and to make them amenable to computation. However, in some cases the representations are very limited.

4.1 Winged-Edge

A widely used data structure in CAD systems is winged-edge data structure. The winged-edge uses edges to keep track of vertices and faces, i.e., each edge keeps information about its two vertices, the two incident faces, and the four edges that immediately follow in the boundary of its two faces. Each vertex/face contains a pointer to an arbitrary incident edge [4]. Euler operations can be used to construct such objects [3]. Winged-edge can represent 2D manifold polyhedral and some non-manifold meshes. Similar approaches are the quad-edge, half-edge and facet-edge data structures, all of which are special cases of the cell-tuple structure (see below).

4.2 Cell-Tuple

The cell-tuple represents a mesh as a pair $(T, switch_i)$ $(0 \leq i \leq d)$ where T is a set of $(d+1)$-tuples and $switch_i$ is a set of operators. Each tuple $(c_0, c_1, \ldots, c_{d-1}, c_d)$ is a maximal chain in the corresponding mesh poset: $c_0 \preceq c_1 \preceq .. \preceq c_{d-1} \preceq c_d$ holds. The switch operator maps each tuple to a unique tuple differing in only one dimension. More formally, the switch operator $switch_k(t)$ is defined as the unique cell tuple that agrees with t in all dimensions except in its k-th dimension [7]. A n-dimensional generalized combinatorial map (GMap) is similar to the cell-tuple [12]. Both are limited to manifold objects. Furthermore, Howe's research shows both models suffer from scalability issues [9].

4.3 Indexed Cell Set (ICS)

ICS is a compact mesh representation which stores coordinates of 0-cells and d-cells as ordered list of its vertices. ICS needs to infer information about k-cells $(0 < k < d)$ which grows fast for $d > 3$ [17]. ICS makes an implicit assumption about combinatorial structure of cells. The assumption does not affect simplicial complexes, because there is only one sensible combinatorial structure. But in other cases such as quad, we must know the order of vertex storage to be able to extract edges. In 3D, if d-cells have 6 vertices then we can either have prism, octahedron, a pyramid. ICS can represent manifold, some non-manifold, and multi-incidence objects.

4.4 GridFields

Howe [10] defines GridFields as a grid (mesh) with data associated to its cells. It is a general combinatorial grid model that separates the topology from the geometry and is not bound to any data structures, i.e., it models grid as a set of cells and incidence relationships between cells. Different operators have been defined which operate on grid (e.g., subgrid) or gridfield (e.g., regrid). Several data structures have been evaluated for implementing gridfield model. Howe's experiments on grid and gridfield show that more general models (e.g., ICS and a variation of IG) perform better. The work provides an algebraic framework for manipulating gridfields, but the model does not provide constraints to support different object classes. Furthermore, there is no declarative query language for gridfields.

4.5 Incidence Graph

Incidence Graph (IG) uses the incidence graph of a mesh as its representation, using the directed side-of relationship (two options are to link all sides of a cell or only those of one dimension less). IGs make it easy to access the boundaries of a cell. Indexed Cell Set (ICS) is a special case of IG with implicit assumption about the cells' combinatorial structure. It stores only coordinates of 0-cells, and each d-cell is stored as ordered list of its vertices. ICS needs to infer information about k-cells $(0 < k < d)$ which grows fast for $d > 3$ [17]. The cell-tuple graph is more redundant than the IG, e.g., in order to store a single tetrahedron, a cell-tuple representation needs $4 \cdot 3 \cdot 2 = 24$ tuples each linked to 3 other tuples, so 72 links in general, while IG needs only $1+4+6+4 = 15$ nodes and $14+(4\cdot 6)+(6\cdot 2) = 50$ links, if all sides are referenced, or $4+(4\cdot 3)+(6\cdot 2) = 28$ links, if only sides of dim $(k-1)$ are stored for each k-cell. In comparison to cell-tuples, IG does not support multi-incidence and does not directly give access to ordering information (such as list of vertices around a face which must be inferred). Furthermore, constraints on combinatorial structures must be added and checked explicitly. IG encompasses all object classes representable with previous models and more. To overcome IG limitations, we generalize IG to support multi-incidences and add checkable constraints.

Table 1 shows a comparison of the models above along the following dimensions: 1) *Object Domain* describes different topological models supported by each model (i.e., how cells are connected to each others), 2) *Dimension Independence (DI)* means that a n-dimensional object is recursively represented as a collection of objects of dimension $(n-1)$ connected through facets of dimension $(n-2)$. The recursive process stops when vertices are reached [11], 3) *Integrity Constraint (IC)* checks if an object belongs to the object domain, i.e., structurally valid, and 4) *Order* describes how lower dimensional cells are located around a higher dimensional cell, e.g., order of vertices around a face.

It is worth mentioning that Geo-databases such as PostGIS (partially) implement *Simple Features*, OGC and ISO standard (ISO 19125), for only two-dimensional geographical data and do not provide unstructured mesh functionalities [15].

5. COMBINATORIAL MESH DATA MODEL

Our data model is built on the incidence graph model. It is a multi-graph data model (i.e., allows multiple links in

Table 1: Summary of mesh representations' features.

Model	Object Domain	DI	IC	Order
IG	Not Multi-incidence	Yes	No	No
Winged-Edge	2D Manifold	No	Yes	Yes
ICS	Manifold	Yes	No	No
Gridfield	Not Multi-incidence	Yes	No	No
Cell-Tuple	Manifold	Yes	Yes	Yes

the incidence graph) which support representation of multi-incidences which can occur in geometric modeling. Examples of multi-incidence objects which cannot be represented by simple graphs are: loop (one edge with one vertex), torus with one cell (one vertex and two edges), pinched torus with one cell, etc. Before defining the ImG-Complex model, we formally define multi-incidence and incidence multi-graph:

DEFINITION 7 (**Multi-Incidence**). *Suppose that c is a cell in a cw-complex and $c_1 \preceq c$, if the pre-image of the neighborhood of any point p in the image of c_1 in D_n consists of k connected components, then c_1 is k-incident or* multi-incident *to c.*

DEFINITION 8 (**Incidence Multi-Graph**). *Incidence multi-Graph (ImG) extends the graph of IG (N, E) to a multi-graph (N, E_m), where each edge $e \in E$ is replicated k times in E_m if the corresponding side-of relationship is a k-incidence.*

This leads to the definition of ImG-Complex data model:

DEFINITION 9 (**ImG-Complex**). *A cell complex is called an Incidence multi-Graph Complex (ImG-Complex) if its cells and incidence relationships form an incidence multi graph.*

The definition of ImG-complex is very general and covers a wide range of mesh application domains. For a given ImG-complex, we need to store all side-of relationships I_{ij} where $0 \leq i \leq (d-1), 1 \leq j \leq d, i < j$. For efficiency reasons, some of the inverse relationships should also be stored. If the shapes of the cells are known (e.g., triangle, tetrahedron, etc.), then it suffices to store only I_{0d} (similar to the ICS representation) and infer other incidence relationship from it.

Although, ImG can represent meshes which incidence graph cannot, topologically different meshes may have the same ImG model, e.g., torus (with one cell) and Kelin bottle. In such cases, users should ensure sufficient subdivision if they need a unique representation

It can be shown that order information can be extracted from ImG model using Brisson's switch operators [7].

The merits of the ImG model are two folds: 1) representing a broad class of meshes, and 2) several admissible mesh classes (see Section 6). The latter is not specific to ImG.

In comparison to specialized models for simplicial meshes, a general model like ImG-Complex cannot help in processing and can exhibit performance problems. We can augment the ImG as a general data structure by maintaining simpler auxiliary structures (depending on additional constraints satisfied by the mesh) which facilitate query processing. For instance, to retrieve neighbors of a given d-cell, we can precompute and store those neighbors for each cell. Thus, query operations can be sped up transparently, at the expense of additional precomputing work and storage. Note that the query interface will hide representation specific details.

6. MESH CONSTRAINTS

ImG-Complex model can represent a large class of objects, but in practice we usually need to restrict ourself to a specific class of objects. One class of objects which plays a central role is manifold. However, in practice, manifolds are too restrictive, i.e., the union of all cells of a mesh may not form a manifold but rather form pseudo-manifold. We built sets of constraints to validate different classes of objects such as manifold, simplicial manifold, and pseudo-manifold. In this section, we show the sets of constraints for manifold and pseudo-manifold objects.

The constraints are not specific to ImG model and can be used with any other general model for validity check.

6.1 Manifold Mesh Constraints

In principle, we must check that each point of a d-mesh fulfills the manifold property. Followings constraints ensure that a given mesh dataset represent a valid manifold:

1. Graded IG: The poset of a manifold IG-Complex must be graded, i.e., every maximal chain has the same length. (Otherwise, there would not be a d-dimensional neighborhood for some points).

2. Diamond Property: If $f_1 \preceq f_2$ and f_1 and f_2 are faces of dimension $(k-1)$ and $(k+1)$ (i.e., $dim(f_2)-dim(f_1) = 2$), then the interval (f_1, f_2) contains exactly two faces f_3, f_4 of dimension k such that $f_1 \preceq f_3, f_4 \preceq f_2$ [7][18]. Figure 3 shows the diamond property for f_1 and f_2.

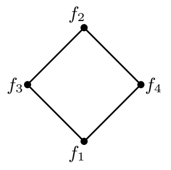

Figure 3: Diamond property between f_1 and f_2, the interval (f_1, f_2) contains only f_3 and f_4 .

The diamond property is proved for convex polytopes [18]. It guarantees that each $(d-1)$-cell f has exactly two incident d-cells and thus each interior point of f is a manifold point.

To check the diamond property in a graph, we focus on $(k+1)$-cell c_1 and $(k-1)$-cell c_2 such that $c_2 \preceq c_1$, and then, if the diamond property holds, there must be exactly two cells of dimension k in interval (c_2, c_1) incident to both c_1 and c_2. (If we look from another $(k+1)$-cell, then there will be 2 different k-cells.)

3. Cell Ring: What is missing now is the manifold property for points on cells of dimension $k \leq (d-2)$. However, we can use some properties stemming from the diamond property: If $c_{d-2} \preceq c_{d+1}$ in a mesh having the diamond property, all the cells in the interval $[c_{d-2}, c_{d+1}]$ form a set of circularly ordered rings. Note that c_{d-2} or c_{d+1} may be improper cells. For instance, edges and faces around a vertex (which are incident to the same 3-cell c in a 3D mesh) form a vertex ring, faces and 3-cells around an edge form an edge ring.

In a 2D mesh, the only points remaining to be checked for manifold property are vertices. We can do so by verifying algorithmically that there is exactly one vertex ring, and hence this ring forms the appropriate neighborhood. Figure 4 shows a vertex with two rings.

In a 3D mesh, we need to check vertex ring and edge ring. For edges, we check that there is a single edge ring around each edge.

Definition 10 (**Vertex Star**). *For vertex v, the (open)* star *of v is the set of cells which v is side-of: $st(v) = \bigcup_{v \preceq c} c$.*

For each vertex v we need to check that $\mathrm{st}(v)$ is a neighborhood of v: first, $v \notin \partial\mathrm{st}(v)$ (v is not side of any cell in $\partial\mathrm{st}(v)$) and second, $\mathrm{st}(v)$ is equivalent to D_3, i.e., $\partial\mathrm{st}(v)$ is a 2D manifold-mesh which is a 2-sphere. Now this is essentially the 2D case, where we can use the 2D Euler characteristic $\chi = V - E + F - 2$ (where V, E, F are the number of vertices, edges, faces of the mesh) to check that in addition $\partial\mathrm{st}(v)$ has the right topological genus and thus is homeomorphic to a sphere. Unfortunately, in higher dimensions, the Euler characteristic is not sufficient to guarantee topological equivalence between manifolds.

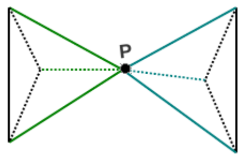

Figure 4: The object is non-manifold in point P, because edges and faces around point P creates two distinct rings.

6.2 Simplicial Manifold Constraints

A simplicial manifold satisfies all manifold constraints plus the following combinatorial constraints:

- Each 2-cell must have three edges.
- Each non-boundary edge belongs to exactly two 2-cells and each boundary edge belong to one or two 2-cells.

6.3 Pseudo-Manifold Mesh Constraints

A manifold with possible singular points not fulfilling the manifold property is called a *pseudo-manifold*. Two components of a pseudo-manifold must not be connected only by singular points.

Pseudo-manifold objects must satisfy graded and diamond property from manifold constraints. Additionally, any component of pseudo-manifold objects must be *strongly connected*. On cells of lower dimension, the manifold property can be violated, but any component of a pseudo-manifold must be strongly connected, i.e. there is a path between any two d-cells of a component consisting alternatively of d-cells and $(d-1)$-cells.

7. COMBINATORIAL MESH QUERIES

We consolidated a list of recurrent mesh combinatorial queries which appear in many domains [5][9].

Q1. *Enumeration of cells in a mesh (e.g., total number of cells, vertices, edges, etc. in a mesh)*
Q2. *Iteration over cells e.g., all cells of a mesh, cells of a particular dimension, etc.*
Q3. *Check if the vertex set of each facet is unique*
Q4. *Get list of cells which a particular cell is side-of*
Q5. *Get list of cells which are side-of a particular cell*
Q6. *Get the number of neighbors (adjacent) of a cell and iterating over them*

8. GRAPH DATABASES FOR MESHES

ImG is a theoretical model to represent a broad class of meshes. Selecting the data structure for ImG is an implementation detail. For instance, one option is to use graph databases. More efficient data structures can be used when the object class is known.

Graph databases model objects and the relationships between objects as graphs. They represent data by nodes, links and properties. Nodes represent objects and links show the relationship between nodes. A typical graph data model contains two main components: 1) nodes with properties, and 2) named relationships with properties. Some graph models also contains hypergraph [2].

8.1 Neo4j

Neo4j is an open source NOSQL graph database under AGPLv3 license which is operational since 2003. Neo4j is schema-less (i.e., allows modeling of complex and densely connected datasets), supports ACID properties, and can do high performance graph operations [1].

Neo4j's data model is property multigraph, i.e., data is stored in nodes and relationships (both with pairs of key-values properties) of a pseudo-graph. Relationships are directed, but graph traversal can be done bidirectional at equal speed.

Cypher, Neo4j's Query Language, is a declarative language, i.e., the user specifies the starting point and the desired outcome using graph pattern matching and Neo4j adapts the algorithm based on the user query. Cypher supports the most important graph algorithms such as shortest path and graph pattern matching. Patterns are very important in Cypher. It allows graph traversal. The user can use *MATCH* clause to describe the shape of the data he/she is looking for. Patterns have starting point(s) which are bound to a set of graph nodes or relationships. A pattern where parts of it are not accessible from any starting point will be rejected. We refer the interested reader to Neo4j documentation for further details [1].

8.2 Combinatorial Queries as Neo4j Queries

We can express only some of the combinatorial queries as single graph queries e.g., **Q1**, **Q2**; **Q4** can be expressed only when the length of the path (i.e., the length of maximal chain) is known; **Q3** needs a Java API and **Q6** in its general form is cumbersome to write as graph query. We formulate **Q4**, **Q6**, and Brisson's switch operator [7] as examples in Cypher.

We assume the ImG model of the tetrahedron in Figure 2. An excerpt of Cypher DDL to define the tetrahedron is shown in following listing:

```
CREATE (t{name:'t',dim:'3'}),
  (f1{name:'f1',dim:'2'}),
  ...
  (f4{name:'f4',dim:'2'}),
  (a{name:'a',dim:'1'}),
  ...
  (X{name:'X',dim:'0'}),
  t-[:INCIDENT]->f1,...,t-[:INCIDENT]->f4,
  f3-[:INCIDENT]->f,f4-[:INCIDENT]->a,
  ...
  a-[:INCIDENT]->X,
  ...
RETURN t;
```

Q4. To get the list of cells which a given edge "e" is side-of, the query below matches a pattern for interval (e,r] where r is the root. It returns all intermediate nodes r and f, i.e., all faces and 3-cells which "e" is side-of:

```
START r=node(0), e=node(12)
MATCH r--f--e
RETURN r,f
```

Symbol "- -" means "related to" which ignores the direction. Node(0) refers to root of the graph.

Q6. This query appears in many mesh domains. It has several variations, namely, vertex-to-vertex, edge-to-edge, face-to-face, and body-to-body adjacency. The query below extracts adjacent edges to edge "a" which share a vertex:

```
START a = node(20)
MATCH a-[:INCIDENT]->v<-[:INCIDENT]-e
RETURN e;
```

The pattern has two parts. The part a-[:INCIDENT]-¿v extracts all the vertices of edge "a". The parts v¡-[:INCIDENT]-e finds all the edges which the vertices are side-of. The set "e" does not contain edge "e". Note that here we used directed relationships, because the direction matters. "INCIDENT" in the query is the type of the relationship between nodes of the graph.

Switch Operator. Switch operators can be expressed as graph pattern matching. For instance, following Cypher query returns answer to $switch_1(tet, f_4, a, X)$:

```
START r=node(0), f=node(5), a=node(10),
      v=node(12)
MATCH  r--f--e--v
WITH collect(e) as c, a
RETURN filter(x in c: x<>a)
```

To sum up, we can write some of the queries as a single graph query, but to express other queries (e.g., **Q6**, **Q3**, and switch operator) several graph queries are needed depending on some input parameter, e.g., parameter k in switch operator. Furthermore, graph databases do not support the constraints and the constraints can not be written as a graph-only query and we need an API for each. Note that general graphs do not support geometric queries.

9. CONCLUSIONS

In this paper, we presented some initial result of the PhD work as follows: 1) discuss and motivate the need for a specialized mesh database, 2) present ImG-Complex model for combinatorial aspect of meshes, 3) introduce sets of constraints which limits ImG to subclasses of objects like manifold, pseudo-manifold, etc., and 4) present graph database potentials and limitation as a base for ImG model.

In the future, we want to extend the ImG model to support geometric embedding and data association, augment the model with indexing techniques (e.g., storing neighbors) to overcome performance issues, validate mesh properties (e.g., non-branching, homogeneous, etc.), and implement the model.

10. REFERENCES

[1] http://neo4j.org.
[2] R. Angles and C. Gutierrez. Survey of graph database models. *ACM Comput. Surv.*, 40(1):1:1–1:39, Feb. 2008.
[3] P. Baumann. A formal specification of a boundary representation. In *Eurographics*, volume 88, pages 141–154, 1988.
[4] B. G. Baumgart. A polyhedron representation for computer vision. In *Proc. National computer conference and exposition*, AFIPS '75, pages 589–596, New York, NY, USA, 1975. ACM.
[5] G. Berti. *Generic software components for Scientific Computing.* PhD thesis, BTU Cottbus, 2000.
[6] G. Berti. Gral - the grid algorithms library. *Future Generation Computer Systems*, 22, 2006.
[7] E. Brisson. Representing geometric structures in d dimensions: topology and order. In *Proc. 5th ann. symp. on Computational geometry*, SCG '89, pages 218–227, New York, NY, USA, 1989. ACM.
[8] V. A. et al. High resolution forward and inverse earthquake modeling on terascale computers. In *SC*, page 52, 2003.
[9] B. Howe. *Gridfields: model-driven data transformation in the physical sciences.* PhD thesis, Portland, OR, USA, 2007. AAI3255425.
[10] B. Howe and D. Maier. Algebraic manipulation of scientific datasets. In *Proc. 30th Int'l Conf. on Very large data bases*, VLDB '04, pages 924–935, 2004.
[11] B. Levy and J.-L. Mallet. Cellular modelling in arbitrary dimension using generalized maps. Technical report, ISA-GOCAD (Inria-Lorraine/CNRS), 1999.
[12] P. Lienhardt. Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Comput. Aided Des.*, 23(1):59–82, Feb. 1991.
[13] P. Lienhardt, L. Fuchs, Y. Bertrand, et al. Combinatorial models for topology-based geometric modeling. *Theory and applications of proximity, nearness and uniformity*, pages 151–198, 2009.
[14] D. W. Moore. *Simplicial mesh generation with applications.* PhD thesis, Ithaca, NY, USA, 1992. UMI Order No. GAX93-00795.
[15] OGC. Opengis simple features specification for sql. Technical report, Revision 1.0, 1998.
[16] C. Pain, M. Piggott, A. Goddard, F. Fang, G. Gorman, D. Marshall, M. Eaton, P. Power, and C. De Oliveira. Three-dimensional unstructured mesh ocean modelling. *Ocean Modelling*, 10(1):5–33, 2005.
[17] C. Silva, Y. jen Chiang, W. Corrêa, J. El-sana, and P. Lindstrom. Out-of-core algorithms for scientific visualization and computer graphics. In *Visualization 2002 Course Notes*, 2002.
[18] G. M. Ziegler. *Lectures on polytopes.* Springer-Verlag, New York, 1995.