UCRL-JC-122298 Abs

MapIt: A new software tool to assist in the transition from conceptual model to numeric simulation models.

Thomas W. Canales, Charles W. Grant,
Lawrence Livermore National Laboratory

Lawrence Livermore National Laboratory (LLNL) is a research and development facility owned by the U.S. Department of Energy (DOE) and operated by the University of California. In 1983, LLNL discovered ground water contamination onsite and offsite that led to LLNL Livermore Site being added to the National Priorities (Superfund) List in 1987. The ground water and vadose zone contaminants of primary concern are volatile organic compounds (VOCs), tritium, and gasoline. Since detection of contamination, LLNL has been working to find the best ways to mitigate the problem and always with the goals of doing the job in the shortest amount of time and in the most cost-effective way possible. Restoration activities have centered on a "smart pump and treat" philosophy which includes: detailed characterization (conceptual model building), validated simulation modeling, directed extraction, and adaptive pumping. To facilitate development of the validated simulation models, LLNL has developed MapIt. MapIt is a new software tool that provides remediation planners with the ability to rapidly produce code specific flow and transport modeling input files. MapIt can read a variety of model independent 1, 2 and 3-d data sources, provide interpolation and extrapolation control, as well as extensive mesh editing capabilities. As a result of integrating these features with an easy to use graphical interface, we have achieved clear cost and time savings during initial model development, conceptual refinement, calibration, and porting to new simulation codes.

As part of the site characterization process, the general structure of relevant hydrogeologic features and the distribution of properties within them gradually emerge. Raw data from which this "conceptual model" is developed can include lithology from core samples; borehole geophysical measurements and seismic analysis; soil sample and ground water chemistry; and hydraulic head data from monitor wells and aquifer tests. A conceptual model is constructed and continuously refined as more information from the site characterization effort becomes available. To share this emerging model with others in such a way that some consensus about its nature is achieved, visualizations are produced. These depictions of the conceptual model can include elaborate 2 and 3-d cross section plots, contour plots, time series plots, volume renderings, etc. and can be produced by technical illustration departments or by the wide variety of visualization software that exists for just this purpose (GMS, AVS, Slice, etc.). At the point that numerical simulation of the site is required, planners are confronted with the problem of how to express the current conceptual model in a form compatible with the specific flow and transport code to be used. The transforming, or mapping of the conceptual model to the specific mesh used by the simulation code has typically been, in the case of 2-d models, tedious, prone to error, and very time consuming. In 3-d the problem can be overwhelming. The difficulty of this task is compounded by the fact that the mapping process is often repeated when significant changes are made to the conceptual model, the underlying mesh is changed during calibration or sensitivity analysis, or when it is decided a new simulation code should be used.

A number of fundamental factors contribute to make this mapping process the most time consuming and expensive task incurred during a ground water modeling effort. First, there is typically no electronic representation of the critical features and property distributions of the conceptual model that exist in a managed, mesh independent form. Next, few integrated tools have been developed specifically for expressing such a representation onto arbitrary meshes in a way that allows direct visualization and manipulation. MapIt has been developed to address these issues and consists of three major components; 1.) a hydrogeologic feature database (HFDB), 2.) an object library containing a hierarchical taxonomy of specific simulation model classes; and 3.) a simple to use graphical user interface. MapIt has been developed for use on SGI and Sun work stations using standard C++ and the OSF Motif toolkit.

With MapIt, we introduce the concept of the hydrogeologic feature database (HFDB), a mesh independent, electronic "encoding" of a conceptual model. The HFDB uses an object oriented approach to represent hydrogeologic features such as hydrostratigraphic units, faults, lakes, rivers, wells, plumes, etc. Complex feature objects can be built up from simpler 1, 2, and 3-d object primitives such as points, lines, polylines, surfaces and volumes. For example, a hydrostratigraphic unit can be represented as a composite object composed of top and bottom surfaces, initial head surface, and a series of volume distributions for properties such as hydraulic conductivity, porosity, transmissivity, etc. The conceptual model can thus be stored and maintained in a form free of constraints or assumptions about any one simulation mesh. The source for feature primitives can include simple user provided ASCII files, or files produced by third party products such as EarthVision (.2grd and .3grd files) and GMS. The user can also create features directly through a grid editing tool. MapIt supports the use of multiple HFDB's. A shared HFDB may contain only features all members of the modeling team have agreed represent the conceptual model. Additionally, individuals may also maintain their own HFDB's containing experimental or less agreed upon features. During the process of building a numerical simulation the user can draw upon both HFDB's.

The second component of MapIt is the simulation model (SM) object class library. An aspect of object oriented systems is that objects contain not only the data structures (member data) that define them but the methods (member functions) used to manipulate and operate on those data structures. In a sense, if an object has a "draw" method, it knows how to draw itself. Within MapIt the member data typically stored for SM objects include a detailed description of the mesh geometry, a series of data arrays containing node or element based initial values (referred to as "components"), and available simulation options. Member functions include, read, write, draw, map, and most importantly, the export methods that produce the code specific input files. The SM library makes extensive use of the inheritance mechanism available in most object oriented programming languages like C++. Specific model classes are "derived" from more general parent classes, inheriting all common member data structures and functions. A specific model class such as the "Modflow" class inherit most of its data structures and functions from its more general parent class, in this case "Finite Difference" models, which in turn inherit from the more general class of "Recto-linear Topology" models and so forth. All that need be added to the Modflow class are those data structures and functions unique to Modflow models, such as the routines that produce the Modflow input files. The benefit is that we can create a new instance of the "Finite Difference" class for a similar modeling code, such as MT3D, by simply adding the data and functionality unique to the MT3D code while inheriting the data structures and methods common to all finite difference models. Development time needed to support new modeling codes is greatly reduced.

The third component of MapIt and the part that ties the other two together is the graphical user interface (GUI). The user interface allows the user to view and manipulate both the contents of the HFDB as well as a graphical representation of the simulation model and its associated components. Features in the HFDB are presented on a palette from which a user can, with a mouse click, select a feature for viewing, or by dragging and dropping, associate a feature with an appropriate SM component. A component operation window allows the user to select components and an operation to be performed. Operations include drawing, mapping, editing, and exporting. The components of the simulation model are represented graphically in the component operations window based on the class of the model. For instance, a layered, finite-element based model such as CFEST is represented as a grid of icons where each column represents a layer, and each row represents some property (i.e., K, R, a_t, a_l). The set of properties available is a function of the model options the user has selected. Also associated with each component is the set of user selected interpolation and extrapolation rules to be used during a mapping operation. Once the user has associated (by drag & drop) all necessary features from the HFDB with their corresponding (SM) components and performed the mapping operation, the model specific input files can be automatically generated. If the draw operation is selected, the user can select any combination of layers and properties for full 3-d visualization, including pinchout visualization. A wide variety of viewing controls lets the user pan/zoom, rotate, z-scale, etc. Typically, components such as the IBOUND array in Modflow and MT3D, models or in some cases certain properties, are either generated manually or must be modified during calibration. For this purpose MapIt also allows the user to graphically edit the component data values directly on the mesh through a variety of techniques (paint brushing, polygon assigns, etc.) before exporting.

By providing data mapping, visualization, refinement, and model input file generation in a single integrated tool, MapIt users dramatically reduce the time and effort needed when producing a new model or calibrating an existing model. By using an electronically encoded and conceptual model a modeling team can manage and document the evolution of the conceptual model and the assumptions upon which simulation results are derived.

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.