Proceedings of the IASTED International Conference
Applied Simulation and Modelling (ASM 2012)
June 25 - 27, 2012 Napoli, Italy

SIMULATION OF A DISTRIBUTED FLOOD CONTROL SYSTEM USING A PARALLEL ASYNCHRONOUS SOLVER FOR SYSTEMS OF ODES

Ricardo Mantilla,
Luciana K. Cunha and
Witold F. Krajewski
IIHR Hydroscience & Engineering
The University of Iowa
Iowa City, Iowa, USA
email: ricardo-mantilla@uiowa.edu

Scott J. Small,
Laurent O. Jay,
Morgan Fonley and
Rodica Curtu
Department of Mathematics
The University of Iowa
Iowa City, Iowa, USA

ABSTRACT
A recently developed parallel asynchronous solver for systems of ordinary differential equations (ODEs) is used to simulate flows along the channels in a river network. In our model, precipitation is applied over the hillslopes adjacent to the river network links and water movement from hillsope to link and along the river network is represented as a system of ODEs. The numerical solver is based on dense output Runge-Kutta methods that allow for asynchronous integration. A static partition method is used to distribute the workload among different processes, enabling a parallel implementation that capitalizes on a distributed memory system. Communication between processes is performed asynchronously. We illustrate the solver capabilities by integrating flow transport equations for a ~32,000 km^2 river basin subdivided into 574,000 sub-watersheds that are interconnected by the river network. We show that the runtime for an eight month-long simulation forced by 1-km resolution NEXRAD rainfall is completed in under 4 minutes using 64 computing nodes. In addition, we include equations to simulate small reservoirs spread throughout the river network and estimate changes in hydrographs at multiple locations. Our results provide a firm theoretical basis for the concept of distributed flood control systems.

KEY WORDS
Rainfall Runoff Models, asynchronous solver, Floods, Reservoirs

1 Introduction

In recent work Small et al. (2012) introduced a parallel implementation of an efficient asynchronous integration method for systems of ordinary differential equations (ODEs) of the form

$$\frac{dy}{dt} = f(t, y) \tag{1a}$$
$$y(t_0) = y_0. \tag{1b}$$

in which the linkage between equations is determined by a directed tree structure. This kind of linkage structure arises

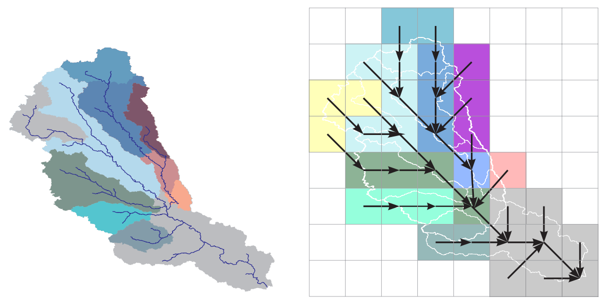


Figure 1: Partitioning schemes in distributed hydrological models that lead to a global system of ODEs. Subwatershed partitioning on the left and square pixel partitioning on the right.

in distributed hydrological models in which a river network of interconnected streams provides a natural tree-like connecting structure (see Figure 1). Under the assumption that the model for the flow of water through a single stream can be written as a system of differential equations, the equations from every stream in the entire network could be viewed as one large system of differential equations (e.g. Mantilla and Gupta (2005); Qu and Duffy (2007)). In general, hydrological models that use a semi-discrete approximation of flow transport of the Saint-Venant equations or the Muskingum method fall in this category (Kampf and Burges, 2007). This type of application is relevant to hydrology of floods (see Mantilla and Gupta (2005); Mandapaka et al. (2009); Mantilla et al. (2011); Cunha et al. (2011)), but the numerical methods developed by Small et al. (2012) are more general and can be extend to other types of physical problems where the connectivity between equations is determined by a directed tree structure.

In this paper we present several cases of application of the algorithm including (1) simulating flows on an artificial river network for which the analytical solution is known, (2) simulating flows along the channels of the Cedar River network in eastern Iowa, that drains an area of ~32,000 km^2, and finally (3) results of simulating the effect of distributed reservoirs in the flows at the outlet of a 250 km^2 basin. First, In section 2 we present the equations used to

DOI: 10.2316/P.2012.776-042

simulate flows and to simulate the behavior of small reservoirs created by small earth dams. Second, in section 3 we give details of the implementation for the cases mentioned above and finally in section 4 we discuss our results.

2 Flow transport on a hillslope-link based river basin model

We consider the link-hillslope decomposition of a river basin described by Mantilla and Gupta (2005). Equations for each link follow the general form described by Mantilla (2007) that has been applied in other contexts of self-similar trees by Menabde and Sivapalan (2001). The topology of a river network can be described as a directed tree structure under the assumption that no loops are present in the network and no channel splits (e.g. braided streams) are considered. For each link (i.e., river segment between two junctions), a differential equation can be written for water discharge by describing the link as a non-linear reservoir. Thus, for link i, the discharge $q(i,t)$ of water (in m^3/s) from the channel, the depth of water (in m) ponded on the channel's surrounding hillslope $s_p(i,t)$, and the volume of water in the hilllslope soil matrix $s_g(i,t)$ at time t (measured in minutes) can be modeled by the system of ODEs

$$\frac{dq}{dt}(i,t) = \frac{1}{\tau} q^{\lambda_1} \left(\sum_{j \to i} q(j,t) - q(i,t) + c_1 s_p^{5/3} \right) \tag{2a}$$

$$\frac{ds_p}{dt}(i,t) = c_2 p(i,t) - c_3 s_p^{5/3} \tag{2b}$$

$$\frac{ds_g}{dt}(i,t) = c_4 p(i,t) - c_5 s_g \tag{2c}$$

where the values τ, c_1, c_2, c_3, and c_4 are constants at each link and are given by

$$\tau = \frac{(1-\lambda_1)L(i)}{60 v_r A(i)^{\lambda_2}} \tag{3a}$$

$$c_1 = \frac{2L(i)}{0.6} \cdot \frac{S(i)^{1/2}}{\eta(i)} \tag{3b}$$

$$c_2 = \frac{10^{-3}}{60} R_C \tag{3c}$$

$$c_3 = \frac{2L(i)}{0.6 A_h(i)} \cdot \frac{S(i)^{1/2}}{\eta(i)} (60 \cdot 10^{-6}) \tag{3d}$$

$$c_4 = \frac{10^{-3}}{60} (1 - R_C) \tag{3e}$$

$$c_5 = K_{sat}. \tag{3f}$$

These constants also handle unit conversion. The sum in equation (2a) is taken over all links "flowing" into link i. The parameters that appear in (3) are described in Table 1. The constants v_r, R_C, λ_1, and λ_2 are universal amongst the entire system. The $c_1 s_p^{5/3}$ term in equation (2a) can be interpreted as the flow of water from the surrounding hillslope into the channel, and $p(i,t)$ is a known function for the precipitation at link i measured in mm/s.

In addition to the equations describing flow in natural conditions we also include equations to describe mass continuity and flow release for water storage in a reservoir. This equation can be expressed as

$$\frac{dV}{dt} = \sum_{j \to i} q(j,t) - q_R(t,V) \tag{4}$$

where V is the volume of water stored in the reservoir; $\sum_{j \to i} q(j,t)$ represent inflows into the reservoir as a function of time; and $q_R(t,V)$ is the outflow from the reservoir, which is determined by the storage volume that is itself a function of the water level in the reservoir. The relationship between the outflow and storage is non-linear and is described by the following set of equations for the outflow,

$$q_R(t,V) = c_6 A_c \sqrt{2gh} \tag{5}$$

if $h \geq H_{spill}$ and,

$$q_R(t,V) = c_6 A_c \sqrt{2gh} + c_7 L (h - H_{spill})^{3/2} \tag{6}$$

if $H_{spill} < h \leq H_{dam}$. Here c_6 is the orifice coefficient; A_c is the orifice cross-sectional area; h is the water level in the reservoir; H_{spill} is the reservoir spill level; H_{dam} is the total dam height; c_7 is the weir coefficient; andL is the length of the weir crest. We assumed the outflow to be equal to the inflow when the reservoir is full.

3 Numerical experiments

In this section we present several applications of the asynchronous algorithm implemented in the C programming language. Communication between processes is done using MPI (see Pacheco (1997)). All experiments are run on a cluster with 200 nodes, and each node has Dual Quad Core Intel(R) Xeon(R) CPU X5550 @ 2.67GHz processors with 24 Gb DDR3 1333MHz memory and 1Tb of storage.

3.1 Validation of numerical results

We apply the algorithm to a particular case of the parameters for which an closed form of the solution of (2) is known. We consider water flows on a Peano network using only the transport equation (2a). Assuming constant velocity (i.e. $\lambda_1 = \lambda_2 = 0.0$), $v_r = 1.0$ m/s, and $L(i) = 500$ m for every link i. In physical terms, this corresponds to a case in which runoff from an instantaneous and uniform storm has made it into the channels of the river network and flows freely downstream. The constant velocity assumption treats each channel link as a linear reservoir. The exact solution at the root node has been recently given by Mantilla et al. (2012) as

Constant	Description	Range
$A_h(i)$	Area of the surrounding hillslope at link i	$(0, 5]$ km^2
$A(i)$	Sum of the A_h's for all links upstream from i	$[0.01, 10^6]$ km^2
$L(i)$	Length of link i	$[10, 1000]$ m
$S(i)$	Average slope of hillslope surrounding i	$[0.0, 0.5]$
$\eta(i)$	Manning coefficient at link i	$[0.1, 0.8]$
v_r	Reference flow velocity	$[0.2, 1.0]$ m/s
R_C	Runoff coefficient	$(0.0, 1.0]$
K_{sat}	Saturated Hydraulic conductivity of the soil	
λ_1	Exponent for flow velocity discharge	$[0.0, 0.7]$
λ_2	Exponent for flow velocity upstream area	$[-0.4, -0.05]$

Table 1: Description of the constants used in the river basin model. Typical values are given under the last column.

$$q_\Omega(t) = q_0 e^{-\frac{t}{\tau}} \sum_{k=1}^{2^{\Omega-1}} \frac{3^{b_k} \left(\frac{t}{\tau}\right)^{k-1}}{(k-1)!} \tag{7}$$

where Ω is the Horton order of the root link in the Peano tree and the sequence b_k takes the integer k into the number of 1's appearing in the binary representation of k (so $b(1) = 1$, $b(3) = 2$, $b(11) = 3$, etc). We have taken the discharge at time 0 in each link to have the same value, q_0. Again, this corresponds to runoff from an instantaneous and uniform pulse of rainfall that instantaneously moves into the channel.

For the experiment, we use the Peano network with a root link having Horton order $\Omega = 10$. This system has a total of $N = 4^9 = 262,144$ links. The initial discharge is taken as $q_0 = 1.0$ m^3/s. For the numerical methods, we apply the 7-stage dense output Dormand and Prince method of order 5 to the ODE at each leaf and the dense output Runge-Kutta method of order 4 to all other links. The true solution and numerical errors from this experiment are given in Figure 2. In these figures, the absolute error tolerance for the local error is progressively reduced by a factor of 10. The numerical solution converges as the tolerance is reduced.

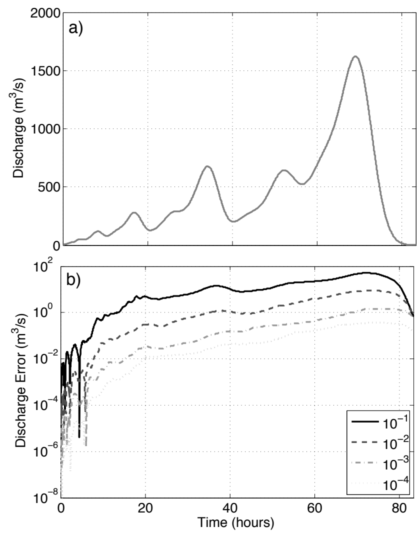


Figure 2: Exact solution and numerical error for the Peano network of Horton order 10.

3.2 Effectiveness of implementation

To test the effectiveness of our implementation, we compare its runtime with that of a standard numerical implementation. For the test problem, we use the full model (2), with $v_r = 0.64$ m/s, $R_C = 0.5$, $\lambda_1 = 0.24$ and $\lambda_2 = -0.12$. The constants specific to each link are again taken from measurements for the 32,375 km^2 basin formed by the Iowa River basin located in Eastern Iowa. Details of the river basin location and locations where streamflow is gauged is shown in Figure 3. The 30-meter resolution digital elevation model (DEM) was obtained from the National Elevation Dataset (Gesch et al., 2002; Gesch, 2007). A 30-meter resolution digital elevation model (DEM) was used to determine geometrical and topological parameters for the river network and adjacent hillslopes. The corresponding system of ODEs is a tree structure with $N = 574{,}780$ links. Initial inputs of $q(i, 0) = 1$ m^3/s and $s_p(i, 0) = 0.0$ m are used.

We apply our implementation to this problem as well as the DOPRI5 routine provided in Hairer et al. (2000). The latter solves the problem as one large system of N ODEs using the Dormand and Prince method of order 5. The DOPRI5 solver is written in the Fortran programming language. For our implementation, we also use the same Runge-Kutta method at each link. The solution is computed for up to 264 days. The runtimes for varying absolute error tolerances are given in Table 2, and the error tolerances are for all components. The runtimes presented include only the time to calculate the numerical solutions at each time step as well as the time to read the rainfall data from disk. All other operations, including partitioning the basin and initializing the system, are independent of implementation and tolerance and took an average of 7 seconds. The results show that our implementation requires significantly less time than the standard DOPRI5 solver. Note that although our implementation has many more step rejections than the DOPRI5 algorithm, a rejection for our im-

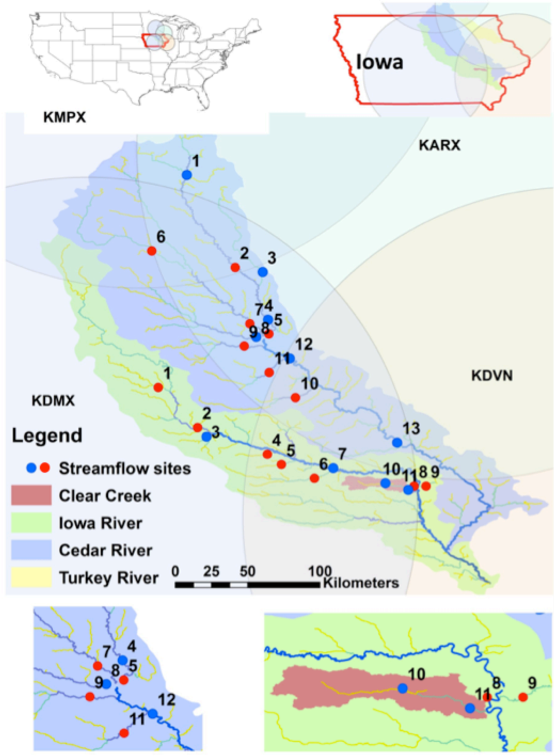


Figure 3: The Iowa River basin including information on radar coverage and streamflow gauging sites

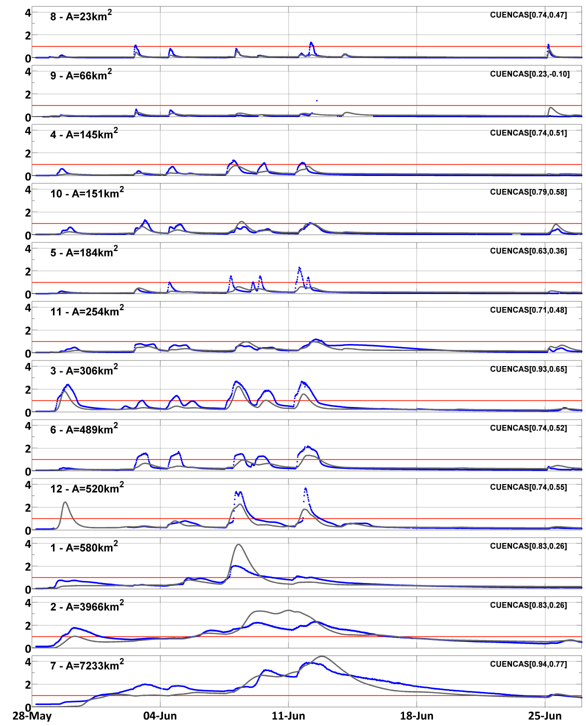


Figure 4: The Iowa River basin including information on radar coverage and streamflow gauging sites

plementation occurs at a single link. Because the DOPRI5 algorithm solves the large system at once, the solution at ALL links must be either accepted or rejected. We also observe that the runtimes of both implementations fluctuate as the tolerance decreases. This results from the fluctuations in the number of rejections that occur balanced with decreases in the step sizes.

3.3 River basin model forced by a radar-derived precipitation product

The full model (2) is tested assuming continuous rainfall. The rainfall product for the year 2008 was obtained from Hydro-NEXRAD (Krajewski et al., 2011) combining information from nearby radars shown in Figure 3. The data is provided as hourly rates in mm/hr with a spatial resolution of 4 km. Therefore, the function $p(i,t)$ from (2b) can be viewed as a step function. For the measured data, for most links, the value of $p(i,t)$ tends to be 0 for most times t. Because step functions are discontinuous, we must not integrate across any jumps, as this will destroy all error controls.

We assume that there is initially no water present on the hillslope of each link ($s_p(i,0) = 0.0$ m). We also assume a small initial discharge of 1 m^3/s at each link ($q(i,0) = 1$ m^3/s). Rainfall measurements taken over the state of Iowa during the spring-summer-fall of 2008 are used for the function $p(i,t)$. The total integration time is 264 days. Hydrographs calculated at the location of the 13 gauging sites are shown in Figures 4 and 5.

The runtimes are given in Figure 6 and decrease by about a factor of 1.5 as the processes are doubled, except for 128 processes. The increase in runtime is a result of the large number of processes reading rainfall data from disk.

3.4 Simulation of the effect of nested reservoirs in outlet hydrograph

As a final application case we studied the effect of 40 nested reservoirs. In our simulation each reservoir is designed to be able to hold 132,000 m^3. We assumed a small circular orifice at the bottom of the dam with 0.75 m in diameter. Dam height and spillway height are chosen to achieve the desired retention volume. Figure 7 shows the location of the 40 reservoirs (top) and the effect that those have on a hydrograph generated by a 100 mm/h event with a duration of 15 minutes. This rainfall amount roughly corresponds with the 100-year storm for this basin. As it can be seen the peak of the hydrograph is reduced by more than half. The values selected here are realistic providing initial evidence on the theoretical feasibility a flood control distributed reservoir system.

	Our Implementation		DOPRI5	
Tolerance	Runtime	Rejections	Runtime	Rejections
10^{-1}	35.30 mins	13,263,336	531.12 mins	6,739
10^{-2}	40.12 mins	13,323,396	631.65 mins	6,066
10^{-3}	35.52 mins	12,290,500	619.58 mins	5,666
10^{-4}	36.55 mins	10,028,964	534.27 mins	5,183
10^{-5}	40.35 mins	11,069,859	612.85 mins	4,433
10^{-6}	41.45 mins	12,048,629	503.48 mins	3,892

Table 2: Runtimes for our implementation versus a standard Dormand and Prince order 5 Runge-Kutta solver. The error tolerances are absolute.

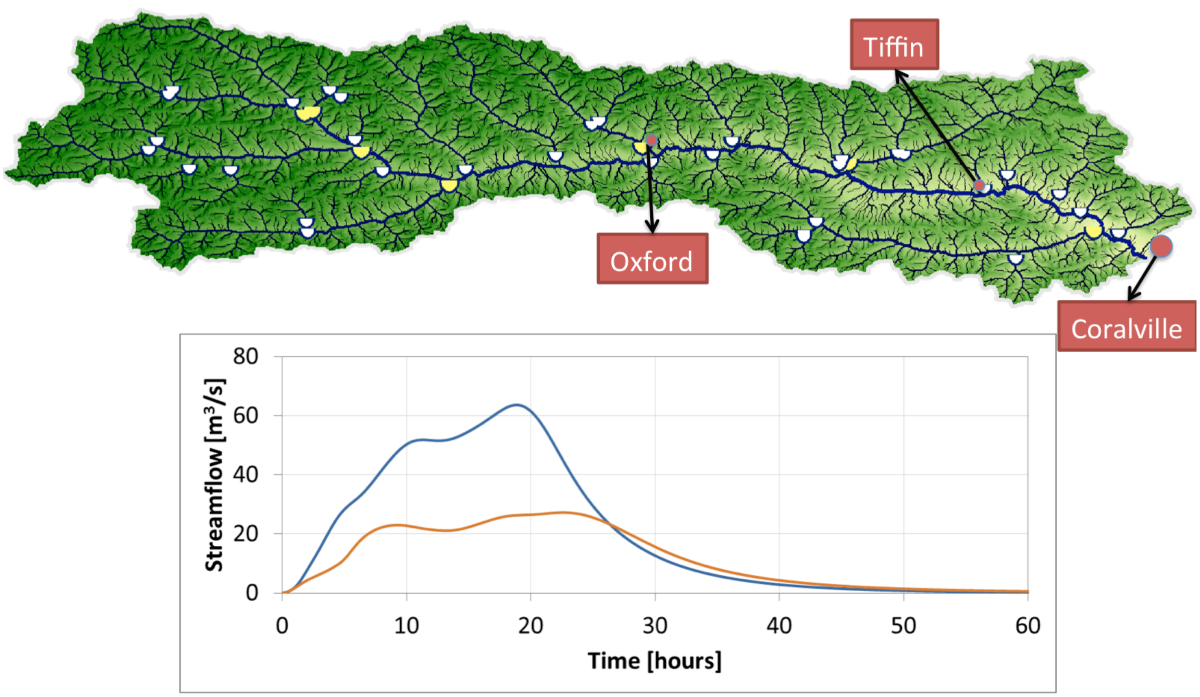


Figure 7: Location of simulated reservoirs (top) and modification of the hydrograph at the basin outlet due to the presence of the reservoirs

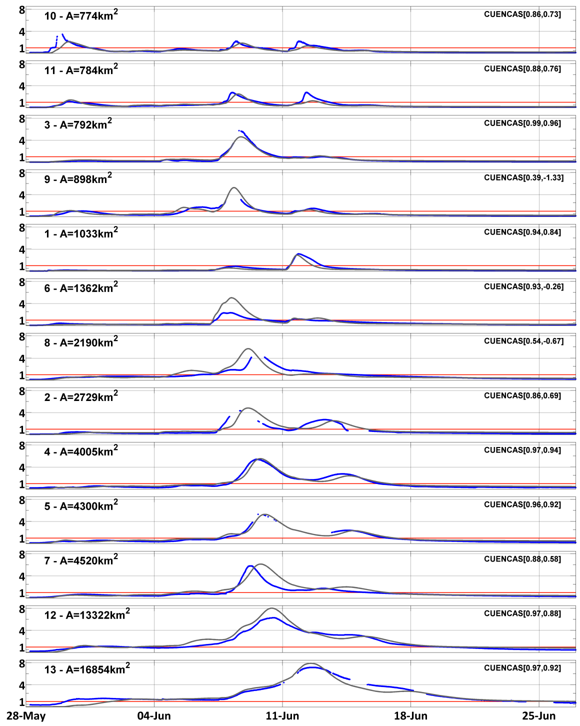
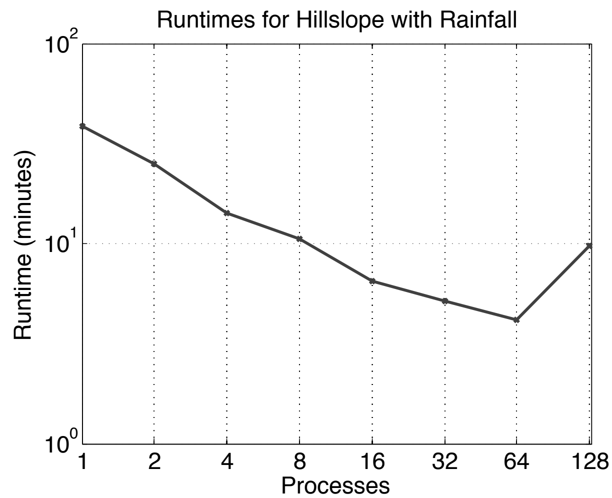


Figure 5: The Iowa River basin including information on radar coverage and streamflow gauging sites

4 Conclusion

We have used a numerical solver for systems of ODEs with a tree structure. The runtimes for these applications give good scalability and load balancing as the number of processes increases. Our results are a great improvement over those of Apostolopoulos and Georgakakos (1997), whose problem is essentially the same as our river basin model of Section 2. Our model presented here does not account for variables such as soil dynamics and evapotranspiration, however, Apostolopoulos and Georgakakos (1997) does solve an ODE having a directed tree structure. The speedup for their model achieved a maximum of around 2.25 using 9 processes compared to up to 25.3 for the algorithm presented here. In addition, using the computing architecture that was available at the time Apostolopoulos and Georgakakos (1997) reported their results, the typical runtimes were about 43 hours for the calculations on a 21 km^2 basin compared to the 16,878 km^2 implemented in this paper.

Efficient numerical integrators such as the one demonstrated here bring closer to reality the goal of implementing fully distributed flood forecasting systems supported by physics based hydrological models and high-quality/high-resolution rainfall products. In addition, it opens the door to more comprehensive Monte-carlo based studies of parameter sensitivity in complex hydrological models, ensemble simulations for data assimilation studies and estimation of long term flood frequencies under realistic physical conditions.



(a)

Procs	Runtime	Speedup
1	38.55	–
2	25.00	1.54
4	14.30	2.70
8	10.52	3.67
16	6.50	5.93
32	5.13	7.51
64	4.13	9.33
128	9.73	3.96

(b)

Figure 6: Runtimes for the river basin model with precipitation. Runtimes are given in minutes.

The results presented in this paper bring closer to reality the ability of running spatially explicit rainfall runoff models for forecasting of floods. Traditionally simulations of flood hydrograph have relied in simplified or lumped models to be able to provide timely predictions. Our results indicate that the computational efficiency illustrated here can serve to deploy such forecasting models and to operate networks of small reservoirs to maximize the reduction of flood peaks at desired locations throughout the river network.

Acknowledgements

Research for this paper was supported by the National Science Foundation grant DMS-1025483.

References

[1] Apostolopoulos, T. K., Georgakakos, K. P., 1997. Parallel computation for streamflow prediction with distributed hydrologic models. Journal of Hydrology 197, 1–24.

[2] Cunha, L., Krajewski, W., Mantilla, R., 2011. A framework for flood risk assessment under nonstationary conditions or in the absence of historical data. Journal of Flood Risk Management 4 (1), 3–22.

[3] Cunha, L. K., Mandapaka, P. V., Krajewski, W. F., Mantilla, R., 2012. Impact of rainfall error structure on estimated flood magnitude across scales: an investigation based on a parsimonious distributed hydrological model. Water Resources Research(*Submitted*).

[4] Gesch, D., 2007. The national elevation dataset. In: Maune, D. (Ed.), Digital Elevation Model Technologies and Applications: The DEM Users Manual, 2nd Edition. American Society for Photogrammetry and Remote Sensing, pp. 99–118.

[5] Gesch, D., Oimoen, M., Greenlee, S., Nelson, C., Steuck, M., Tyler, D., 2002. The national elevation dataset. Photogrammetry Engineering and Remote Sensing 68 (1), 5–11.

[6] Hairer, E., Nørsett, S., Wanner, G., 2000. Solving Ordinary Differential Equations I, Nonstiff Problems. Springer New York.

[7] Kampf, S. K., Burges, S. J., 2007. A framework for classifying and comparing distributed hillslope and catchment hydrologic models. Water Resources Research 43, W05423.

[8] Krajewski, W. F., Kruger, A., Smith, J. A., Lawrence, R., Gunyon, C., Goska, R., Seo, B.-C., Domaszczynski, P., Baeck, M. L., Ramamurthy, M. K., Weber, J., Bradley, A. A., DelGreco, S. A., Steiner, M., 2011. Towards better utilization of nexrad data in hydrology: an overview of hydro-nexrad. Journal of Hydroinformatics 13 (2), 255–266.

[9] Mandapaka, P. V., Krajewski, W. F., Mantilla, R., Gupta, V. K., 2009. Dissecting the effect of rainfall variability on the statistical structure of peak flows. Advances in Water Resources 32 (10), 1508 – 1525.

[10] Mantilla, R., 2007. Physical basis of statistical scaling in peak flows and stream flow hydrographs for topologic and spatially embedded random self-similar channel networks. Ph.D. thesis, University of Colorado.

[11] Mantilla, R., Gupta, V. K., 2005. A GIS framework to investigate the process basis for scaling statistics on river networks. Geoscience and Remote Sensing Letters, IEEE 2 (4), 404–408.

[12] Mantilla, R., Gupta, V. K., Troutman, B. M., 2011. Scaling of peak flows with constant flow velocity in random self-similar networks. Nonlinear Processes in Geophysics 18 (4), 489–502.

[13] Mantilla, R., Navarro, W., Ramirez, J.M., 2012. An analytic expresion for a scale dependent geomorphic instantaneous unit hydrograph (GIUH) and implications for scaling of peak flows. Advances in Water Resources Research(*Submitted*).

[14] Menabde, M., Sivapalan, M., 2001. Linking space-time variability of river runoff and rainfall fields: a dynamic approach. Advances in Water Resources 24, 1001–1014.

[15] Pacheco, P., 1997. Parallel Programming with MPI. Morgan Kaufmann Publishers, Inc.

[16] Qu, Y., Duffy, C. J., 08 2007. A semidiscrete finite volume formulation for multiprocess watershed simulation. Water Resources Research 43 (8).

[17] Small, S. J., Jay, L. O., Mantilla, R., Curtu, R., Cunha, L. K., Fonley, M., Krajewski, W. F., 2012. An asynchronous solver for systems of odes linked by a directed tree structure. Advances in Water Resources Research(*Submitted*).

[18] Vivoni, E. R., Mascaro, G., Mniszewski, S., Fasel, P., Springer, E. P., Ivanov, V. Y., Bras, R. L., 2011. Realworld hydrologic assessment of a fully-distributed hydrological model in a parallel computing environment. Journal of Hydrology 409, 483–496.