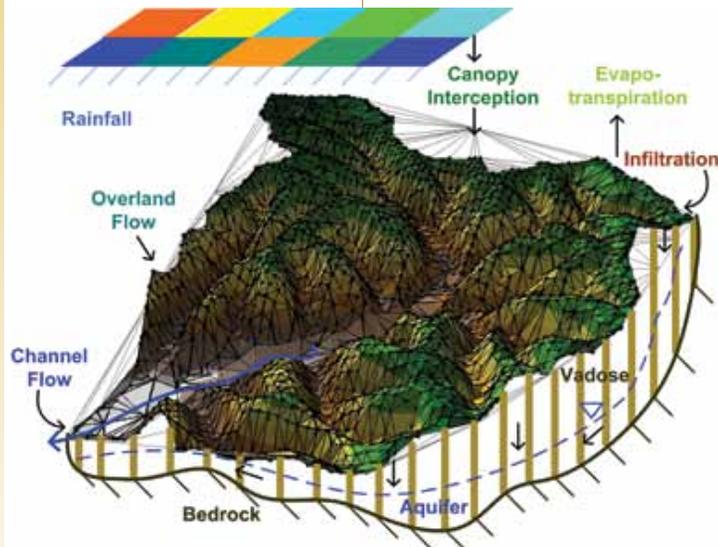


## Subbasin-based Parallel Hydrological Models for Semi-arid Regions

Sue Mniszewski, Patricia Fasel, CCS-3; Everett Springer, STBPO-PRM;  
Enrique Vivoni, Arizona State University; Amanda White, New Mexico Institute  
of Mining and Technology

High-resolution models of river basin hydrology can provide an important tool in assessing the impacts of global change, climate variability, and land-use changes on hydrologic response, leading to a more informed decision-making process about water resources and associated environmental issues. One of the primary obstacles towards advances in high-resolution watershed simulations has been the limited computational capacity available to most models. The growing trend of model complexity, data availability, and physical representation has not been matched by adequate developments in computational efficiency. This situation has created a serious bottleneck that has limited hydrologic models to small domains and short durations.

Fig. 1. The tRIBS model



A novel parallel approach has been applied to the TIN-based Real-Time Integrated Basin Simulator (tRIBS) [1,2], a physically based, distributed hydrologic model originally developed at the Massachusetts Institute of Technology. It is a C++ code that provides continuous simulation of watershed hydrology using an adaptive, multiple resolution representation of complex topography based on a triangulated irregular

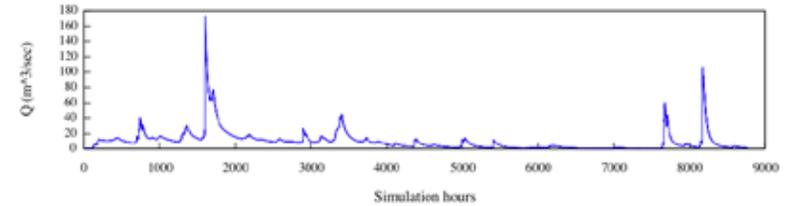


Fig. 2. Baron Fork hydrograph for November 1997–December 1998

network (TIN), modeling processes such as rainfall interception, evapotranspiration, moisture dynamics in the unsaturated and saturated zones, and runoff routing (see Fig. 1). In conjunction with the Surface Hydrology Research Group at Arizona State University and the NSF Science and Technology Center for Sustainability of Semi-Arid Hydrology and Riparian Areas (SAHRA) a parallel version of the tRIBS has been developed.

The parallel tRIBS approach [3] utilizes domain decomposition based on subbasins of a watershed. A stream reach graph of the channel network structure is used to determine each subbasin and its connectivity. Individual subbasins or subgraphs of subbasins are assigned to separate processors to carry out internal hydrologic computations (e.g., rainfall-runoff transformation). Routed streamflow from each subbasin forms the major hydrologic data exchange along the stream reach graph. Individual subbasins also share subsurface hydrologic fluxes across adjacent boundaries. The parallel tRIBS code has been shown to run on high-resolution or large-sized basins (~1 million nodes).

In parallel tRIBS, as in the sequential version, hydrological processing occurs in three major steps: 1) unsaturated zone, 2) saturated zone, and 3) river routing. In the unsaturated zone step water flowing across the land surface and a short distance below surface is simulated. While all subbasins are being processed, flow data is sent from upstream ghost outlet nodes to downstream head nodes. After subbasin processing, state information from downstream head nodes is updated in the upstream ghost outlet nodes to be used in the next iteration. Next, in the saturated zone step, underground flow is simulated. The direction of flow

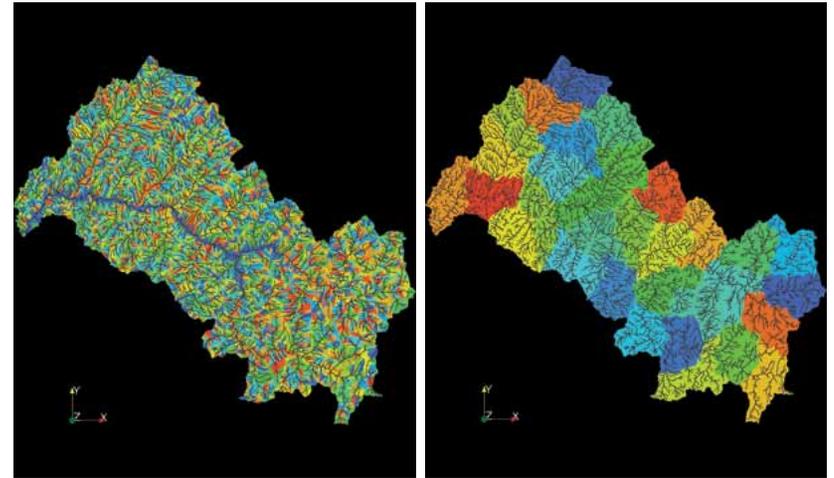
underground can change at each time step. Initial water table levels and ground water change are exchanged between processors that contain adjacent reaches. The river routing step simulates water flow through the stream reach network. Flow data from upstream ghost outlet nodes are sent to downstream head nodes during the calculation. Parallelization gains are achieved more from overlapping step processing than within step processing due to the dependencies between the related subbasins within the unsaturated zone and river network.

To improve usage of the parallel model, three supporting capabilities have been developed: 1) preprocessing construction of numerical meshes for large domains, 2) a mechanism to restart a long-running simulation, and 3) visualization for model debugging and results presentation. To speed up the mesh generation on all processors, a separate executable, MeshBuilder, is used to build input files containing the mesh, channel network, and ghost cell information specific to each processor. In the parallel operation, each processor is then able to read in and build only the assigned subbasins. Additional improvements related to faster calculations and sorting of the mesh generation also allow for the construction of very large domains ( $>10^6$  nodes). Given time limitations in the use of cluster resources, a restart capability allows saving the entire model state at user-specified time intervals. This capability is useful for long (multiyear) simulations or for multiple scenarios from a given state. Finally, a tRIBSReader Visualizer plugin library for the ParaView visualization toolkit [4] allows for the rapid display, inspection, and animation of the static basin characteristics and dynamic model output.

An efficient partitioning balances the computational load and minimizes message passing between processors. The number of nodes per reach contributes to the computational load. Connections between reaches in the stream network and subsurface flux network contribute to the messaging. Four methods were considered to improve load balancing. The default method slices up the reaches as created by tRIBS. Three others were produced using METIS multiconstraint graph partitioning. The flow method balances

the number of nodes (computational load) and stream reach network connectivity. The flow-flux method additionally considers the subsurface flux connectivity. And the flow-flux-upstream method also balances the number of reaches without upstream reaches.

A high-resolution Oklahoma Baron Fork basin model (~900K nodes) was run for a 1-year period from November 1997 to December 1998 (see Fig. 2) for performance and load balancing experiments on the Institutional Computing Coyote cluster. The default method randomly distributes reaches across processors (see Fig. 3), does not balance the computational load, and results in excessive message exchanges, especially for the subsurface flux. The flow, flow-flux, and flow-flux-upstream methods all produce geographic distributions of subbasins based on the flow network (see Fig. 3). The computational load as nodes is distributed more evenly for these methods, while flow minimizes the river routing exchanges and flow-flux minimizes the subsurface flux exchanges between processors. Run times are greatly improved compared with the default method by up to a factor of 10 for 32, 64, 128, and 256 processors. The best performance is seen for flow-flux on 128 processors with 2.186 min/simulated day. Compared with run time per simulated day on one processor, this method results in a 42X speedup.



**Fig. 3. Baron Fork sub-basins colored by partition on 32 processors for default (left) and flow-flux (right) load balancing methods**

**For more information contact  
Sue Mniszewski at  
smm@lanl.gov.**

- [1] V.Y. Ivanov et al., *Water Resour. Res.* **40**, W11102 (2004).
- [2] E.R. Vivoni et al., *Hydrol. Earth Sys. Sci.* **11**, 1683 (2007).
- [3] S. Mniszewski et al., *Increased Efficiency for High-Resolution Baron Fork Simulations Using Basin Structure Characteristics*, 9th SAHRA Annual Meeting (2009).
- [4] J. Ahrens et al., *ParaView: An End-User Tool for Large-Data Visualization in The Visualization Handbook*, Academic Press, 717 (2005).

#### **Funding Acknowledgments**

LANL Directed Research and Development Program