Separating Software and Science in the Adaptive Hydraulics Suite

Lucas Pettey, Engility Corp
Corey J. Trahan, U.S. Army Engineer Research and Development Center
Gajanan K. Choudhary, The University of Texas at Austin
Clint N. Dawson, The University of Texas at Austin
Gaurav Savant, U.S. Army Engineer Research and Development Center

Key words: software engineering, parallel computing, optimization

The Adaptive Hydraulics (AdH) Suite is a U.S Department of Defense software package for solving the shallow water, transport and groundwater equations. The suite has been developed over several decades and expanded greatly throughout its development cycle. In addition to high fidelity, finite element based modeling of shallow water systems, it supports a host of features vital to most hydraulic and transport-engineering applications, including baroclinic capabilities, surface wave and wind-wave stress coupling, flow through hydraulic structures (weirs, flap gate, etc.) and vessel flow interactions. The package is parallelized through MPI message passing protocols.

Recently, a decision was made to refactor the software and use modern software engineering concepts to streamline future additions and error trapping. One of the driving forces behind the refactoring decision was to allow independent models within the AdH suite to be coupled to one another. The internal model coupling can be either monolithic or via flux, as will be described in this presentation. To this end, a hierarchical data structure consisting of supermodels and submodels was implemented in AdH to separate various aspects of the software and allow developers to concentrate on their areas of interest.

Perhaps one of the most flexible data structures in AdH is the “smpi” structure. The smpi data structure contains all of the necessary variables and methods for communication. Since everything is contained within the data structure, it can be attached anywhere within the supermodel/submodel hierarchy. This allows separate communication between processors working different supermodels, within the same supermodel or within the same submodel. Since the communication software development is separated from the discrete mathematics, a software specialist can attempt new communication protocols without disturbing the finite element engine. Conversely, applied mathematicians can implement new finite element methods without concerning themselves with the parallelization scheme.

By separating the computer science from the applied mathematics through data structures, software optimization can be performed and new features can be added without the fear of breaking unrelated parts of the code. The new AdH suite framework has resulted in performance gains at high processor counts and development modularization, paving the way for future development as AdH looks to move into an open source environment.

CMWR 2018