Title
Enhanced fixed-size parallel speedup with the Muskingum method using a trans-boundary approach and a large subbasins approximation

Journal
Water Resources Research, 51(9)

ISSN
0043-1397

Authors
David, Cédric H
Famiglietti, James S
Yang, Zong-Liang
et al.

Publication Date
2015-09-01

DOI
10.1002/2014wr016650

Peer reviewed

AGU PUBLICATIONS

Water Resources Research

RESEARCH ARTICLE
10.1002/2014WR016650

Key Points:
- A new parallel routing algorithm for the Muskingum method is presented
- The algorithm uses a domain decomposition based on a topological sort
- The trans-boundary Muskingum method leads to valuable speedup of computations

Correspondence to:
C. H. David,
cedric.david@jpl.nasa.gov

Citation:
David, C. H., J. S. Famiglietti, Z.-L. Yang, and V. Eijkhout (2015), Enhanced fixed-size parallel speedup with the Muskingum method using a trans-boundary approach and a large subbasins approximation, *Water Resour. Res.*, *51*, 7547–7571, doi:10.1002/2014WR016650.

Received 8 NOV 2014
Accepted 20 JUN 2015
Accepted article online 24 JUN 2015
Published online 15 SEP 2015

Enhanced fixed-size parallel speedup with the Muskingum method using a trans-boundary approach and a large subbasins approximation

Cédric H. David[1,2], James S. Famiglietti[1,2,3], Zong-Liang Yang[4], and Victor Eijkhout[5]

[1]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, USA, [2]University of California Center for Hydrologic Modeling, University of California, Irvine, California, USA, [3]Department of Earth System Science, University of California, Irvine, California, USA, [4]Department of Geological Sciences, Jackson School of Geosciences, The University of Texas at Austin, Austin, Texas, USA, [5]Texas Advanced Computing Center, The University of Texas at Austin, Austin, Texas, USA

Abstract This study presents a new algorithm for parallel computation of river flow that builds on recent work demonstrating the relative independence of distant river reaches in the update step of the Muskingum method. The algorithm is designed to achieve enhanced fixed-size parallel speedup and uses a mathematical approximation applied at the boundaries of large subbasins. In order to use such an algorithm, a balanced domain decomposition method that differs from the traditional classifications of river reaches and subbasins and based on network topology is developed. An application of the algorithm and domain decomposition method to the Mississippi River Basin results in an eightfold decrease in computing time with 16 computing cores which is unprecedented for Muskingum-type algorithms applied in classic parallel-computing paradigms having a one-to-one relationship between cores and subbasins. An estimated 300 km between upstream and downstream reaches of subbasins guarantees the applicability of the algorithm in our study and motivates further investigation of domain decomposition methods.

1. Introduction

The global network of rivers is the primary mechanism for terrestrial water transport in the Earth system. Rivers sustain the health of humans, terrestrial ecosystems, transport of nutrients to the oceans; and their extreme variations during floods and droughts result in billions of dollars of damage annually. Given the natural and human significance of global river networks, the understanding of river dynamics at continental-to-global scales is essential. River gauges are of essence to terrestrial hydrology despite their declining availability [*The Ad Hoc Group et al.*, 2001], and the expected Surface Water and Ocean Topography (SWOT) mission [*Alsdorf et al.*, 2007a, 2007b; *Durand et al.*, 2010] promises to further enhance the study of the global terrestrial water cycle. While the combination of in situ and remotely sensed observations will remain key to research in water resources, continental-to-global scale river network models [e.g., *Miller et al.*, 1994; *Olivera et al.*, 2000; *Oki et al.*, 2001; *Lohmann et al.*, 2004; *Yamazaki et al.*, 2011, 2012] will still be required to interpolate between space-time acquisitions of observations, to better understand process interactions, to support water management decisions, for climate simulations, and for prediction. However, the development of river transport models having the ability to simulate the past, present, and future states of surface water bodies at continental-to-global scales is progressing slowly compared to the advances in the oceanic and atmospheric sciences [*Arrigo*, 2011; *Famiglietti et al.*, 2011].

The ever-increasing availability of massively parallel super-computers has fostered the application of parallel computing in many fields of geosciences. Examples of such applications include global climate models [e.g., *Bamzai*, 2012], regional models of the atmosphere [e.g., *Michalakes et al.*, 2005] and of the oceans [e.g., *Wang et al.*, 2005], and one-dimensional (vertical) modeling of the land surface [e.g., *Peters-Lidard et al.*, 2007]. The use of parallel computing for modeling horizontal transfers on the land surface (including flow of water within river networks) and underneath the land surface is, however, still limited in comparison to other fields of geoscience. Of particular note are efforts related to the development of hydrologic models that are parallel from inception [e.g., *Ashby and Falgout*, 1996; *Wang et al.*, 2007] and others focusing on parallel enhancements to existing models [e.g., *Neal et al.*, 2009; *Vivoni et al.*, 2011; *Hwang et al.*, 2014].

One of the key aspects of high-performance programming is the decomposition and orchestration of the work load to several computing cores [*Culler et al*., 1997a]. When modeling horizontal water dynamics on parallel computers, decomposition generally focuses on the study domain (i.e., domain decomposition) and involves classifying the subbasins of a large river basin (or the reaches of a large river network); and orchestration consists of instructing a given computing core to address a subset of all subbasins (or all river reaches) [e.g., *Kollet and Maxwell*, 2006; *Neal et al*., 2009; *Li et al*., 2010; *David et al*., 2011b, 2013a; *Vivoni et al*., 2011; *Hwang et al*., 2014]. Interestingly, decomposition methods that are suitable for parallel computing of the horizontal movements of water differ from the traditional hydrological approaches to codifying subbasins [e.g., *Seaber et al*., 1987; *Verdin and Verdin*, 1999] or classifying river reaches [e.g., *Horton*, 1945; *Strahler*, 1952] as further developed in this study.

David et al. [2013a] demonstrated that the relative influence of a given river reach on its downstream reaches, when updating simple Muskingum routing computations, decreases with increasing distance until it becomes too small to be accounted for by floating-point arithmetic. Hence, downstream reaches can be updated without waiting for prior update of upstream reaches, which is rather counterintuitive. This result allows for the design of new parallel-computing algorithms that take advantage of the relative independence, even with connected river reaches. This relative independence in turn has impacts on decomposition options for large river networks.

The goal of this study is to present a new algorithm for calculating flow in large river networks within a parallel-computing environment and a new domain decomposition method. This paper builds on a brief background (section 2) on classification and sort techniques for hydrologic features, on domain decomposition approaches for parallel computing of horizontal water transfers, and on existing methods for studying parallel speedup. The new algorithm (section 3) based on a trans-boundary approach to the Muskingum method is designed in hope for substantial savings in computation time, and uses a simple technique for balanced decomposition of large river networks. The largest river basin of the United States, the Mississippi River Basin (Figure 1) is used in this study (section 4) because the main assumption of the algorithm is based on basin size. The parallel performance of the new routing algorithm and domain decomposition method is then presented (section 5), followed by a conclusion (section 6). Supporting information concerning the analysis of flow hydrographs in our modeling system is also given (Appendix A) to merely illustrate some of the challenges of continental-scale river network routing.

2. Background

2.1. Classifying and Sorting Hydrological Subdomains

The most well-known method for classification of river basins is perhaps the Hydrologic Unit Codes (HUCs) of *Seaber et al*. [1987] in which boundaries match with topographic divides; and which was developed to facilitate the work of U.S. water resources agencies in inventorying, storing, and exchanging hydrologic data. Another typical approach is that of the Pfafstetter codification system [e.g., *Verdin and Verdin*, 1999] which—similarly to HUCs—uses topographic divides for its boundaries, but also has a numbering system that recognizes upstream/downstream positions, hence accounting for the topology of river basins.

The traditional hydrological approaches to classifying the reaches of river networks are the stream ordering methods introduced by *Horton* [1945] and adapted by *Strahler* [1952]. These stream orders were designed to better understand erosion processes and hence increased downstream: a value of one is assigned to the headwater reaches and the largest value occurs at the outlet of a river network. Note that several river reaches can hence have the same stream order.

Around the same period, computer scientists interested in the scheduling of large projects involving a series of interdependent tasks introduced the concept of topological sorts [e.g., *Lasser*, 1961; *Kahn*, 1962]. In a topological sort, each task is given a unique integer value. By organizing tasks in increasing or decreasing value of their topological sort, a sequence of all tasks for completion of the entire project can be determined.

A related concept of classification of river reaches is the topological path length that was first brought from mathematics and computer science to the study of channel networks by *Werner and Smart* [1973]. A path along the river network is defined as the shortest route between the outlet point of a river network and the upstream point of a river reach. The topological path length is then defined as the number of river reaches

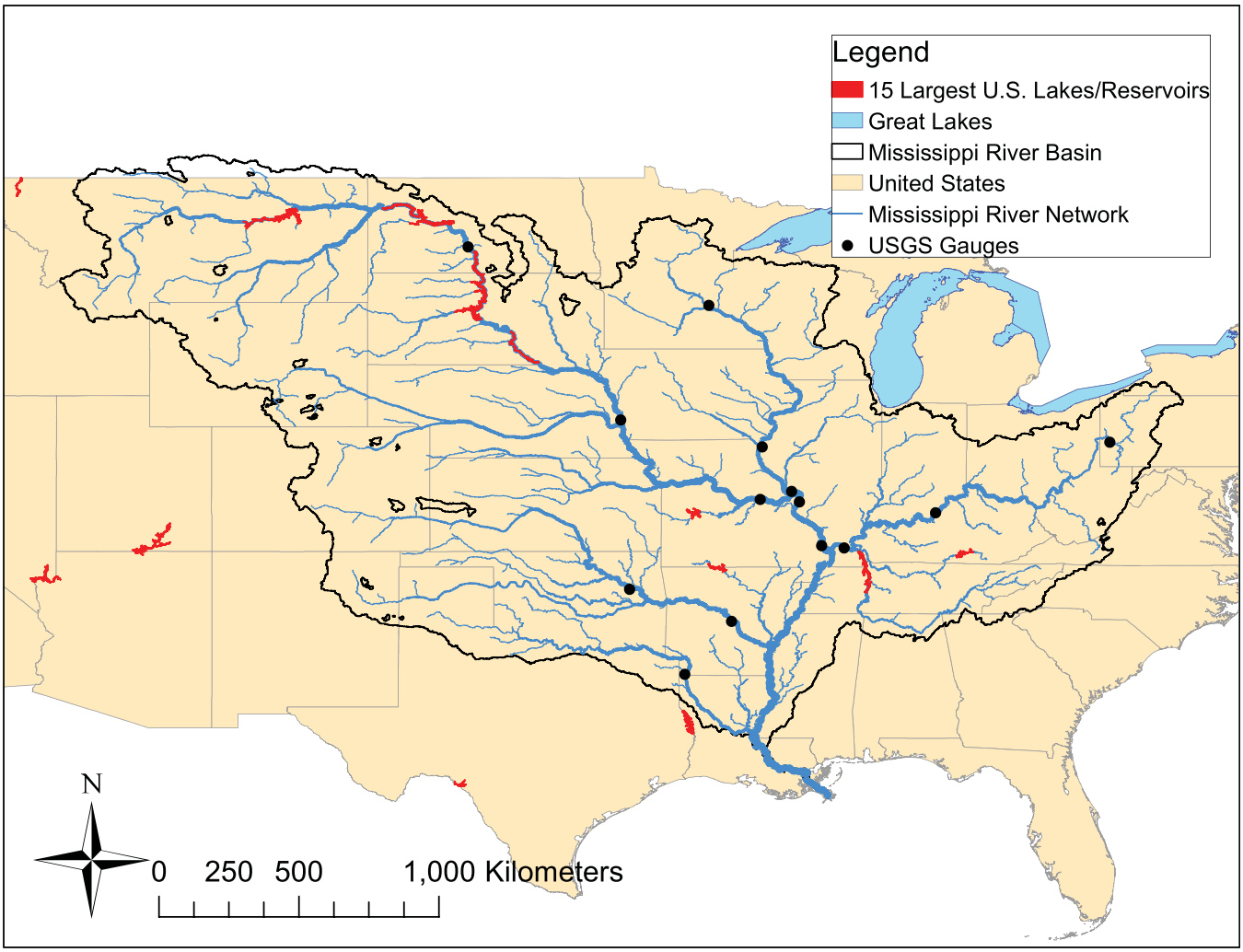

Figure 1. The HydroSHEDS representation of the Mississippi River network includes 102,229 river reaches (blue) although only the 10% with largest contributing catchments are shown here for clarity. Fourteen USGS gauges (black) are used in this study for calibration and validation. Eight of the 15 largest U.S. lakes and reservoirs (red) are located in the Mississippi River Basin.

(the links of the network) traversed in a path. As in the Strahler stream order (but unlike topological sorts), two different river reaches can have the same topological path length. However, unlike the Strahler stream order, consecutive reaches on a river stem have different topological path lengths even without the presence of confluences. Despite being easy to compute, the topological path length is advantageous in this study because it provides a direct way to calculate the topological distance between two connected upstream/downstream reaches, as will be of importance in section 3.

Because water mainly flows downstream, many studies organize hydrologic features with a topological sort, and such a topological sort is often generated based on the topological path length. Various types of topological sorts have recently been used for river network routing: that of the SIM-France modeling system [*Habets et al.*, 2008; *David et al.*, 2011a] and the hydrologic sequence number [*USEPA and USGS*, 2010] for river network routing in U.S. river basins [*David et al.*, 2011b, 2013a, 2013b]. Topological sorts are also valuable for domain decomposition in parallel computing, as further developed below.

2.2. Existing Approaches for Horizontal Hydrologic Transfers Within a Parallel-Computing Environment

As mentioned in section 1, the development of a parallel-computing model is concerned in part with two interconnected goals: partitioning the problem and orchestrating computations. Partitioning involves splitting the problem into small tasks and combining the tasks into groups of tasks. Orchestration involves

allowing synchronization of the groups of tasks and communication among them. Readers are referred to *Culler et al*. [1997b] for further details on the parallelization process. Existing studies of horizontal hydrologic transfers involve these two interconnected aspects of parallel computing.

Neal et al. [2009] adapted the version of the LISFLOOD model that is capable of computing floodplain inundation (LISFLOOD-FP) [*Bates and De Roo*, 2000] to run in a parallel environment. In *Neal et al*. [2009], the entire computing domain is partitioned in rectangular subdomains that are then each addressed simultaneously by different computing cores. Note that while floodplain inundation processes are parallelized in *Neal et al*. [2009] river routing processes are not. A similar approach to domain decomposition is applied in papers by Kollet, Maxwell, and collaborators [*Kollet and Maxwell*, 2006; *Kollet et al*., 2010] who use the Parallel Flow simulator (ParFlow) [*Ashby and Falgout*, 1996]. ParFlow allows for the movement of water underneath the land surface and therefore differs from LISFLOOD-FP, and the computations for surface and subsurface flows in ParFlow are both parallelized [*Kollet and Maxwell*, 2006]. Despite these differences, an approach for domain decomposition that uses rectangular boxes as subdomains is used in ParFlow and LISFLOOD-FP studies. *Hwang et al*. [2014] modified HydroGeoSphere [*Therrien et al*., 2006] for parallel computations of coupled surface-subsurface flow in which domain decomposition is made of slices in the North-South direction and subdomains are not necessarily rectangular. *Vivoni et al*. [2011] adapted the TIN-based Real-time Integrated Basin Simulator (tRIBS) of *Ivanov et al*. [2004] for use on parallel computers. Like ParFlow and HydroGeoSphere, tRIBS simulates coupled surface/subsurface interactions. The study of *Vivoni et al*. [2011] emphasizes domain decomposition the most because three different techniques are used: one based on the relative upstream/downstream position of subbasins, and two using a classic graph partitioning software called METIS [*Karypis and Kumar*, 1999]. One important road block in the methods of *Vivoni et al*. [2011]—compared to the studies of *Neal et al*. [2009], *Kollet and Maxwell* [2006], and *Hwang et al*. [2014]—is that computing loads remain unbalanced in all methods tested. One valuable advantage though, is that the decomposition of *Vivoni et al*. [2011] follow natural topographic divides and hence limits intercore communications. Note that the aforementioned orchestration techniques are all static, i.e., each computing core addresses a unique subdomain and all computations are simultaneous.

A different orchestration paradigm is used in the Digital Yellow River Model (DYRIM) of *Wang et al*. [2007] which was developed to study erosion processes. DYRIM uses the domain decomposition of *Li et al*. [2010] that is based in part on the topological path length. A dynamic orchestration paradigm based on the algorithm of *Li et al*. [2011] is used in DYRIM. Once subbasins are determined, computations are carried from upstream to downstream with less computing cores than subbasins, and any core is allowed to address any subbasin as soon as it becomes available for computation. Such a dynamic algorithm allows for valuable gains in time. More information on the significant capabilities of dynamic algorithms is given in an unrelated study by *Wang et al*. [2012] which also provides notable theoretical background on the limitations of dynamic parallel algorithms.

2.3. Parallel Speedup in Horizontal Terrestrial Hydrology

One valuable advantage of parallel computers is their ability to save their users' time. These temporal savings are typically quantified using a metric called parallel speedup. Two approaches are generally used to compute speedup. The first investigates if parallel computers can reproduce a given simulation in less time. The parallel speedup for these fixed-size problems with increasing computing power is known as fixed-size speedup (or "strong" speedup) and is governed by Amdahl's law [*Amdahl*, 1967]. The study of fixed-size speedup is the most prevalent in the aforementioned hydrologic literature [*Neal et al*., 2009; *David et al*., 2011b, 2013a; *Li et al*., 2011; *Vivoni et al*., 2011; *Wang et al*., 2012; *Hwang et al*., 2014]. The second approach to quantifying temporal savings is based on a different paradigm in which parallel computers are used to address problems one would not otherwise tackle because simulations times would be prohibitive on regular computers. In other words, one can solve bigger problems in a given amount of time with more computing power [e.g., *Ashby and Falgout*, 1996; *Kollet and Maxwell*, 2006; *Kollet et al*., 2010]. The corresponding temporal savings are measured in studies where computing power increases linearly with the number of unknowns using a quantity referred to as scaled-size speedup (or "weak" speedup) and governed by the Gustafson-Barsis Law [*Gustafson*, 1988]. Despite their seeming similarities and equal maximum (ideal) value, the upper practical limits for fixed-size and scaled-size speedup approaches are quite different, the latter often leading to larger values [e.g., *Gustafson*, 1988].

Recent work by *David et al*. [2013a] demonstrated the rather counterintuitive fact that the relative upstream/downstream influence within a river network can become smaller than floating-point precision if river basins are large enough, hence motivating the development of alternative algorithms for parallel computing of river flow, which is the main purpose of this paper. Because such relative independence entails a virtual decoupling of connected river reaches (section 3), it also permits the use of an increased number of computing cores for a given study domain, therefore justifying a fixed-size speedup approach in this paper.

3. Theoretical Approach

The theoretical approach presented in this study is directly motivated by the overarching goal to develop simple routing algorithms that can be applied at scales ranging from continents to the full globe and at high-spatial resolutions (1–10 km) while efficiently using parallel computing. For this reason, focus is made here on a classic river routing scheme called the Muskingum method [*McCarthy*, 1938] acknowledging that many efforts have since its inception focused on enhanced hydraulics of Muskingum-type algorithms [e.g., *Cunge*, 1969; *Miller and Cunge*, 1975; *Koussis*, 1978; *Ponce and Yevjevich*, 1978; *Todini*, 2007]. Despite its simplicity, the Muskingum method allows for downstream propagation of flow waves, and hence provides a good test case for studying means of addressing wave propagation in large networks of rivers and streams within a parallel-computing environment; with potential applications to more advanced river routing schemes.

3.1. The Muskingum Method as a Linear System of Equations

Adapting the Muskingum method [*McCarthy*, 1938] using a matrix notation, *David et al*. [2011b] showed that this method can be represented by equation (1), and solved on parallel computers by splitting the vectors and matrices on multiple cores and allowing for intercore communication:

$$\begin{aligned}(\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N})\cdot\mathbf{Q}(t+\Delta t)&=\mathbf{C_1}\cdot\mathbf{Q^e}(t)+\mathbf{C_2}\cdot[\mathbf{N}\cdot\mathbf{Q}(t)+\mathbf{Q^e}(t)]+\mathbf{C_3}\cdot\mathbf{Q}(t)\\&=\mathbf{b}(t)\end{aligned} \tag{1}$$

In equation (1), t is time and Δt is the routing time step. The bold notation is used for vectors and square matrices which are all of size m the total number of river reaches in a river network. $\mathbf{I}$ is the identity matrix. $\mathbf{N}$ is the network matrix in which—assuming a maximum of one downstream river reach being allowed for each river reach—a value of one is used at row i and column j if reach j flows into reach i and zero is used elsewhere. $\mathbf{C_1}$, $\mathbf{C_2}$, and $\mathbf{C_3}$are diagonal matrices composed of the classic Muskingum parameters C_{1_j}, C_{2_j}, and C_{3_j}, respectively. The vector $\mathbf{Q}$ is made of elements Q_j corresponding to the river flow rate at the downstream point of each river reach j. The vector $\mathbf{Q^e}$ is made of elements Q_j^e corresponding to the flow rates from outside the river network that are estimated using a land surface model and added upstream of each river reach j. The derivation of equation (1) assumes the partial temporal uniformity of $\mathbf{Q^e}$ in order to simplify the mathematical formulation, to limit the quantity of input data, and to facilitate the coupling with land surface models. Such an assumption is valid if land surface model data are made available less often that the river routing time step as in this study. All elements on the right-hand side of equation (1) are taken at time t and can be combined into a single vector $\mathbf{b}(t)$ to highlight the difference between the data and the unknown $\mathbf{Q}(t+\Delta t)$. One can hence solve for $\mathbf{Q}(t+\Delta t)$in equation (1) by providing the highly sparse matrix $\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N}$ and the right-hand side $\mathbf{b}(t)$ to a linear system solver.

Equation (1) is valid regardless of how river reaches are sorted. However, sorting in an upstream-to-downstream manner allows that $\mathbf{N}$ is strictly lower triangular (i.e., all elements on and above the diagonal are zero) which will be assumed in the following. Such sorting hence makes the linear system matrix of equation (1) a lower unit triangular matrix (i.e., a lower triangular matrix for which all diagonal elements have a value of one) and therefore simplifies the solving procedure [*David et al*., 2011b, 2013a]. More information on the derivation and parallel performance of this matrix-based Muskingum method can be found in *David et al*. [2011b, 2013a].

3.2. Introduction of a Parallel Trans-Boundary Matrix in the Matrix-Based Muskingum Method

On parallel computers with N cores, a matrix is usually split so that each core addresses one of N subsets of consecutive rows of the initial matrix. These rows are then further divided in one square submatrix, the diagonal of which coincides with that of the initial matrix; and one rectangular submatrix containing the

remaining elements of the subset. The elements in the off-diagonal submatrix correspond to places where intercore communication is necessary for operations in a parallel-computing environment.

In the case of a river network in a basin, a subset of consecutive rows of the network matrix **N** corresponds to the river network of a given subbasin although there is no a priori necessity for the subbasins to be contiguous in this context of parallel matrices as there is generally in hydrology. Once consecutive subsets of the network matrix are selected, the elements of an off-diagonal submatrix of **N** correspond to places where water flows across the boundaries separating the corresponding subbasin from others. Isolating the off-diagonal submatrices from the network matrix therefore allows highlighting the trans-boundary flows among subbasins in the river network. Let **T** be such a parallel trans-boundary matrix in which the elements of the off-diagonal submatrices are those of **N** and the elements of the diagonal submatrices are all null. Like **N**, **T** and **N**−**T** are both strictly lower triangular. Also, by construction, **T** is strictly lower triangular by blocks, and **N**−**T** is diagonal by blocks. Introducing the matrix **T** into equation (1) leads to

$$(\mathbf{I}-\mathbf{C_1}\cdot(\mathbf{N}-\mathbf{T}))\cdot\mathbf{Q}(t+\Delta t)=\mathbf{b}(t)+\mathbf{T}\cdot\mathbf{C_1}\cdot\mathbf{Q}(t+\Delta t) \tag{2}$$

The comparison of equations (1) and (2) leads to two remarks. First, in addition to being a lower unit triangular matrix (like $\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N}$) the matrix $\mathbf{I}-\mathbf{C_1}\cdot(\mathbf{N}-\mathbf{T})$ is also diagonal by blocks, which allows for faster resolution in a parallel-computing environment because intercore communication is not required within the linear system solver. Second, equation (2) is implicit in time because its right-hand side includes the unknown vector $\mathbf{Q}(t+\Delta t)$. However, the number of elements of the unknown $\mathbf{Q}(t+\Delta t)$ that are included in the product $\mathbf{T}\cdot\mathbf{C_1}\cdot\mathbf{Q}(t+\Delta t)$ is very limited because $\mathbf{T}\cdot\mathbf{C_1}$ is sparse. Therefore, the estimation of the few nonzero elements in $\mathbf{T}\cdot\mathbf{C_1}\cdot\mathbf{Q}(t+\Delta t)$ could allow for taking advantage of potentially faster parallel computations in equation (2) compared to equation (1).

3.3. The Limited Spatial Propagation of Flow Waves During Each Time Step of the Muskingum Method

David et al. [2013a] showed that the relative influence of one given river reach on another during the update step of the Muskingum method decreases with increasing distance separating the two reaches until it becomes smaller than can be accounted for by floating-point arithmetic. The minimum distance above which the flow of a given river reach does not influence the update of the distant downstream reach is referred to as the "radius of downstream influence" in *David et al*. [2013a] and this quantity varies in space and time. *David et al*. [2013a] provided a conservative estimate of the maximum value of this topological distance (i.e., a distance expressed as a number of river reaches) for each reach of a large network over an entire year of simulation. In the study herein, we will focus on the maximum value of the radius of influence in time but also in space—for simplicity—and use $R_{\max}^{down}$ to denote such quantity.

Let D_k be the minimum topological distance separating any two connected upstream and downstream boundary reaches of a given subbasin k. Assuming that a type of domain decomposition can be designed assuring that D_k is always greater than $R_{\max}^{down}$, the downstream-most river reach in each subbasin (at least) can be calculated exactly without having to account for intercore communication. One could therefore neglect the term $\mathbf{T}\cdot\mathbf{C_1}\cdot\mathbf{Q}(t+\Delta t)$ in equation (2) to compute $\hat{\mathbf{Q}}$, an estimate of **Q**:

$$(\mathbf{I}-\mathbf{C_1}\cdot(\mathbf{N}-\mathbf{T}))\cdot\hat{\mathbf{Q}}(t+\Delta t)=\mathbf{b}(t) \tag{3}$$

In the case of large subbasins, $\hat{\mathbf{Q}}$ is an accurate estimate of **Q** only for the downstream-most river reaches of each subbasin, all other reaches being inaccurate. Fortunately, the downstream-most reaches are the only ones involved in the computation of $\mathbf{T}\cdot\mathbf{C_1}\cdot\mathbf{Q}(t+\Delta t)$, and the following equality ensues

$$\begin{gathered}\forall k\in[1,N], D_k > R_{\max}^{down}\\ \mathbf{T}\cdot\mathbf{C_1}\cdot\mathbf{Q}(t+\Delta t)=\mathbf{T}\cdot\mathbf{C_1}\cdot\hat{\mathbf{Q}}(t+\Delta t)\end{gathered} \tag{4}$$

Again, such equality is only valid when subbasins are large enough with regards to the radius of influence.

3.4. A Trans-Boundary Muskingum Method With Large Subbasins Approximation

While $\mathbf{T}\cdot\mathbf{C_1}\cdot\mathbf{Q}(t+\Delta t)$ is neglected above when solving for the downstream-most reaches, it cannot be neglected when computing all other reaches. To obtain a valid computation of all river reaches in the network, the substitute $\mathbf{T}\cdot\mathbf{C_1}\cdot\hat{\mathbf{Q}}(t+\Delta t)$ of equation (4) must be reinjected into equation (2). Combining

equations (2–4), one obtains what will be referred to here as a trans-boundary Muskingum method with large subbasins approximation:

$$\forall k \in [1,N], D_k > R_{\max}^{down}$$
$$\begin{cases} (\mathbf{I}-\mathbf{C_1}\cdot(\mathbf{N}-\mathbf{T}))\cdot\hat{\mathbf{Q}}(t+\Delta t)=\mathbf{b}(t) \\ (\mathbf{I}-\mathbf{C_1}\cdot(\mathbf{N}-\mathbf{T}))\cdot\mathbf{Q}(t+\Delta t)=\mathbf{b}(t)+\mathbf{T}\cdot\mathbf{C_1}\cdot\hat{\mathbf{Q}}(t+\Delta t) \end{cases} \quad (5)$$

When computing $\hat{\mathbf{Q}}$ and $\mathbf{Q}$ sequentially, equation (5) is explicit in time, and can therefore be solved with linear system solvers. One must note here again that equation (4) will fail and hence equation (5) will not produce accurate results if the subbasins are not sufficiently large when compared to the radius of influence. Therefore, this method must be used with caution.

Finally, equation (5) is derived here for parallel-computing environments, but collapses into equation (1) repeated twice when using one unique computing core because $\mathbf{T}$ becomes a null matrix in which case no approximation is made. Equation (5) can therefore be used on one or multiple cores although using it on one core only is unnecessarily inefficient.

3.5. Fixed-Size Speedup of Computations

As mentioned earlier, the ability to solve a mathematical problem faster by increasing the number of computing cores is generally referred to as fixed-size speedup. Given $\tau(N)$, the time needed to solve a mathematical problem on N cores, the fixed-size speedup $S(N)$ is defined by

$$S(N)=\frac{\tau(1)}{\tau(N)} \quad (6)$$

Ideal speedup is obtained when $S_{ideal}(N)=N$, i.e., when computing time decreases exactly with the number of cores used. Because simple routing schemes such as the Muskingum method are updated in an upstream-to-downstream manner, topological constrains make it a challenge to achieve ideal fixed-size speedup [*David et al*., 2013a]. However, assuming balanced computing loads and accounting for the limited spatial propagation of flow waves during the update step of the Muskingum method, *David et al*. [2013a] hypothesized that one could design computing algorithms allowing for a fixed-size speedup $S_2(N)=N/2$, although the method they used failed to achieve such a feat.

All matrices and vectors of equation (5) are of the same size as those in equation (1). Since equation (5) consists of twice the number of systems of linear equations as equation (1), one should expect the resolution of equation (5) to require approximately twice the amount of time necessary to solving equation (1) when using one unique computing core for resolution. However, in both systems of linear equations used in equation (5), the linear system matrix is diagonal by blocks. Hence, when solving on multiple cores, the resolution of these two linear systems can be done independently among all computing cores (i.e., without intercore communication); and equation (5) will likely reach high fixed-size speedup capabilities. The only place where communication among cores is necessary is in the computation of $\mathbf{T}\cdot\mathbf{C_1}\cdot\hat{\mathbf{Q}}(t+\Delta t)$, so the number of elements in $\mathbf{T}$ should be kept small to guarantee best-possible fixed-size speedup. In cases where equation (5) is valid, it can therefore be expected that fixed-size speedup will be close to $S_2(N)=N/2$ if loads are balanced. Accounting for the initial overhead of a factor of two due to the mathematics of equation (5) compared to equation (1) and for potential load imbalance L_N, one can therefore hope that the proposed trans-boundary Muskingum method with large subbasins approximation will achieve the following fixed-size speedup:

$$S(N)=L_N\cdot\frac{N}{2} \quad (7)$$

In order to do so, careful decomposition into subbasins must be undertaken.

3.6. Sorting and Decomposing the River Network Using the Topological Path Length for Eased Computations and Comparisons

The proposed work is concerned with evaluating a new method for parallel computations and hence comparisons among results obtained with various numbers of computing cores are valuable. Such comparisons

are eased if a unique global pattern is used for sorting all river reaches—regardless of the number of cores used for resolution—because such guarantees similarity among output files.

The choice of a unique sorting method partly enforces how the basin is decomposed into subbasins because parallel matrices are split so that each computing core addresses consecutive rows of the initial matrix (section 3.2). The number of river reaches in each subbasin, however, remains to be chosen and load balancing is of importance (section 3.5). In order to keep the load imbalance minimum ($L_N=1$), each core should be assigned m/N river reaches. Note here that an attempt at using a classic domain decomposition based on topographic divides failed to provide balanced subbasins (see section 5.4) as was previously experienced by *Vivoni et al*. [2011] hence motivating the development of an alternate method in this study.

Sorting river reaches in an upstream-to-downstream manner makes linear system matrices lower triangular and decreases the resolution time in linear system solvers [*David et al*., 2011b, 2013a]. Such is true at the basin level (global sorting) and at the subbasin level (local sorting).

A direct way to allow for such sorting—both globally and locally and regardless of the size chosen for each subbasin—is to arrange river reaches in decreasing value of their topological path length. However, there are multiple ways to do so because two or more river reaches can have the same topological path length, but one can enforce a unique sorting method by further sorting reaches with the same topological path length in increasing value of their unique identifying integer.

In this study, river reaches are therefore sorted in decreasing value of their topological path length and reaches with equal topological path length are further sorted in increasing value of their unique identifier. Each computing core is then assigned m/N river reaches to keep the computing loads balanced.

Again, one has to check that the proposed method for decomposing the river network allows for subbasins that are large enough with regards to the radius of influence (i.e., $\forall k \in [1,N], D_k > R_{\max}^{down}$) otherwise equation (5) might fail to provide accurate results. Additionally, the corresponding amount of trans-boundary flows needs be calculated in order to quantify intercore communication.

3.7. Precision Expected From Computer Implementations of the Muskingum Method

Applying the formulation of *Higham* [1990] to equation (1), an estimate of the numerical precision to be expected from computations obtained using the Muskingum method, regardless of what algorithm is used for resolution, can be obtained by relating small variations of $\mathbf{b}$ and small variations of $\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N}$ to small variations of $\mathbf{Q}$ [*David et al*., 2013a]:

$$\frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2}\leq\varepsilon,\frac{\|\Delta(\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N})\|_2}{\|\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N}\|_2}\leq\varepsilon,$$
$$\frac{\|\Delta\mathbf{Q}\|_2}{\|\mathbf{Q}\|_2}\leq\frac{2\cdot\varepsilon\cdot\kappa_{\|\|_2}(\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N})}{1-\varepsilon\cdot\kappa_{\|\|_2}(\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N})} \tag{8}$$

where ε is a small real number, $\|\|_2$is the two-norm, and $\kappa_{\|\|_2}(\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N})$ is the two-norm condition number of $\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N}$ computed using

$$\kappa_{\|\|_2}(\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N})=\frac{\sigma_{\max}}{\sigma_{\min}} \tag{9}$$

where $\sigma_{\min}$ and $\sigma_{\max}$ are, respectively, the minimum and maximum singular values of $\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N}$.

4. Application to the Mississippi River Basin

We test our theoretical approach in a 10 year simulation of discharge in the largest river basin of the United States, the Mississippi River Basin. At the core of this application is a river network routing model called RAPID [*David et al*., 2011b] simulating discharge in all river reaches of the Mississippi River Basin as described in a near-global hydrographic data set called HydroSHEDS [*Lehner et al*., 2008]. Estimates of surface and subsurface runoff were derived from second phase of the North American Land Data Assimilation System (NLDAS2), [*Xia et al*., 2012a, 2012b] and estimates of river discharge from in situ observations were obtained from the U.S. Geological Survey. The temporal range of this study is chosen as 1 January 2000 to 31 December 2009. Each component of the simulation is further described in the following sections.

4.1. RAPID

The Routing Application for Parallel computatIon of Discharge (RAPID) [*David et al*., 2011b] is a river network routing model based on a matrix version of the Muskingum method given in equation (1). The principal modification made to RAPID compared to previous studies [*David et al*., 2011a, 2011b, 2013a, 2013b] is the addition of the trans-boundary Muskingum method with large subbasins approximation given in equation (5). Additional information on RAPID—including download links for the source code corresponding to this study [*David*, 2013] as well as versions corresponding to previous studies—is available at http://rapid-hub.org.

4.2. Using HydroSHEDS in RAPID

HydroSHEDS [*Lehner et al*., 2008] is a near-global hydrographic data set that was derived based on the digital elevation model (DEM) obtained from the Shuttle Radar Topography Mission (SRTM). Of particular interest for this study are the HydroSHEDS files that contain vectorized river networks including the cumulative drainage area contributing to each river reach. These river network files are available on a per continent basis and the highest existing available resolution (derived from the 15 arc sec DEM) is used in this study. One of the valuable attributes of the network file is the cumulative catchment area for each river reach. HydroSHEDS also includes a preliminary basin file focusing on basins corresponding to the outlet points and the outline of the Mississippi Basin used here is extracted from this basin file. This study focuses on the Mississippi River Basin which corresponds to 102,229 river reaches of the 467,820 available for North America in HydroSHEDS (Figure 1).

The current version of HydroSHEDS does not include connectivity information among river reaches, nor does it provide the noncumulative contributing catchment area or the length corresponding to each river reach. These three pieces of information are needed in RAPID and were therefore built here.

To determine the connectivity among reaches of the river network, a two-step process was used. First, the latitude and longitude of the start and end points of each reach were determined using a geographic information system (GIS) called ArcGIS. Second, a Fortran program was written to identify that reach j flows into reach i when the geographic coordinates of the end point of reach j match those of the start point of reach i. This two-step process showed that, in the HydroSHEDS representation of the Mississippi River Basin, each river reach has a maximum of one downstream reach and a maximum of four upstream reaches. From a graph theory perspective, this network is a connected graph with 102,229 vertices (i.e., river reaches) and 102,228 edges (i.e., connections) and is therefore a tree [*Berge*, 1962]. From a hydrology perspective, this means that the water in every river reach of the domain will eventually flow to a unique outlet. The topological path length that is useful for domain decomposition and sorting of river reaches (see section 3.6) was determined based on this network connectivity and its value ranges between 1 and 1044.

The catchment area contributing to each reach was estimated with a Fortran program using the connectivity information derived above and the native HydroSHEDS field called "UP_CELLS" that gives the maximum flow accumulation (in number of 15 arc sec grid cells) of each reach. The maximum flow accumulation of all upstream reaches was subtracted from the maximum flow accumulation of each given reach to compute the number of 15 arc sec grid cells in any given catchment (Figure 2). The area of a 15 arc sec grid cell located at the downstream end of each reach was then computed assuming a spherical Earth with radius the arithmetic mean radius [see e.g., *Moritz*, 1980, for definition] of the WGS84 spheroid [*NIMA*, 2000] that serves as the geographic coordinate system of HydroSHEDS. Finally, the catchment area was computed by multiplying the number of grid cells by the area of one local grid cell. Based on this series of calculations, the sum of all contributing catchment areas in the entire Mississippi River Basin is 3,179,875.9 km^2 which is within 0.01% of the 3,179,517.1 km^2 provided in the preliminary basin layer of HydroSHEDS and is therefore deemed satisfactory. The benefit of this procedure is that an estimate of the noncumulative catchment area for *every* river reach is provided—which is useful for river routing—instead of the overall basin area currently included in HydroSHEDS. Based on these computations the contributing catchments vary in size from 0.14 to 542.78 km^2 (mean: 31.11 km^2, median: 24.79 km^2, standard deviation: 25.66 km^2). The probability density function and the cumulative distribution function of contributing catchment sizes are shown in Figure 3a).

The length of each river reach in the network was computed in ArcGIS through a projection from the native geographic coordinate system used in HydroSHEDS (WGS84) to a projected coordinate system adapted to

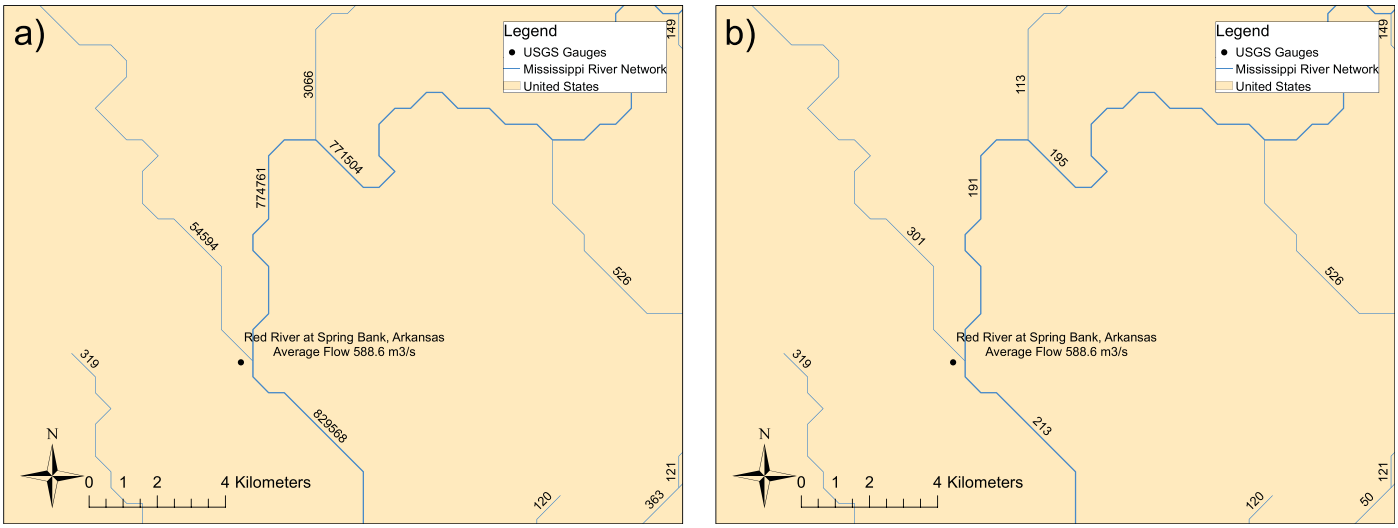

Figure 2. (a) The cumulative drainage area of each reach of the HydroSHEDS river network is available in number of 15 s grid cells as part of the data set. (b) The noncumulative drainage area is computed in this study based on network connectivity. Also note the USGS Gauge located on the Red River at Spring Bank, AR does not exactly match that of the nearest reach.

North America (an Albers Equal Area Conic Projection with standard parallels 20°N and 60°N and based on NAD83). It was determined that the 102,229 river reaches of the Mississippi River Basin vary in size from 0.29 to 101.50 km (mean: 6.20 km, median: 4.79 km, standard deviation: 5.29 km). The probability density function and the cumulative distribution function of river reach sizes are shown in Figure 3b).

4.3. Inflow From Land to HydroSHEDS Rivers

An estimate of the water flowing on and underneath the land surface into the river network was obtained using data generated with the VIC model [*Liang et al*., 1994; *Wood et al*., 1997] and of which outputs from version 4.0.3 were obtained from NLDAS2 [*Xia et al*., 2012a, 2012b]. Three hourly inflows of water into HydroSHEDS river reaches were estimated from the sum of VIC surface and subsurface runoff in a way similar to that used by *David et al*. [2013a, 2013b] and in which the runoff value corresponding to a single point in the catchment is multiplied by the catchment area. The single point used here is not the catchment centroid as in previous studies because catchment layers are not available in HydroSHEDS. Instead, the downstream point of each reach was picked as the single point of reference for consistency with the location used in the estimation of the catchment area (see section 4.2). This conversion method between the gridded environment of NLDAS2 and the vector environment of HydroSHEDS effectively assumes that each

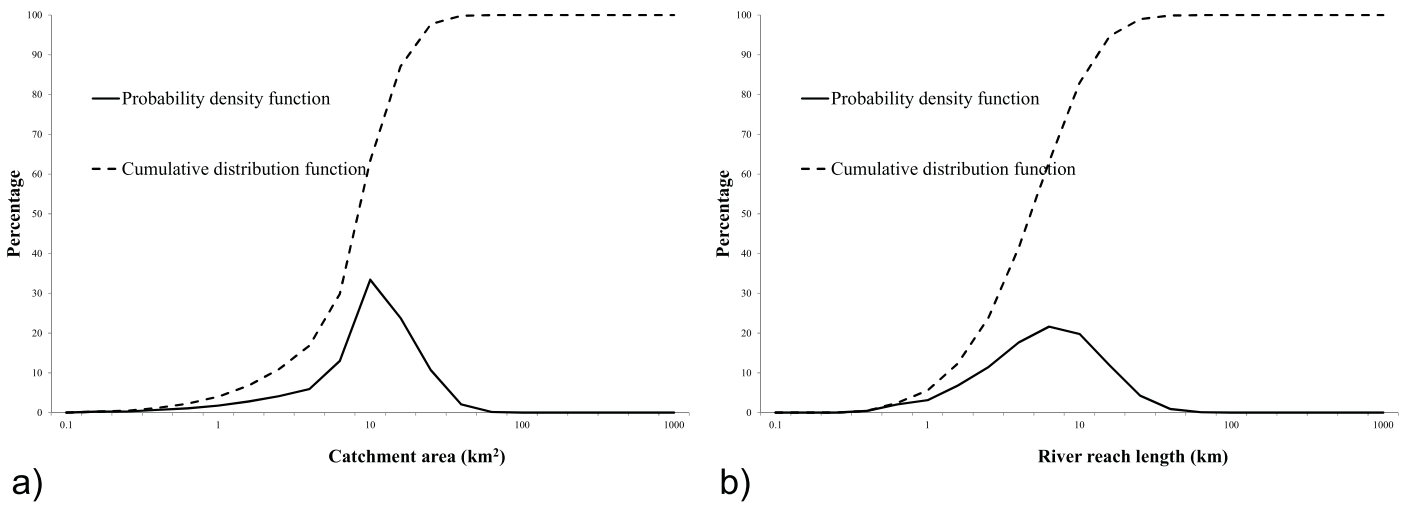

Figure 3. Probability density function and cumulative distribution function of (a) catchment area and (b) river reach length.

catchment is located in a single NLDAS2 grid cell which is a valid approximation if catchments are much smaller than grid cells as is mostly the case in the domain considered. As in previous studies [*David et al*., 2011b, 2013a, 2013b], the time lag between runoff generation and inflow to the river network is assumed null. Such neglecting of the horizontal routing time for transport of water on and underneath the land surface outside the river network could be challenged with the larger catchment sizes used here (approximately 30 km^2) compared to previous studies (approximately 3 km^2). A single land surface model was used here (VIC) following a previous comparative study of the four NLDAS2 land models used with RAPID [*David et al*., 2013b]. Note that land-river coupling is done in a one-way sense in this study although RAPID can be used within a coupled surface-subsurface modeling system [e.g., *David et al*., 2011a] if the land model has such capability. One might expect that the inclusion of surface routing processes and river-aquifer interactions could contribute to the betterment of simulations in some parts of the Mississippi River Basin. However, the main focus of this study is on parallel speedup of computations within the river network so these simplifications were kept, for simplicity of the modeling approach.

4.4. Observations of Discharge and Relation to HydroSHEDS River Reaches

Estimates of river discharge from in situ observations were obtained from the U.S. Geological Survey National Water Information System (USGS NWIS). Because the HydroSHEDS river networks are derived from a DEM, the reach locations do not exactly coincide with those of mapped rivers. Hence gauges must be virtually moved from their actual location to the closest river reach with similar drainage area (e.g., Figure 2) in order to accurately relate gauges and their corresponding HydroSHEDS reach. This procedure is often referred to as "snapping" in GIS software and requires manual checking. Therefore, a limited number of gauging stations is used in this study for calibration of model parameters and validation of model simulations.

Three criteria were chosen for selection of which gauging stations to use in this study. The first criterion was that stations capture the main hydrologic features across the Mississippi River and its main tributaries. Focus was therefore made on gauges located on the Missouri River, the Tennessee River, the Ohio River, the Red River, the Arkansas River, and the Mississippi River itself. It was attempted to capture the flow before and after the main confluences, and before and after the main lakes and reservoirs, when possible. A variety of gauging stations with mean discharge spanning from 400 to 8000 m^3/s (approximately) were hence selected. The second criterion for selecting stations was to—when possible—use gauges that have already been chosen in existing hydrologic studies of the Mississippi Basin or its subbasins [i.e., *Abdulla et al*., 1996; *Wood et al*., 1997; *Lohmann et al*., 1998, 2004; *Maurer et al*., 2001; *David et al*., 2013a]. The third criterion for selection was the availability of daily data every day throughout the temporal range of this study. The stations used in the published literature were often initially selected for their hydrologic significance and are therefore likely still functioning today. If such stations had been decommissioned prior to or during the study period, alternative stations in their vicinity and that captured similar flows were used here instead. If two or more stations located on the same river measured comparable flow magnitudes, the station with the longest record was picked.

Table 1 shows the list of stations used in this study and includes their name, USGS code, geographic coordinates, drainage area, and a summary of the two initial criteria used for selection of each station. Figure 1 shows the location of these stations in the study domain. Note that no station of the main stem of the Tennessee River fit the aforementioned selection criteria.

5. Results

5.1. Time Step of RAPID

The river routing time step used for RAPID is determined based on the smallest of the mean and the median reach length (i.e., the median length of 4.79 km in this study) and an overestimated flow wave celerity (here 2.5 m/s) leading to 1916 s. In order to keep an integer conversion factor between the routing time step and the period of time at which inflow data are obtained, the value of $\Delta t = 1800$ s was chosen.

5.2. Calibration of RAPID

As in previous studies [*David et al*., 2011b, 2013a, 2013b], an optimization procedure designed to minimize the square errors between observations and simulations at the daily time step was used to determine the parameters of RAPID simulations. The optimal set of parameters was chosen by determining the values of two scalars λ_k and λ_x such that

AGU Water Resources Research

10.1002/2014WR016650

Table 1. USGS Gauges of the Mississippi River Basin Selected for This Study

USGS Gauge Name	USGS Station Code	USGS Longitude	USGS Latitude	USGS Coordinate System	USGS Drainage Area (km^2)	Hydrologic Features Observed by Gauge (Selection Criteria 1)	Gauge Also Used in the Published Literature (Selection Criteria 2)
Missouri River at Bismarck, North Dakota	06342500	−100.82	46.81	NAD83	482,771	Downstream of Fort Peck Lake and Lake Sakakawea	Close to Missouri River at Garrison Dam, North Dakota [*Wood et al*., 1997]
Missouri River at Omaha, Nebraska	06610000	−95.92	41.26	NAD27	836,044	Downstream of Lake Oahe and Lake Francis Case	*Wood et al*. [1997]
Missouri River at Hermann, Missouri	06934500	−91.44	38.71	NAD83	1,353,262	Upstream of confluence with Mississippi River	*Wood et al*. [1997], *Maurer et al*. [2001], and *Lohmann et al*. [2004]
Mississippi River at Saint Paul, Minnesota	05331000	−93.09	44.94	NAD83	95,311	Close to large metropolitan area, centennial gauge	
Mississippi River at Keokuk, Iowa	05474500	−91.37	40.39	NAD27	308,207		*Maurer et al*. [2001] and *Lohmann et al*. [2004]
Mississippi River at Grafton, Illinois	05587450	−90.43	38.97	NAD83	443,663	Downstream of confluence with Illinois River	
Mississippi River at Saint Louis, Missouri	07010000	−90.18	38.63	NAD83	1,805,213	Centennial gauge, downstream of confluence with Missouri River	
Mississippi River at Thebes, Illinois	07022000	−89.46	37.22	NAD83	1,847,170	Upstream of confluence with Ohio River	*David et al*. [2013a]
Ohio River at Sewickley, Pennsylvania	03086000	−80.21	40.55	NAD27	50,505	Close to large metropolitan area (Pittsburg)	
Ohio River at Louisville, Kentucky	03294500	−85.80	38.28	NAD27	236,128	Close to large metropolitan area	
Ohio River at Metropolis, Ohio	03611500	−88.74	37.15	NAD27	525,765	Downstream of confluence with Tennessee River, upstream of confluence with Mississippi River	*Maurer et al*. [2001] and *Lohmann et al*. [2004]
Arkansas River near Haskell, Oklahoma	07165570	−95.64	35.82	NAD83	195,007		*Abdulla et al*. [1996]
Arkansas River at Murray Dam near Little Rock, Arkansas	07263450	−92.36	34.79	NAD83	409,574		Close to Arkansas River at Little Rock, Arkansas [*Abdulla et al*., 1996; *Lohmann et al*., 1998]
Red River at Spring Bank, Arkansas	07344370	−93.86	33.09	NAD83	N/A		Close to Red River at Shreveport, Louisiana [*Abdulla et al*., 1996; *Lohmann et al*., 1998]

$$\begin{gathered}\forall j \in [1, 102229] \\ k_j^\alpha = \lambda_k^\alpha \cdot \frac{L_j}{c^0} \\ x_j^\alpha = \lambda_x^\alpha \cdot 0.1\end{gathered} \tag{10}$$

where k_j^α (time) and x_j^α (nondimensional) are the traditional parameters of the Muskingum method for each river reach j and used to compute the elements C_{1_j}, C_{2_j}, and C_{3_j}. L_j is the reach length. $c^0 = 1$ km×h^{-1} = 0.28 m×s^{-1} is a constant flow wave celerity. The year 2008 was chosen as the optimization period because it is the wettest year out of the proposed simulation period. In each experiment, six sets of initial values for $\left[\lambda_k^\alpha; \lambda_x^\alpha\right]$ were used ([2; 3], [4; 1], [1; 1], [0.3; 3], [0.5; 3], and [0.7; 3]), and only the values that had a physical meaning (i.e., $\lambda_k^\alpha > 0$ and $\lambda_x^\alpha \in [0, 5]$) [*Cunge*, 1969] obtained throughout the search were retained. Finally, the couple $\left[\lambda_k^\alpha; \lambda_x^\alpha\right]$ leading to the smallest value of the square error cost function was kept as optimal.

An initial optimization using all available gauging stations led to

$$\begin{gathered}\forall j \in [1, 102229] \\ \lambda_k^{\alpha,0} = 0.406250, \qquad k_j^{\alpha,0} = \lambda_k^{\alpha,0} \cdot \frac{L_j}{c^0} \\ \lambda_x^{\alpha,0} = 0.296875, \qquad x_j^{\alpha,0} = \lambda_x^{\alpha,0} \cdot 0.1\end{gathered} \tag{11}$$

This corresponds to a flow wave celerity of $c^{\alpha,0} = c^0/0.406250 = 0.68$ m×s^{-1} which is comparable to the results of previous studies also using the VIC model for runoff production [*David et al*., 2013a, 2013b].

As in *David et al*. [2013b], a second optimization using only the stations leading to a positive value of the efficiency [*Nash and Sutcliffe*, 1970] was performed, and led to

$$\begin{gathered}\forall j \in [1, 102229] \\ \lambda_k^{\alpha,1} = 0.210876, \qquad k_j^{\alpha,1} = \lambda_k^{\alpha,1} \cdot \frac{L_j}{c^0} \\ \lambda_x^{\alpha,1} = 0.341400, \qquad x_j^{\alpha,1} = \lambda_x^{\alpha,1} \cdot 0.1\end{gathered} \tag{12}$$

Here the flow wave celerity of $c^{\alpha,0} = c^0/0.210876 = 1.32$ m×s^{-1} is about twice as fast as in previous studies.

Because faster flow waves lead to better timing (Appendix A), and because the speed of computations is more crucial for emergency management when flow waves are moving faster, the remainder of the study will focus on the parameters $\left[\lambda_k^{\alpha,1}; \lambda_x^{\alpha,1}\right]$ corresponding to the faster flow wave.

5.3. Radius of Influence

Following *David et al*. [2013a], a conservative estimate for the radius of downstream influence R_j^{down} for each river reach was computed using the worst-case values corresponding to the data $\mathbf{b}(t)$ and to the unknown $\mathbf{Q}(t+\Delta t)$ for any 30 min time step during the 10 year simulation. Figure 4 shows the spatial distribution of R_j^{down} which varies between 0 and $R_{\max}^{down} = 50$. Multiplying the radius of downstream influence by the mean river reach length leads to a distance of 320 km. A previous estimate of the radius of downstream influence was 155 reaches for the Upper Mississippi Basin at a 15 min time step, with a mean river reach length of 1.87 km, and using a flow wave celerity of 0.78 m/s [*David et al*., 2013a]. Note that despite different values for the flow wave celerity and for the routing time step in the previous study, multiplying the radius of downstream influence by the mean river reach length leads to approximately 290 km. The similarity of these two distances is striking although it is not clear whether such resemblance is due to the common geographical location, a yet-to-be-determined physical quantity, or chance.

Regardless, one can expect that simulations in this study using the trans-boundary Muskingum method with large subbasins approximation will provide accurate results—at least—when the topologic distance separating any two connected boundary reaches of any subbasin is greater than 50 (i.e., $D_k > R_{\max}^{down}$).

5.4. River Network Decomposition for Parallel Computing

The methods presented in section 3.6 for sorting and decomposing the river network into subbasins were applied on the HydroSHEDS representation of the Mississippi River Basin to 1, 2, 4, 8, 16, and 32 computing cores.

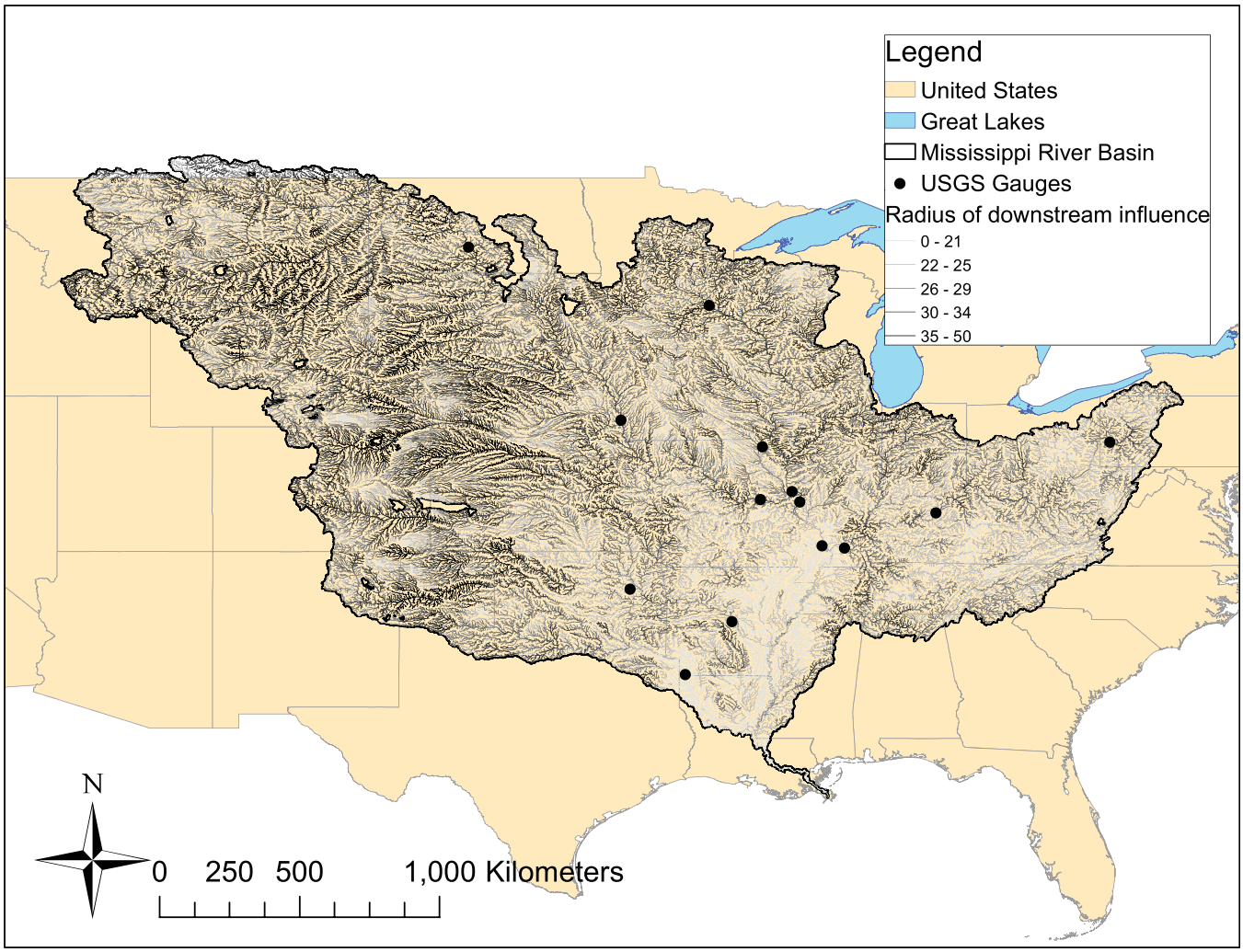

Figure 4. The radius of downstream influence varies between 0 and 50 river reaches.

For each domain decomposition and for each subbasin k, a conservative estimate of the topological distance D_k was obtained by subtracting the minimum topological path length of any reach of k that has a connection with any upstream subbasin from the maximum topological path length of any reach of k that has a connection with any downstream subbasin. This simplified estimate allows avoiding the need for tracing connectivity among all reaches of a given subbasin which is computationally demanding. A value of $D_k = \infty$ was used if a subbasin has no upstream connection, or if a subbasin has no downstream connection. The conservative estimate of the topological distance corresponding to each given decomposition was then obtained by taking the minimum of all estimated values of D_k.

Table 2 shows the load imbalance, the conservative estimate of the topological size D_k, and the number of elements in the trans-boundary matrix **T**; for all domain decompositions used in this study. Figure 5 shows the domain decomposition obtained for 4, 8, 16, and 32 computing cores.

Given the conservative estimates for $R_{\max}^{down}$ and D_k presented above, one should expect that the trans-boundary Muskingum method with large subbasins approximation will provide accurate results when running on 1, 2, 4, and 8 computing cores because $D_k > R_{\max}^{down}$ (Table 2). Note that the proposed sorting and domain decomposition schemes lead to $D_k = \infty$ on 1 and 2 computing cores because no single subbasin has both upstream and downstream connections with other subbasins. Simulations run on 16 or 32 cores may lead to accurate computations but such cannot be guaranteed given that the estimated D_k is larger than the estimated $R_{\max}^{down}$.

Table 2. Information Related to the Domain Decomposition Over 1–32 Computing Cores Corresponding to the HydroSHEDS River Network of the Mississippi River Basin

Number of Cores Used (N)	Heaviest Load	Lightest Load	Load Imbalance (L_N)	Conservative Estimate of Minimum Topological Size (D_k) in Any Subbasin	Number of Elements in T
1	102,229	102,229	1.00	∞	0
2	51,115	51,114	1.00	∞	199
4	25,558	25,557	1.00	133	483
8	12,779	12,778	1.00	59	1029
16	6,390	6,389	1.00	27	2282
32	3,195	3,194	1.00	12	4424

Note that Table 2 also shows that the total number of elements in **T** increases almost linearly with the number of computing cores until it becomes greater than the number of river reaches in any subbasin when running on 32 computing cores. One can therefore expect (see section 3.5) that intercore communication might limit fixed-size speedup capabilities with the domain decomposition used here. Such limitation should become particularly overwhelming when the total number of river reaches involved in intercore communication (the trans-boundary reaches) becomes greater than the computing load addressed by any given core. An alternative way to avoid this large amount of intercore communication while keeping

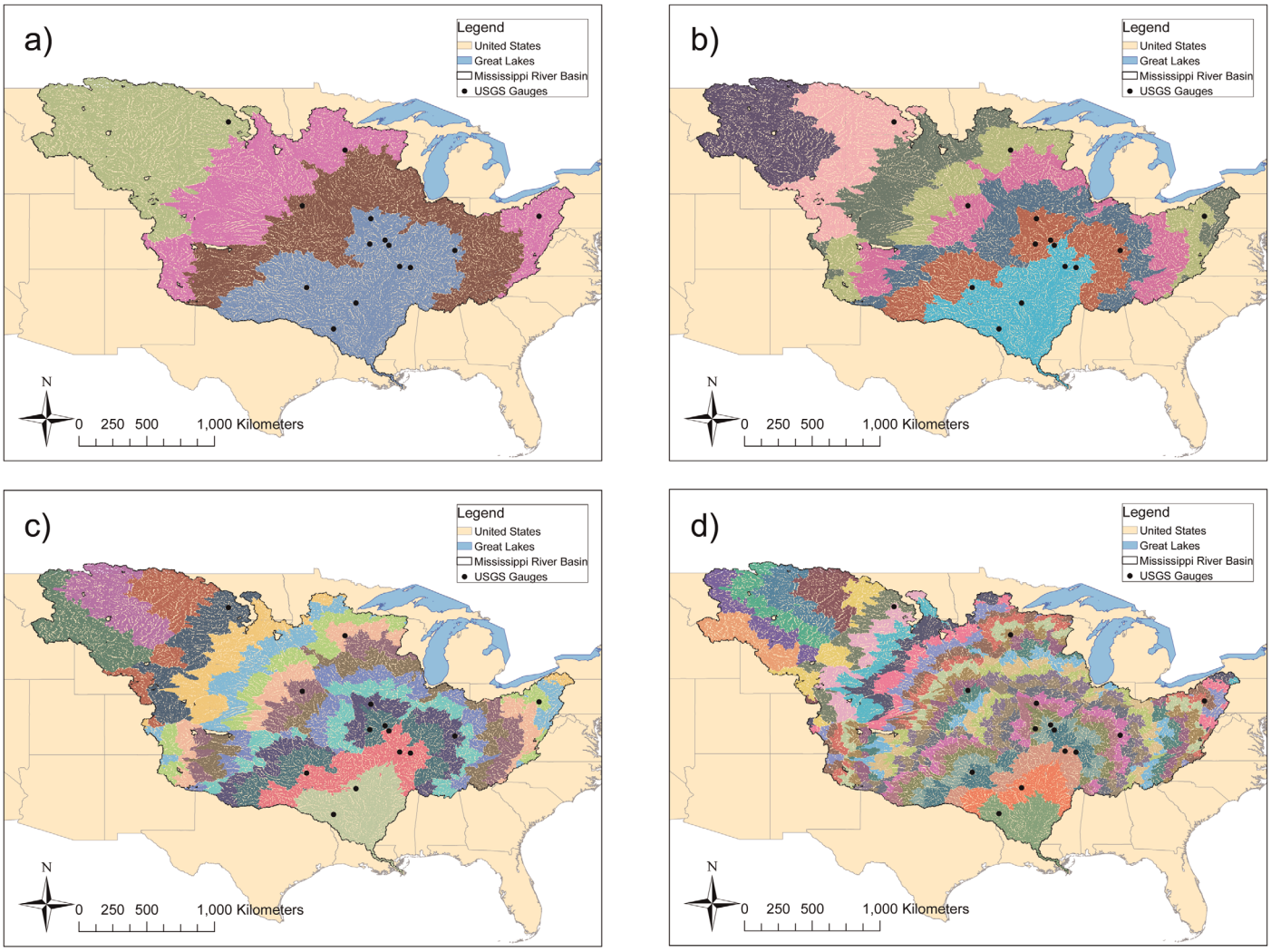

Figure 5. In this study, the domain decomposition for reaches in the HydrosHEDS river network of the Mississippi River Basin is based on the topological path length and, is balanced. Each color represents a different computing core for each decomposition on (a) 4 cores, (b) 8 cores, (c) 16 cores, and (d) 32 cores.

Table 3. Differences in 3 Hourly Averaged Results Among Computing Methods Used

Method 1		Method 2		Maximum Value of Relative Difference in the Flow Rate Vector Using the Two-Norm Between Method 1 and Method 2, for Any 3 Hourly Average During Entire Simulation (Dimensionless)	Maximum Value of Absolute Difference in Flow Rate Between Method 1 and Method 2, for Any River Reach and for Any 3 Hourly Average During Entire Simulation (m^3/s)
Method Name	Number of Computing Cores Used	Method Name	Number of Computing Cores Used		
Matrix-based Muskingum	1	Traditional Muskingum	1	1.83E-15	2.76E-10
Matrix-based Muskingum	1	Iterative Matrix-based Muskingum	1	0.00E+00	0.00E+00
Matrix-based Muskingum	1	Iterative Matrix-based Muskingum	2	2.05E-15	2.76E-10
Matrix-based Muskingum	1	Iterative Matrix-based Muskingum	4	1.95E-15	2.76E-10
Matrix-based Muskingum	1	Iterative Matrix-based Muskingum	8	1.95E-15	2.76E-10
Matrix-based Muskingum	1	Iterative Matrix-based Muskingum	16	2.08E-15	2.91E-10
Matrix-based Muskingum	1	Iterative Matrix-based Muskingum	32	1.77E-09	9.88E-05
Matrix-based Muskingum	1	Trans-boundary Muskingum with large subbasins approximation	1	0.00E+00	0.00E+00
Matrix-based Muskingum	1	Trans-boundary Muskingum with large subbasins approximation	2	1.04E-15	9.46E-11
Matrix-based Muskingum	1	Trans-boundary Muskingum with large subbasins approximation	4	8.22E-16	1.46E-10
Matrix-based Muskingum	1	Trans-boundary Muskingum with large subbasins approximation	8	8.04E-16	1.60E-10
Matrix-based Muskingum	1	Trans-boundary Muskingum with large subbasins approximation	16	8.80E-16	2.04E-10
Matrix-based Muskingum	1	Trans-boundary Muskingum with large subbasins approximation	32	1.77E-09	9.88E-05

subbasins as large as possible in order to satisfy constrains related to the radius of influence would be to enforce contiguous subbasins in the domain decomposition. Such was attempted using METIS [*Karypis and Kumar*, 1999] as part of this study. However, the resulting domain decompositions are heavily unbalanced ($L_N=1.53$) as was previously encountered by *Vivoni et al*. [2011], hence justifying the approach used here.

Finally, Figure 5 shows that in all decompositions of the Mississippi River Basin used, all subbasins have connections with upstream reaches except the upstream-most subbasin. This upstream-most subbasin is addressed by the first computing core and is the only one that could be solved in one unique iteration with the algorithm proposed. Idling—or in our method: unnecessary duplicate operations (see section 3.4)—is therefore limited to only one core here.

5.5. Comparison Among Parallel Computations

The minimum and maximum singular values of the linear system matrix of equation (1) and built with the Muskingum parameters of equation (12) were obtained using the Scalable Library for Eigenvalue Problem Computations (SLEPc) [*Hernandez et al*., 2005] and allow computing the condition number of $\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N}$:

$$\kappa_{\|\|_2}(\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N})=\frac{\sigma_{\max}}{\sigma_{\min}}\approx\frac{1.8655}{0.4302}\approx 4.34 \tag{13}$$

Applying the concepts presented in section 3.7 to double-precision floating-point operations ($\varepsilon=1.11\times10^{-16}$), relative errors in $\|\mathbf{b}\|_2$ and $\|\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N}\|_2$on the order of ε will lead to relative differences in flow rate computations on the order of $\|\Delta\mathbf{Q}\|_2/\|\mathbf{Q}\|_2=2\cdot\kappa\cdot\varepsilon/(1-\kappa\cdot\varepsilon)\approx 9.63\times10^{-16}$. Differences of such magnitude among results obtained with various resolution schemes are hence also to be deemed expected and acceptable.

Table 3 shows the differences obtained among 3 hourly averaged output files for a series of resolution schemes including the traditional Muskingum method on one core [*McCarthy*, 1938], the matrix-based Muskingum method of equation (1) solved directly on one core [*David et al*., 2011b], the matrix-based Muskingum method solved iteratively on 1–32 cores [*David et al*., 2011b, 2013a], and the trans-boundary Muskingum method with large subbasins approximation solved with two iterations on 1–32 cores (this study).

As expected, the trans-boundary Muskingum method with large subbasins approximation provides results within the acceptable accuracy on 1–8 cores. In addition, Table 3 shows that the results obtained on 16 cores are also within the expected range. This suggests that the combined conservative estimates of $R_{\max}^{down}$ and D_k provide a safe approach to determining whether or not subbasins are large enough to warrant the use of the proposed trans-boundary Muskingum method. However, the differences in results obtained on 1

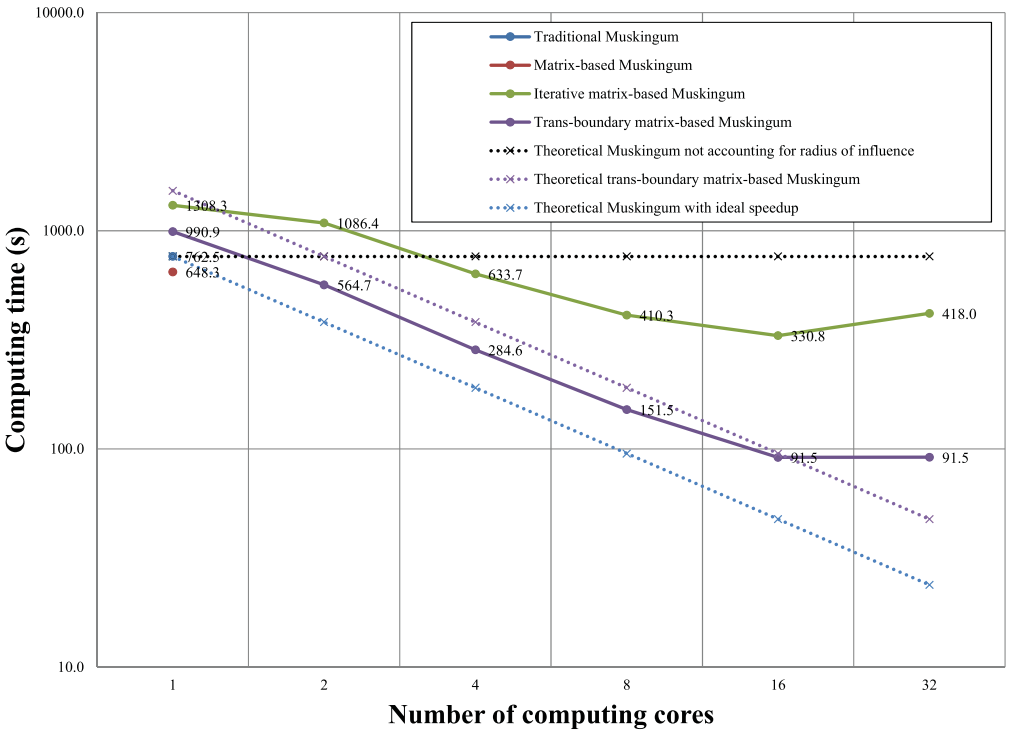

Figure 6. Computing time as a function of the number of computing cores for four algorithms all based on the Muskingum method and for three theoretical curves.

and 32 cores is six orders of magnitude higher than differences obtained among other simulations suggesting that the subbasin sizes are no longer large enough with regards to the radius of downstream influence. Even then, the maximum value of the absolute error made on any river reach is on the order of $10^{-6}m^3 \times s^{-1}$ which is much smaller than the model bias or the accuracy resolved by gauging stations. Note that, surprisingly, the proposed trans-boundary method also leads to smaller differences with the matrix-based Muskingum method than the iterative matrix-based method.

Addressing concerns related to the precision of results is crucial in this study because the proposed algorithm is developed based on the assumption that subbasins are large enough with regards to the radius of influence. The experimental differences obtained here among multiple simulations are always within the magnitude predicted by floating-point arithmetic as long as this assumption is satisfied. It can therefore be concluded that the algorithm developed in this study conserves mass and can be used safely. Even when the large subbasin assumption fails, the differences among results are several orders of magnitude smaller than the model bias and the accuracy of observations. The risks of failing mass conservation are hence inexistent when the assumption of the algorithm is verified, and are very limited when the assumption fails. We also expect that such precision of results will persist even if replacing modeled flows by observed flows at a few locations within the river network—as done in *David et al.* [2011a]—because the upstream-to-downstream connectivity would purposely be broken at these selected locations hence further relaxing the assumption of the algorithm.

5.6. Fixed-Size Speedup

The computing times related to each experiment of Table 3 are shown in Figure 6. Times corresponding to three theoretical curves are also included in Figure 6: a null fixed-size speedup resulting from not accounting for the existence of radii of downstream influence [*David et al.*, 2013a], the fixed-size speedup of equation (7), and a hypothetical ideal fixed-sized speedup. For full disclosure [e.g., *Bailey*, 1992], all simulations share the following features: double-precision floating points were used, all results presented are from actual numerical experiments (no projections were made except for theoretical values), only the times corresponding to the slowest of all cores (not the average times) are reported, compiler optimization options were always the same, and a unique but shared 32 core computer was used with null to minimal impact from other users. As in our previous studies [*David et al.*, 2011b, 2013a] all computing times in Figure 6 correspond exclusively to

solving the routing equations—i.e., the focus of this study—and do not include initial model setup, although a common setup program is used in all experiments. Finally, note that in order to provide fair comparisons, all theoretical and experimental fixed-size speedup values reported in this study are calculated against the computing time of the traditional Muskingum method with no parallel overhead.

As in previous studies [*David et al*., 2011b, 2013a], the traditional Muskingum method and the matrix-based Muskingum method can be solved in comparable times (Figure 6). Also like in our previous research, the iterative matrix-based Muskingum method suffers from an initial handicap factor of approximately 2 (Figure 6) from an added iteration and from the computation of the initial error necessary in automated iterative methods; and shows limited fixed-size speedup from 1 to 16 cores after which communications likely become overwhelming [*David et al*., 2011b].

As expected, the trans-boundary Muskingum method is more computationally expensive than the matrix-based Muskingum method when running on one unique computing core (Figure 6) although experimental results show a factor of 1.5 (not 2) in computing time. This suggests that the added burden from doubling the number of linear systems to be solved in the trans-boundary Muskingum method therefore does not lead to a doubling in computation time, which is advantageous compared to our theoretical estimate of equation (7).

More importantly, Figure 6 shows that the factor of 1.5 is mostly stable as the number of cores increases, and the computing time of the trans-boundary Muskingum method decreases almost linearly going from 1 to 16 computing cores until it reaches a fixed-size speedup of 8.3. Despite being inferior to a hypothetical ideal speedup, the experimental results of the trans-boundary approach consistently outperform the theoretical estimate of equation (7). One must emphasize here that the Muskingum method is in its defining equation heavily serialized, which is why Muskingum-type computations have traditionally been solved in sequence from upstream to downstream. This topological limitation in turn has direct consequences on fixed-size speedup capabilities unless alternative algorithms are used. In this study, the largest number of subbasins (i.e., computing cores) crossed—out of all possible paths going from upstream to downstream—in the entire river network is always equal to the total number of cores. If one were to solve the Muskingum equations separately on each subbasin with the static domain decomposition developed here, each core would have to wait for all upstream computations to be completed prior to starting its task resulting in no gains in time at all. The reader is referred to *David et al*. [2013a] for further information on this topological limitation. The mere fact that experimental fixed-size speedup is observed in this study can hence be considered a success. Additionally, our experimental fixed-size speedup results outperform the theoretical estimates of equation (7) over 1–16 cores.

To the best of our knowledge, such values obtained for fixed-size speedup with the Muskingum method over 1–16 cores using a static domain decomposition over a fully interconnected river basin are unprecedented. There remains, however, a strong limitation to this study. Despite providing balanced subbasins, the domain decomposition approach used here triggers an increasing need for intercore communication (related to the number of elements in $\mathbf{T}$) when the number of cores increases (Table 2). The increasing population size of the trans-boundary matrix might be responsible for the slope of the trend being less than one. Starting at 32 cores, the fixed-size speedup of the trans-boundary Muskingum stalls (in addition to being inaccurate) which is likely due to the large number of elements in the trans-boundary matrix $\mathbf{T}$ compared to the computing load of each core (see sections 3.5 and 5.4). The investigation of balanced domain decomposition methods allowing for large subbasins and for limited connections among subbasins would therefore be valuable as they should alleviate the flattening on the speedup curve.

Finally, the main characteristics of the matrices $\mathbf{I}-\mathbf{C_1}\cdot\mathbf{N}$ and $\mathbf{I}-\mathbf{C_1}\cdot(\mathbf{N}-\mathbf{T})$, that are used, respectively, in the matrix-based Muskingum method and in the trans-boundary matrix-based Muskingum method, do not change with the problem size. Their lower unit triangular shape, the total number of nonzero elements per row, and their ability to be solved in a few iterations has been stable in all existing RAPID studies [*David et al*., 2011a, 2011b, 2013a, 2013b]. One could therefore expect that the mathematical approaches used in this study would behave well from a scaled-size speedup perspective, although such investigation is admittedly beyond the scope of this study.

5.7. Implications for the Study of Riverine Water Resources

The flow of water in rivers is a causal phenomenon that mostly goes downstream and many river routing schemes—including the Muskingum method—update a given river reach partly based on the prior update

of all upstream river reaches [*David et al*., 2013a]. The most intuitive way to solve such schemes is hence to proceed sequentially from upstream to downstream (as done traditionally) because the equations suggest that upstream river reaches must be updated prior to downstream reaches. This topological constraint limits one's ability to achieve any fixed-size parallel speedup when using traditional solving methods. In fact, attempting to obtain fixed-size parallel speedup for static orchestration paradigms in which each computing core is assigned a unique subbasin was until recently thought to be an unworthy endeavor for the Muskingum method [*David et al*., 2011b, 2013a]. This study avoids the perceived topological constraint by building an iterative method informed by the limited upstream-to-downstream propagation of flow waves. The results obtained here not only show experimental fixed-size speedup; they also outperform the theoretical speedup of equation (7) when applied from 1 to 16 cores (Figure 6). Further, the theoretical speedup of equation (7) is much beyond what was previously thought achievable [*David et al*., 2013a]. This implies that new paradigms can be developed for solving other similar flow wave propagation algorithms in order to obtain results faster.

The only assumption of the solving algorithm is that subbasins are large enough when compared to the radius of downstream influence. The domain decomposition method presented here satisfies this condition and is also advantageous because only one computing core ever idles (corresponding to the upstream-most subbasin) and its idling only lasts for half of the simulation. Despite admitted limitations due to increasing communication that seems to limit performance starting at 32 cores, the domain decomposition method used here allows demonstrating the validity of the solving algorithm. Note however that, while their joint use in this study allows for a proof of concept, the computational algorithm and the domain decomposition method are independent. Further study of decomposition approaches could therefore help alleviating the limited performance starting at 32 cores obtained here.

The work presented in this study uses a 10 year study of the Mississippi River Basin. The timing of numerical experiments demonstrates that the new computation method allows saving a factor of 8, i.e., almost an order of magnitude, while conserving expected numerical precision. This suggests that substantial gains in simulation time can be obtained when performing retrospective studies of the world's largest river basins. Additionally, such temporal gains promise to be valuable for time-critical operational flood forecasting efforts such as the National Flood Interoperability Experiment [*Maidment*, 2015] in which ensembles of RAPID simulations are used (A. D. Snow et al., A new high-resolution national-scale ensemble hydrologic forecast model and dataset, submitted to *Journal of American Water Resources and Association*, 2015). The research in this paper therefore has implications on the study of the world's largest river basins both for retrospective studies and for large ensemble predictions.

6. Conclusion

The main purpose of this study is to develop and assess a new algorithm for river routing with the Muskingum method that allows for large gains in computing time when run on parallel computers. The method presented builds on recent work demonstrating that the flows in distant upstream reaches have an influence on the Muskingum update of flow in a given river reach that is so small it cannot be accounted for by floating-point operations in computers [*David et al*., 2013a]. To the best of our knowledge, the fixed-size speedup performance of the proposed trans-boundary Muskingum method is unprecedented for a fully interconnected river basin with static orchestrations, and should allow large gains in computing time when studying the world's largest river basins retrospectively as done here or for real-time flood forecasting [e.g., *Maidment*, 2015]. Our mathematical approach uses a matrix-vector notation because such allows easily describing the trans-boundary flows using what is referred to here as a trans-boundary matrix. However, a similar concept can be applied to the traditional algorithm used in the Muskingum method or to the many other routing algorithms in which flow rates are updated sequentially from upstream to downstream.

The foundation of the proposed trans-boundary approach to the Muskingum method lies in the assumption that a large interconnected network of river reaches can be decomposed into subbasins that are large enough with regards to the radius of downstream influence. Such constraints on domain decomposition need be considered with two other crucial aspects of parallel computing:

load balancing and intercore communication. This study uses the topological path length to sort and decompose large river networks into subbasins of balanced size. However, the resulting subbasins have a small topological size which limits the number of computing cores that can effectively be used for resolution with the trans-boundary approach. Additionally, the large number of trans-boundary reaches generates much intercore communication which also likely limits fixed-size parallel speedup capabilities. Our work therefore only provides an initial step towards understanding best-suited river network decomposition methods and suggests that more advanced techniques may lead to further fixed-size speedup performance. Such advanced domain decompositions can be expected to be valuable to other river routing algorithms.

Our quantitative comparison of estimated values for the radius of downstream influence and for the topological distance separating consecutive subbasins sheds some light on when and why the proposed trans-boundary method works. However, the calculation of such quantities is computationally intensive, and remains approximate. Future studies may consider instead a trial-and-error method in which results obtained among various domain decompositions are compared to determine when subbasins become too small to warrant the use of the trans-boundary Muskingum method.

This research also presents a similarity in the numerical value of a distance (300 km)—obtained by multiplying the radius of downstream influence (a number of river reaches) by the mean length of the river reaches—in this study and a previous study [*David et al*., 2013a], despite different spatial and temporal resolutions. It may be of interest to further study the meaning of this distance to see if hard limits exist in parallel computing of river flow based on a physical distance that does not depend on resolution.

Appendix A: River Flow Simulations

Table A1 shows the statistics of the simulations obtained with the two sets of parameters given in equations (11) and (12) and includes the mean observed flow, the mean simulated flow, the root-mean-square error (RMSE) and the efficiency [*Nash and Sutcliffe*, 1970] for both sets of model parameters used. The statistics of Table A1 were computed using daily data for the entire period of simulation (1 January 2000 to 31 December 2009). Figure A1 shows daily hydrographs for the driest year (2000) of the 10 year study and corresponding to the seven stations with the highest observed mean flow; and Figure A2 shows the same hydrographs for the wettest year (2008).

The flow statistics (Table A1) obtained for the gauging stations located on the main stem of the Mississippi River in the northern part of the Upper Mississippi River Basin (i.e., St. Paul, MN, Keokuk, IA, and Grafton, IL) show good agreement with observations when the slower flow wave celerity is used.

Table A1. Statistics of Flow Simulations Obtained For the Gauges of Table 1 With Two Sets of Parameters Used in RAPID

		Initial Optimization ($\lambda_k^{\alpha,0}$ and $\lambda_x^{\alpha,0}$)			Subsequent Optimization ($\lambda_k^{\alpha,1}$ and $\lambda_x^{\alpha,1}$)		
Name	Mean Flow (Observation)	Mean Flow (Model)	RMSE	E	Mean Flow (Model)	RMSE	E
Missouri River at Bismarck, North Dakota	466.9	907.3	843.4	−55.32	908.5	885.8	−61.14
Missouri River at Omaha, Nebraska	755.6	1,919.2	1809.9	−50.45	1,921.8	2000.6	−61.87
Missouri River at Hermann, Missouri	2083.1	4,969.0	4048.3	−7.23	4,976.8	4276.9	−8.19
Mississippi River at Saint Paul, Minnesota	401.5	477.0	380.0	0.24	477.1	520.3	−0.42
Mississippi River at Keokuk, Iowa	2188.4	2,974.8	1299.4	0.26	2,977.9	1539.4	−0.04
Mississippi River at Grafton, Illinois	3417.3	4,439.1	1678.7	0.46	4,445.6	1962.6	0.27
Mississippi River at Saint Louis, Missouri	5546.1	9,504.1	5381.2	−1.37	9,520.4	5617.9	−1.58
Mississippi River at Thebes, Illinois	6078.5	10,237.9	5669.8	−1.20	10,257.4	5788.0	−1.30
Ohio River at Sewickley, Pennsylvania	989.5	1,002.6	573.9	0.53	1,003.3	457.1	0.70
Ohio River at Louisville, Kentucky	3622.7	4,357.8	2695.2	0.28	4,369.3	2170.3	0.54
Ohio River at Metropolis, Ohio	8127.9	10,375.1	4580.0	0.42	10,413.0	3577.2	0.64
Arkansas River near Haskell, Oklahoma	303.4	1,019.5	997.3	−6.49	1,020.3	1030.7	−7.00
Arkansas River at Murray Dam near Little Rock, Arkansas	1294.1	2,986.1	2355.2	−1.77	2,990.0	2442.9	−1.98
Red River at Spring Bank, Arkansas	588.4	1,418.0	1195.6	−1.73	1,420.0	1310.5	−2.28

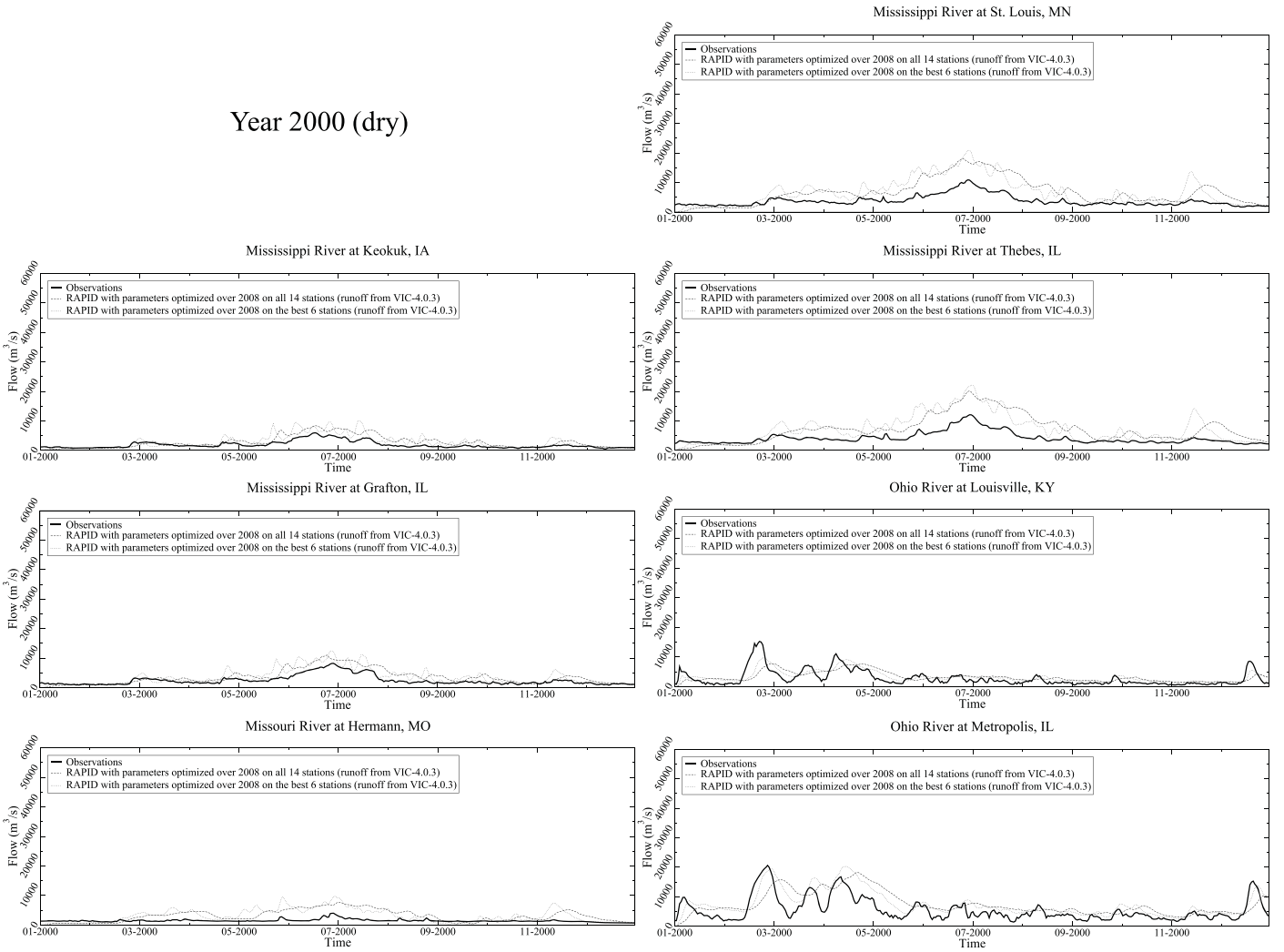

Figure A1. Comparison of daily averaged observed hydrographs with those obtained by RAPID with the two sets of parameters used in this study, for the year 2000 (dry year), and for the seven stations with highest 10 year average flow.

The corresponding efficiencies range between 0.24 and 0.46 and the mean flow is well captured by the modeling system. The faster flow wave celerity leads to poorer results. The comparison of hydrographs (Figures A1 and A2) at the nearby stations of Keokuk, IA (upstream) and Grafton, IL (downstream) show that the downstream flow is much more diffused in the observations than in the model results. Such is particularly apparent during the high flows of June 2008 (Figure A2). The station located on the Mississippi River at Grafton, IL is located just downstream of the confluence of the Illinois River and the Mississippi River. Hence, such enhanced diffusive processes may be related to backwater flows at the confluence, but such backflows cannot be represented by the traditional nor the variable-parameter Muskingum methods [*Todini*, 2007].

The stations on the main stem of the Missouri River (Bismarck, ND, Omaha, NE, and Hermann, MO) consistently show poor flow statistics (Table A1) regardless of which wave celerity is used. The value of the mean modeled flow is approximately twice that of the mean observed flow for these three locations. As a consequence, the efficiencies are always substantially negative, hence suggesting poor simulations. The study of all corresponding hydrographs (Figures A1 and A2) shows that the flow is much more dampened in the observations than in the simulations. Such dampening suggests that long residence times are present in the Missouri River Basin. Figure A3 focuses on the Missouri River Basin and shows that four of the fifteen largest man-made reservoirs of the U.S. (Lake Sakakawea,

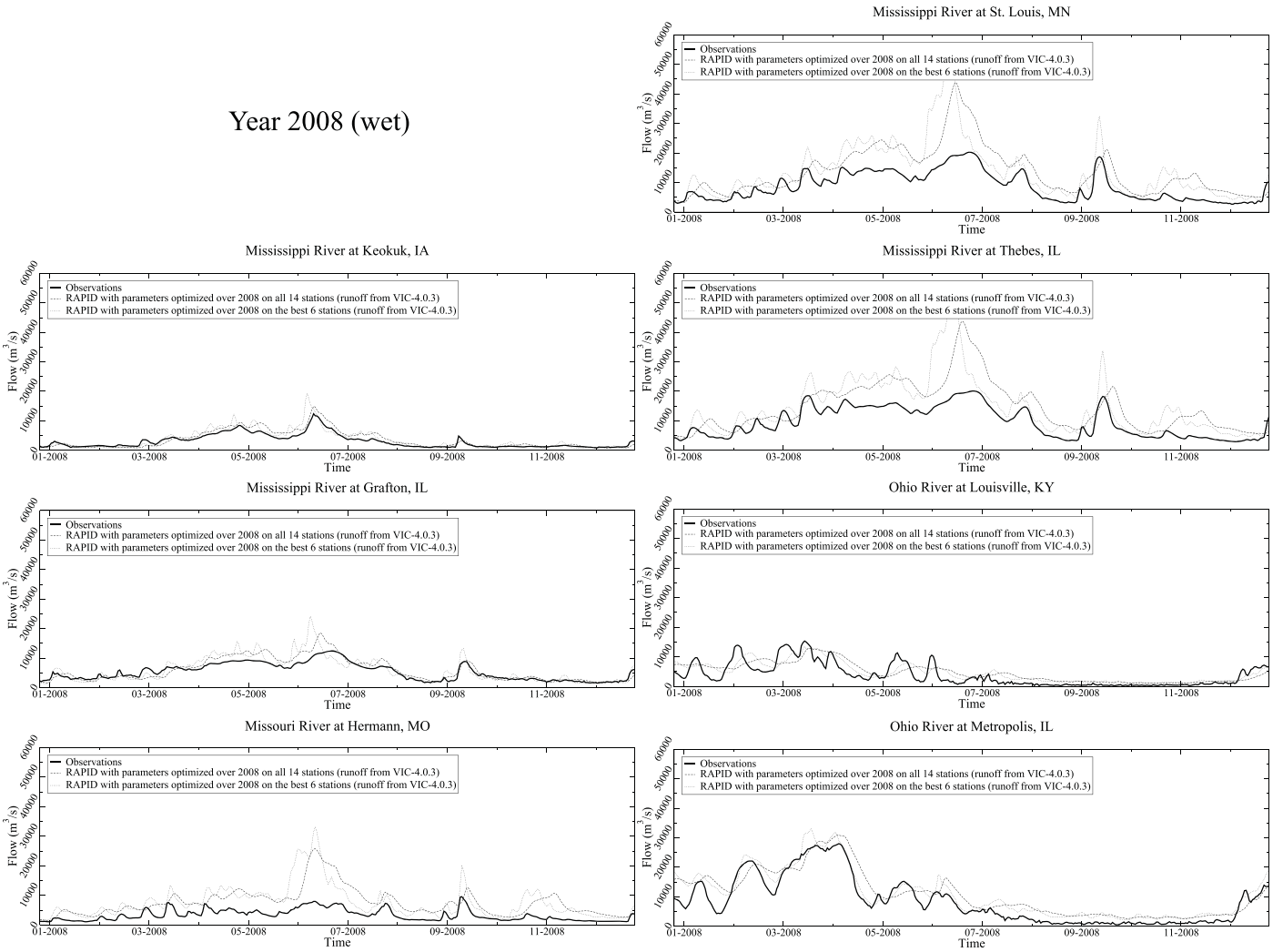

Figure A2. Same as Figure A1 but for 2008 (wet year).

Lake Oahe, Fort Peck Lake, and Lake Francis Case) are located upstream of the Missouri River at Omaha, NE and five are upstream of Hermann, MO (Truman Reservoir added to the above list). The lack of representation of storage processes in reservoirs within the modeling system used in this study can therefore partly explain the poor dynamics of simulations in the Missouri River Basin. Additionally, enhanced evaporation from large open water bodies may reveal potential causes for the wide overestimation of the mean flow in the model compared to observations. The examination of the timing of observed and modeled peaks (Figures A1 and A2), however, suggests that the faster flow wave celerity is the most appropriate for the Missouri River at Hermann, MO.

Approximately, 40% of the flow measured downstream of the confluence of the Mississippi River and the Missouri River (at St. Louis, MO) is coming from the Missouri River (Table A1). Therefore, the poor model simulations of the Missouri River have a direct impact on the modeled hydrographs (Figures A1 and A2) and corresponding statistics (overestimated mean and negative efficiencies in Table A1) for the Mississippi River at St. Louis, MO and at Thebes, IL.

The simulations of the Ohio River appear to be the best in the study domain. Efficiencies for the Ohio River at Sewickley, PA, Louisville, KY, and Metropolis, IL (Table A1) are all positive regardless of which of the two values of wave celerity is used here, although higher efficiencies are obtained with the faster wave (Table A1) and better hydrograph dynamics (Figures A1 and A2).

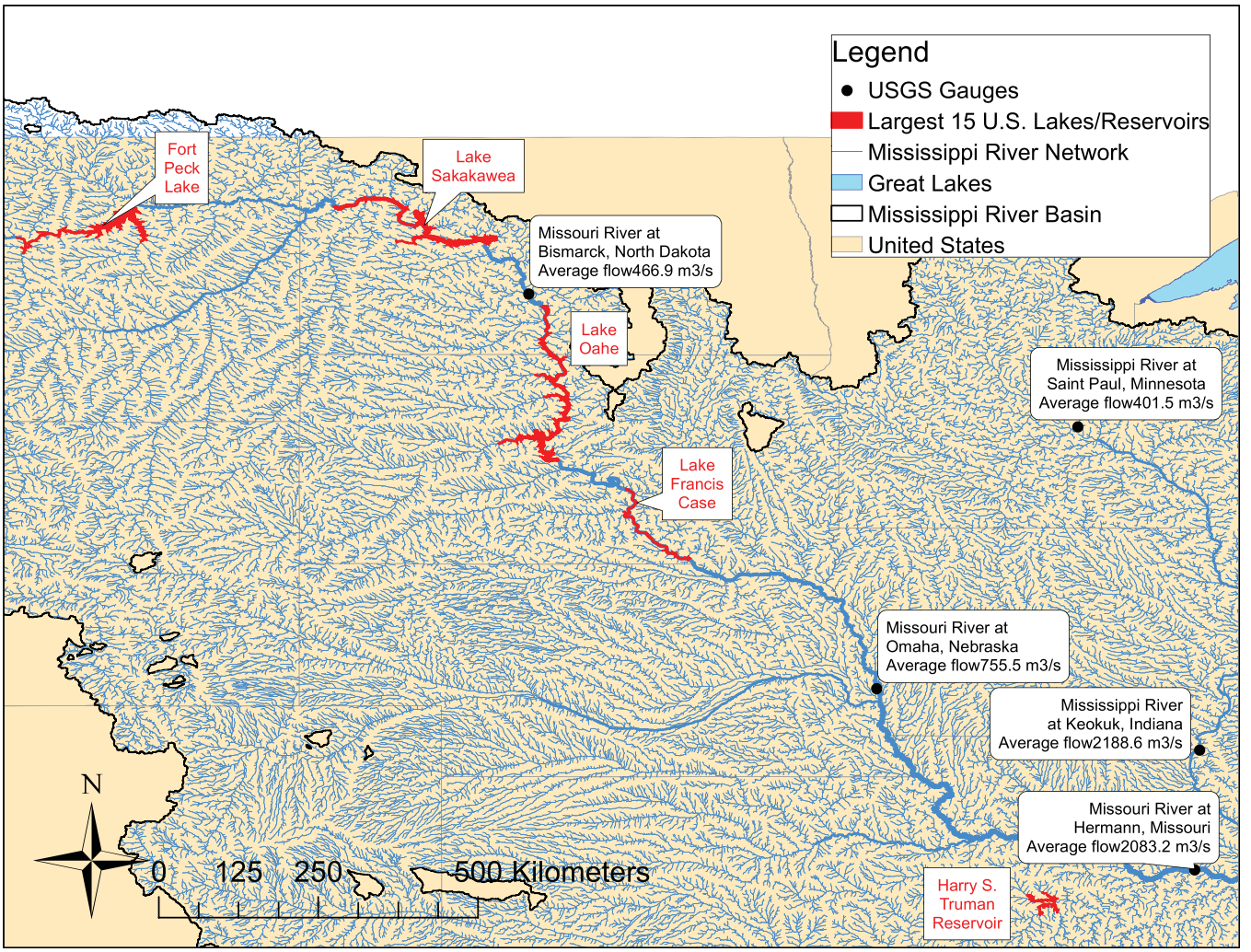

Figure A3. Five of the 15 largest U.S. lakes and reservoirs are located in the Missouri River Basin.

The flow statistics (Table A1) for the Arkansas River (Haskell, OK and Little Rock, AR) and Red River (Spring Bank, AR) show that the simulated mean flow is approximately 3 times higher than the observed mean flow leading to poor efficiencies of simulations. Here again, one must note the presence of enhanced evaporation and storage capacity in surface water bodies such as Lake Dardanelle (Arkansas River) and Lake Texoma (Red River) that could explain the poor simulations.

Therefore, the examination of flow statistics (Table A1) and flow hydrographs (Figures A1 and A2) suggests that the modeling system has some skill in reproducing the daily variability of observed river flow despite significant limitations in locations where storage in surface water reservoirs—currently lacking from the modeling system—has strong impacts on surface water dynamics. Additionally, a key contributor to reproducing the observed hydrograph is the runoff generated by the land surface model. No attempt is made here to improve the parameterization of runoff in land models [e.g., *Famiglietti and Wood*, 1994; *Niu et al*., 2011; *Cai et al*., 2014] as this is far beyond the scope of the current paper, which is strictly focused on improving the computational efficiency of the routing model. Note that, contrary to previous work [*David et al*., 2013b], the optimization of the Muskingum parameters focusing only on the stations where positive efficiency had been obtained did not lead to overall improvement of metrics. However, the faster flow wave celerity obtained through such focused optimization generally seems to be leading to better timing in hydrograph peaks (Figures A1 and A2) than the slower wave although the statistics of flow are hence deteriorated (Table A1).

Acknowledgments
This work was supported by the University of California Office of the President Multicampus Research Programs and Initiatives and by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration; both institutions are gratefully acknowledged. The practical application in this study was made possible using the following freely available data: river network information from the Hydrological data and maps based on SHuttle Elevation Derivatives at multiple Scales (HydroSHEDS), estimates of runoff from phase 2 of the North American Land Data Assimilation System (NLDAS2) and gauge observations from the U.S. Geological Survey National Water Information System (NWIS). The authors are thankful to the Editor, Associate Editor, Reed Maxwell, Ezio Todini, and one anonymous reviewer for their valuable comments on earlier versions of this manuscript.

References

Abdulla, F. A., D. P. Lettenmaier, E. F. Wood, and J. A. Smith (1996), Application of a macroscale hydrologic model to estimate the water balance of the Arkansas Red River basin, *J. Geophys. Res.*, *101*(D3), 7449–7459.

Alsdorf, D. E., E. Rodriguez, and D. P. Lettenmaier (2007a), Measuring surface water from space, *Rev. Geophys.*, *45*, RG2002, doi:10.1029/2006RG000197.

Alsdorf, D. E., E. Rodriguez, D. Lettenmaier, and J. Famiglietti (2007b), Surface Water Ocean Topography (SWOT) mission, in *Earth Science and Applications From Space: National Imperatives for the Next Decade and Beyond*, pp. 354–359, Natl. Res. Counc., Washington, D. C.

Amdahl, G. M. (1967), Validity of the single processor approach to achieving large scale computing capabilities, in *Proceedings of the April 18–20, 1967, Spring Joint Computer Conference*, pp. 483–485, ACM, Atlantic City, N. J.

Arrigo, J. (2011), Developing community infrastructure for hydrologic modeling, *Eos Trans. AGU*, *92*(31), 260.

Ashby, S. F., and R. D. Falgout (1996), A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations, *Nucl. Sci. Eng.*, *124*(1), 145–159.

Bailey, D. H. (1992), Misleading performance reporting in the supercomputing field, *Sci. Program.*, *1*(2), 141–151.

Bamzai, A. (2012), Preface, *Int. J. High Perform. Comput. Appl.*, *26*(1), 3–4, doi:10.1177/1094342012437194.

Bates, P. D., and A. P. J. De Roo (2000), A simple raster-based model for flood inundation simulation, *J. Hydrol.*, *236*(1–2), 54–77.

Berge, C. (1962), Trees and arborescences, in *The Theory of Graphs and Its Applications*, pp. 152–164, Methuen and Co. Ltd., London, U. K.

Cai, X., Z.-L. Yang, C. H. David, G.-Y. Niu, and M. Rodell (2014), Hydrological evaluation of the Noah-MP land surface model for the Mississippi River Basin, *J. Geophys. Res. Atmos.*, *119*, 23–38, doi:10.1002/2013JD020792.

Culler, D. E., A. Gupta, and J. P. Singh (1997a), Partitioning for performance, in *Parallel Computer Architecture: A Hardware/Software Approach*, pp. 133–145, Morgan Kaufmann, San Francisco, Calif. [Available at http://dl.acm.org/citation.cfm?id=550071.]

Culler, D. E., A. Gupta, and J. P. Singh (1997b), The parallelization process, in *Parallel Computer Architecture: A Hardware/Software Approach*, pp. 95–104, Morgan Kaufmann, San Francisco, Calif. [Available at http://dl.acm.org/citation.cfm?id=550071.]

Cunge, J. A. (1969), On the subject of a flood propagation computation method (Muskingum method), *J. Hydraul. Res.*, *7*(2), 205–230.

David, C. H. (2013), RAPID v1.4.0. Zenodo, doi:10.5281/zenodo.24756.

David, C. H., F. Habets, D. R. Maidment, and Z.-L. Yang (2011a), RAPID applied to the SIM-France model, *Hydrol. Processes*, *25*(22), 3412–3425.

David, C. H., D. R. Maidment, G.-Y. Niu, Z.-L. Yang, F. Habets, and V. Eijkhout (2011b), River network routing on the NHDPlus dataset, *J. Hydrometeorol.*, *12*(5), 913–934.

David, C. H., Z.-L. Yang, and J. S. Famiglietti (2013a), Quantification of the upstream-to-downstream influence in the Muskingum method and implications for speedup in parallel computations of river flow, *Water Resour. Res.*, *49*, 2783–2800, doi:10.1002/wrcr.20250.

David, C. H., Z.-L. Yang, and S. Hong (2013b), Regional-scale river flow modeling using off-the-shelf runoff products, thousands of mapped rivers and hundreds of stream flow gauges, *Environ. Modell. Software*, *42*, 116–132.

Durand, M., F. Lee-Lueng, D. P. Lettenmaier, D. E. Alsdorf, E. Rodriguez, and D. Esteban-Fernandez (2010), The Surface Water And Ocean Topography mission: Observing terrestrial surface water and oceanic submesoscale eddies, *Proc. IEEE*, *98*(5), 766–779.

Famiglietti, J. S., and E. F. Wood (1994), Multiscale modeling of spatially-variable water and energy-balance processes, *Water Resour. Res.*, *30*(11), 3061–3078.

Famiglietti, J. S., L. Murdoch, V. Lakshmi, and J. Arrigo (2011), *Report From the 3rd Workshop on a Community Hydrologic Modeling Platform (CHyMP): A Strategic and Implementation Plan, Establishing a Framework for Community Modeling in Hydrologic Science*, Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI), Boston, Mass.

Gustafson, J. L. (1988), Reevaluating Amdahl's law, *Commun. ACM*, *31*(5), 532–533.

Habets, F., et al. (2008), The SAFRAN-ISBA-MODCOU hydrometeorological model applied over France, *J. Geophys. Res.*, *113*, D06113, doi:10.1029/2007JD008548.

Hernandez, V., J. E. Roman, and V. Vidal (2005), SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems, *ACM Trans. Math. Software*, *31*(3), 351–362, doi:10.1145/1089014.1089019.

Higham, N. J. (1990), How accurate is Gaussian elimination?, in Numerical Analysis 1989, Proceedings of the 13th Dundee Conference, edited by D. F. Griffiths and G. A. Watson, *Volume 228 of Pittman Research Notes in Mathematics*, pp. 137–154, Longman Scientific and Technical, Essex, U. K.

Horton, R. E. (1945), Erosional development of streams and their drainage basins: Hydrophysical approach to quantitative morphology, *Geol. Soc. Am. Bull.*, *56*(3), 275–370, doi:10.1130/0016–7606(1945)56[275:EDOSAT]2.0.CO;2.

Hwang, H.-T., Y.-J. Park, E. A. Sudicky, and P. A. Forsyth (2014), A parallel computational framework to solve flow and transport in integrated surface-subsurface hydrologic systems, *Environ. Modell. Software*, *61*, 39–58, doi:10.1016/j.envsoft.2014.06.024.

Ivanov, V. Y., E. R. Vivoni, R. L. Bras, and D. Entekhabi (2004), Catchment hydrologic response with a fully distributed triangulated irregular network model, *Water Resour. Res.*, *40*, W11102, doi:10.1029/2004WR003218.

Kahn, A. B. (1962), Topological sorting of large networks, *Commun. ACM*, *5*(11), 558–562, doi:10.1145/368996.369025.

Karypis, G., and V. Kumar (1999), A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM J. Sci. Comput.*, *20*(1), 359–392, doi:10.1137/S1064827595287997.

Kollet, S. J., and R. M. Maxwell (2006), Integrated surface-groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model, *Adv. Water Resour.*, *29*(7), 945–958.

Kollet, S. J., R. M. Maxwell, C. S. Woodward, S. Smith, J. Vanderborght, H. Vereecken, and C. Simmer (2010), Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources, *Water Resour. Res.*, *46*, W04201, doi:10.1029/2009WR008730.

Koussis, A. D. (1978), Theoretical estimations of flood routing parameters, *J. Hydraul. Div. Am. Soc. Civ. Eng.*, *104*(1), 109–115.

Lasser, D. J. (1961), Topological ordering of a list of randomly-numbered elements of a network, *Commun. ACM*, *4*(4), 167–168, doi:10.1145/355578.366314.

Lehner, B., K. Verdin, and A. Jarvis (2008), New global hydrography derived from space borne elevation data, *Eos Trans. AGU*, *89*(10), 93–94.

Li, T., G. Wang, and J. Chen (2010), A modified binary tree codification of drainage networks to support complex hydrological models, *Comput. Geosci.*, *36*(11), 1427–1435.

Li, T., G. Wang, J. Chen, and H. Wang (2011), Dynamic parallelization of hydrological model simulations, *Environ. Modell. Software*, *26*(12), 1736–1746.

Liang, X., D. P. Lettenmaier, E. F. Wood, and S. J. Burges (1994), A simple hydrologically based model of land-surface water and energy fluxes for general-circulation models, *J. Geophys. Res.*, *99*(D7), 14,415–14,428.

Lohmann, D., et al. (1998), The project for intercomparison of land-surface parameterization schemes (PILPS) phase 2(c) Red-Arkansas River basin experiment. 3: Spatial and temporal analysis of water fluxes, *Global Planet. Change*, *19*(1–4), 161–179.
Lohmann, D., et al. (2004), Streamflow and water balance intercomparisons of four land surface models in the North American Land Data Assimilation System project, *J. Geophys. Res.*, *109*, D07S91, doi:10.1029/2003JD003517.
Maidment, D. R. (2015), *A Conceptual Framework for the National Flood Interoperability Experiment*, Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI), Boston, Mass.
Maurer, E. P., G. M. O'Donnell, D. P. Lettenmaier, and J. O. Roads (2001), Evaluation of the land surface water budget in NCEP/NCAR and NCEP/DOE reanalyses using an off-line hydrologic model, *J. Geophys. Res.*, *106*(D16), 17,841–17,862.
McCarthy, G. T. (1938), The unit hydrograph and flood routing, paper pre– sented at Conference of the North Atlantic Division, U.S. Engineer Department, New London, Conn.
Michalakes, J., J. Dudhia, D. Gill, J. Henderson, J. Klemp, and W. Skamarock (2005), The weather research and forecast model: Software architecture and performance, in *Proceedings of the Eleventh ECMWF Workshop on the Use of High Performance Computing in Meteorology*, pp. 156–168, World Sci, Reading, U. K. [Available at http://www.worldscientific.com/doi/abs/10.1142/9789812701831_0012.]
Miller, J. R., G. L. Russell, and G. Caliri (1994), Continental-scale river flow in climate models, *J. Clim.*, *7*(6), 914–928, doi:10.1175/1520-0442(1994)007<0914:CSRFIC>2.0.CO;2.
Miller, W. A., and J. A. Cunge (1975), Simplified equations of unsteady flow, in *Unsteady Flow in Open Channels*, vol. 1, edited by K. Mahmood and V. Yevjevich, pp. 216–232, Water Resour. Publ., Fort Collins, Colo.
Moritz, H. (1980), Geodetic reference system 1980, *J. Geod.*, *54*(3), 395–405.
Nash, J. E., and J. V. Sutcliffe (1970), River flow forecasting through conceptual models. Part I—A discussion of principles, *J. Hydrol.*, *10*(3), 282–290.
Neal, J. C., T. J. Fewtrell, and M. A. Trigg (2009), Parallelisation of storage cell flood models using OpenMP, *Environ. Modell. Software*, *24*(7), 872–877.
NIMA (2000), *Department of Defense World Geodetic System 1984*, Natl. Imagery and Mapping Agency, Bethesda, Md. [Available at http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf.]
Niu, G.-Y., et al. (2011), The community Noah land surface model with multiparameterization options (Noah-MP): 1. Model description and evaluation with local-scale measurements, *J. Geophys. Res*, *116*, D12109, doi:10.1029/2010JD015139.
Oki, T., Y. Agata, S. Kanae, T. Saruhashi, D. W. Yang, and K. Musiake (2001), Global assessment of current water resources using total runoff integrating pathways, *Hydrol. Sci. J.*, *46*(6), 983–995.
Olivera, F., J. Famiglietti, and K. Asante (2000), Global-scale flow routing using a source-to-sink algorithm, *Water Resour. Res.*, *36*(8), 2197–2207.
Peters-Lidard, C. D., et al. (2007), High-performance Earth system modeling with NASA/GSFC's land information system, *Innov. Syst. Software Eng.*, *3*(3), 157–165, doi:10.1007/s11334-007-0028-x.
Ponce, V. M., and V. Yevjevich (1978), Muskingum-Cunge method with variable parameters, *J. Hydraul. Div. Am. Soc. Civ. Eng.*, *104*(12), 1663–1667.
Seaber, P. R., F. P. Kapinos, and G. L. Knapp (1987), Hydrologic unit maps, *U. S. Geol. Surv. Water Supply Pap.*, *2294*, 1–63.
Strahler, A. N. (1952), Hypsometric (area-altitude) analysis of erosional topography, *Geol. Soc. Am. Bull.*, *63*(11), 1117–1142, doi:10.1130/0016–7606(1952)63[1117:HAAOET]2.0.CO;2.
The Ad Hoc Group, et al. (2001), Global water data: A newly endangered species, *Eos Trans. AGU*, *82*(5), 54–58, doi:10.1029/01EO00031.
Therrien, R., R. Mclaren, E. Sudicky, and S. Panday (2006), *HydroGeoSphere: A Three-Dimensional Numerical Model Describing Fully-Integrated Subsurface and Surface Flow and Solute Transport*, Groundwater Simulations Group, University of Waterloo, Waterloo, Ontario.
Todini, E. (2007), A mass conservative and water storage consistent variable parameter Muskingum-Cunge approach (vol 11, pg 1645, 2007), *Hydrol. Earth Syst. Sci.*, *11*(6), 1783–1783.
USEPA and USGS (2010), *NHDPlus User Guide*, Horizon Systems Corporation, Herndon, Va. [Available at ftp://ftp.horizon-systems.com/NHDPlus/NHDPlusV1/documentation/NHDPLUSV1_UserGuide.pdf.]
Verdin, K. L., and J. P. Verdin (1999), A topological system for delineation and codification of the Earth's river basins, *J. Hydrol.*, *218*, 1–12.
Vivoni, E. R., G. Mascaro, S. Mniszewski, P. Fasel, E. P. Springer, V. Y. Ivanov, and R. L. Bras (2011), Real-world hydrologic assessment of a fully-distributed hydrological model in a parallel computing environment, *J. Hydrol.*, *409*(1–2), 483–496.
Wang, G., B. Wu, and T. Li (2007), Digital Yellow River Model, *J. Hydro-environ. Res.*, *1*(1), 1–11.
Wang, H., Y. Zhou, X. Fu, J. Gao, and G. Wang (2012), Maximum speedup ratio curve (MSC) in parallel computing of the binary-tree-based drainage network, *Comput. Geosci.*, *38*(1), 127–135.
Wang, P., Y. T. Song, Y. Chao, and H. Zhang (2005), Parallel computation of the regional ocean modeling system, *Int. J. High Perform. Comput. Appl.*, *19*(4), 375–385, doi:10.1177/1094342005059115.
Werner, C., and J. S. Smart (1973), Some new methods of topologic classification of channel networks, *Geogr. Anal.*, *5*(4), 271–295.
Wood, E. F., D. Lettenmaier, X. Liang, B. Nijssen, and S. W. Wetzel (1997), Hydrological modeling of continental-scale basins, *Annu. Rev. Earth Planet. Sci.*, *25*, 279–300.
Xia, Y., et al. (2012a), Continental-scale water and energy flux analysis and validation for the North American Land Data Assimilation System project phase 2 (NLDAS-2): 1. Intercomparison and application of model products, *J. Geophys. Res.*, *117*, D03109, doi:10.1029/2011JD016048.
Xia, Y., et al. (2012b), Continental-scale water and energy flux analysis and validation for the North American Land Data Assimilation System project phase 2 (NLDAS-2): 2. Validation of model-simulated streamflow, *J. Geophys. Res.*, *117*, D03110, doi:10.1029/2011JD016051.
Yamazaki, D., S. Kanae, H. Kim, and T. Oki (2011), A physically based description of floodplain inundation dynamics in a global river routing model, *Water Resour. Res.*, *47*, W04501, doi:10.1029/2010WR009726.
Yamazaki, D., H. Lee, D. E. Alsdorf, E. Dutra, H. Kim, S. Kanae, and T. Oki (2012), Analysis of the water level dynamics simulated by a global river model: A case study in the Amazon River, *Water Resour. Res.*, *48*, W09508, doi:10.1029/2012WR011869.