SC2008 Tutorial Proposal

Interoperable Mesh and Geometry Tools for Advanced Petascale Computing

Presented by
Some subset of the ITAPS Team

Primary Contact: Lori Diachin, diachin2@llnl.gov

Abstract

Advanced simulations need advanced software tools to manage the complexities associated with sophisticated geometry, mesh, and field manipulation tasks, particularly as computer architectures move toward the petascale. The Center for Interoperable Technologies for Advanced Petascale Simulations (ITAPS) is creating interoperable and interchangeable mesh, geometry, and field manipulation services that are of direct use to applications. The premise of our technology is that such services can be provided as libraries that can be used with minimal intrusion into application codes. In this tutorial, we give an overview and introduction to the ITAPS philosophy as well as several examples of how our technology has been used to impact existing application codes. We briefly describe several services that are available to application scientists including dynamic partitioning, mesh adaptation, mesh quality improvement, and front tracking. Underlying these services are a common data model and interfaces that are key to providing uniform access to all ITAPS tools. We will teach attendees basic use of the ITAPS interfaces through simple examples and provide advice on best practices for use of ITAPS software. We conclude the tutorial with a description of how to build your own ITAPS compliant implementation and detailed information on obtaining ITAPS software.

Description

Goals and Target Audience. The primary goal for this tutorial is to introduce application scientists to the tools and libraries that are available for advanced mesh, geometry, and field manipulations on large-scale parallel computers. These scientists will generally have an application code in hand along with the realization that advanced tools such as adaptive mesh refinement or front tracking are required to help them reach their scientific goals. The first part of the tutorial will introduce the audience to the wide variety of available software libraries that provide such services and include examples of their use in application codes. The second part of the tutorial will target application scientists who are interested in more details of the ITAPS philosophy and who may be ready to try one or more of the available tools. The detailed discussion of ITAPS interfaces and the hands-on exercises are aimed primarily at this group, but will also give those new to ITAPS technologies a clear view of what's involved in writing, building, and using these packages.

Content Level. 30% beginner, 45% intermediate, 25% advanced

Audience Prerequisites. Attendees should have familiarity with the numerical solution of PDEs. Examples will be shown in various programming languages, including Fortran, C, and C++ so specific language background should not matter. To participate in the hands-on portion, a unix-based computer with network connectivity is required to connect to the off-site tutorial server to download and build the basic examples. These examples will also be projected on the front screen for those without computer or network access.

Duration. Full day is preferred, although this tutorial can be given as either a half or full day. If selected for a half day tutorial, we will reduce focus on the introductory material and provide examples of the ITAPS interfaces and their use in applications. The hands-on portion of the tutorial will be eliminated.

Relevance. The advent of petascale computing will enable increasingly complex, realistic simulations of PDE-based applications. Numerous software tools are available to help manage the complexity of these simulations, including computer-aided design systems used to represent the geometry of the computational domain, mesh generation tools to discretize those domains, solution adaptive methods (AMR) to improve the accuracy and efficiency of simulation techniques, and parallel tools such as dynamic partitioning to ease implementation on today's computer architectures. The ITAPS center focuses on providing tools and technologies to increase the levels of interoperability of mesh-based methods for the analysis of PDEs and to fill specific technology gaps needed to increase the level of automation and reliability of these simulations. We believe the topic of advanced tools that enable PDE simulation and that will improve application scientist productivity is of significant interest to many SC conference attendees. There are already many examples of application scientists leveraging the technologies that ITAPS provides to increase their simulation accuracy, allow them to operate more effectively on complex computational domains, or reduce the total time to solution. Our tools have been used in a wide variety of applications ranging from accelerator and fusion modeling to nuclear reactor and groundwater flow simulations.

Brief Description of Tutorial.

The tutorial is divided into 6 basic modules as described below:

1. *Overview and introduction:* Introduces students to the need for advanced mesh and geometry libraries for petascale simulation and provides example that motivate their use. We also introduce the ITAPS philosophy; namely that such services can be provided in an interoperable way that provides maximum flexibility to the application programmer. We compare the ITAPS approach with other commonly used software development methods.

2. *ITAPS data model:* In this part of the tutorial we introduce the ITAPS data model and its three core data types: mesh geometry, and fields. This abstract data model is based on information flow in PDE simulations and supports a wide array of technologies and a broad spectrum of usage scenarios. We also describe how the core data types associate with each other through the concept of data relation managers and the building blocks within the data types: namely entities, entity sets and tags.

3. *ITAPS services and tools:* There are two basic models for use of ITAPS tools in applications and we give a high level description of each. We describe several mesh and geometry services that are available for use including dynamic partitioning, mesh adaptation, front tracking, and mesh quality improvement. In each case we give examples of how applications can benefit from their use.

4. *ITAPS interfaces:* We describe the design philosophy for the development of the ITAPS interoperable tools, a key component of which is that we do not enforce any particular data structure or implementation with our interfaces, requiring only that certain questions about the geometry, mesh, or field data can be answered through calls to a common interface. We provide examples of accessing basic information through these interfaces using both arrays and iterators, how to modify the underlying database, and what is needed to support parallel computations. Finally, we discuss the design decisions made to support language interoperability through both a C interface (which interoperates with C, C++, and Fortran) and through SIDL/Babel (which provides additional interoperability with Java and Python).

5. *Using ITAPS: Approaches & Experience*: We provide several case studies of use of ITAPS interfaces in advanced application settings. The first focuses on the development of a new application for nuclear reactor modeling, the second on the insertion of ITAPS mesh adaptation software into a fusion simulation, and the third on the use of several ITAPS tools in concert for shape optimization of accelerator cavities. Based on these and other applications, we provide a list of 'best practices' for use of ITAPS interfaces and software.

6. *ITAPS software:* We provide the necessary information for students to download, use and build ITAPS implementations and tools. We give the status of the available software and show examples of developing and testing a user-specific ITAPS implementation.

Hands-on exercises. A series of individual exercises utilizing ITAPS interfaces and tools in simple PDE simulations, described in detail below.

Coordination of the Presentation. The ITAPS team has been working together since July of 2001 and has been working to develop tutorial materials since November of 2005. Since that time we have given two half day tutorials at the CCA quarterly meeting (Jan 2006) and the SciDAC 2007 conference (July 2007), and plan to give two additional half day tutorials prior to SC08 at the SciDAC 2008 meeting (July 2008) and the ACTS toolkit meeting (Aug 2008). During this period, we have developed slide materials that are consistent across different presenters and use an informal review process by the entire ITAPS team of all newly developed materials prior to use. All slides follow the same template for visual consistency, and are presented as a single file from one laptop so that there are no interruptions due to switching of presentations or computers. For each tutorial given to date, we have requested and received feedback and refined our material to reflect this input.

Development Plans. Our tutorial continues to evolve as new technologies and interfaces are defined by various ITAPS working groups. The most substantial addition for SC08 will be a detailed description of the interfaces needed for parallel mesh manipulation services. Our hands on exercises will be updated to reflect these additions as well. In the description of ITAPS services, we will focus more heavily on their use in distributed memory environments as well as their performance on large-scale machines.

Description of Hands-On Exercises

Content. Roughly two hours of the tutorial is devoted to a series of in depth examples and hands-on exercises in through which the participants will gain a good understanding of the ITAPS data model, interface design philosophy and use of ITAPS implementations and tools in basic applications. Initially, participants will assemble and build a simple "Hello ITAPS" application using pre-defined, pre-built implementation. This will ensure they understand the basic structure of an ITAPS application and will provide the building block for later exercises. Once this is working correctly, students will be able to develop a simple driver application that accesses simple mesh information such as vertex coordinate locations and element adjacency information using both array and iterator access. More advanced exercises include the use of the data relations manager interfaces to work with geometry and meshes together, the use of the parallel interfaces to access off-processor information, and an example use of one of the ITAPS services (for SC08 we will focus on mesh partitioning). We do not expect students to complete all of these exercises in the time given, but rather to focus on the one or two of most interest to them. Finally, for students interested in creating their own ITAPS implementation to take advantage of ITAPS tools for their application, we provide a framework and step by step instructions for building and testing the compliance of the implementation. The students are encouraged to work at their own pace, and to skip around to exercises that particularly interest them. A written "Hands-on Guide" will provide detailed, step-by-step instructions for each exercise. The planned exercises are as follows:

1) Assembling an ITAPS application "Hello ITAPS" (from pre-built components)
2) Accessing basic mesh information: vertex coordinate information
3) Accessing basic mesh information: element adjacency information
4) Using meshes and geometry together: data relations manager
5) Accessing off processor information in parallel meshes
6) Using an ITAPS service (partitioning a mesh)
7) Creating an ITAPS implementation and compliance testing

Development Plans for SC2007. Many of these exercises have been provided in the past as in depth examples as part of the lecture portion of the ITAPS tutorial. For SC08, we will develop the associated software modules and provide written documentation in the "Hands-on Guide".

Presentation Approach. The most effective approach to a hands-on session is to provide the students with a complete set of written instructions and let them work at their own pace. Instructors roam the room, both answering questions and simply checking on student progress to insure that they're not getting stuck. Some students will naturally form small groups (2-3 people) to work together on the exercises, while others will work on their own. Students can pick and choose among the exercises based on their level of experience and interests.

Facilities. To make this work we need to provide some sort of pre-built set of tools on a remote system – can anyone provide guest accounts for tutorial attendees?

Detailed Outline of Tutorial

1. Overview and introduction *(lecture format)*
    a. The need for advanced mesh and geometry libraries
    b. Motivating examples
    c. Overview of the ITAPS philosophy
    d. Comparison with other development approaches
2. ITAPS data model *(lecture format)*
    a. Core data types: Mesh, Geometry and Field
    b. Basic Building blocks: Entities, Entity Sets, Tags
    c. Managing the relationships between core data types
    d. Parallelism in the ITAPS data model
3. ITAPS services and tools *(lecture format)*
    a. Two models for ITAPS use in applications
    b. Available ITAPS services (dynamic partitioning, mesh adaptation, front tracking, mesh quality improvement)
    c. Examples of impact on applications
4. ITAPS interfaces *(lecture format)*
    a. Design philosophy and basic tenets
    b. Accessing information using arrays and iterators
    c. Modifying the underlying database
    d. Parallel operations
    e. Language interoperability through the C interface and through SIDL/Babel
5. Using ITAPS: Approaches & Experience *(lecture format)*
    a. Case Study 1: New application for nuclear reactor modeling
    b. Case Study 2: Inserting adaptive meshing into fusion simulation
    c. Case Study 3: Using several tools in concert for shape optimization of accelerator cavities
    d. Best practices for use of ITAPS software and interfaces
6. ITAPS software *(lecture format)*
    a. Accessing, downloading, and building ITAPS implementations
    b. Accessing, downloading, and building ITAPS services
    c. ITAPS compliance testing
    d. Simple tutorial examples
7. Hands-On Exercises *(individual online exercises)*

    Described in previous section