The APF Library

Dan Ibanez

March 27, 2014

1 Overview

The Attached Parallel Field library is a C++ library that supports the manipulation of finite element fields. It supports a wide variety of finite element basis functions and numerical integration rules. Field storage is provided for multiple mesh databases, including support for frequent migration and modification of the mesh. Some common finite element field operations are implemented in the core library, while large or user-specific libraries are included, called modules.

2 Notation

$f^e(\xi)$	The value of a field f in mesh element e at local coordinates ξ
$W(e)$	The set of multipliers associated with mesh entity e
$W(\bar{e})$	The set of multipliers affecting element e
w_a	The distribution function of multiplier a
f_a	The value of field f at multiplier a
L_t^p	The set of fields distributing t-order tensors using Lagrange polynomials of order p
S_t^d	The set of fields distributing t-order tensors step-wise over elements of dimension d

3 Field

Each field represents a distribution of a tensor value over a domain. A field is defined by a independent tensor type, distribution, and domain.

4 Tensor

Field values are by definition tensors, and any tensor types required by finite element applications are potentially within the scope of APF. At the moment three types of tensor are supported and distinguished since they are those most used in target applications:

name	order	space
scalar	0	$\mathbb{R}$
vector	1	$\mathbb{R}^3$
matrix	2	$\mathbb{R}^3 \times \mathbb{R}^3$

5 Distribution

There are many kinds of distributions available just within the context of finite element meshes. APF's support is initially restricted to the class of weighted-sum element-local distributions. Specifically, APF supports fields such that

$$f^e(\xi) = \sum_{a \in W(\bar{e})} w_a(\xi) \cdot f_a$$

Where the affecting multipliers $W(\bar{e})$ are those associated with entities on the closure of e, Thus:

$$W(\bar{e}) = \cup_{c \in \bar{e}} W(c)$$

Each multiplier is associated with one mesh entity, and can affect multiple elements. While the distribution function of a multiplier depends on the element, its reference value f_a does not.

5.1 Lagrange

The first distributions supported by APF are first and second order Lagrangian shape function over simplex elements. These distributions are well described in other literature. This document offers a notation for a field which distributes a t-order tensor using p-order Lagrange polynomials:

$$f \in L_t^p$$

5.2 Step-wise

When dealing with derivative quantities of fields, it becomes convenient to define fields with less-than-linear polynomial order. Such fields have C^{-1} continuity between elements and are constant within an element. The element distribution function, in this case, is the higher-dimensional analogue of a step function. Technically these fields fit the description for an APF distribution:

$$f^e(\xi) = f_0 = \sum_{a \in W(\bar{e})} w_a(\xi) \cdot f_a$$

$$|W(\bar{e})| = 1, w_0(\xi) = 1$$

A Lagrange polynomial distribution over a face is also well-defined over its edges, but step-wise distributions are valid only within mesh elements and discontinuous at all adjacent entities of lower dimension. Thus the defining parameter in a step-wise distribution is the dimension d of elements. From this the notation for a field distributing a t-order tensor over elements of dimension d could be written as follows:

$$f \in S_t^d$$

6 The Coordinate Field

Coordinates are treated as a separate field $x^e(\xi)$, whose evaluation defines a mapping and whose local gradient defines the Jacobian.

$$J^e(\xi) = x^e_{,\xi}(\xi) = \sum_{a \in W(\bar{e})} w_{a,\xi}(\xi) \otimes x_a$$

The coordinate field has several conditions that make it special:

1. Each mesh has one and only one coordinate field
2. All other fields query the coordinate field
3. Coordinates are stored implicitly in the mesh database

The first condition requires a tight coupling between the mesh and the coordinate field: that field must be retrievable given the mesh, and irreplaceable over the mesh lifetime. If the coordinate field were treated like all other fields, the second condition would cause a problematic infinite loop of queries. Lastly, the final condition makes implementation troublesome since the mesh database is not purely topological and coordinates cannot be handled entirely by the field library. For practical reasons, this will likely remain the case for some time.

7 Operators

The core of APF provides a set of general operators for the different supported fields. Operators which are more application-specific will be organized into modules that make use of the core. The following core operators are available in the first release:

7.1 Gradient

The gradient of a field f at a point ξ is a tensor describing the partial derivatives of the field along each spatial dimension:

$$\mathrm{grad}(f) = \nabla \otimes f = f_{,x}$$

The formula in terms of distribution functions for computing a scalar gradient is:

$$(\nabla \otimes f)^e(\xi) = \sum_{a \in W(\bar{e})} w_{a,x}(\xi) \otimes f_a$$

Where $w_{a,x}$ is the gradient of multiplier a's distribution function with respect to global spatial coordinates. These, in turn, are computed from the local gradients of f and the Jacobian given by the coordinate field:

$$w_{a,x} = J^{-1} w_{a,\xi}$$

7.2 Divergence

Given a vector field, the divergence describes how much outward flow occurs from each point in the field:

$$\mathrm{div}(f) = \nabla \cdot f = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z}$$

The divergence can be computed from multipliers as follows:

$$(\nabla \cdot f)^e(\xi) = \sum_{a \in W(\bar{e}} w_{a,x}(\xi) \cdot f_a$$

7.3 Curl

The curl of a vector field describes the twist or spin of the field around a particular point. Curl is a vector along the axis of spin whose magnitude indicates the amount of spin.

$$(\nabla \times f)^e(\xi) = \sum_{a \in W(\bar{e})} w_{a,x}(\xi) \times f_a$$

8 Implementation

8.1 Linear Algebra

APF provides classes representing vectors and matrices. A study of other libraries shows that many of them implemented similar classes out of a universal need for linear algebra in scientific computing. APF's classes aim to be good enough that any package can use them, and are provided publicly along with the API.

8.2 Mesh

APF maintains an abstract mesh interface in the form of a purely virtual base class. This can be implemented separately for each mesh database. All reference values are associated with mesh entities as attached data. The tensor data is packed as efficiently as possible within the constraints of the mesh API and the requirements for adaptivity and field operations. Modules around APF for high-level operations such as error estimation can use the mesh interface and gain portability across mesh databases.

The APF mesh object does not have to represent the entire mesh; it could provide a clean solution to fields defined over subsets of the mesh. Once this extension becomes necessary, we may rename these objects Domains...

8.3 Mesh Element

The APF Mesh Element object represents the coordinate field over an element. These objects exist to cache coordinate multiplier data in the case of multiple queries. A Mesh Element provides the following information:

1. Local to global coordinate mapping

2. Jacobian at a point

3. Differential volume (Jacobian determinant) at a point

4. Integration points given an order of accuracy

An APF Mesh constructs a Mesh Element from a mesh entity at the user's request.

8.4 Field

APF's field objects are created given an APF mesh and tensor type. Different creation functions exist for each family of distribution functions due to the variation in parameters. Upon creation, a field defines itself over the entire given mesh.

8.5 Field Element

A Field Element represents its parent field over a particular element. Once created, the Field Element provides the following:

1. Field evaluation at a point

2. Gradient evaluation at a point

3. Operator evaluation at a point (curl, div, etc.)

Field Elements are created by a field given a Mesh Element. This allows multiple Field Elements (representing different fields defined over the same mesh entity) to share one Mesh Element object for their coordinate field queries.

9 Upcoming Considerations

1. geometric classification and attributes