Simulation Information Structures

M.S. Shephard, P. Fischer, K.K. Chand and J.E. Flaherty

Abstract. This document discusses the information components used in mesh-based simulation procedures. A set of information structures are outlined to deal with the various informational components and the interactions between them. The emphasis of this discussion is the functional level of the information components and their interactions.

1. Introduction

The process of solving problems in mathematical physics using generalized numerical analysis procedures has matured to the point that there are a wide variety of tools and techniques available. As part of this maturation process, a clearer understanding has emerged of the key informational components associated with these processes. In the area of grid/mesh-based solution to PDE's two such components are the grid/mesh structures and the non-linear algebraic systems of equations arising from the PDE discretization. Within the DOE SciDAC program, the TSTT center has been formalizing structures and tools for meshes and the TOPS center is providing structures and tools for solving systems of non-linear algebraic equations.

As efforts toward increasing the interoperability of simulation technologies moves forwards, we need to qualify, in as generalized a manner as possible, the key information structures. Before giving a high level definition of those structures it is important to first indicate the view of interoperable simulation taken here. This view assumes:

- The simulation process can require the application of multiple interacting discretization technologies.
- The discretizations employed during the simulation can evolve based on the use of a posteriori information (i.e. the application of adaptive discretization methods).
- It is desirable to be able to automate the steps in the simulation process to the greatest possible extent.

To support this view, the methods and structures used must be able to provide the required information at any point in the process "in a manner independent of any of the specific discretization processes". It is important to indicate that the statement "in a manner independent of any of the specific discretization processes" is only with respect to the means by which the information is provided. It does not mean that the evaluation of the information is independent of the discretizations upon which it is constructed. A consequence of this stated view of interoperability is there must be a high level definition of the problem being simulated that is independent of the discretization methods that can be used in its approximate solution.

The structures presented in this document are defined to support mesh-based simulation processes. Although specific suggestions for structures are biased toward unstructured meshes typical of finite element methods, the overall structures defined here are applicable to all classes of methods where a double discretization process is used. A double discretization process is one in which the domain is discretized into a set of piecewise components and the equations to be solved are discretized over these individual piecewise components. Methods covered by such approaches include finite difference, finite volume, finite element, boundary element and partition of unity (so-called meshfree). The key difference in the information structures given below is that the details to the “mesh” structure may be different. However, the basic relationships and interactions of all the structures remains the same.

Section 2 provides an brief introduction of the information flow within a mesh-based simulation process while Section 3 defines a set of information structures for dealing with the representation and flow of this information. This is followed by a section on each of the informational structures which considers:

- The functions that must be supported and the sources of information for that structure.
- The interactions of that information structure with other information structures with an emphasis on the key associations that must be supported.
- Overview of a basic representational approach for that structure.

This paper focuses on the functional requirements of information with the eventual goal of developing specific implementations of these structures to support the development of interoperable numerical analysis procedures. Essentially no consideration is given herein to the effective implementation of these structures. Experience indicates that a variety of implementations can meet these the functional requirements with relative advantages and disadvantages depending on the types of numerical methods they most commonly support.

This document presents a perspective of the discretization process that begins with a generalized form of overall problem representation that will support not only the various levels of adaptivity desired, by the effective integration with other tools used to define information important to the execution of simulation processes (e.g., a CAD model representation). As critical as it is that the discretization methods and associated library functions developed as part of this effort are consistent with the integration and use of such high level information, it is a necessary requirement that the methods be useful in supporting the development of discretization procedures where not all of this information is available at the highest level. For example, a majority of current simulation environments will associate geometric shape and physical attribute information at only the levels of mesh patches or individual elements. Since having the information at this level is sufficient for at least executing simulations on the given mesh patch structure or mesh, they must be supported.

2. Information Flow in Mesh-Based Simulations

Before stating a specific set of information structures and the interactions between them, it is useful to provide a high level view of the flow of information within mesh-based simulation processes. The input to the simulation process must include:

- The space/time domain of over which the simulation is to be run.
- The mathematical form governing the simulation (PDE’s, variational principle, weak form).
- Specification of the parameters, to be referred to as physical attributes, associated with the governing mathematical form in terms of:
    - material properties
    - forcing functions
    - boundary conditions
    - initial conditions

Given this input the space/time domain is discretized in a piecewise manner into a mesh, difference grid and/or overlapping partitions. Often one form of discretization is used for the space domain and another for the time domain.

The mathematical form is then discretized over the individual pieces of the space/time domain using finite dimensional approximations written in terms of basis functions and multipliers (to be referred to as dof in this document). The substitution of the finite dimensional approximation into the mathematical form will produce a local set of algebraic (typically non-linear) equations written in terms of the local dof's over the individual pieces of the space/time domain. The specifics of the mathematical form and finite dimensional approximations define the "rules" by which the local dof's of "neighboring domain pieces" relate to each other to define the dofs of the complete system (the global dofs).

The rules by which local dof's interact and define the global dof's are used to control the process of assembling the local sets of algebraic equations into the global algebraic equations. The global algebraic equations are then solved to produce the values of the global dof's. Some key points about this process are:

- This is typically the dominate part of the process in terms of computational time.
- The process of assembly and solution is rarely carried out such that all the dof over the space/time domain are determined at once.
- Given the dof and associated basis functions, it is possible to evaluate the physical fields over the domain.

Since the mesh is defined in terms of the input domain, it represents the key linkage between the input information and the other components of the numerical analysis process. Since the evaluation of the mesh level algebraic equations requires knowledge of the appropriate physical attribute information and since the mesh is derived from the domain definition, the physical attributes need to be associated with the appropriate entities in the domain definition. The dof and associated basis functions represent the key linkage from the mesh to the algebraic system.

Historically, most mesh-based simulations solved one set of equations on a single mesh. In these cases all information flow was in a single direction so all that needed to be supported is the ability to perform a single mapping of information forms. The application of adaptive methods and solution of coupled multiphysics systems using alternative discretization technologies requires maintaining knowledge of the appropriate mapping processes so that appropriate information can be transferred back and forth in the information chain. Maintaining the consistency of these mappings is critical to avoiding the introduction of additional errors into the process.

In cases where there are a specific limited number of such linkages, it has been common to develop specific ad-hoc procedures to support that transfer process. Such an approach has the advantage of being efficient and providing the most straight forward means to ensure the operations are performed consistent with the discretization processes performed. If has the disadvantage of not supporting other situations.

To support the needs of adaptive multiphysics simulations using alternative discretization technologies, a more formal set of information structures and their interactions is needed that will allow the consistent transfer of information between different structures interacting with different discretization technologies while maintaining the needed consistency.

3. A Set of Information Structures

This section introduces a set of information structures that can meet the needs of adaptive multiphysics simulations. The seven information structures are:

- The space/time domain
- The physical and mathematical model attributes
- The mesh
- The discretization operators and dof
- The algebraic systems contributors
- The algebraic system
- The fields

Space/Time Domain. The space/time domain represents a key portion of the problem definition. It must provide a complete understanding of the domain to be accounted for in the simulation. The geometric domain definition is also the appropriate structure to which all other problem definition information is defined (e.g., a traction on a model face).

Physical and Mathematical Model Attributes. The physical attributes represent the second component of the problem definition. In general the physical attributes are tensoral quantities that can be general functions of the independent variables of space and time as well as other dependent variables including physical attributes and fields. Within the definition given here the physical attributes are defined in terms of known distributions that are given as input to the simulation process. The mathematical model attributes indicate the mathematical model representation to be solved over the various portions of the domain.

Mesh. The mesh is a piecewise decomposition of the space/time domain. The mesh represents the first of the two discretization steps carried out. The specifics of the definition of the mesh is a function of the methods used for the second discretization process.

Discretization Operators and dof. Represent the second component of the discretization process in which the mathematical equations are discretized over the mesh entities. The two common methods of equation discretization are:

- Direct operator discretization (e.g., difference equations)
- Functional operator discretization in which finite dimensional basis functions are substituted into a weak form of the original equations (e.g., finite elements)

In either case the equation discretization is in terms of a finite number of parameters (the dof) over the appropriate mesh entities. The dof are the discrete parameters resulting from the equation discretization. The dof arise from the mesh level discretization processes and must be related to the global algebraic system.

Algebraic System Contributors. The algebraic system contributors are the result of the application of the discretization of the mathematical problem over the mesh the mesh entities.

Algebraic System. The algebraic system is the assembled system contributors into a global system. The actual numerical procedures used to solve the global system will dictate the level to which the full global system is assembled. For purposes of this discussion, the critical informational aspect of this process is how the dof associated with the contributors interrelate in the definition of the global algebraic system.

Fields. Are the description of physical parameters determined as part of the simulation process. As with the physical attributes these are tensorial quantities associated with entities in the domain model. The key difference is that the underlying distribution of these tensors is in terms of the basis functions (that are often operated on) and dof defined over mesh entities.

4. Space/Time Domain Definition

The material in this section focuses on high level representation which can effectively support a full set of automated and adaptive methods with no loss of information at any level of the process. In cases where there is not sufficient information to support this perspective completely, the discretization library capabilities developed must be able to interact with the information at the level at which it is defined.

4.1 Functional Requirements and Sources of Information

Form the functional viewpoint of supporting a numerical simulation, the space/time domain representation must be able to:

- Support the process of creating the first discretization step which is a mesh that properly represents the domain of the simulation.
- Support the ability to address any domain interrogation required during the numerical analysis procedures
- Support the association of the physical and mathematical attributes with the mesh discretization in a manner consistent with the simulation process.
- In the cases where the domain evolves as part of the solution process, be able to track the domain evolution.

In cases where a single fixed mesh is used for all aspects of the simulation process, the historic process of defining the domain in a bottom-up fashion directly in terms of the mesh entities with all attributes associated with the mesh was fully satisfactory. For purposes supporting interoperable simulation technologies using adaptive techniques this is not satisfactory since:

- Different discretization technologies that must exchange information between different domain discretizations (meshes).
- The applications of higher order and adaptive methods are not well satisfied when the only specification of the problem is in terms of an initial mesh where based on the mesh entities geometry and basis functions the domain geometry and representation of physical attributes can have a level of approximation that is not acceptable and can not be recovered.

Although there are a small number of discretization procedures that directly deal with the space/time domain in uniform manner, the vast majority of procedures employ separate discretizations of the spatial and temporal domains to account for the fact that the geometric domain it typically irregular while the temporal domain is regular. The discussion that follows focuses on the representation of the spatial (geometric) domain.

Another reason that defining the problem in terms of a single mesh is often not satisfactory is that many user communities are demanding simulation processes be able to be run automatically given a problem definition in terms of the domain definition they are provided. One popular form of such an independent definition is for the spatial domain to be defined in terms of a CAD model and to associate all other physical information, such as boundary conditions, loads, material properties and initial conditions, with the CAD model. However, a CAD model is not the only way to construct a high level domain definition. Not only is it desirable to support a high level definition of the problem given only a discretized definition of the domain, there are cases where the domain evolves based on the results of the simulation and therefore, the current definition of the domain must be constructed based on the domain discretization existing at the time. Again the issue is that the high level definition of the problem that can provide information "in a manner independent of any of the specific discretization processes" that may be involved with its current definition. And again, this information should always be determined in a manner consistent with any discretization used in its construction.

4.2 Key Associations

The geometric domain definition represents the key input structure to which spatially-based problem input must be associated. The entities defining the geometric model also provide a convenient means of requesting high level information on the solution results. For example to request the "lift" on an airfoil, one must integrate the correct solution parameters over the appropriate surfaces in the domain definition.

Most of the physical attribute information is directly associated with entities in the geometric model. Typical examples are material properties for a region, pressure load on a face, etc.

The relationship between the mesh and the geometry is critical, due to the special nature of this relationship and it associated rules, this specific association is referred to as classification (see Section 6) which will refer to the relationship of a mesh entity and the model entity to which it is associated. The relationship of a model entity to the mesh entities associated with is referred to as reverse classification.

In cases where the mesh is developed through a hierarchal decomposition of the domain, the maintenance of information relating the lowest level "mesh entities" to the top level domain representation will be through this hierarchy. For example in the case of the structured grid methods the domain is decomposed in terms of the a set of mesh patches, with a structured set of mesh entities within the patch. In this case the most effective means to maintain the geometric relationships is to (i) maintain the grid cell to mesh patch relationship in a form best suited to its representation and (ii) maintain the relationship of the mesh patch entities to the geometric domain entities.

Although the solution fields are defined at the base level in terms of dof and basis functions, the need to treat fields as physical attributes to either answer a user request or to represent input boundary or initial conditions for a multiphysics analysis dictates there is an association of the fields to geometric domain entities. So long as the dof and basis functions are associated with mesh entities, and the mesh entities are classified with respect to the model entities, it is possible to treat fields in the same way as any other attribute tensors.

4.3 Overview of a Representational Approach

As indicated above, there are multiple sources for the high level domain definitions with CAD models, image data and cell-based (mesh-based) being the most common forms for the definition of domains to be simulated. Each of these sources has one or more representational forms. Historically, CAD systems used various forms of boundary representations or volume based forms defined in terms of positioned primitive shapes combined by a set of Boolean operations. Image data is defined using a volumetric form such as voxels or octrees. Depending on the configuration of the cells (mesh entities) a variety of implicit and explicit boundary or volumetric representations have been used to represent these domains.

Except in cases of image data and when all aspects of the simulation process can be effectively defined in terms of volume metric, it is generally accepted that the use of a boundary representation is well suited for the domain definition in simulations based on solving partial differential equations over general domains. There is a substantial computer-aided design literature on the various boundary representations. Common to all of these representations is the use the abstraction of topological entities and their adjacencies to represent the various model entities of different dimensions. In a boundary representation the information defining the actual shape of the topological entities can be thought of as attribute information associated with the appropriate entities. The ability to interact with the domain definition in terms of the topological entities provides an effective means to develop abstract interfaces to the domain definition allowing the easy integration of multiple domain definition sources.

An important consideration in the selection of a boundary representation is its ability to represent the classes of domain needed. In the case of numerical simulations the domains to be meshed can be general combinations of 0-, 1-, 2- and 3-D entities in general configurations. Figure 1 shows a typical analysis domain that may be used for the structural analysis of a portion of a piping system. The analysis domain to be meshed is an idealization of the real geometric domain where portions of the pipes can be properly idealized by 1-D beam members and the support bracket can be properly idealized by 2-D plate members. However, the need to determine the 3-D stresses at the pipe juncture requires that the portion of the model near the juncture be fully represented in 3-D. This also necessitates the addition of idealized plate structures to tie the beam members to the 3-D portion of the model.

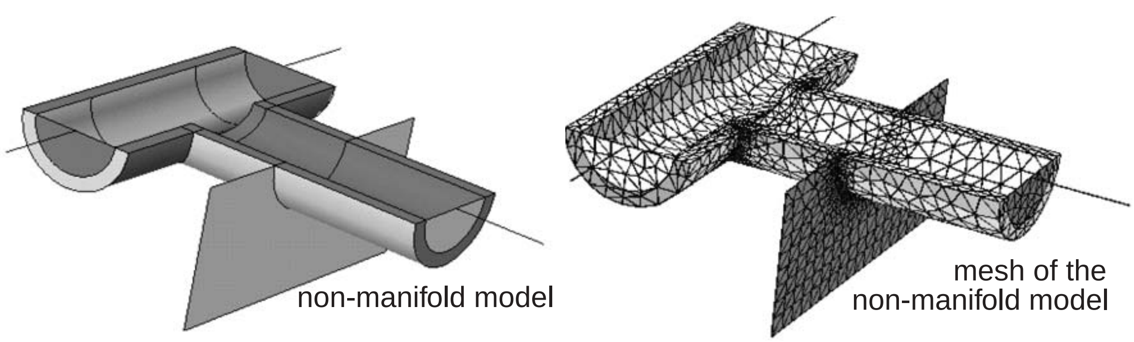

Figure 1. Example of a non-manifold model and associated mesh used in simulation.

The boundary representations that can fully and properly represent such geometric domains, as well as other situations such as multi-material domains, are referred to as a non-manifold boundary representations [8,26]. In addition to the basic the base 0-3 dimensional topological entities of the vertices, edges, faces and regions, boundary representations used for general geometric domains include loop and shell entities. A loop is a closed circuit of edges. Faces are bounded by one or more loops. A shell is a closed circuit of faces. Regions are bounded by one or more shells. In the case of non-manifold models the representation must also indicate how topological entities are used by bounding higher order entities. For example, each side of a face may be used by a different region. Therefore, faces have two uses. Another terminology for the use of a topological entity by higher order topological entities is co-entities [23]. Figure 2 shows the basic structure of one of the most commonly used non-manifold topological representations [26].

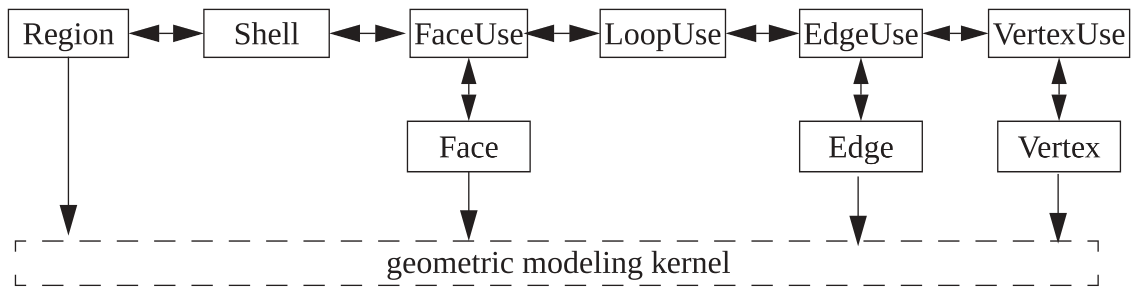

Figure 2. Model topological adjacency information and relationship to model geometry.

In addition to the abstraction of topological entities (which indicates how things are connected) and geometry (the information that defines shape), geometric modeling systems must maintain tolerance information giving numerical information on how well the entities actually fit together. The inclusion of this tolerance information is necessitated by the fact that to function properly geometric modeling systems must employ finite tolerance information. The algorithms and methods within the geometric modeling system are able to use the tolerance information to effectively define and maintain a consistent representation of the geometric model. (The vast of majority of what various geometry-based applications have referred to as dirty geometry is caused by a lack of knowledge or proper use of the tolerance information [2].)

The abstraction of topology provides an effective means to develop operator driven interfaces to boundary-based modelers that are independent of any of the specific shape information. The ability to generalize these interfaces is further enhanced by the fact that in the vast majority of cases, the geometry shape information actually needed by simulation procedures (normals, nearest point, various derivatives) consists of pointwise interrogations that can be easily answered in a method independent of any shape representation used by the modeler. An examination of more advanced situations, like evolving geometry simulations or automated geometric domain idealization processes, indicates that they can also be satisfied using methods independent of any shape representation. Although there are some situations where the simulation procedure is actually changing the model topology that operators may need to deal with loop and shell entities, it is possible to support the geometric interrogations used by simulation procedures focused on the basic model entities of vertices, edges, faces and regions.

The developer of CAD systems have recognized the possibility of supporting geometry-based applications through general API's. This has lead to the development of geometric modeling kernels like ACIS and Parasolid which are now used as the geometry engines for the majority of geometric modeling systems. Even those systems that do not use one of these kernels have made operator driven API's available (Granite from PTC). These geometric modeling API's have been successfully used to developed automate finite element modeling processes [13,21] and are the basis for commercial automatic mesh generators and simulation-based design procedures [2].

In some cases the only known representation of the domain is a mesh. In other cases, such as large deformation forming problems, fragmentation, etc., the shape of the overall domain evolves based on how the mesh moves and deforms. In these cases it is still desirable to construct a high level topological representation of the problem domain. This topological representation must be built based on information available from the simulation which is limited to the deformation of the mesh (e.g., node point coordinates), the model topology before the current set of analysis steps (in the case of an evolving geometry simulation), and simulation specific information such as contacting mesh entities, entities that have separated due to fracture, etc. In these cases the process of constructing or updating the topological entities associates with the domain geometric model for these cases is focused on determining the appropriate sets of mesh faces, mesh edges, and mesh vertices to associate with the model faces, edges and vertices respectively. Algorithms to do this based on mesh based geometry parameters and/or simulation contact or fracture information have been developed [11,12,18,25]. Once the model topology has been defined, the geometric shape information can be defined directly in terms of the mesh facets, or can be made higher order using subdivision surfaces [6,12] or higher order triangular patches [16,17].

5. Physical and Mathematical Model Attributes

5.1 Functional Requirements and Sources of Information

An examination of the properties of analysis attributes indicates those that qualify the physical problem are tensorial quantities, and therefore require a general scheme to specify various order tensors. Tensorial quantities [5] must be defined with respect to a coordinate system and may include various forms of symmetry which must be taken into account for efficiency. Therefore, the components of the structure used to define tensor attributes are (i) the order of the tensor, (ii) the coordinate system the attribute is defined in, (iii) the symmetries possessed, and (iv) the dependence of the tensor on other quantities.

Tensors are specified using values based on a specific coordinate system; however, by maintaining the coordinate system information, the tensor information can be transformed into any requested coordinate system.

In addition to tensors, other common forms of attribute information used in numerical simulations include integers, reals, and character strings. Another type is a reference to a model entity (or entities) that can be used to define other information. For example, the loading on one surface may be related to the minimum distance relative to another surface that is represented as a model entity attribute. Another example of using model references is in the case of describing the properties of a fan in a CFD analysis. One component of the fan's properties is the identification of the fan's inlet and outlet surfaces which would be represented as references to the model's surfaces. Similarly the system provides a mechanism to model information by referencing other attributes in the system.

The discretization library should be constructed such that is can fully support a complete set of attribute specification and manipulations. In addition, it must be able to effectively support the level of simulation processes that can be supported by the capabilities available. For example, if the functionality to do coordinate systems conversion is not supported, one can still solve in a single system given the needed information in that system, or if a full set of spatial variations of the attributes are not supported, support the specific ones that are.

5.2 Key Associations

Analysis attributes are associated with a specific portion of the domain being analyzed. In the problem definition case where analysis attributes are primarily associated with the specification of boundary value problems, the association of the attributes with the various topological entities in the boundary representation of the geometric model is a natural choice.

In addition to the original geometric entities in the model, auxiliary geometric information, which aids in attribute specification, may be required to fully capture the analyst's intentions. Common examples of auxiliary geometries are scribed edges on a model face used to indicate that a traction is only applied to a portion of a face, or projection geometry that aids in the specification of an attribute, such as a wind load distribution on a vertical plane. To properly reflect the association of such attributes the geometric model requires augmentation to account for the auxiliary geometry. For example, the curve used to denote the end of a traction on a face must be used to split the face during augmentation.

In those cases where the attribute information is defined in terms of another level of the geometric discretization library, the discretization procedures should support the needed operations from that level. For example, if the simulation tools assume the specification of physical attributes at the level of the patch structure in a structured grid procedure, the ability to associate the needed attribute information from the patch entity level to the grid entity level needs to be supported.

5.3 Overview of a Representational Approach

An attribute may be a function of space, time or some other attribute. In addition attributes can get their definition from a variety of other sources, such as:

- The results of a previous analysis where the spatial description is given in terms of quantities defined over some discretized version of the geometry.
- A function which depends on geometric operators (e.g. a traction may depend on the distance between two model entities).

Supporting mathematical expressions that consist of operators, constants and user defined variables allows the specification of general attribute distributions.

The functions desired to support mathematical expressions include:

- Arithmetic: + - * / X^Y (including vector and matrix versions)
- Trigometric: sin() cos() tan() arcsin() arccos() arctan()
- Nesting of functions (e.g. (), {}, etc.)
- Interpolations: Linear, Bi-Linear, Tri-Linear, Spline-based, etc.
- Integration and Differentiation
- Conditional
- User Defined Functions

In the case of modeling expressions, modeling operators such as closest point, intersection, union, and subtraction are also included.

To support the effective specification of attributes for the complete set of related analyses, while at the same time making it efficient to collect the attributes required for each specific analysis, an organizational structure is useful for the purpose of describing sets of attributes. The organizational structure should effectively support processes where multiple physical behaviors must be evaluated. In many cases, the results of one analysis represents part of the problem definition of another. For example, consider the situation of performing thermal, electrical, and thermal-mechanical analyses of an electrical component. Though the three analyses are quite different, there is an overlap of attribute information. The base materials are the same for all three analysis types, while the boundary conditions and loading conditions vary among the three. The thermal analysis would study various thermal load distributions. The thermal-mechanical analysis would use the resulting temperature fields as input to its load cases. The ability to effectively organize hierarchies of attributes as needed for each of these analyses is critical to a useful attribute management system.

To meet these needs the attribute manager should able to structure information:

- Consisting of several pieces of specific information which must remain together in order for a specific analysis to be well defined.
- To support relationships between information components. For example, a load on a given model entity may be represented by the vector expression {a, 0, 0}, while another load on a different entity is defined by {0, b, 0}. Using hierarchical information, one can impose a relationship between the variables a and b such that a = 2b.

6. Mesh

6.1 Functional Requirements and Sources of Information

The mesh is a piecewise decomposition of the space/time domain that represents the first step of the double discretization process. Simply stated, the mesh is needed because there is a lack of techniques to directly discretize the PDE's over general domains, while there are a number of approaches that support the definition and assembly of discretization operators over a mesh of the domain in a manner that provides the desired levels of accuracy.

It is common to employ different discretizations for the spatial and temporal domains. Since the definition of the mesh over the spatial domain is typically the more complex of the two, it will be the focus of this discussion. As a piecewise decomposition of the domain, basic requirements of the mesh are:

- To have the appropriately defined union of the mesh properly represent the domain of interest.
- To maintain sufficient geometric shape information so that processes such as differentiation and integration can be properly executed.
- To support the specification discretization operators for the PDE's of interest over the mesh entities.
- To maintain an understanding of the relationship of the mesh entities such that processes of defining dof and assembling contributors can be effectively carried out.
- To maintain a relationship to the overall problem definition (domain and analysis attributes) so that the appropriate information can be obtained.

The requirements on the mesh structure is a function of the requirements of the method of equation discretization used and the way in which dof are defined and related. One common form is the conforming mesh where the intersections of two mesh entities is null and the intersections of their closure is either the null or the closure of a common boundary mesh entity (face, edge or vertex). A minor variant of this is the non-conforming mesh where the intersections of the closure of two mesh entities is null, or different parts of the boundaries of the two mesh entities. Other mesh structures employ mesh patches that can overlay each other in a variety of ways. Finally, other methods, often referred to a mesh-free methods, are defined in terms of overlapping regions (e.g., spheres or cubes). In each of these cases there are specifics of rules associated with how the mesh regions interact, how discretizations are constructed in them including the assignment of dof, and how the system contributors defined from them are constructed and assembled.

In all these cases knowledge of the geometric shape of the mesh entities must be understood so as to support the construction of the contributors. In many methods the mesh geometric information is implied form a discrete set of parameters (e.g., node point coordinates) which is satisfactory for fixed mesh simulations. An alternative, that can be used to support all geometric operations is to maintain a linkage to the high level domain definition an use it to supply all geometric information. This alternative tends to be expensive for supporting all the required geometric interrogations so that a mesh based geometric definition (either implied or explicit) is typically used. However, in the case of adaptive mesh improvements it is necessary to use the links back to the original domain definition to ensure the mesh geometric approximation improves in a manner consisted with the other improvements. For example, as piecewise linear elements approximating curved portions of the geometry are refined, the new nodes are placed on the boundary, or as the polynomial order of the basis functions of an element is increased, the geometric approximation of the closure of that entity is increased to the correct order.

6.2 Key Associations

When the simulation is driven by a high level space/time domain definition, it is necessary to maintain the association of the mesh to the domain definition. Because of the central nature of this association and the importance that it follow appropriate rules, it is referred to as classification and is explicitly defined in Section 6.3 for the case where the model and mesh both maintain a topological representation.

In some cases the mesh is actually associated with a hierarchical structure in its definition. Two common examples when the mesh is defined in terms of a set of mesh patches (either for the purpose of mesh construction and/or to support the equation discretization methods used, or the mesh is collected into sets of mesh partitions that are distributed over the processors in a parallel computer. From a conceptional point of view, one can view these decomposition as part of a hierarchical domain composition in which the mesh entities are classified against the next level in the hierarchy, that level is classified with the next higher level and the highest level of decomposition is classified against the geometric model. Note that in actual implement, how formally one wants to support the classification hierarchy, and what relationships are needed, is driven by the needs of the meshing and simulation procedures interacting with them.

The mesh entities must also maintain appropriate associations with the basis functions and associated dof associated with them when they are used as contributors (Section 8). This association information must also capture the rules of how the dof of neighboring mesh entities interact. In simple cases of C^0 continuous functions defined with conforming basis, this information can be captured implicitly just based on the dof local labeling. In more complex situations such as the use of discontinuous functions of mesh based traversal operations, more explicit mean of qualifying the neighboring mesh entities is needed.

6.3 Overview of a Representational Approach

The data model for the representation of the mesh must maintain an association with the domain representation and with the discretization functions and dof. Therefore the details of it representation is a function of both of these structures. From the perspective of maintaining its relationship to the geometric domain, the use of an appropriate set of topological entities and their adjacency is ideal. In the case of unstructured meshes, particularly when there is the need to support adaptivity and higher order basis functions, it is ideal to apply such a topological relationship directly to the mesh [1,7,23]. In this manner it is possible to:

- Directly associate the mesh entities to the domain entities to obtain needed attributes and geometric information.
- Associated appropriate discretization functions and dof with the particular mesh entities.
- Support a variety of mesh level traversals (i.e., dof ordering to minimize global matrix skyline) and search operations using mesh topological entities.

In other cases, such a representation is not ideal for the lowest order entities in the mesh. For example, something like an octree, or some other spatially-based structure, is appropriate for the partition of unity (so call meshfree) methods. In the case of structured meshes maintaining a complete topology down to the individual cell entities would be overkill since all the information needed for the discretization process is well qualified by mesh patch ordering. However, in both of these cases there are additional structures desired that are well suited to a topological representation that can be associated with the domain topology. For example, the boundaries of the mesh patches in an unstructured mesh definition are ideally defined in terms of a topological structure augmented with the rules of mesh patch interaction. In the case of partition of unity methods, maintaining mesh like topological entities within the cells of an octree effectively supported the process of integrating those cells to the domain boundary and associating the boundary conditions [10].

Consider the case of using a topological structure for the definition of the mesh entities. Under the assumption that each topological mesh entity of dimension d, M_i^d, is bounded by a set of topological mesh entities of dimension $d-1$, $\{M_i^d\{M^{d-1}\}\}$, the full set of mesh topological entities are:

$$T_M = \{\{M\{M^0\}\}, \{M\{M^1\}\}, \{M\{M^2\}\}, \{M\{M^3\}\}\} \quad \text{(EQ 1)}$$

where $\{M\{M^d\}\}$, $d = 0, 1, 2, 3$ are respectively the set of vertices, edges, faces and regions which define the primary topological elements of the mesh domain. It is possible to limit the mesh representation to just these entities under the following set of restrictions [1]:

1. Regions and faces have no interior holes.
2. Each entity of order d_i in a mesh, M^{d_i}, may use a particular entity of lower order, M^{d_j}, $d_j < d_i$, at most once.
3. For any entity $M_i^{d_i}$ there is a unique set of entities of order $d_i - 1$, $\{M_i^{d_i}\{M^{d_i-1}\}\}$ that are on the boundary of $M_i^{d_i}$.

The first restriction means that regions may be directly represented by the faces that bound them, faces may be represented by the edges that bound them, and edges may be represented by the vertices that bound them. The second restriction allows the orientation of an entity to be defined in terms of its boundary entities (without the introduction of entity uses). For example, the orientation of an edge, M_i^1 bounded by vertices M_j^0 and M_k^0 is uniquely defined as going from M_j^0 to M_k^0 only if $j \neq k$.

The third restriction means that an entity (defined as $M_i^{d_i} \sqsubset G_i^{d_j}$ where, $d_j \geq d_i$ and at least one of $\partial(M_i^{d_i}) \sqsubset G_i^{d_j}$) is uniquely specified by its bounding entities. Most representations including the TSTT representation use this requirement. There are representational schemes where this condition only applies to interior entities; entities on the boundary of the model may have a non-unique set of boundary entities as illustrated with a model and a coarse mesh of a plate with a hole in Figure 3 can be bounded by the same entities [1]. The two mesh edges, M_1^1 and M_2^1, on the hole boundary have the same set of vertices, M_1^0 and M_2^0.

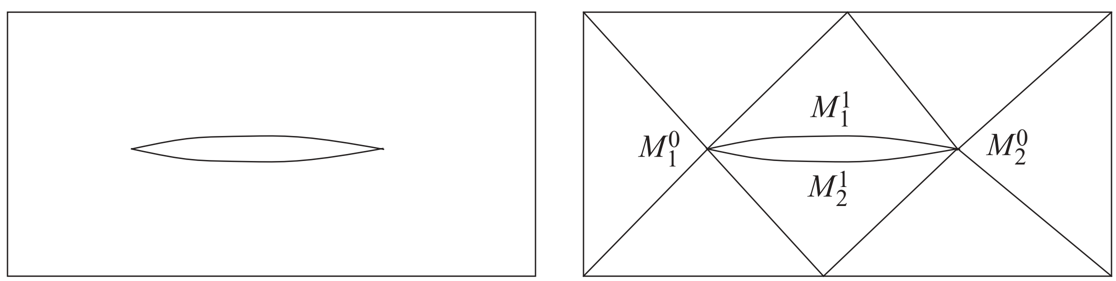

Figure 3. Example of mesh entities on boundary having non-unique boundary entities.

A key component of supporting mesh-based automated simulation is the association of the mesh with respect to the geometric model [1,20,21]. This association is referred to as classification in which the mesh topological entities are classified with respect to the geometric model topological entities upon which they lie.

Definition: Classification *- The unique association of mesh topological entities of dimension d_i, $M_i^{d_i}$ to the topological entity of the geometric model of dimension d_j, $G_j^{d_j}$ where $d_i \leq d_j$, on which it lies is termed classification and is denoted $M_i^{d_i} \sqsubset G_j^{d_j}$ where the classification symbol, $\sqsubset$, indicates that the left hand entity, or set, is classified on the right hand entity.*

Definition: Reverse Classification *- For each model entity, G_j^d, the set of equal order mesh entities classified on that model entity define the reverse classification information for that model entity. Reverse classification is denoted as:*

$$RC(G_j^d) = \left\{ M_i^d \middle| M_i^d \sqsubset G_j^d \right\} \quad \text{(EQ 2)}$$

Functions to provide the reverse classification, that is the equal order mesh entities classified on a model entity, are useful in various steps of simulation processes such as constructing system contributors (see Section 8).

The concept of the classification of mesh entities to a higher level model can be extended to include additional levels of model decomposition. Two important cases of this type are parallel mesh partitions and structured mesh partitions. In the cases when these partitions are non-overlapping, it is easy to view the associations of the models and having the mesh topological entities classified against the partition topological entities, and the partition boundary entities classified against the model topological. These concepts can be extended to the case of when there are overlapping partitions, each of which has been meshed. In this case one must define the appropriate rules of the interaction of entities in the different models.

Mesh shape information can be effectively associated with the topological entities defining the mesh. In many cases this is limited to the coordinates of the mesh vertices and, if they exist, higher order nodes associated with mesh edges, faces or regions. In addition, it is possible to associate other forms of geometric information with the mesh entities. For example, the association of Bezier curves and surface definitions with mesh edges and faces for use in p-version finite elements [14]. The mesh classification can be can be used to obtain other needed geometric information such as the coordinates of a new mesh vertex caused by splitting a mesh edge classified on a model face or to support the calculation of the geometric Jacobian information when doing an element stiffness integration.

7. Discretization operators and dof

7.1 Functional Requirements and Sources of Information

The partial differential equations (PDE) being solved are written in terms of a set of dependent variables that are functions of the independent variables of the space time domain. For purposes of generality, it assumed that the partial differential systems is written in terms of multiple sets of dependent variables which may be vector and or scalar quantities. For example, the dependent variables in compressible Navier-Stokes written in conservative form include the components of the velocity vector, $\underset{\sim}{u}$, the density, ρ, and an total energy term, e. For purposes of this discussion, a consider the set of PDE's being solved are written in the form:

$$\mathcal{D}(\underset{\sim}{u}, \sigma) - f = 0 \quad \text{(EQ 3)}$$

where

$\mathcal{D}$ - represents the appropriate differential operators.

$\underset{\sim}{u}(\underset{\sim}{x}, t)$ - represents one of more vector dependent variables which are functions of the independent variables of space, $\underset{\sim}{x}$, and time, t.

σ - represents one of more scalar dependent variables which are functions of the independent variables of space, $\underset{\sim}{x}$, and time, t.

f - represents the forcing functions.

(Note that the complete statement of PDE problem must include a set of boundary and, for time dependent problems, initial conditions.)

In the double discretization process used in mesh-based PDE solvers, the dependent variables are discretized over the individual mesh entities either by direct operator discretization (e.g., difference equations) or in terms of a set of basic function. In both cases the undetermined multipliers associated with the discretization are the degrees of freedom (dof). Specific mesh entities are used to construct the algebraic system contributors and are subsequently referred to as contributors. For those that think in the context of finite element methods, an element is a contributor.

In direct operator discretization processes discrete operators are directly substituted for the differential operators in the PDE's to yield a discrete form which can symbolically be written as:

$$\bar{\mathcal{D}}(\{\underset{\sim}{d}\}^e) - f = 0 \qquad \text{(EQ 4)}$$

where

$\bar{\mathcal{D}}$ - represents the discretized differential operators written in terms of appropriate discrete parameters $\{\underset{\sim}{d}\}^e$ which represent the dof associated with the discretization of the PDE over a mesh contributor, e.

In this process the discretization operators are defined over mesh entities and do correspond to a specific distribution of the primary variables over those mesh entities. When that dof associated with that distribution process correspond to the values of the dependent variables at specific points the distribution function is an interpolating function. Since it is not uncommon for some equation discretization methods to use dof that are not the values of the dependent variables at specific points, the term basis function is used to define the distribution functions.

In those methods based on the discretization of weak forms (e.g., finite element, boundary element and PUM) the dependent variables are discretized in terms of basis functions over a contributor, e, can be written as:

$$\underset{\sim}{u}^e = \sum_{i=1}^{ndof(\underset{\sim}{u})^e} \underset{\sim}{d}_i^{(\underset{\sim}{u})} N_i^{(\underset{\sim}{u})} \qquad \text{(EQ 5)}$$

$$\sigma^e = \sum_{i=1}^{ndof(\sigma)^e} d_i^{(\sigma)} N_i^{(\sigma)} \qquad \text{(EQ 6)}$$

where

$\underset{\sim}{u}^e$ - a typical vector dependent variable written over the contributor e.
$N_i^{(\underset{\sim}{u})}$ - is the i^{th} $\underset{\sim}{u}$ basis functions on contributor e.
$\underset{\sim}{d}_i^{(\underset{\sim}{u})}$ - it the vector of dof associated with $N_i^{(\underset{\sim}{u})}$ in the construction of $\underset{\sim}{u}^e$.
σ^e - a typical scalar dependent variable written over the contributor e.

$N_i^{(\sigma)}$ - is the i^{th} σ basis functions on contributor e.

$d_i^{(\sigma)}$ - it the dof associated with $N_i^{(\sigma)}$ in the construction of σ^e.

These discretized forms are substituted into the weak form which, with appropriate integration processes, with yield the final discrete form and thus the discretization operator for the associated entity.

In the fully discretized system the dof, $d_i^{(*)}$, and the shape function are functions of the independent variables, $N_i^{(*)}(\underset{\sim}{x}, t)$. In the commonly applied semi-discrete methods, the shape functions are only a function of the spatial variables, $N_i^{(*)}(\underset{\sim}{x})$, and the dof are functions of time, $d_i^{(*)}(t)$, where the time is independently discretized later. In both cases it is common the think of the dof as multipliers (within a time step) and the basis functions as the distribution of the dependent variables. The current discussion will take that perspective. However, any actually implementation needs to properly differentiate between the dof (which are simply multipliers) and the basis functions. Representing the expansion of the dependent variable in terms of basis functions and dof given in EQ 5 and EQ 6 can be used for the common discretization methods of finite element, finite difference[1], finite volume, partition of unity and boundary element.

The discretization operators applied to the dependent variables dictate the continuity requirements that must be met by their discrete forms. For example, if we consider the case of solving a second order PDE represented in variational form using a Galerkin finite element method, the resulting integral operators require C^1 continuity within the element (intraelement) and C^0 continuity between elements (interelement) of the discretization of the dependent variables. This can be met by using continuous linear or higher basis functions on the elements written in terms of "properly defined shared" dof associated with the closure of the element. This will meet the C^1 interelement continuity requirement. To meet the C^0 interelement continuity requirement requires the appropriate selection and "sharing" of dof that includes the following:

- The dof associated with each boundary entity of the contributor's mesh entity are common (shared) by all the contributors that the entity is in the closure of.
- The basic function and dof must be constructed such that the value of the discrete distribution of the dependent variable over each such entity is as function of only the dof associated with that entity and that those dof uniquely define that distribution.

If the same set of PDE were operated on by a discontinuous Galerkin variational form the introduction of the additional interelement flux operator reduces the interelement continuity requirement to C^{-1}. In this case the basis function on each element can be written in terms of independent dof over each element.

The combination of the PDEs, discretization operators on the PDEs, basic functions and dof over the individual elements will dictate the processes associated with the construction of the algebraic system contributors. The combination of the operators, basis functions and dof will also dictate the rules associated

1. Note that in the case of finite difference operators EQ 5 or EQ 6 are not used in the discretization process. The basis functions are only used in the contest of defining the variations of the fields over the mesh (typically in terms of interpolants) in field operations (Section 10). In these cases compact structures are often used to define this information.

7.2 Key Associations

Form the perspective given in EQ 5 and EQ 6, the discretization operators, basis functions and dof are associated with the contributors which are mesh entities. However, there is typically a very specific discretization operators and sets of basis functions used that are written in a general form that can be applied to any element of a given topology. Therefore, a general way to view the association of discretization operators and basis functions with the contributors is to assume there is a library of each are available and each contributor will employ an appropriate members of those libraries.

In addition to being associated with the individual contributors, the dof have an association with the global systems of equations. Note that this association exists independent of the methods used to solve the global system which range from complete global assembly processes to so called “element-by-element” methods where portions of the solve that can be performed are carried as soon as the contributors algebraic systems are formed. In any case, the ordering of the dof with respect to the global system has a substantial influence on the computational effort required to solve the system.

7.3 Overview of a Representational Approach

There are a wide variety of representational schemes that will satisfy the requirements and associations indicated above. In the vast majority of implementations they are ad-hoc implicit representations that are specific to the functions in the analysis code. The ability to support a multiple analysis code options requires a representational approach that explicitly considers the requirements and associativities given above.

In the case of finite difference methods there is a strong correspondence between the discretization operators (the difference equations) and basis functions used in field evaluations. Difference operators can be defined in a compact forms that account for the mesh topology, the order or the difference operators and issues associated with the representation of boundary conditions. The Overture system includes a set of such discretization operators [15].

The discretization operators used in weak forms are independent of the specifics of mesh topology and basis functions. In this case the discretization operators will contain information on the specific of integral form the basis functions are to be substituted into.

At this point we only address the basic representation of the basis functions and the dof. The complete representation will include the ability to support operations in the construction of system contributor (covered in Section 9) and manipulating the fields defined by the combination of the basis functions and dof (covered in Section 10). To support a variety of implementations, a high level approach is taken here to the specification of basis function and dof in terms of a basis function library and a dof manager.

A key attribute of the basis function library is that the individual entities in the library can be written in a form that is independent of the actual size and shape of the mesh entities they will be associated with. (However, they are typically not independent of the topology of the mesh entity.) This is accomplished by either writing the basis function in a local coordinate system that is a function of the mesh entity topology or directly in the global coordinated of the entire system. In the case where a local coordinate system is used, the ability to actually carry out the needed operations does require the introduction of mappings (see Sections 8 and 10). Figure 4 shows some example basis function library entries used in the Trellis software [19,24]. The use of a basis function library coupled with the use of mesh topological entities provides a flexible means for constructing algebraic contributors. For example, Trellis supports variable order C^0 p-version finite elements by assigning polynomial orders to the mesh topological entities of edge, face and region (mesh vertices are assumed to contribute to the "linear" basis only). In this case the process of constructing the appropriate left-hand contributor includes examining the polynomial orders of the mesh entity set defined by the closure of the contributor and collecting the associated entity level p-version basis functions for the library [22].

Functions	DofKey's
$1-u-v$	$\{ \mathsf{M}_1^0 , \texttt{field->getTag()} , 1 , 1 \}$
u	$\{ \mathsf{M}_2^0 , \texttt{field->getTag()} , 1 , 1 \}$
v	$\{ \mathsf{M}_3^0 , \texttt{field->getTag()} , 1 , 1 \}$

Table 1: functions and keys for a 1^{st} order Lagrange interpolation on a triangle

Functions	DofKey's
$1-2v$	$\{ \mathsf{M}_1^1 , \texttt{field->getTag()} , 1 , 1 \}$
$2(u+v)-1$	$\{ \mathsf{M}_2^1 , \texttt{field->getTag()} , 1 , 1 \}$
$1-2u$	$\{ \mathsf{M}_3^1 , \texttt{field->getTag()} , 1 , 1 \}$

Table 2: functions and keys for a 1^{st} order non-conforming Crouzeix-Raviart interpolation on a triangle

Functions	DofKey's
w	$\{ \mathsf{M}_1^0 , \texttt{field->getTag()} , 1 , 1 \}$
u	$\{ \mathsf{M}_2^0 , \texttt{field->getTag()} , 1 , 1 \}$
v	$\{ \mathsf{M}_3^0 , \texttt{field->getTag()} , 1 , 1 \}$
wu	$\{ \mathsf{M}_1^1 , \texttt{field->getTag()} , 1 , 1 \}$
wv	$\{ \mathsf{M}_2^1 , \texttt{field->getTag()} , 1 , 1 \}$
uv	$\{ \mathsf{M}_3^1 , \texttt{field->getTag()} , 1 , 1 \}$
$s_1 wu(u-w)$	$\{ \mathsf{M}_1^1 , \texttt{field->getTag()} , 2 , 1 \}$
$s_2 uv(u-v)$	$\{ \mathsf{M}_2^1 , \texttt{field->getTag()} , 2 , 1 \}$
$s_3 wv(v-w)$	$\{ \mathsf{M}_3^1 , \texttt{field->getTag()} , 2 , 1 \}$
uvw	$\{ \mathsf{M}_i^2 , \texttt{field->getTag()} , 1 , 1 \}$

Table 5: functions and keys for a 3^{rd} order interpolation on a triangle. Note that $w = 1-u-v$.

Figure 4. Some example basis function library members [24].

When weak forms are used there is additional information that is required to carry out the integration processes required to create the discrete form. This includes information on integration rules and methods to handle the geometric Jacobain information. Again appropriate library entries can be defined for these items.

The dof manager is responsible to controlling the dof during the various steps in the process. In addition to having the actual value of the dof[1], the structure for each dof needs to include:

1. In the case of vector dof, $\underset{\sim}{d}_i^{(\underset{\sim}{u})}$, it is assumed that there is one value per component.

- dof handle
- status (evaluated or not)
- global equation number(s) (one for each component in the vector case)
- mesh entity the dof is associated with
- field the dof is associated with (this may be redundant information depending on the definition of other structures)

8. Algebraic Systems Contributors

8.1 Functional Requirements and Sources of Information

The contributors are mesh entity level algebraic expressions resulting from the discretization of the PDEs using the appropriate discretization operations and associated basis functions when needed. The global algebraic systems is defined by the appropriate assembly of the contributors. In a classic mesh based analysis code, there is an implied correspondence between the mesh entities in the input file and the contributors. In this case the information associated with the contributors and the mesh are obtained simultaneously and simply supplemented with some additional control information. Such a mode of operation must continue to be effectively supported. However, it is also important to support simulations when a mesh independent high level problem definition is provided and automated adaptive techniques are applied.

The construction of the system contributors can be controlled by the appropriate traversal of information in the high level problem definition, or at a level above the mesh such as the mesh patch level for structured methods. For purposes of this discussion the types of information defined in the specific representational approaches in subsections 4.3 and 5.3 are assumed[1]. The first step is determining the mathematical model to be solved and the properties of the discretization methods to be employed by accessing the overall control information in the problem definition case. The system contributors are then calculated by an appropriate traversal of the problem definition information. This can be done by the properly traversing the physical attributes in the problem definition case, and the discretization and solution information in the strategy case. Another option would be to traverse the model entities and use links to the appropriate attribute information. In either case, as the traversal process must:

- Determine the mesh entities that will be used in contributor construction.
- Get the appropriate discretization operator sets for that contributor.
- Construct the list of dof associated with the contributor.
- Apply the appropriate differential and/or integral operations to the basis functions as defined for the contributor to construct the resulting matrix and/or vector contributions.
- Provide the resulting matrix and/vectors with needed dof ordering to be processed in the global system.

---

1. It is important to note that although the discussion presented here is based on a high level problem definition with defined components not readily identifiable in classic mesh based codes, the actual steps are in fact much same and classic codes can be effectively supported using the same overall approach. In that case, the problem definition case is simply the overall process control information typically at the beginning of an analysis code input file and the contributors are identified by traversing the mesh information in that file.

In the case when the system contributors are processed by a traversal of the domain entities, reverse classification and mesh entity adjacencies can be used to obtain the lists of any mesh entities that will have contributors processed for them. The analysis attribute information is used to determine what contributors, if any, are associated with the model entity. In the three-dimensional case the domain regions will lead to the construction of contributor matrices for the left-hand side of the global system and any contributions to the right hand vector due to domain contributions (e.g., body forces). The traversal of the domain boundary entities will lead to the construction of the contributions to the global system associated with boundary conditions. The natural boundary conditions will contribute to the right-hand side vector. The essential boundary conditions define constraints that eliminate or match (as in the case of periodic boundary conditions) possible dof and, in the case of non-zero essential boundary conditions, contributions to the right-hand side[1]. Three technical points are:

- Processing essential boundary condition information does require a proper interpretation of the boundary condition with respect to the closure of the entity to which it is applied.
- The efficient implementation of the a mesh-based simulation procedure requires the processing of the essential boundary conditions before processing contributors that contribute to the left-hand side for mesh entities that are bounded by that boundary condition. (The relates to the issue possible dof on the entity are eliminated from the global system and in that case the matrix terms will be scaled by non-zero essential boundary condition values and added to the right-hand side vector [9].)
- The process of accounting for boundary conditions when non-interpolating basis functions are used can require the introduction of additional operations (e.g., projection operations) that define relationships between dof. Such operations are common is spectral and p-version FE methods.

Initially information on the appropriate discretization operators and basis functions will be constructed from the attribute information. In the case where they are adaptively set, each contributor can have its own set and in the case of methods like variable p-version methods information on the basis functions will be associated with individual mesh entities defining the closure to the contributor.

As part of the process of setting-up the contributor, the list of dof associated with the contributor has to be determined/defined. One option for doing this is to track that through the mesh entities in the closure of the contributor since it provides an effective way to deal with common dof between entities in $C^{\alpha}, \alpha \geq 0$ continuous discretizations as well as when the basis are associated with mesh entities (e.g., C^{0} p-version adaptive).

A number of capabilities are needed to effectively support the application of the required differential and/or integral operators to the basis functions. Standardized procedures for gradients, divergence, curl and other differential operators to operate on the basis functions are needed. Methods to effectively implement integration processes are needed. Since many of these integrals follow specific standard forms, it is possible to generalize

1. Properly dealing with non-zero essential boundary conditions does require an interaction with the contributions of the domain entities that boundary condition bounds [9]. It can shown that this is easily handled as part of the assembly process given knowledge of the possible dof that were eliminated by that boundary condition.

the standard cases. In the most general cases where the integrands can be rational functions (typically caused by Jacobian mappings) numerical integration rules must be supported. In the case of general mesh entity shapes, information of the geometric mappings for the mesh entities being integrated is needed. Note that in the case where numerical integration is used the operators need only support pointwise evaluations of the operations needed to evaluate the integrand at specific points.

Given a contributor and its list of dof, a general assembler can support its inclusion into the global system.

8.2 Key Associations

The contributors have associations with all the other information structures. The problem definition information provides the base level information on the contributors needed. This includes information on the mathematical problem to be solved and the values of parameters associated with the PDE's being solved. In the case on multiphysics analyses some of these parameter evaluations may be provided from fields of other analyses. The linkage to the mesh structure is critical to the proper definition and construction of the contributor. In combination with the appropriate mesh information, the contributor must get the appropriate basis functions, dof and operators as needed to define the contributor vectors and matrices which must be properly associated with the global algebraic system.

Since the associations are often over logical groups of mesh entities the associations may be defined at a level where mesh entities are grouped such as at the mesh patch level in structured grid methods or general mesh set structures.

8.3 Overview of a Representational Approach

The representation of the contributor must indicate the mesh entity it is associated with, be able to obtain the list of dof associated with the contributor and maintain access to the resulting matrix or vector.

While the contributor is being constructed, the information on the basis functions and any mappings are needed as well as the operations applied to the basis functions and mappings in the construction of the contributor. Depending on the needs of the simulation process, this information may be explicitly associated with the contributor, or simply determined form the appropriate attribute information at the time of formation.

9. Algebraic System

9.1 Functional Requirements and Sources of Information

The algebraic system represents the fully discretized system to be solved. Since the vast majority of the computation time of a numerical analysis is spent in the solution of the algebraic system this area has received considerable attention by the numerical analysis community. These efforts have resulted in a wide variety of "solver libraries" that are becoming mainstay of numerical analysis codes.

Although there are a wide variety of specific forms possible for the algebraic system, key characteristics is that it will consist of one or more matrices of evaluated coefficients multiplying vectors of the dof on the left-hand side of the equation. This is equated to a vector of evaluated parameters on the right-hand side of the equation. This resulting system is linear when all the left-hand matrix and right-hand vector coefficients are independent of the dof. If any of these terms are a function of the dof the system is nonlinear.

The coefficients in the matrices and vectors are determined by the appropriate “assembly” operations applied to the coefficients of the algebraic systems contributors defined over the mesh entities. The assembly operations are defined in terms of the relationship of the coefficients of the algebraic systems contributors to the system dof and the assembly requirements of the solver used.

9.2 Key Associations

The key associations with the global algebraic system and the processes associated with their solution are the contributors, associated dof lists and dof structure. This information is used by the assembler(s) to construct the forms needed by the solver(s). Once the solution process is complete the dof manager is responsible for up-dating the dof structures indicating the dof that are evaluated and providing the appropriate values.

9.3 Overview of a Representational Approach

There are a variety of representational forms for the algebraic system by the various solver packages. An example that has an effective set of methods to provide the needed information is the PETSc package [3,4].

10. Fields

10.1 Functional Requirements and Sources of Information

Fields represent the solution information from a simulation in a form useful for queries and operations. Since the solution information determined in the simulation is the approximation for physical parameters, fields are various order tensors that generally vary over the domain of the problem. Therefore, the most basic requirement of the field structure is to maintain a representation of the physical parameters solved for (the dependent variables) over the problem domain in a manner that is consistent with the solution process.

Since the fields associated with the dependent variables are defined in terms of the basis functions (that may be interpolants) and dof over contributors, they were already constructed during the solution process. These fields will be referred to as the primary variable fields. In addition to the primary variable fields, there is often the desire to define additional fields, to be referred to as secondary variable fields. Common examples of secondary fields are ones associated physical parameters like fluxes and stresses defined by the appropriate differentiation of the primary variables and application of appropriate material parameters. Functionality similar to those needed to supported differential an integral operations applied to contributors need to be supported. These operations can include ones that are consistent with the original basis functions used for the primary dependent variables, or can include the introduction of additional basis functions, field dof and operations. A typical example of this type is the construction of a stress fields to be provided to visualization tools. Often the stress field evaluated using a consistent definition of the original basis function will have discontinuities between mesh entities it is to be displayed over. If a smooth stress field is desired for the purpose of display, it can be constructed from the original stress field using a set of continuous basis functions and new stress dof employing projection operations that may be supplemented by functions like equilibrium iterations.

In addition to supporting structures for the definition of the fields and the operations to construct them, additional structures may be introduced to support operations on them. Form example, if one want to support the general interrogation of a field for its value any-where over the domain, various search structures may be introduced to efficiently identify a small list of candidate mesh entities to check for the containment of the point of interest. This can be supported by mesh adjacency structures and/or added structures like a regular tree. The basic interrogation operations should be structured such that the field can be interrogated in both an evaluated and unevaluated form, where in the evaluated form the values of any needed dof are available and in the unevaluated form they are not yet available. Since the cost of execution of specific interrogations can vary dramatically with discretization method, specific care should be exercised to construct means to extract the needed field information in ways most appropriate to the current discretization method.

Since in the definition of fields are in terms of coordinate systems, the structures need to understand the coordinate system the field is defined in and support the transformation of the field's tensor into other coordinate systems as desired.

It should be noted that the need to construct various fields goes well beyond providing output to a graphical postprocessor. They play key roles coupled multiphysics analyses and in operations such as a posteriori error estimation. It is important to note that the basis functions, projections and constraints used in the construction of fields for such uses must be carefully matched to the specific procedure being implemented. Therefore, it is important to support flexibility in the definition of various fields.

Another functional requirement is the ability to derive fields, or in the limit scalar parameters, from other fields (both primary and secondary) based on requests at either the mesh or model level. At the mesh level, the fluxes on a mesh region's faces is needed for specific error estimation procedures. At a model level, the determination of the lift over an airfoil requires the integration of the pressure on the appropriate model faces.

10.2 Key Associations

Since the fields are defined by solutions over the mesh they must be associated with the mesh, basis functions and dof. When the field of interest corresponds to the primary variables it is directly associated with the appropriate contributors.

In the cases where the field is for a secondary variable, the field must also be associated with the operations and associated information used in its construction. The construction of these fields often employ different basis functions than used for the primary variables that must also be available in the basis function library. In addition, there is a need to associate the resulting discrete multipliers associated with these fields.

Finally, there will be many cases where the fields are associated with the appropriate entities in the problem definition. Some common examples are the field over the entire spatial domain at a particular point in time and the normal flux over a given set of domain faces.

10.3 Overview of a Representational Approach

Clearly supporting the full range of function and associations indicated above will require a sizable representation structure. In cases where there are a very well understood set of basis functions over major portions of the mesh, the use of a compact structure that focuses on just the needed discrete parameters is desirable. Effective support of structured mesh methods required that information below the patch level be based on a compact form.

Figure 5 graphically depicts a field description for a set of mesh faces. The definition of a mesh entity based structure to define this field in this case would maintain information of the basis functions, discrete parameters, mappings and operators applied in the construction of the field.

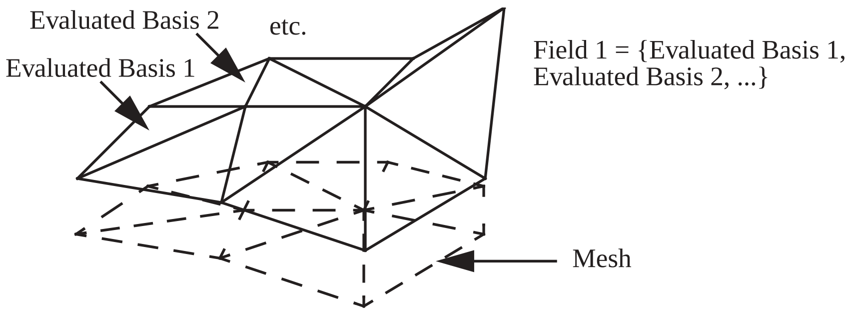

Figure 5. Representation of a field defined over a mesh.

The structured needed to support the definition of fields is much the same as used in the definition of the contributors (Section 8).

11. References

[1] Beall, M.W. and Shephard, M.S., “A General Topology-Based Mesh Data Structure”, *Int. J. Num. Meth. Engng.*, 40(9):1573-1596, 1997.

[2] Beall, M.W., Walsh, J. and Shephard, M.S., “Accessing CAD Geometry for Mesh Generation”, Report 2003-1, Scientific Computation Research Center, RPI, Troy, NY, 2003.

[3] Balay, S., Buschelman, K., Gropp, W.D., Kaushik, D., Knepley, M., McInnes, L.C., Smith, B.F. and Zhang, H., "{PETSc} home page", http://www.ncs.anl.gov/petsc, 2001.

[4] Balay, S., Gropp, W.D., McInnes, L.C. and Smith, B.F.,“Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries”, *Modern Software Tools in Scientific Computing*, E. Arge, A.M. Bruaset and H.P. Langtangen, Ed., Birkhauser Press, pp. 163-202, 1997.

[5] Beju, I., Soos, E. and Teodorescu, *Euclidean Tensor Calculus with Applications*, Abacus Press, 1983.

[6] Cirak, F., Ortiz, M. and Schroder, “Subdivision surfaces: a new paradigm for thin shell finite-element analysis”, Int J. Num. Meth. Engng, 47:2039-2072, 2000.

[7] Dey, S., O’Bara, R.M. and Shephard, M.S., “Curvilinear mesh generation in 3D”, Computer-Aided Design, 33:199-209, 2001

[8] Gursoz, E.L., Choi, Y. and Prinz, F.B., “Vertex-Based Representation of Non-Manifold Boundaries”, *Geometric Modeling Product Engineering*, North Holland, Amster

[9] Hughes, T.J.R., *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover, 2000.

[10] Klaas, O. and Shephard, M.S., “Automatic generation of octree-based three-dimensional discretizations for partition of unity methods”, *Computational Mechanics,* 25(2-3):296-304, 2000.

[11] Krysl, P. and Ortiz, M., “Extraction of boundary representation from surface triangulations”, Int J. Num. Meth. Engng, 50:1737-1758, 2001.

[12] Lee, C.K., “Automatic metric 3-D surface mesh generation using subdivision surface geometry model. Part 1: Construction of underlying geometric model”, Int J. Num. Meth. Engng, 56:1593-1614, 2003.

[13] Li, X., Shephard, M.S. and Beall, M.W., “Accounting for curved domains in mesh adaptation”, International Journal for Numerical Methods in Engineering, 2002.

[14] Luo, X., Shephard, M.S., Remacle, J.-F., O’Bara, R.M., Beall, M.W., Szabó, B.A. and Actis, R., “p-Version Mesh Generation Issues”, 11th International Meshing Roundtable, Sandia National Laboratories, pp. 343-354, 2002.

[15] Overture webpage: http://www.llnl.gov/CASC/Overture/

[16] Owen, S.J. and White D.R., “Mesh-based geometry: A systematic approach to constructing geometry from a finite element mesh”, Proc. 10th Int. Meshing Roundtable, Sandia report SAND 2001-2967C, pp. 83-96, 2001.

[17] Owen, S.J., White D.R. and Tautges, T.J., “Facet-based surfaces for 3-D mesh generation”, Proc. 11th Int. Meshing Roundtable, pp. 297-311, 2002.

[18] Pandofi, A. and Ortiz, M., “An efficient procedure for fragmentation simulations”, Engng. With Computers, 18(2):148-159, 2002.

[19] Remacle, J.-F., Flaherty, J.E. and Shephard, M.S., “An adaptive discontinuous Galerkin technique with an orthogonal basis applied compressible flow problems”, *SIAM Review*, 45(1):53-72, 2003.

[20] Schroeder, W.J. and Shephard, M.S., “On rigorous conditions for automatically generated finite element meshes”, Turner, J.U., Pegna, J. and Wozny, M.J., eds., *Product Modeling for Computer-Aided Design and Manufacturing*, North Holland, Amsterdam, 267-291, 1991.

[21] Shephard, M.S. and Georges, M.K., “Reliability of Automatic 3-D Mesh Generation”, *Comp. Meth. Appl. Mech. and Engng.*, 101:443-462, 1992.

[22] Shephard, M.S., Dey, S. and Flaherty, J.E., “A straight forward structure to construct shape functions for variable p-order meshes”, *Comp. Meth. Appl Mech. Engng.*, 147:209-233, 1997.

[23] Tautges, T.J., “The common geometry module (CGM): A generic, extensible geometry interface”, Proc. 9th Int. Meshing Roundtable, Sandia report SAND 2000-2207, pp. 337-359, 2000.

[24] “Trellis”, SCOREC, RPI, Troy, NY, 2003.

[25] Wan, J., Kocak, S, Shephard, M.S. and Mika, D., Automated adaptive forming simulations”, submitted to 12th Int. Meshing Roundtable, 2003.

[26] Weiler, K.J., “The radial-edge structure: A topological representation for non-manifold geometric boundary representations”, M.J. Wozny, H.W. McLaughlin, J.L. Encarnacao, editors. *Geometric modeling for CAD applications,* North Holland, pp. 3-36, 1988.