Environmental Modeling Research at the University of North Carolina at Chapel Hill: Final Report
Grant Number: FG02-02ER63369
C.T. Miller

Abstract

Mechanistic mathematical models of environmental systems are used routinely to assess our understanding of the operative complex processes in nature. As our understanding matures, the complexity of these models increases and so too does the effort required to construct such models. This effort can be person years in some cases, and changes in model formulations or methods frequently leads to the need to either perform major revisions of existing codes or to abandon an existing code and recode the majority of the simulator. This project was intended to be a proof of concept approach aimed at developing a problem solving environment for the development of environmental models. The domain of focus was fluid flow and species transport in subsurface, porous medium systems. An approach was developed in which a mathematical model formulation was specified in LaTeX and this text document was processed, or compiled, multiple times to ultimately result in a computational simulator or model. The DOE developed Common Component Architecture paradigm was leveraged to implement solvers for reactions, integrators, algorithms, and discretization methods. A set of test problems was solved and the overall conclusion of the project is that a problem solving environment to support environmental modeling is certainly within scientific reach at this time.

Nature of Project

This project investigated the design and implementation of a problem-solving environment (PSE) for subsurface flow and transport phenomena. Broadly, the goal of the PSE is to transform a range of questions about subsurface phenomena into numerical simulations. The questions are expressed using a mathematical model, and the simulations must incorporate appropriate algorithms and computational methods to efficiently yield reliable answers.

Subsurface systems are of interest for a number of reasons, including the fact that the Department of Energy (DOE) must assess and manage DOE installations where past waste disposal practices have contaminated the subsurface environment. Computational models do and will continue to play an important role in efforts to characterize subsurface systems, including the management and remediation of hazardous waste sites. More effective and reliable decisions will result as models become increasingly realistic and accessible.

Unfortunately, the ability of computational models to characterize subsurface systems adequately and provide the information necessary for decision-making is limited at best. There are several reasons for the current state of subsurface

modeling, including model complexity, development cost, and the computational demands placed by simulations of realistic multiphase systems. Moreover, the physical models themselves are still evolving as scientific understanding of subsurface systems improves. Numerical methods are also advancing as researchers attempt to address the computational requirements of large-scale simulations.

A PSE would improve the current state of subsurface modeling by accelerating the process of developing simulators, facilitating the introduction of new scientific advancements into models, and encouraging the application of more realistic models. The key to this approach is to combine a high-level problem specification with a framework that is sufficiently flexible to include advanced solution methods suited to large-scale problems, incorporate legacy solvers, and introduce new methods as they become available. A vital requirement is that the PSE architecture be able to produce code that can run efficiently on target platforms.

Scenario problems

Developing a PSE capable of fully realizing the above demands is a difficult task requiring many person-years of effort, and necessitating research on many fronts. As a first step, we focused our efforts on developing a PSE framework flexible enough to produce efficient solvers for a restricted but significant and meaningful set of problems as a proof of concept. We selected several model subsurface problems to guide the development of the PSE. The model scenarios included a range of characteristics and difficulty in terms of the dimensionality, steady and transient behavior, single or coupled equations, linear and nonlinear problems. Using a broad range of physically meaningful problems helped insure that the PSE included a sufficient and robust language for problem specification and was flexible enough to include a set of advanced numerical techniques required for large-scale simulations. Our target scenarios include:

1. Batch sorption using a distributed first-order model.

2. A batch model of the nitrogen cycle, including reactions and interphase mass transfer.

3. Transient groundwater flow in a confined aquifer.

4. Transient, variably saturated groundwater flow.

5. Non-reactive contaminant transport.

6. Nitrogen cycling in the vadose zone, including the solution of both groundwater flow and transport of multiple nitrogen species.

Initial Approach

The specific features and emphases of PSE's vary with the application domain and target audience. However, there are two common approaches that can be used to characterize PSE's. The first is a library-based approach. As its name suggests, a library-based approach provides a simulation environment built on a substantial kernel of existing library routines providing a range of solution methods and functionality. The user then orchestrates the library routines inside the PSE's simulation environment to solve his or her problem.

This approach has the drawback that it does not scale very well: as the range of problems and sophistication of solution methods increases, the interaction between methods becomes more complex and simple compositionality of the library routines is easily lost. Moreover an increasingly large burden is placed on the user of such a system to formulate even simple problems.

On the other hand, a software synthesis, or transformational, approach constructs a simulator directly from the high-level problem description given by the user. Producing a simulator using this approach involves the application of a set of transformations by which algorithms are introduced and data structures are refined from an abstract, high level into a concrete implementation to be executed on a given platform.

To provide a simple example, consider a problem posed as a set of ordinary differential equations (ODE's). Depending on the form of the equations, the most appropriate solver may be an implicit ODE solver, an explicit ODE solver, or a differential algebraic equation (DAE) solver. The library approach would provide each of these solvers, but the PSE user would have to construct the appropriate function arguments. The arguments required are quite different for the three solvers, and must be extracted from the problem description. A synthesis approach automates this process, and may automate (or at least simplify) the selection of the appropriate method, by inspection of the form of the ODE. The synthesis approach is complex, and is only feasible for a limited domain of problems. Thus its applicability will be more restricted than a library-based approach.

Our initial work focused on evaluating the relative merits of a library or transformational approach for developing a PSE. We decided to take a hybrid approach that offers the ability to refine a problem specification from an initial high-level description but is not required to generate the entire set of solution methods as would typically be done in a transformational approach. The key to realizing this goal is to provide sufficiently rich internal representations of the underlying mathematical problem as well as the algorithms and solution methods necessary to achieve a solution. Using an initial prototype, solutions to the first

two model scenarios, as well as the last which includes both multiphase flow and reactive transport were presented at the 2003 SIAM Conference on Computational Science and Engineering.

Second Approach

Our initial prototype relied on Mathematica to provide a high-level mathematical description of a problem, while the solvers were assembled directly from object-oriented C++ libraries. While Mathematica provides a way to express the mathematical models, it is not easy to read. Moreover, our initial focus was on solver construction, and the putative advantages of integrating the simulator with other parts of the problem expressed in Mathematica are a more distant goal.

Consequently our current notation for problem specification is in LaTeX. In analogy to ``literate programming'' approaches in which pieces of a program are incorporated as text in a written document describing the development of the program (and extracted from such a document as input for a compiler), we expect a PSE problem specification to be developed in a written document culminating in the formal equations. Our current working documents follow this approach. Where needed, we use some LaTeX macro definitions to convey semantic information about equations beyond that needed by LaTeX for equation layout.

Thus the starting point for the PSE is a set of differential equations written in a coordinate-free operator form using LaTeX.

Equation analysis: A compiler reads the LaTeX equations and constructs an abstract syntax tree. Within this tree, all definitions are expanded (i.e., occurrences of named functions are replaced with their definitions), and the set of resultant equations is analyzed to determine the following attributes for each equation:

Type: Is the equation algebraic, an ODE, or a PDE?

Order: The highest derivative that occurs in the equation.

Linearity: For each variable that occurs in the equation, whether the equation is linear or non-linear in this variable.

For the set of n equations as a whole, a coupling matrix C is constructed. C_{ij} is nonzero iff a variable derived in equation i is used in equation j.

At this point a knowledge base of rules are applied to the equations and the attributes to determine a set of applicable solution methods. The set of rules must be easily expandable to accommodate new forms of problems and additional solution methods. It must also provide a ranking of applicable solvers. Thus far

we have a very limited set of problems and solvers and our selection method is ad-hoc.

The solvers are named components in a library as explained next. Expressing solvers using the Common Component Architecture: In an effort to provide greater flexibility, we have adopted a component-based approach for assembling numerical solution methods. Component-based approaches such as the Object Management Group's Common Object Request Broker Architecture (CORBA) have been popular for many years in a number of arenas outside scientific computing. In very broad terms, such an approach assembles applications out of existing software elements (components) that interact through well-defined interfaces. The specification of component interfaces is independent of the actual language used to implement the components. Moreover, components interact via a framework provided by a component architecture and so are more loosely coupled than one would commonly find within an object-oriented paradigm.

The component architecture that we are using is targeted toward scientific computing and is under development by the Common Component Architecture (CCA) Forum based out of the DOE's SciDAC program.

The output of the compiler is a script file which CCaffeine uses to assemble the collection of interacting components to solve the problem. Solver components come from a library, but some problem specific components (e.g. specific residual functions or linear systems) are output by the compiler as (C++) components.

Conclusions

Work accomplished on this project was sufficient to demonstrate a proof of concept: we have produced a working, but limited scope, system. We have used a LaTeX formulation coupled with a CCA approach to produce working models of flow and transport phenomena in porous medium systems. The approach is sufficiently robust so that a much more complete system could be built upon the notions advanced in this work.

To be clear, however, the development of a full-featured problem solving environment for environmental modeling would require many additional person years' worth of effort. This is so because of the level of technical detail involved which ranges from compiler construction to methods development and target platform object code generation.

Project Products

Miller, C.T., M.W. Farthing, J.F. Prins, J.F. Kanney, D. Sassan, and H.E. Jeffries (2003) A PSE for Modeling Transport Phenomena in Porous Medium Systems, SIAM Conference on Computational Science and Engineering, San Diego, California, 10--13 February 2003.

Miller, C.T. (2003) Computational Science Research and Education: Perspectives Evolving from Modeling Porous Medium Dynamics, Society for Industrial and Applied Mathematics Workshop on Computational Science, Mathematics, and Engineering, Arlington, Virginia, 23--25 March 2003.

Miller, C.T., J.F. Kanney, M.W. Farthing, H.E. Jeffries, D. Sassen, and J.F. Prins (2003) A PSE for Modeling Transport Phenomena in Porous Medium Systems, Workshop on Simulation and Optimization, Statistical and Applied Mathematical Sciences Institute, Research Triangle Park, North Carolina, 28--30 April 2003.

Farthing, M.W., D. Sassen, J.F. Prins, and C.T. Miller (2004) A Problem Solving Environment for Subsurface Flow and Transport Phenomena, Proceedings of the 15th International Conference on Computational Methods in Water Resources (CMWR XV), 13--17 June 2004, Chapel Hill, North Carolina, Computational Methods in Water Resources, Volume 2, Developments in Water Science 55, Edited by: C.T. Miller, M.W. Farthing, W.G. Gray, and G.F. Pinder, Elsevier Science, Amsterdam, The Netherlands, pp. 1117--1130.

Miller, C.T., C. Abhishek, and M.W. Farthing (2006) A Spatially and Temporally Adaptive Solution of Richards' Equation, Advances in Water Resources, Vol. 29, No. 4, pp. 525--545.

Farthing, M.W., C.E. Kees, T.F. Russell, and C.T. Miller (2006) An ELLAM Approximation for Advective-Dispersive Transport with Nonlinear Sorption, Advances in Water Resources, Vol. 29, No. 5, pp. 657--675.