#### Simulation Fields - WORKING DRAFT RPI Version 0.08

# 1. Introduction

Simulation fields are one of the seven information structures associated with the numerical solution of PDE's using mesh-based methods [4]. Simulation fields represent tensor quantities defined in terms of numerical analysis discretizations in a form useful to support queries and operations by other functions or simulations. Two common examples where fields are used are (i) multiphysics analysis where the solution fields from each physics analysis represents a forcing function or boundary condition for another, and (ii) the construction of external adaptive control loops where the solution fields are used by error estimation procedures to obtain estimates of the discretization errors and to construct new mesh size field.

The tensors defining input fields associated with loads, material properties and boundary conditions of a PDE can be defined directly in terms of general distributions over the entities in the high level problem definition. The definition of these tensors can be supported by the use of attribute structures (see reference [3]). This document focuses on the methods and functions to support the definition and use of tensors discretized over numerical analysis discretizations, subsequently referred to as fields, that discretize tensors defined over the meshes of the domain.

The definition of a tensor for a numerical analysis discretization has the obvious two main components:

- The definition of the tensor.
- The discretization of a tensor.

## 1.1 Definition of a Tensor

Tensor quantities [1] used in the quantification of problems of mathematical physics are of order  $n \ge 0$ . A zeroth order tensor is referred to as a scalar and a first order tensor is referred to as a vector, etc. Since our efforts are focused on the solution of PDE's over domains, this document assumes the tensor is defined over the dimensions of the spatial domain which is assumed to be a physical space. Space/time domains will also be considered. The use of more general spaces over which to tensors is also possible but only space/time domains are considered here.

In our case where a tensor is defined in physical space, the spatial domain of the tensor is  $\Omega$  and if it exists, the temporal domain of the tensor is  $\Upsilon$ . Knowledge of the order of a tensor, n, and the dimension, d, of the spatial domain it is defined over, defines the number of components needed to uniquely define the tensor, which is  $d^n$ . The symmetries, for tensors of order 2 or greater, define those components that are identical to, or negative of (antisymmetric), other components. The components of the tensor are in general functions of the domain parameters as well as other problem parameters. The ability to understand and use a tensor at any particular instant requires knowledge of the coordinate system,  $\overline{X}$ , in which the components of the tensor are referred. Tensors can be represented in other coordinate systems of equal or lower order through appropriate coordinate transformations. In the case of transformation to a lower order coordinate

system (e.g., projection of a 3-D stress field onto a model face), the transformation yields a lower order tensor tensor and thus requires the definition of a new field.

Tensors can also have constraints on them. Examples of constraints include boundary conditions applied on the tensor, divergence free properties of the tensor, etc.

## 1.2 Discretization of a Tensor

The discretization of a tensor over a domain is called a field. The field inherits the tensor order and spatial domain dimension from the tensor along with any symmetries and constraints. We are concerned about the case where the spatial domain of the tensor is discretized over is an appropriate set of mesh entities. The field discretizes the tensor component values over the mesh entities in terms of distributions and degrees of freedom (dofs). The distributions are defined over the mesh entities (and temporal discretization entities as needed) and give the variation of the components of the field. The dofs multiply the distributions and set the magnitude of the variation of the individual distributions. The tensor's constraints which limit the variation of the tensor's components are inherited by the field.

# 2. Overall TSTT Field Design

To this point, the discretization of tensors into fields has been independent of any implementations. In this section we begin discussion of the components of the TSTT Field Library.

## 2.1 Field Instances

A complex simulation process can involve a number of fields defined over various portions of the domain of the simulation. A single field can be used by a number of different analysis routines that interact, and the field may be associated with multiple meshes having alternative relationships between them. In addition, different distributions can be used by a field to discretize its associated tensor.

The ability to have a specific tensor defined over multiple meshes and/or discretized in terms of multiple distributions is handled by supporting multiple instances. A field instance has a single set of distributions over a given mesh. These distributions are defined over mesh entities of dimension d, referred to as the *field instance mesh entities*, where d is the same dimension as the tensor. A field instance can exist in an evaluated form where the dof have been determined, or in an unevaluated form where the dof are not yet determined. Ultimately the TSTT field services will be designed to support appropriate operations on both evaluated and unevaluated fields.

# 2.2 TSTT Field Functionalities

There is a wide variation in the nature and complexity of functionalities needed by field services that strongly influences (i) the amount of information the function must be provided, (ii) the amount of information that is produced, and (iii) the computational effort required to execute the operation. The interactions between these influences does not lend itself to an easy categorization that is both logical and provides a natural level of increasing complexity. Thus the categorization provided here for purposes of introducing the types of field functions needed first considers the level of entities at which the fields information is requested which are:

• At specific points in the domain

- Over mesh entities
- Over mesh entity groups to define new field instances or new fields

The information returned by functions requesting information at points will always be limited to values of the components of the field in a requested coordinate system at those locations. Therefore, these functions do not have to explicitly expose the distribution information associated of the field instance being considered. (The distribution functions are used in the process, but the details of the distribution need not be exposed in the functional request.)

Functions that evaluate over mesh entities or groups will often require a more explicit consideration of distribution information. The groups of mesh entities can be defined useinf mesh sets or based on the classification of mesh entities against model entities. Since the functions in the mesh entity group level will create new fields or field instances, they must be driven by a maintained mesh set, or model entities and reverse classification.

# 3. Fields

A field is a representation of a tensor over a domain. The first level of information defining a field level includes:

- *name* which is a unique string (unique to fields only) which is used to identity a specific field.
- *n* which is the tensor order of the field.
- *d* which is the dimension of the spatial domain the field acts on.
- *geometric model* which is an opaque pointer to the geometric model the field is defined over as described in further detail in Section 3.1. If a *geometric model* is not specified, the spatial domain of the tensor is specified at the field instance level through a mesh.
- Y which is the temporal domain the tensor is defined over. This is specified as an initial and a final time.
- symmetries which is the information indicating the symmetries of the tensor.
- *constraints* which is a list of constraints acting on the tensor.
- *field instance list* which is a list of opaque pointers to the field instances (described in Section 4) owned by the field. \*\*\*\* MAY WANT TO START WITH STATING THE NUMBER OF CURRENT INSTANCES \*\*\*\*\*

n and d are static and must be specified during instantiation of a field. A field can have multiple field instances and the number of field instances is dynamic.

## 3.1 Geometric Model

The information which is stored in the geometric model is:

- *model* which is a TSTT Geometry opaque pointer for the spatial domain the field is defined over. If the spatial domain that the field is defined over is all the geometry entities of dimension *d* in *model*, no more information needs to be specified and the *model specification parameters* contains no information.
- *model specification parameters* which is information used to narrow down which *model* entities of dimension *d* are to be used to specify the spatial domain the field is defined over. This information includes:
  - *list of model entities* which is a list of entities of dimension *d* which are in *model* which comprise the spatial domain.

\*\*\*\* WHAT IS CURRENTLY HERE DOES NOT DEAL WELL WITH USING BOTH MODEL CLASSIFICATION AND MESH SETS (EVEN IF THE ONLY THING USED FROM THE MODEL IS AN IDENTIFICATION OF IT.) \*\*\*\*\*\*

# 4. Field Instance

A field instance describes the discretized representation of the tensor over an appropriate set of mesh entities. The information at the field instance level of the library includes:

- *name* which is a unique string which is used to identify a specific field instance.
- *mesh* which is the TSTT Mesh opaque pointer for the mesh the field instance is defined over. The field can be defined over some or all of the mesh entities of dimension *d* in the specified mesh as described in Section 4.1.
- *coordinate system* identifier that will provide access to the information defining the coordinate system the dof values are defined with respect to. If not specified it is assumed to be the coordinate system the *mesh* is defined over. Coordinate systems are discussed in more detail in Section 4.2. Note the distribution functions need not be defined in the same coordinate system.
- *distribution identifier* that will provide access to the first level of information on the field distribution information. The information defined in the distribution is given is Section 4.3
- *order of continuity* which indicates to continuity order of the field over the mesh entities it is defined over.

# 4.1 Mesh Entities the Field is Defined Over

In the case when the field is referenced to a geometric model, the field instance is defined over the d dimensional mesh entities classified on the indicated model entities. If no model is specified that mesh parameter indicates a TSTT mesh set which must contain d dimensional mesh entities and/or other mesh sets containing d dimensional mesh entities.

\*\*\*\* NOTE - We can define a broader set of options here, what is defined now is a minimal set that can meet the needs of those that prefer mesh based and those that prefer model based methods of interactions. It is not clear that a broader set would be helpful, or instead would lead to confusion. This is an area TSTT needs to discuss.

# 4.2 Coordinate Systems

A key aspect of interacting with tensor fields has to do with dealing with multiple coordinate systems and the transformations, including reductions, performed on them. Thus we will need to add an appropriate mechanism for defining coordinate systems to meet the needs of these processes.

The top level of information in a coordinate system is:

- *type* which is the type of coordinate system. Typical examples are Cartesian, cylindrical, parametric, etc.
- *association* which gives the information on what the coordinate system is associated with. Typical examples include *model* and *mesh* as defined above, as well as with specific mesh or model entities. Setting *association* automatically sets *dimension* which is the spatial dimension of the coordinate system which is the same dimension as its associated object.
- ...more stuff

\*\*\*\* NEED TO HAVE SOMEONE WORK THIS PROPERLY. ISSUES:

- EXACTLY WHAT INFORMATION IS NEEDED TO DEFINE A COORDINATE SYSTEMS AND ITS TRANSFORMATION. DOES THIS REQUIRE THERE IS ONE MASTER ONE.
- NEED TO DEAL WITH CURVILINEAR ONES. DOES THIS MEAN DEALING WITH CO- AND CONTRA-VARIENT COORDINATE SYSTEMS.
- HAVE TO LINK FROM HIGHER TO LOWER ORDER ONES SURFACE COORDINATES.
- NOTE THE DISTRIBUTIONS AND FIELDS CAN BE IN DIFFERENT COORDINATE SYSTEMS

#### 4.3 Distributions

The are a wide variety of distribution functions and associated dof that can be used in the definition of fields. It is possible to define a generalization that can be used to represent all possible situations. However, such a generalization would introduce a substantial overhead both in terms efficiency and complexity and would not effectively meet the needs of common cases such as fixed finite difference stencils and fixed order finite elements. Therefore, the TSTT field library will support multiple forms for the definition of the distribution functions and dof. Although this approach will allow greater efficiency of operation and implementation in the common cases, it will require the introduction of multiple forms (which can indude a general one). It will also introduce added complexity in implementation of such functions will be less straight forward. These complications will be reduced, to an extend, by defining the distribution information in levels.

\*\*\*\*\* NOTE TO TSTT TEAM - THIS IS THE AREA WHERE MOST OF THE COMPLICATION IS AND WHERE WE HAVE GONE THOUGH MULTIPLE OPTIONS AND STILL HAVE NOT FOUND ONE THAT APPEARS TO BE OPTIMAL TO ALL. \*\*\*\*

#### 4.3.1 Relationships that can be used for General Distribution Specification

Before defining the specifics of the how to define the distribution functions and their dof, a couple of properties of distribution functions and dof are stated for the case of meshbased fields. The distribution function can be associated with a key mesh entity while the dof can be associated with a single mesh entity. Rules defined in terms of mesh adjacency information can provide all the information on the full set of mesh entities the distribution acts over as well as the mesh entities that have dof associated with them.

**Finite difference based or a vertex stencil.** In finite difference methods the distribution functions are difference stencils typically written in terms of *dofs* that are associated with vertices on the closure of the mesh entities the distribution are defined over. In this case the distribution function is the difference stencil which acts over the *d*-dimensional mesh entities the mesh vertex associated with the dof bounds.

Finite elements with common *dof* between neighboring mesh entities. Finite element distribution functions, referred to as shape functions, are written over individual mesh entities, referred to as elements. In cases where  $C^m$ ,  $m \ge 0$  continuity is required, the element level shape functions associated with neighboring elements are made  $C^m$ ,  $m \ge 0$  continuous by having common *dofs* associated with the bounding mesh entities that are

common to the neighboring elements. For example, a  $C^0$  field between two neighboring quadratic face finite elements can be constructed by sharing three *dof* associated with the closure of the common edge where two of *dofs* are associated with the vertices bounding that edge, and one is associate with a point on the edge. In the case interpolating  $C^0$  shape functions the full set of *dofs* used by the element distribution are the *dofs* associated with the mesh entities in the closure of the *d*-dimensional mesh entities. There are other means to meet such continuity requirements to even higher order. In each of these cases there are specific rules on the sharing of *dof* on the common boundaries (unless one wishes to deal with expensive constraint equations, etc.).

Finite volume methods using discontinuous functions. Finite volume methods are constructed in terms of distribution function written over individual *d*-dimensional mesh entities, referred to as cells. In most cases the field being defined is  $C^{-1}$  and the *dof* are associated with the *d*-dimensional mesh entities the distribution is defined over.

Although these relationships can be used to define one uniform means to define the distribution functions and dof, it would be too costly for the some common cases. The compromised approach taken below is to define a set of standard methods that effectively meet the needs of the standard cases and have one general case that can be used for any other.

#### 4.3.2 Information Qualifying the Distribution

Since the detailed information defining a field instance distribution will not be defined in the same manner for all distribution types, the first component of the definition of a field instance is a set of parameters that will identify the type of field distribution be to used. These parameters will indicate:

- The key mesh entities for the distribution functions. Options include:
  - The *d*-dimension mesh entities the distribution is defined over (as in standard finite elements when looking at it from an element shape function perspective).
  - Specific *d*-dimensional mesh entities bounding the mesh entity identified as the key to the distribution (e.g, mesh vertices for finite difference stencils for which the distribution acts over the *d*-dimensional mesh entities that bound it, or the mesh entities associated with the finite elements nodes when viewing the finite element shape functions form the global level instead of the element level).
  - non-mesh entities, just a given set of points.
  - more general sets of mesh adjacencies as would be needed to represent spline functions of various types (see the recnt work of T.J.R. Hughes which intriduces a wide number of possibilites.
- The distribution form information. Here we need a means to support a set of known methods and then allow a means for people to add more to support those developing new methods (it is not only academics working on spectral, p-version, DG and spline type methods that are introducing new distribution functions. SciDAC people regularly do it Steve Jardin for example). Distribution forms to be included are:
  - Common FD stencils. (Are there complications here with respect to dealing with the stencils to take care of the various boundary conditions?)
  - Lagrange polynomial finite elements
  - Standard finite volume functions

- p-version finite element and spectral elements (may not be so simple in that there are multiple forms of shape function used here)
- Spline type functions

(This part can be even nastier than indicated so far. For example, the shape functions for 4-noded finite elements in structures finite element codes are not really bi-linear Lagrange shape functions. They include specific condensed quadratic terms and reduced integration is used on specific terms, etc. The vendors do not share those details and even if they did, it is not clear we would know how to properly use it in dealing with the fields.)

\*\*\*\* EUNYOUNG - I WOULD LIKE TO THINK ABOUT AN EFFECTIVE WAY TO IMPLEMENT THIS WERE USERS CAN EASLIY STICK THEIRS IN \*\*\*\*\*\*

- Specification of the coordinate system the distribution is written in. It is quite common that the coordinate system the field in represented in, which defines the coordinate system the components of the dof are in, is different from the coordinate system the distribution functions are written in. The most common example of this would be finite element shape functions that are typically written in local parametric (curvilinear) coordinates defined over the *d*-dimensional mesh entities while the dof are in a global coordinate system.
- Indicate if the distributions functions are the same for all components of the tensor or independent sets of distribution functions for each component. When independent distribution functions are used for each component, the tensor component that the distribution function is used to discretize must be specified.
- Indication if the distribution has been evaluated or not, that is have the values of the dof associated with the dof been evaluated, or are they yet to be evaluated.

#### 4.3.3 The dof

The parameters qualifying the distribution functions do fix aspects of the dof. However, even within that there are multiple options as to how the dof are related to mesh and distribution information and how they are to be grouped for accessing the information.

As indicated previously, each dof is uniquely associated with mesh entities (not that we always explicitly store such a relationship). Options include:

- The *d*-dimension mesh entities they are defined on (as in some FV and DG methods)
- Specific order mesh entities in the closure of the mesh entities the field is defined over (e.g, mesh vertices for finite difference stencils or  $C^0$  finite element entities.
- \*\*\*\* EFFECTIVE MEANS FOR LAGRANGE FINITE ELEMENTS IE NODES WE HAVE TO COMPLETE THE NODE THING IN THE MESH FIRST.
- All mesh entities within the closure of the d-dimensional mesh entity (e.g., as in the case of  $C^0$  p-version finite element mesh methods)
- non mesh entities, just a given set of points.

The options for storing the dof range from light weight storage of ordered vectors, or arrays, of dof to the use of dof objects. In a light weight storage schemes knowledge of the distribution and dof qualification parameters, and the ordering used provide the desired information. Examples include:

• A vector of degrees of freedom in a scalar field case or a "matrix" of dof in the case of higher order tensor fields. The vector and matrix analogy is not to imply that is the specific required implement, but provides a simple means to explain grouping and

ordering. Take the simple case where the dof are associated with mesh vertices, the row number could correspond to the vertex and the columns could hold the values of the dof for the components of the tensor at that point. This approach can easily used for a number of the common forms including those where every dof holder has the same number of dof associated with it, and the dof are one type of mesh entity or other entity type like nodes for high order Lagrange finite element.

- Degree of freedom objects with one dof per object, or the set of degrees of freedom for all the components needed. Each object could include:
  - Mesh entity the dof is associated with.
  - Coordinates of the point in space the dof is associate with when the dof represents the value of a quantity, or some derivative of it, at that location.
  - Value of the dof.
  - Value type which specifies the type of number that value is. Options for this include integer, real, double and complex.
  - *State* which specifies the state of the dof. Options for this include: (i) *unevaluated* where *value* has not yet been set, (ii) *fixed* where *value* is specified and cannot be changed, or (iii) *free* where *value* is specified and can be changed.
  - In the case where there is one dof per object, the tensor component which specifies which component of the tensor the dof and its corresponding distribution contributes to. Alternatively, the dof object could the set of dof used to define the components of a tensor quantity when the same distribution function is used
  - *Tag* which is user specified information that can be attached to a dof (identical to the tags specified in the TSTT Mesh interface). An example of its usefulness is by attaching an integer tag which would be used to order the dof *values* in an array as is often done during equation solving.

A form of dof object has been found to be advantageous for things like variable p-version finite elements. In this case the relationship to the mesh entity is heavily used to reduce some of the overhead of having both mesh entity and dof objects. The dof object approach is also advantageous for where there are unequal numbers of dof on a particular mesh entity.

# \*\*\*\*\*\* WHAT IS HERE NOW ARE TWO MORE OR LESS THE TWO LIMITS OF DEALING WITH DOF - IS THERE A NEED FOR ANYTHING IN-BETWEEN?

#### 4.3.4 Relationship between distributions and dof

The qualification of the relationship between the distribution function and the dof is strongly influenced by the manner used to define the dof. In the cases where the distribution function is the same over every *d*-dimensional mesh entity in the field instance, using a vector or matrix for storing the dof coupled with the rules implied by the field instance qualification parameters associated with distribution and dof, plus mesh connectivity information (or equivalent) can be used to understand the relationship of distribution over each mesh entity and the dof. If on the other hand, the distribution functions can vary from mesh entity to mesh entity, then using the combination of the mesh topology and dof objects in conjunction with the field instance qualification parameters can effectively deal with the process.

# 5. Operations Performed on Fields

## 5.1 Pointwise Interrogation Functions

A pointwise interrogation is a request to evaluate the field at one or more specific locations in the domain. Included in pointwise request are requests for various order derivatives of the field at the point.

The points of the field is to be evaluated at is required input for all the pointwise functions as well and an indication of which field and instance of that field is being referred to. Since types of operations required to do this determination are a strong function of what is known about the points, the input to the function should also provide information useful for differentiating the cases. For example at one extreme, the field is known to be an interpolating field and the desired evaluation point are known to be at the interpolating points, in which case all that has to be returned are the appropriate dof that define the components of the tensor field. At the other extreme the only thing known about the point is its coordinates. In this case the element the point is associated must be determined (requires time proportional to the log of the number of elements), the coordinates of the point in the element coordinates that the distribution is written must be found (typically required non-linear iteration). The other two common cases are when one knows which mesh entity the point is associated with, this saving the search, or there is an ordered set of point where the consecutive points are in the neighborhood of the other (thus indicating adjacency based searches to find elements would be efficient).

The output is the values of the components of the tensor, or derivatives, at the points.

The effective implementation will require either the definition of a single function that is provided the full set of input for the most general case, or a set of functions based on the knowledge of the class of points provided. Since the algorithms used for the various cases vary dramatically, a set of functions will be defined.

To evaluate a field at point the mimimal input is:

- the location the point
- the coordinate systems the point is defined in
- the field to be evaluated
- the coordinate systems the field is to be evaluated in and the components of the field to be evaluated
- the field instance it is to be evaluated for.

Additional information that would be useful making the process efficient:

- knowledge if the evaluation is at an existing interpolating point (in which case the dof define the field in the coodrinate system of the dof
- knowledge of which mesh entity it is in
- knowledge if the point is "topologically" near the previous evaluation point
- knowledge that the coordinate systems the is given in is the same as that the distribution is defined in.

# \*\*\*\*\* NEED TO START DEFINING A SET OF FUNCTIONS AND OPTIONS FOR THEIR DEFINITION \*\*\*

## 5.2 Mesh Entity Interrogation Functions

The most common mesh entity level interrogation is to request some integral measure of the field, or integral of its derivatives, where the integration is carried out over the mesh entity. These requests can range for determination of specific norms used in error estimation to the integration of a quantity such as the pressure over the mesh faces classified on a model face to calculate lift.

\*\*\* SHOULD NOT BE TOO HARD TO GENERALIZE THE INTEGRATION PROCESS - OTHER STUFF MAY BE HARDER \*\*\*\*

### 5.3 Field Coordinate Transformations

It is quite common that multiple coordinate systems are used in the process of dealing with the set of fields in a simulation. In those cases the field may is to be represented in multiple coordinate systems over the d-dimensional entities it is defined over. In this case, the transformation involves the application of a standard coordinate transformation and yields a new field instance (if both are to be maintained).

## 5.4 Field Reductions

A common field reduction is to request the representation of a field defined over a set of lower entities on the closure of the d-dimensional mesh entities the field is defined on. A common example of this is requesting tractions on a model or mesh face due to the stresses over the model or mesh region the face bounds. In this case, a new field is defined since the dimension of the mesh entities it is defined over is lower and the number of components is less. In the case of the tractions on a face based on the stresses in a region the face bounds there is a reduction of one in the field dimension with a tensor being reduced to a vector.

## 5.5 Field Modifications

Their are a number of operations that can be performed that modify the field. Many of them maintain the characteristics of the tensor the field is representing and can this be represented as a new instance of the field. Some common examples include:

- Introducing alternative distribution functions over the same mesh and representing the field with those new distribution functions. A very common specific example of this type is defining an L<sub>2</sub> project of a C<sup>-1</sup> field onto a set of C<sup>0</sup> distribution functions defined over the same d-dimensional mesh entities.
- Transferring the field from one mesh to another using the same or different distribution functions. This is commonly done in multiphysics analysis when a solution field for one analysis is used as a forcing function in another analysis. Another case would be in adaptive analysis where history dependent fields have to be transferred from between meshes (either entirely new meshes, or locally modified meshes).

# 6. References

- [1] Beju, I., Soos, E. and Teodorescu, P.P., *Euclidean Tensor Calculus with Applications*, Abacus Press, 1983.
- [2] Niu, Q. and Shephard, M.S., "Transfer of solution variables between finite element meshes", SCOREC report 4-1990, Scientific Computation Research Center, RPI, Troy, NY, 1990.

- [3] O'Bara, R.M., Beall, M.W., and Shephard, M.S., "Attribute management system for engineering analysis", *Engineering with Computers*, 18:339-351, 2002.
- [4] Shephard, M.S., Fischer, P., Chand, K.K. and Flaherty, J.E., "Simulation Information Structures" TSTT, 2003.
- [5] 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.
- [6] Chand, C., Fix, B., Dahlgren, T., Freitag Diachin, L., Li, X., Olliveir-Gooch, C., Seol, E., Shephard, M., Tautges, T. and Trease, H., "The TSTT Mesh Interface, Version 0.6" http://www.tstt-scidac.org/software/software.html.
- [7] Beall, M.W. and Shephard, M.S., "A General Topology-Based Mesh Data Structure", *Int. J. Num. Meth. Engng.*, 40(9):1573-1596, 1997.
- [8] Weiler, K.J., "The radial-edge structure: A topological representation for nonmanifold geometric boundary representations", M.J. Wozny, H.W. McLaughlin, J.L. Encarnacao, editors. *Geometric modeling for CAD applications*, North Holland, pp. 3-36, 1988.
- [9] Wan, J., Kocak, S, Shephard, M.S. and Mika, D., Automated adaptive forming simulations", submitted to 12th Int. Meshing Roundtable, 2003.
- [10] Nedelec, J.C., Mixed Finite Elements in R<sup>3</sup>, Numerische Mathematik, 35:315-341, 1980.
- [11] Silvestri, P.P. and Ferrari, R.L., *Finite Elements for Electrical Engineers, 3rd, Ed.*, Cambridge University Press, 1996.
- [12] Shephard, M.S., Tautges, T.T., Freitag Diachin, L. and Seole, E., TSTT Geometry Interface, V 0.6, *in preparation*.