Consistent Initial Conditions for the Saint-Venant Equations in River Network Modeling

Cheng-Wei Yu
Center for Water and the Environment, The University of Texas at Austin
email: yuchanway@utexas.edu

Frank Liu
IBM Research Austin
email: frankliu@us.ibm.com

Ben R. Hodges
Center for Water and the Environment, The University of Texas at Austin
email: hodges@utexas.edu

Friday 23rd June, 2017

Running head: Initial Conditions for Saint-Venant Equations

Abstract

Initial conditions for flows and depths (cross-sectional areas) throughout a river network are required for any time-marching (unsteady) solution of the one-dimensional (1D) hydrodynamic Saint-Venant equations. For a river network modeled with several Strahler orders of tributaries, comprehensive and consistent synoptic data are typically lacking and synthetic starting conditions are needed. Because of underlying nonlinearity, poorly-defined or inconsistent initial conditions can lead to convergence problems and long spin-up times in an unsteady solver. Two new approaches are defined and demonstrated herein for computing flows and cross-sectional areas (or depths). These methods can produce an initial condition data set that is consistent with modeled landscape runoff and river geometry boundary conditions at the initial time. These new methods are: (1) the Pseudo-Time-Marching Method (PTM) that iterates toward a steady-state initial condition using an unsteady Saint-Venant solver, and (2) the Steady-Solution Method

(SSM) that makes use of graph theory for initial flow rates and solution of a steady-state 1D momentum equation for the channel cross-sectional areas. The PTM is shown to be adequate for short river reaches, but is significantly slower and has occasional non-convergent behavior for large river networks. The SSM approach is shown to provide rapid solution of consistent initial conditions for both small and large networks, albeit with the requirement that additional code must be written rather than applying an existing unsteady Saint-Venant solver.

Keywords: Flood modeling, One-dimensional models, River channels, Streams and rivers Initial condition, Model spin up

1 Introduction

1.1 Motivation

Setting initial conditions for unsteady simulations of the Saint-Venant equations (SVE) across large river networks can be challenging. Every element of the river network must be given initial values of flow and depth, and these values should be consistent with the inflow boundary conditions (e.g. from a land surface model) at the starting time to prevent instabilities. This issue has not been previously addressed in the literature, arguably because (i) adequate initial conditions are fairly trivial for small SVE reaches, (ii) hydrological models with large river networks often do not use the full SVE (e.g. Beighley et al., 2009), or use it over a smaller set of reaches (e.g. Paiva et al., 2012), and (iii) time-marching models only consider results after spin-up time is complete (i.e. after the effects of the initial conditions have been washed out of the system solution), which implies the initial conditions are entirely irrelevant in analyzing the model results. However, for SVE solutions of river networks the initial conditions can dramatically affect both spin-up time and convergence. Indeed, in our experience näive initial conditions can cause the spin-up time to be longer than time-marching period of interest. In extreme cases this can result in divergence and complete failure of an SVE solver. Consistency between the initial conditions and the boundary conditions appears to be necessary for short spin-up times, and thus is the focus of this study. Note that the need for consistency in an SVE model initialization is due to the coupling of nonlinearity and the water surface slope in the momentum equation, so common reduced-physics models (not discussed herein) may not be as sensitive to consistent initial conditions.

Saint-Venant equation modeling arguably dates from Priessmann's seminal work (Preissmann, 1961; Preissmann and Cunge, 1961), followed by decades of advances in techniques and applications (Cunge, 1974; Ponce et al., 1978; Cunge et al., 1980; Abbott et al., 1986; Zhao

et al., 1996; Sanders, 2001; Pramanik et al., 2010). These models focused on hydraulics of short river reaches or main stem rivers that are easy to initialize for flow and depth. It is only recently that the solvers for large river networks have become practical (Hodges, 2013; Liu and Hodges, 2014), and it is with large networks that initial conditions are problematic. Indeed, initial conditions and associated spin-up problems have been recently acknowledged and investigated for hydrological models (e.g. Ajami et al., 2014), but without consideration of a separate river network model. Work by Seck et al. (2015) and Rahman and Lu (2015) show that hydrological model spin-up computational times could be significant and were dominated by the selected initial hydrological conditions.

Our experience with Saint-Venant river network modeling is that simple approaches to initial conditions often cause localized numerical instabilities, slow convergence of the time-marching numerical solution, and long model spin-up times. Herein, we investigate the initial condition problem for a Saint-Venant river network model for a given set of inflows from a hydrological model.

1.2 Synthetic *v.* observed initial conditions

Model spin-up time is completed when the effects of initial conditions cannot be observed in the model results. Thus, by definition the initial conditions cannot affect the unsteady solution beyond spin-up time. It follows that initial conditions are irrelevant to the quality of the time-marching simulation and are only important in how they affect the spin-up duration. We can imagine a "perfect" set of initial conditions with zero spin-up time, which would require initial flows and depths consistent with (i) the actual unsteady behavior prior to the model start time and (ii) the model boundary conditions – the latter includes both the bathymetric model for the river channels and the coupled hydrological model providing runoff and base flows. Such perfect initial conditions are practically unattainable due to the sparsity of synoptic flow/depth data as well as unavoidable uncertainty and errors in both bathymetric and hydrological models. Another way to think of this is that perfect initial conditions also requires perfect boundary conditions (perfect bathymetry and hydrology), or else some spin-up time is required to wash out inconsistencies. In general, the spin-up time will be affected by how far the initial conditions are from the theoretical perfect conditions.

The key point is that the exact observed river initial conditions (if such were available throughout a network) will *not* eliminate or necessarily reduce spin-up time if the observed data are inconsistent with the model boundary conditions. Similarly, interpolations of sparse synoptic data will not be *a priori* consistent with the boundary conditions, and thus cannot eliminate spin-up time. Inconsistency is a critical concept: the mismatches between the initial conditions and the boundary conditions can lead to unrealistic destabilizing impulses in time-

marching the SVE solution. Such impulses can require extensive spin-up time to damp their effects. An extreme example is a high runoff rate into an almost dry stream that can cause a Gibbs phenomenon at a wave front and negative values for the computed cross-sectional area (Lax, 2006; Kvočka et al., 2015; Yang et al., 2012). Although several studies show such numerical discontinuities can be resolved (Kazolea and Delis, 2013; Caleffi et al., 2003; Liang et al., 2006), the high computational cost of damping or resolving is a burden (Kvočka et al., 2015) that seems unnecessary during spin-up since it cannot affect the time-marching results.

We argue that the primary goal of initial conditions is providing consistency with the boundary conditions to allow smooth, convergent spin-up of an unsteady solver. This task can be accomplished with synthetic initial conditions that are independent of observations. The only practical discriminators between using observed and synthetic initial conditions are (i) the effort required to prepare the initial condition data, and (ii) the length of spin-up time. As demonstrated herein, consistent synthetic initial conditions can be readily generated for even large complex networks – a task that is daunting for interpolation/extrapolation of sparse synoptic observations. Indeed, developing consistent synthetic initial conditions only requires the channel geometry and hydrological model that are used for the SVE time marching, supplemented by a steady-state SVE solver (§2.3). In contrast, interpolation/extrapolation of synoptic data requires analysis of the data locations and model geometry, which is likely to require customization for each river network.

Unfortunately, we cannot definitively prove our second discriminator; the wide range of possible hydrological models, channel geometry models, and synoptic observation systems make it impossible to prove that initial conditions based upon observations will always require longer spin-up times. However, no one has proposed an approach for interpolation/extrapolation of observed synoptic data that provides simultaneous consistency with a hydrological model, a river geometry model, and the SVE. It follows that we are on firm ground in stating that any existing approach using observed data as SVE initial conditions will result in inconsistencies. Finally, as our work demonstrates that inconsistent initial conditions cause long spin-up times in SVE models of sufficiently large river networks, it can be argued that observed data should be deprecated as initial conditions until a consistent approach is demonstrated and the spin-up can be shown to be shorter than provided by synthetic initial conditions.

1.3 Initial condition approaches

Approaches for specifying initial conditions for the SVE can be grouped into three main categories: (i) a "synoptic start" applying an interpolated/extrapolated set of sparse observational data, (ii) a "cold start" with initial flow rates and flow depths prescribed either as zero (e.g. Chau and Lee, 1991) or from some analytical values, e.g. mean annual flows and depths, and

(iii) a "steady-state" start, which we describe herein. The metric for evaluating initial conditions is *not* how well they reflect available real-world observations, but how effective they are in efficiently providing a consistent set of initial conditions.

Based on our discussion above, the first approach (synoptic start) is unlikely to be efficient for SVE initial conditions in a large river network due to inconsistencies between observations and model boundary conditions as well as inconsistencies caused by interpolating/extrapolating sparse observations throughout a network. There are no proven approaches to analyzing consistency and melding observations to hydrological model runoff, so the river network model spin-up will be subject to random inconsistencies and instabilities that can delay or prevent convergence.

The second approach, a cold start, provides innumerable possible ways to create initial conditions. For example, mean annual flows and depths (e.g. from the NHDplus data in the USA) can provide a smooth and consistent set of flows and elevations throughout a network. Although such cold start initial conditions can be internally consistent, they may be far from the flows/depths implied by the initial hydrological forcing. For example, a river network model that is started with mean annual values would be substantially in error if the initial hydrological inflows were from the monsoon season. As a result of inconsistencies between the selected cold start values and the hydrological inflows, a cold start can require extensive spin-up time to dilute or wash out the error. Indeed the spin-up time dominated the computational time for the large SVE networks we previously modeled in Liu and Hodges (2014) when we used a cold start with mean annual values. It might be possible to design a cold start approach that is consistent with the hydrological inflows; however, we suspect that any such approach is likely to be merely a variant of the steady-state approach discussed herein.

Herein we investigate the third approach, steady-state initial conditions, as a preferred method for initializing a large river network model. With this idea, a set of consistent initial conditions is one that satisfies both the $t = 0$ hydrological forcing and the *steady-state* Saint-Venant equations at $t = 0$. That is, we know that we cannot match the *unsteady* SVE at $t = 0$ as we cannot perfectly know the time-varying nonlinear effects before the model starts (unless, of course, we run a model for the prior period, which would be simply time-shifting the initial condition problem). The steady-state approach has the advantage of providing flows and depths that are consistent across the entire network with all the boundary conditions (inflows and channel geometry) as well as consistent with the nonlinear governing equations. These consistencies eliminate destabilizing impulses otherwise caused by mismatches between the flow/depth in a river reach and the runoff, so subsequent time-marching of the unsteady solution is smooth. Furthermore, the steady-state solution is the closest available proxy to the unknown unsteady solution at $t = 0$, so this approach should minimize the spin-up time required to reach an

unsteady time-march that is independent of the initial conditions.

1.4 Overview

Herein we present an efficient approach to establishing a set of steady-state conditions that provides a consistent and smooth starting point for time-marching an unsteady Saint-Venant simulation. A full model initialization problem has two parts: (i) determine a set of flows and water surface elevations that are consistent steady solutions of the SVE for starting an unsteady solver, and (ii) determine the spin-up time needed to ensure errors in the initial conditions are washed out of the unsteady solution. The second problem is highly dependent on the network characteristics and the particular flow and boundary conditions during spin-up, so for brevity, this work deals quantitatively with solving the first problem, and then illustrates the effects on the second problem.

2 Methods

2.1 Saint-Venant equations

The Saint-Venant equations for temporal (t) evolution of flow and water surface elevation along one spatial dimension (x) following a river channel are generally derived using the hydrostatic and Bousinesq approximations applied to the incompressible Navier-Stokes and continuity equations (de Saint-Venant, 1871). Cross-section averaging to obtain the 1D equations is considered reasonable where cross-sectional gradients are smaller than along-channel gradients. However, the equations are widely used even where such assumptions are questionable (e.g. near a bridge with multiple immersed piers), with the effects of significant cross-section gradients or non-hydrostatic behavior being represented as empirical energy losses. A number of conservative and non-conservative equation forms have been used, with different advantages and disadvantages (Hodges and Liu, 2014). Herein we follow Liu and Hodges (2014) in using cross-sectional area (A) and flow rate (Q) as principle solution variables of the numerical system and the local water depth (h) and friction slope (S_f) as a secondary variables (i.e. variables that depend on A and Q through auxiliary relationships). The equation set can be written as:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = q_l \tag{1}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) + gA\frac{\partial h}{\partial x} = gA(S_0 - S_f) \tag{2}$$

where boundary conditions are the local channel bottom slope (S_0) and the local lateral net inflow (q_l), the latter representing both inflows from the landscape and outflows to groundwater.

Auxiliary equations for $h = h(A)$ are derived from river cross-section data. The Chezy-Manning equation can be used to provide the friction slope as:

$$AS_f = \tilde{n}^2 Q^2 F \tag{3}$$

where $\tilde{n}$ is the standard Manning's n roughness coefficient and F is a convenient equivalent friction geometry (Liu and Hodges, 2014), which subsumes the conventional hydraulic radius (R_h) using a definition of

$$F = \frac{1}{AR_h^{4/3}} = \left(\frac{P^4}{A^7}\right)^{1/3} \tag{4}$$

with $P = P(A)$ is the wetted perimeter and $R_h = AP^{-1}$. Note that eq. (4) fixes a typographical error in eq. (10) of Liu and Hodges (2014) and eq. (3.55) of Hodges and Liu (2014). Required boundary conditions for the unsteady Saint-Venant solution are $q_l(t)$ for each stream segment, $Q_{bc}(t)$ at the furthest upstream node (headwater) in river branches with a Strahler order of one, and h with an $h(A)$ relationship at the downstream boundary (assumed subcritical). The time-marching unsteady solution requires initial conditions for (Q, A), which can also be given as (Q, h) with $A = A(h)$. Implementation details of the unsteady solver used herein can be found in Liu and Hodges (2014) and Liu (2014).

2.2 Pseudo time-marching approach

The most obvious approach for finding steady-state initial conditions is to time-march an unsteady solver until a steady state is achieved. That is, we apply the unsteady solver with time-invariant boundary conditions of $q_l(t) = q_l(0)$ and $Q_{bc}(t) = Q_{bc}(0)$ for $t_0 \leq t < 0$ where t_0 is our pseudo-time start and $t = 0$ is the time for which we want a set of initial conditions. We call this the "pseudo time-marching method" (PTM). The initial condition for PTM is a set of $Q(t_0)$ and $A(t_0)$ for each stream segment (e.g. some cold start method as described above). At first glance, the logic here might seem circular: we are trying to solve for initial condition set $\{Q(0), A(0)\}$ of the unsteady model and PTM requires specifying $\{Q(t_0), A(t_0)\}$. This begs the question as to why PTM should be used rather than simply apply a cold start of the unsteady solver with $Q(0) = Q(t_0)$ and $A(0) = A(t_0)$. The answer is that the key difference between the PTM using $Q(t_0)$ and $A(t_0)$ and a cold start of the unsteady solver with the same values is that the former has time-invariant boundary conditions while the latter's are time-varying. Thus, an unsteady solver with time-varying boundary conditions is trying to take an inconsistent starting condition and converge it to a moving target. In contrast, the PTM takes the inconsistent starting conditions and attempts to converge them to a time-invariant target, which is more likely to

be successful. However, the PTM does not *a priori* ensure consistency between the $Q(t_0)$ and $A(t_0)$ starting conditions and the $t = 0$ boundary conditions. It follows that PTM performance can be subject to the same type of problems as a cold start depending on the choice of $Q(t_0)$ and $A(t_0)$ and the skill of the modeller in their selection. In §4.5 we show that PTM typically has problems for large river systems with complex geometry because the complexity of selecting a reasonable set of $\{Q(t_0), A(t_0)\}$ to ensure convergence.

The PTM is outlined as Algorithm 1. A user-selected parameter (ϵ) is used as a threshold tolerance value for declaring convergence to the steady state. A typical choice of the tolerance ϵ is the square root of the computer hardware tolerance. For example, on a 64-bit Intel architecture, the hardware tolerance for a double precision floating point floating number is 2.2204×10^{-16}, which means a good choice of ϵ is 1.4901×10^{-8}. As a practical matter, ϵ of 10^{-6} or even 10^{-4} is likely to be sufficient for initial conditions; that is, as further spin-up time is still required to dilute initial condition errors the convergence needs only to be sufficient for consistency across the network. The method can use a time-step size that is either constant or varying, with an automatic reduction in step size when convergence is not achieved in a given time step (Liu and Hodges, 2014). To avoid infinite runtimes for non-convergent behavior (e.g. due to instabilities developed with inconsistent starting conditions), the solution is terminated (failure to converge) in Algorithm 1 after a user-selected N_{max} iterations. The starting conditions for $\{Q(t_0), A(t_0)\}$ are discussed in Appendix A.

Algorithm 1 Pseudo Time-Marching Method

procedure PSEUDOTIMEMARCHING($\boldsymbol{A}_{ini}$, $\boldsymbol{Q}_{ini}$, ϵ, N_{max}) ▷ $\boldsymbol{A}_{ini}$,$\boldsymbol{Q}_{ini}$: initial guesses of A and Q; ϵ: tolerance; N_{max}: maximal iteration number

    $\boldsymbol{A} \leftarrow \boldsymbol{A}_{ini}$

    $\boldsymbol{Q} \leftarrow \boldsymbol{Q}_{ini}$

    $i \leftarrow 0$

    $t_0 \leftarrow 0$

    for $i = 1$ to N_{max} **do**

        Solve SVE at time point t_i using unsteady method

        Compute error: $e \leftarrow |\boldsymbol{Q}_t - \boldsymbol{Q}_{t-1}| + |\boldsymbol{A}_t - \boldsymbol{A}_{t-1}|$

        if $e < \epsilon$ **then**

            return *Success*

        end if

        $t_{i+1} \leftarrow t_i + \Delta t_i$

        $i \leftarrow i + 1$

    end for

    return *Failure*

end procedure

2.3 Steady-solution method

The PTM approach (above) results in a steady solution of the unsteady Saint-Venant equations that satisfies both momentum and continuity for time-invariant $q_l(0)$ and $Q_{bc}(0)$ boundary conditions in the unsteady solver. However, we can achieve a similar effect more directly by writing a steady-state version of the Saint-Venant equations as:

$$\frac{\partial Q}{\partial x} = q_l \tag{5}$$

$$\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) + gA\frac{\partial h}{\partial x} = gA(S_0 - S_f) \tag{6}$$

A key point, implied by eq. (5), is that the spatial gradient of steady-state Q over a stream segment is entirely due to the lateral inflow (q_l) without any influence of A. It follows that for steady q_l and Q_{bc} boundary conditions, the flow in the i^{th} river segment (Q_i) that has Strahler order S_i must be the sum of all the Q_j for all the j connected reaches of Strahler order $\mathsf{S}_j < \mathsf{S}_i$. That is, the steady flow at any point is simply the sum of all the connected upstream $t = 0$ boundary conditions. The corresponding A (and hence depth h) can then be computed with a numerical PDE solution of eq. (6) for known Q values. Note that for large river networks, the natural downstream boundary condition is subcritical, which requires specification of h and the corresponding A as the starting point. We call this a "steady-solution method" (SSM). To look at this from another viewpoint, if A is uniform then eq. (6) devolves the fundamental equation of gradually varying flow, $dE/dx = S_0 - S_f$, where E is the specific energy. Thus, the SSM corresponds to using the steady-state flow based on all boundary conditions and solving for surface elevations with gradually-varying flow solution for non-uniform cross-sections.

To efficiently compute the conservative initial Q throughout the river network, it is useful to apply graph theory as discussed in Hodges and Liu (2014). A river network can be classified as a "direct acyclic graph" (DAG) as a river may split upstream or downstream at a junction, but the flow cannot loop back to a starting point. The connectivity of a DAG can be efficiently computed by applying existing graph methods, such as depth-first-search (DFS) or breadth-first-search (BFS), which provide simple and efficient approaches to computing $Q(0)$ for each stream segment over an entire network. Note that these methods were designed and named by computer scientists, so "depth" in DFS and "breadth" in BFS do not refer to hydraulics or river geometry, but instead are jargon referring to the graph network characteristics. For simplicity in the present work, we confine ourselves to the subset of DAG systems that are simply-connected trees, i.e. where there is never more than a single downstream reach from any junction (as shown in Fig. 1) so that there are no uncertainties in flow directions or magnitudes. To extend the method to geometry with multiple downstream reaches (e.g. braiding, canals, deltas) requires additional rules for downstream splitting of flows that are beyond the scope of the present work.

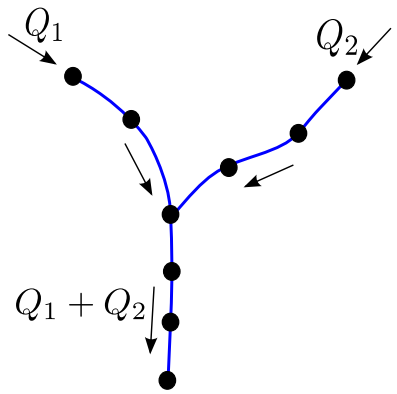


Figure 1: Propagation of flow rate Q at a junction.

A simple DFS traversal (Cormen et al., 2001) for Q is shown in Algorithm 2. From each headwater node (Q_j), the inflow boundary condition is propagated downstream by adding the value to the downstream node and including any lateral q_l from the upstream reach (stored in Q_k). For river networks, the DFS traversal is highly efficient and requires negligible computational time for river networks of 10^5 computational nodes (e.g. Liu and Hodges, 2014). Based on our experience, the DFS computational costs should be essentially trivial for even continental-scale systems of 10^7 nodes.

Algorithm 2 DFS traversal for Q

procedure QTRAVERSAL
  for all i **do** ▷ initialization
    $Q_i \leftarrow 0$
  end for
  for each headwater node j with BC $Q_j(t)$ **do**
    $Q_j \leftarrow Q_j(t = 0)$
    $k \leftarrow$ downstream node of node j
    while k is not empty **do**
      $Q_k \leftarrow Q_k + Q_j(t = 0)$
      $k \leftarrow$ downstream node of node k
    end while
  end for
  return
end procedure

After the steady Q_i for each stream segment is computed, eq. (6) can be solved for the corresponding A_i. We discretize this equation with the Preissmann scheme, similar to the approach used for the unsteady Saint-Venant equation in Liu and Hodges (2014). The value and derivative for any term are approximated as:

$$f(x,t) \simeq \frac{1}{2}(f_{j+1} + f_j) \tag{7}$$

$$\frac{\partial}{\partial x} f(x,t) \simeq \frac{1}{\Delta x}(f_{j+1} - f_j) \tag{8}$$

where subscripts indicate a node in the discrete system. Using $j + 1/2$ to represent geometric data that is logically between nodes (i.e. roughness $\tilde{n}$ and S_0), eq. (6) becomes:

$$\begin{aligned} &\frac{2}{\Delta x}\left[\frac{(Q_{j+1})^2}{A_{j+1}} - \frac{(Q_j)^2}{A_j}\right] \\ &+\frac{g}{\Delta x}\left(A_{j+1} + A_j\right)\left(h_{j+1} - h_j\right) - g\left[A_{j+1} + A_j\right] S_{0(j+1/2)} \\ &+g\tilde{n}^2_{j+1/2}\left[(Q_{j+1})^2 F_{j+1} + (Q_j)^2 F_j\right] = 0 \end{aligned} \tag{9}$$

These nonlinear equations are similar to the unsteady discrete equations, except that Q for each computational node is known from the DFS traversal. Newton's method is used to solve this system for A without linearization, similar to the approach in Liu and Hodges (2014). The SSM requires a starting guess for A to solve the steady-state problem. Herein we use a bisection method with the Chezy-Manning equation for normal depth conditions, discussed in Appendix A. The overall algorithm for SSM is illustrated in Algorithm 3.

Algorithm 3 Steady-Solution Method

```
procedure SteadySolution
    Call QTraversal()
    for all node j in network do                    ▷ Initial guess of A
        Call bisection routine BiSection(Q_j)
    end for
    Solve steady version of dynamic eqn in eq. (9)
    return
end procedure
```

3 Computational Tests

3.1 Overview

The performance of PTM and SSM are examined with a series of test cases ranging from simple uniform cross-sections over short river reaches to 15,000 km of a real river network. To demonstrate the robustness and performance of the SSM, we conduct tests from three perspectives: (i) effects of different cross-section geometries; (ii) scalability with an increasing number of computational nodes; and (iii) real-world river networks. Two different computers are used: the cross-section and scalability tests are run on a computer with 2.00GHz Intel Xeon D-1540 CPU's and 64GB of RAM, while the large network tests are run on a computer with 2.52GHz Intel i7-870 CPU's and 8GB of RAM. In both cases Ubuntu Linux is the operating system and GNU C++ compiler is used.

3.2 Effects of cross-section geometry

Test cases for cross-section geometry effects were conducted for synthetic geometry of simple river reaches without tributaries. Cases included rectangular, parabolic, trapezoidal, and non-uniform cross-sections, with a range of channel lengths, widths, and computational nodes, as provided in Table 1.

test case	channel length (km)	number of computational nodes	cross-section shape type	cross-section shape detail
Case 1	3.1	78	Uniform rectangular	$W_B = 20$ m
Case 2	0.2	6	Uniform trapezoidal	$W_B = 1$ m $S_{sw} = 0.5$
Case 3	0.3	6	Uniform trapezoidal	$W_B = 0.1$ m $S_{sw} = 1.5$
Case 4	5.6	71	Uniform trapezoidal	$W_B = 10$ m $S_{sw} = 0.5$
Case 5	10	167	Uniform quasi-parabolic	$f = 37.8$
Case 6	10	1664	Surveyed bathymetry	Unsymmetrical cross-section
Case 7	122	31	Surveyed bathymetry	Unsymmetrical cross-section

Table 1: Cross-section geometry test cases. W_B and S_{sw} represent bottom width and sidewall slope respectively, f represents the focal length of parabolic shape.

3.3 Scalability

To demonstrate the scalability as the number of computational nodes increases, we use the geometry and flow conditions of Case 4 in Table 1 and generate synthetic test cases with increasing

numbers of nodes from a few hundred to over a million in the set: { 560, 2800, 5600, 11200, 22400, 44800, 89600, 179200, 358400, 716800, 1433600 }.

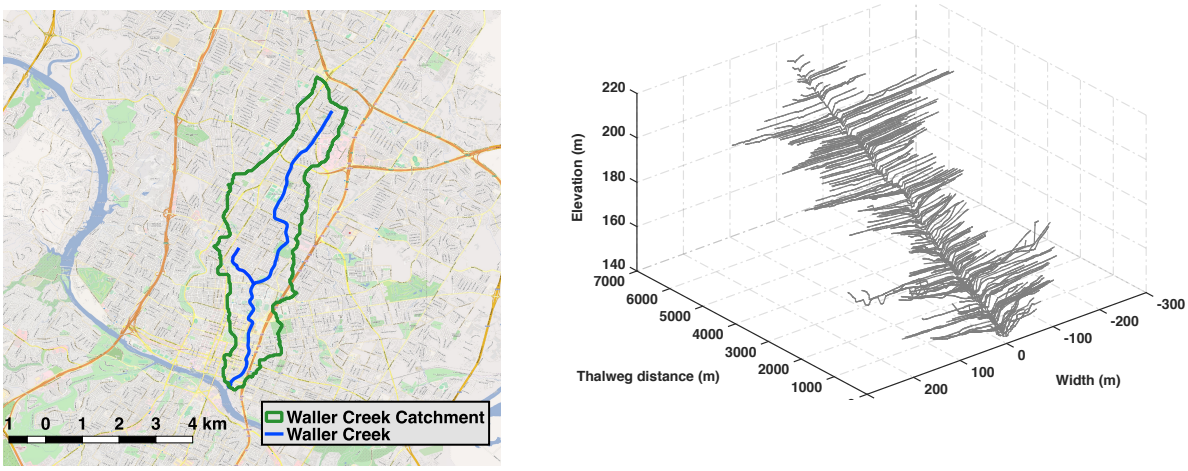
3.4 Large river networks

To examine the robustness of PTM and SSM for more realistic conditions over both small and large scales, we use a section of Waller Creek (Texas, USA) as well as the entire watershed of the San Antonio and Guadalupe River basins (Texas, USA). The former is a small urban watershed for which dense cross-section survey data is available, whereas the latter is a large river basin that has been previously modeled with the RAPID Muskingum routing model (David et al., 2011) and the SPRNT Saint-Venant model (Liu and Hodges, 2014).

The Waller Creek study includes two stream reaches and the catchment area illustrated in Fig. 2. The total stream length is 11.6 km, which drains an area of 14.3 km^2. The layout of Waller Creek is shown in Fig 2(a), and parts of the bathymetry surveyed data from City of Austin is shown in Fig. 2(b) for clarity. Two different model geometries were considered, which are designated as WCA and WCB. For WCA the stream is discretized by 373 computational nodes based on separation of the surveyed cross-sections. WCA neglects the minor tributary of Waller Creek and includes the full complexity of the surveyed cross-sections shown in Fig. 2(b). In contrast, WCB includes both tributaries, but uses wider computational node separation with only 30 of the 373 surveyed cross-sections.

To test the initial condition approach for a large river network, we use the San Antonio and Guadalupe River basins (Fig. 3), which have a combined total stream length of 12,728 km (excluding some minor first-order segments). The model herein uses 63,777 computational nodes, 59,594 segments, and 2,643 junctions, but is otherwise similar to the model setup with 1.3×10^5 nodes used in Liu and Hodges (2014). Although the unsteady SPRNT model is typically run by coupling with a land-surface model for headwater and lateral inflows, for the present steady-state tests we used a synthetic inflow data set for the headwater inflows. The synthetic flow at each headwater stream was computed based on a downstream peak flow rate distributed uniformly across all the headwater reaches. We used the peak flow rate recorded on the main stem of Guadalupe River at Victoria (Texas) on January 19th, 2010 by USGS gauge 08176500. As this gauge does not include the San Antonio River flows, we divided the peak flow rate (453 m^3s^{-1}) by the total number of headwater streams in the Guadalupe River (815) to get a single inflow value that was applied to each headwater reach (0.55 m^3s^{-1}). The same flow rate was used for the 725 headwater reaches of the San Antonio River network. This approach ensures that there is flow in every branch in the river network.

As is often the case in large river networks, comprehensive cross-section geometry data were not available for the San Antonio and Guadalupe Rivers. Indeed, Hodges (2013) noted data



(a) (b)

Figure 2: (a) Waller Creek and catchment in Austin (Texas, USA) (b) Surveyed cross-sections of main channel for Waller Creek (Texas). Only 149 of 327 cross-sections are shown for clarity. Elevations are relative to mean sea level. Data courtesy of City of Austin.

availability and our ability to effectively use synthetic geometry as one of seven fundamental challenges to Continental River Dynamics modeling. Because the geometry affects both PTM and SSM solutions, we tested four different estimation approaches for synthesizing geometry (Cases A, B, C, and D). Cases A uses synthetic trapezoidal cross-sections using the approach applied in Liu and Hodges (2014) based on Western et al. (1997). In this method, trapezoidal widths (W) are computed from mean annual flows (Q_m) from the NHDPlus dataset as $W = \alpha Q_m^{0.5}$ with $\alpha = 1.5$. For the side slope of trapezoidal cross section, an identical sidewall slope (45 degrees) is used throughout the river network. Case B channels were similar to Case A, but included some minor changes to Manning's n, inflow boundary conditions, and channel bottom slopes in reaches where instabilities occurred, which was necessary to provide convergence for the PTM (see §4.5). Case C channels were based on work of Santibanez (2015), who used USGS streamflow measurements in the San Antonio and Guadalupe River network along with the at-a-station hydraulic geometry approach (Rhodes, 1977) to find the best trapezoidal cross-section approximation for the drainage area. Using this approach, the bed width (b_0) is an exponential function of cumulative drainage area (A_D) as:

$$b_0 = \gamma A_D^{\lambda} \tag{10}$$

where b_0 is meters, A_D is km^2, and the coefficients are $\gamma = 12.59$ and $\lambda = 0.382$. The Santibanez

(2015) approach provides reasonable values for trapezoidal channel sidewall slopes over most of the basin, but fails in many of the first-order streams with small drainage areas (< 25 km^2) where the computed sidewall slopes are near zero. For simplicity in the present test cases, a uniform value of 45 degrees is used for the sidewall slopes throughout the river network. Case *D* uses channel bathymetry data generated from Zheng (2016), which uses a Height Above Nearest Drainage (HAND) analysis (Nobre et al., 2011) applied to the National Elevation Dataset (NED) to provide an automated approach for estimating trapezoid-based composite cross-sections.

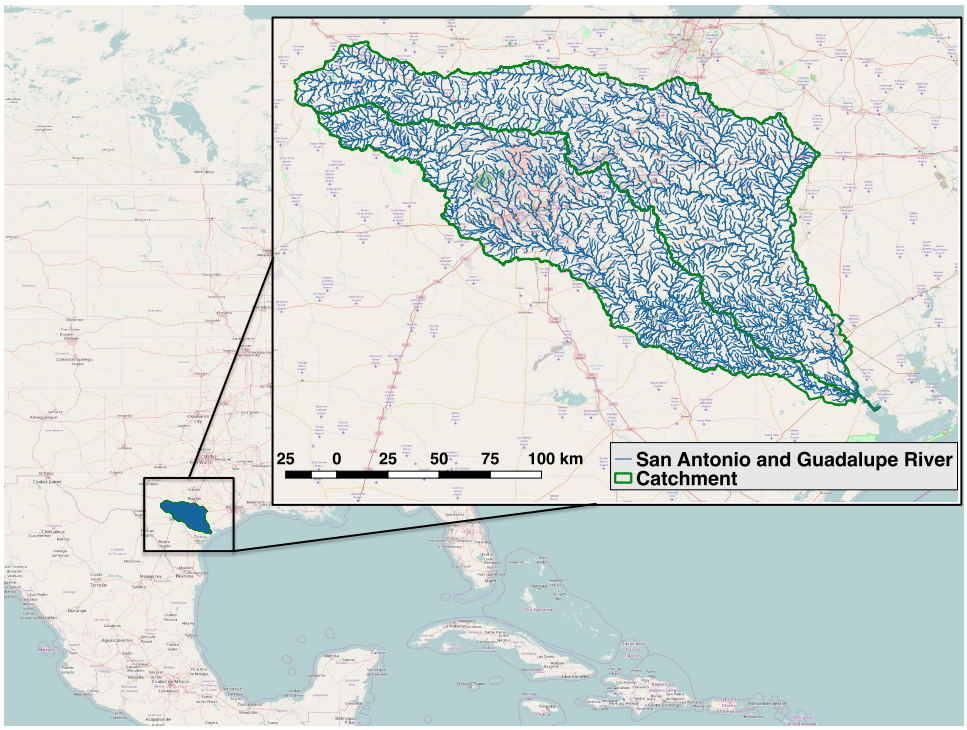


Figure 3: San Antonio and Guadalupe River network, from NHDPlus V2 Flowline

4 Results

4.1 Comparison metrics

The overall algorithm efficiency is evaluated by the number of Newton iterations required for convergence to steady-state. The number of Newton iterations reflects the difficulty in converging the nonlinear solution and is proportional to the simulation runtime. As this metric is independent of computer architecture it provides a universal measure of algorithm performance. For SSM we use the number of iterations to converge the area (A) solution of eq. (6), which is the dominant computational cost (i.e. the non-iterative graph-traversal solution for Q is negligible in comparison). For PTM, we use the cumulative sum of Newton iterations for the (Q, A) solution over all pseudo-time steps. Where converged solutions of PTM and SSM both exist, comparisons (not shown) indicate the resulting (Q, A) steady-state results are identical within the convergence tolerance ($\epsilon = 10^{-6}$).

4.2 Effects of cross-section geometry

Table 2 provides a comparison of Newton iterations for the test cases of Table 1 for single reaches with different channel cross-sections. The SSM converges quickly across all cases, whereas the performance of the PTM is always substantially slower than the SSM. The performance of the PTM appears somewhat erratic, which is likely because the overall number of pseudo-time steps depends on how far the starting guess is from the converged answer and the size of the time step used in the PTM pseudo time march.

test case	PTM iterations	SSM iterations	relative speed-up of SSM
Case 1	327	6	54×
Case 2	73	4	18×
Case 3	136	8	17×
Case 4	773	9	85×
Case 5	8634	76	113×
Case 6	13 765	4	3441×
Case 7	91 234	30	3041×

Table 2: Newton iterations required to achieve convergence for benchmark geometry test cases. The converged results are identical for both methods.

By comparing the geometric data from Table 1 with the results in Table 2, it can be seen that the largest discrepancies between PTM and SSM performance (Cases 6, 7) are with non-

uniform cross-sections. In both of these the SSM performs $O(10^3)$ times better, compared to $O(10)$ to $O(10^2)$ improvements for simple geometry cases. This result is consistent with the idea that the performance of PTM depends on how close the starting guess for $\{Q, A\}$ is to the steady-state solution. With non-uniform cross-section geometry, the starting guess is generally a quite far from the steady-state condition as it is difficult to *a priori* estimate gradients of the water surface that match the nonlinear acceleration associated with cross-section variability. In contrast, the benchmark tests with simple cross-section geometry (Cases 1-5) show more modest speed-up by SSM, which is consistent with the steady-state solution for PTM with simple geometry being closer to the starting guess. For short reaches with simple geometry and only a few computational nodes (Cases 2, 3) the speed-up by SSM is essentially irrelevant.

4.3 Scalability

Computing initial conditions using models with varying numbers of computational nodes for Case 4 in Table 1 provides the speed-up results shown in Fig. 4. These tests use simple trapezoidal cross-sections and, consistent with the results above, the speed-up advantage of the SSM is relatively modest with less than 10^3 nodes. However, beyond this point, the effective speed-up with SSM is quite dramatic. It appears that the SSM method becomes more effective than PTM both with increasing complexity of the cross-sectional geometry and the increasing number of computational nodes.

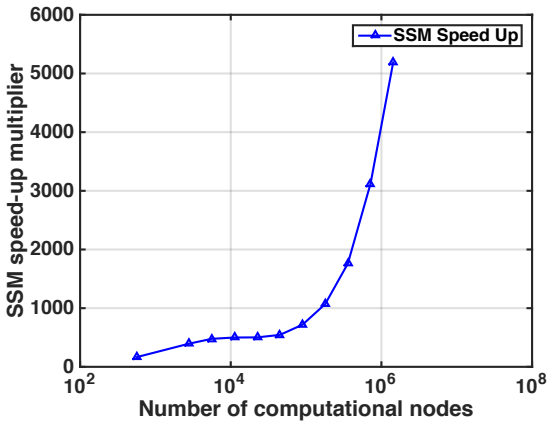


Figure 4: Speed-up multiplier of SSM compared to PTM for Case 4 as a function of the number of computational nodes.

4.4 Waller Creek test cases

The results of initial condition convergence for two Waller Creek simulations are shown in Table 3. The SSM method dramatically reduces the total number of iterations to convergence, which is also reflected in reducing the computer runtime by 99% and 92% for WCA and WCB, respectively. Although the absolute runtime for this small system is trivial by either PTM or SSM, the disparity provides insight into the performance that is confirmed with the more complicated river network, discussed below.

configuration	pseudo time-marching iterations (PTM)	steady-solution iterations (SSM)	relative speed-up of SSM	PTM runtime	SSM runtime
WCA	2900	23	130×	1.570 sec	0.011 sec
WCB	890	13	70×	0.037 sec	0.003 sec

Table 3: Total Newton's iterations required to achieve convergence of the Waller Creek Creek test case.

4.5 San Antonio and Guadalupe River basins

The results of the full river network computations are provided in Table 4, which shows the SSM was successful and used a relatively small number of Newton iterations despite the complexity of the system. Although results in §4.2 and §4.4 indicate that PTM method is acceptable for small systems, results for large-scale river network simulations are less promising. In the San Antonio and Guadalupe River network configurations, three out of four PTM solutions failed; that is, the method diverged from any selected starting condition and finally caused convergence failure. Configurations A, C and D all showed evidence of numerical instabilities leading to divergence. For example, configuration D in PTM method failed at convergence after 497 iterations, the maximum L2 convergence norm of the matrix reached up to $1.51e^{235}$, which is unquestionably divergent. Our inability to converge PTM with any A, C, or D cases led to development of the ad hoc B setup that provided the only PTM solution on the full river network. To obtain the B configuration we identified reaches where instabilities developed and made minor ad hoc adjustments for Manning's n, inflow boundary conditions, and channel bottom slopes until the model showed reasonable convergence behavior (see below). Note that the modeler's time to tune the system for the PTM method to successfully converge is not included in the comparisons of Table 4.

The convergence behavior of the PTM for configuration B is shown in Fig. 5. It can be seen that for several hundred time marching steps the solution was oscillating rather dramatically, but eventually settled down to a slow, smooth behavior. We believe this is evidence of the PTM trying to overcome inconsistencies between the $\{Q(t_0), A(t_0)\}$ starting conditions and the boundary conditions in the network. Note that PTM for B was not converged to the same $\epsilon = 10^{-6}$ tolerance used for SSM. Instead, the solution was manually terminated after 9+ hours, when the convergence norm reached 1.6×10^{-4} and was sufficiently smooth so that it was clear that the method would eventually converge.

configuration	PTM Newton iterations	SSM Newton iterations	relative speed-up of SSM	PTM runtime*	SSM runtime
A	convergence failure	61	–	–	3 sec
B	192,527	51	$> 3775\times$	9 hr 5 min 8 sec	3 sec
C	convergence failure	29	–	–	6 sec
D	convergence failure	46	–	–	14 sec

Table 4: Total Newton's iterations required to achieve convergence for four configurations of the San Antonio and Guadalupe River network. *The PTM method was terminated after the L2 convergence norm reached 1.6×10^{-4} whereas the SSM was converged to the predefined tolerance of 10^{-6}

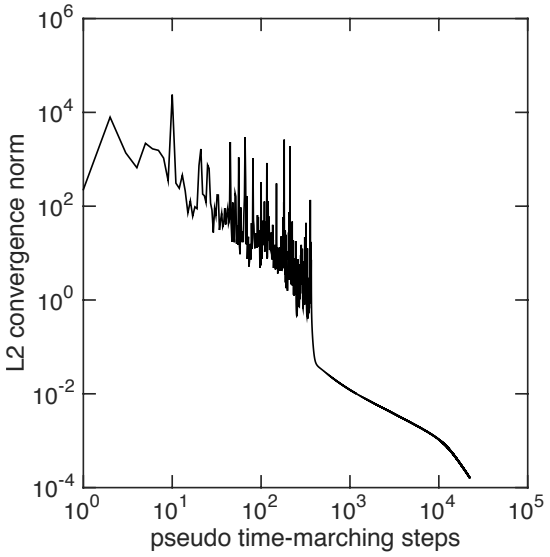


Figure 5: Convergence of the L2 norm between consecutive pseudo-time marching solutions for the PTM with configuration B of the San Antonio and Guadalupe River network. Note the above uses the number of time-marching steps as compared to the larger number of Newton iterations provided in Table 4.

5 Discussion

5.1 Effects on spin-up

As alluded to in the Introduction, obtaining an effective model initial condition is only one step in the initialization of an unsteady model. A second step is understanding at what time the model results are independent of any errors or inconsistencies in the initial conditions – i.e. the spin-up time. Some model spin-up time is generally unavoidable as we never have exactly the correct spatially-distributed initial conditions that are exactly consistent with spatially-distributed boundary conditions. In effect, eliminating spin-up time requires a set of initial conditions that are not only consistent with the boundary conditions at $t = 0$, but also consistent with the boundary conditions for $t_m < t < 0$, where t_m represents the system "memory" (or the time interval to wash out a transient impulse).

As an illustration of the scale of the spin-up problem compared to the initial condition problem, we have run the SPRNT unsteady SVE model (Liu and Hodges, 2014) for the San Antonio and Guadalupe River network using 30000+ data points of unsteady lateral inflows for 14 days in January 2010. These boundary condition data were generated from NLDAS. The initial conditions were generated using SSM, as described above. The initial conditions were then perturbed by $\pm 20\%$ in every first-order reach, which provides two slightly different initial condition data sets to compare to the baseline. In Figure 6, the time-marching results for the perturbed and baseline initial condition cases reach the same state throughout the network (0.001% threshold value) at 152 and 154 hours of simulation time, respectively. Thus, approximately 160 hours represents a conservative estimate of the expected time for errors in first-order streams to be diluted in the higher-order (larger) river branches. Note that it only takes 3.8 seconds of CPU time to compute initial conditions using SSM and an additional 5 minutes of CPU time to compute the time-marching during the spin-up interval with the SPRNT unsteady model. This is two orders of magnitude faster than the 9+ hours of CPU time required just to compute initial conditions using PTM for the same system.

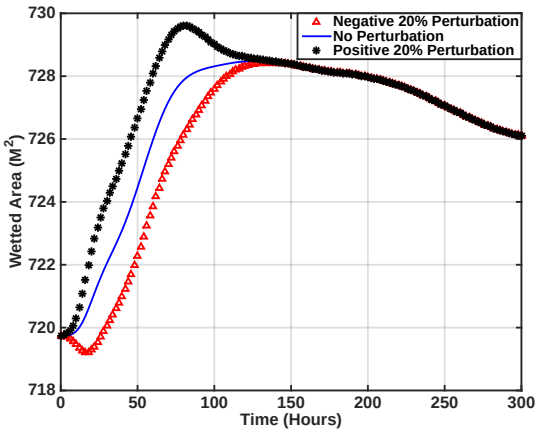


Figure 6: Spin-up for San Antonio and Guadalupe River network with the SPRNT unsteady SVE model initialized using the SSM approach. The positive and negative 20% perturbations are to the Q initial conditions in first-order reaches.

5.2 Model performance

In general, the PTM performed poorly except on very simple systems. As the river network complexity increases the PTM changes from being somewhat slower than SSM to being non-convergent. Indeed, the PTM has only one advantage over the SSM in providing initial conditions to an unsteady SVE solver: specifically, no new code is needed as PTM uses the same unsteady SVE code. However, using PTM for large systems requires a frustrating trial and error approach to tuning the system to obtain convergence. In contrast, the SSM provides a rapid solution to the initial condition problem because Q is computed from simple graph traversal (once through the network), and the subsequent computation of A is "local" (in the sense there is no coupling between distant computational nodes). Note that in contrast to PTM, the SSM does not require the modeler to select a set of starting conditions; i.e. the SSM starting condition (Q) is defined solely by the $t = 0$ boundary conditions. Thus, different modelers will produce exactly the same $Q(0)$ and $A(0)$ with the same number of iterations when using the SSM on identical geometry and boundary conditions. This same cannot be said for PTM as modelers must select their $\{Q(t_0), A(t_0)\}$ starting condition and may need to resort to custom model tuning to obtain convergence.

Herein we only tested two methods for initial conditions, both based on finding the steady-state $\{Q, A\}$ that are consistent with the boundary conditions. However, we can also argue that the cold start and synoptic start (see §1.3) would likely perform as bad or worse than PTM. That the cold start would perform poorly follows from the fact that it has the exact same problem as the PTM (converging over time from inconsistent starting data), but increases the difficulty by trying to converge to the unsteady boundary conditions. A cold start effectively turns the initial condition problem into a spin-up problem. For a cold start model performing similar to the PTM for the San Antonio and Guadalupe River network, we can expect spin-up to require more than 10^4 time-steps of the unsteady solver.

Although it is possible that a synoptic start could perform better than PTM or a cold start, it seems likely that any approach to interpolating/extrapolating sparse observational data across a larger river network will necessarily result in inconsistencies between the initial $\{Q, A\}$ and the boundary conditions. If such inconsistencies result in model instabilities (a difficult thing to predict), the overall model spin-up time could be extensive. The key problem for the synoptic start is that it requires judgment as to how to best interpolate/extrapolate observational data for initial conditions, which is contrasted to the SSM approach of simply using the actual $Q(0)$ boundary conditions and the steady solver without any further choices by the modeler.

Note that the poor performance of the PTM cannot be attributed to inefficiencies in the unsteady solver. As discussed in §5.1, the unsteady solver computed 150 hours of unsteady simulation from the SSM in about 5 minutes, or 1800× faster than real time. This implies that

the unsteady solver computed 114 minutes of real-world unsteady flows in the 3.8 s required for the SSM to compute the steady-state initial condition. Thus, the SPRNT unsteady solver is quite efficient – except when given an inconsistent set of initial and boundary conditions as in the PTM solution.

5.3 Limitations

For simplicity, the SSM algorithms presented herein are for common river networks with only a single reach downstream of each junction. Such tree networks are a subset of the more general DAG river networks. Extension to more complex DAG geometry requires the definition of splitting rules to uniquely define partitioning of Q where the flow splits into multiple downstream branches. Search algorithms for more complicated DAG forms are available in the graph theory literature (e.g. Cormen et al., 2001), and could be readily extended to the SSM.

Fundamental to the SSM approach is the presumption that there are no places of net flow accumulation or loss throughout the network. All reservoirs and hydraulic structures are treated as pass through so that all the upstream flow is propagated through the downstream reaches. However, if there are known locations of accumulation or loss, the river network could be divided into subnetworks with separate SSM solutions. For example, upstream of a reservoir an SSM solution can be used to obtain the inflow to the reservoir, which can be used with operating information from the reservoir to provide the correct outflow Q for use in the SSM solution for downstream reaches. Hydraulic structures that affect h (or dh/dx) as a function of Q can be readily included in the SSM. To do so merely requires changing the momentum equation in a reach to model the physics of the structure (as is similarly done in unsteady SVE models).

6 Conclusion

It is demonstrated that inconsistencies between initial conditions and boundary conditions for a large river network solver of the Saint-Venant equations can lead to long spin-up times or solution divergence. We that synthetic initial conditions are preferred over observed synoptic initial conditions due to the ability of the former to provide smooth and consistent spin-up. Two methods to compute synthetic initial conditions for flow (Q) and cross-sectional area (A) for an unsteady Saint-Venant river network model have been presented. Both approaches use the steady-state solution for $t = 0$, which provides initial conditions that are smooth and globally consistent with the boundary conditions for the model start time. The Pseudo Time-Marching Method (PTM) is likely similar to undocumented approaches previously used; i.e. application of an unsteady model with constant boundary conditions to achieve a steady solution consistent with initial land surface inflows. For large river networks, the PTM method is slow and inconsistent, arguably depending on the quality of the first guesses for Q and A and the size of the time step required for a stable pseudo-time march. A new Steady-Solution Method (SSM) is developed to address these issues. The SSM computes the initial condition Q in each reach from the inflow boundary conditions of the entire network at $t = 0$ by applying a mass-conservative graph traversal technique. The initial condition A in each reach is found from the

solution of the steady-state 1D momentum equation with known Q. The first-guess problem for A is solved as a normal-flow problem with the Chezy-Manning equation and the Q from graph traversal. Although SSM requires writing an additional numerical solver rather than relying on an existing unsteady solver (as used in PTM) our numerical experiments show that SSM is more robust and consistently faster than PTM. Code for both initial condition solvers is publicly available at GitHub (Liu, 2014).

7 Acknowledgements

This material is based upon work supported by the U.S. National Science Foundation under Grant No. CCF-1331610.

8 Notation

A cross-sectional area (m^2)

b_0 trapezoidal channel bed width (m)

b_1 trapezoidal channel sidewall slope

DA drainage Area (mile^2)

g gravitational acceleration (ms^{-2})

f generic function

F equivalent friction geometry ($\mathrm{m}^{-10/3}$)

h depth (m)

$\tilde{n}$ Manning's roughness ($\mathrm{m}^{-1/3}\mathrm{s}$)

P wetted perimeter (m)

Q volumetric flow rate ($\mathrm{m}^3\mathrm{s}^{-1}$)

q_l flow rate per unit length through channel sides ($\mathrm{m}^2\mathrm{s}^{-1}$)

r residual function

R_h hydraulic radius (m)

S_0 channel bottom slope

S_f channel friction slope

S Strahler order

t time (s)

W channel width (m)

x along-channel spatial coordinate

Appendix A: Starting conditions

PTM requires starting conditions (or a first guess) of $\{Q(t_0), A(t_0)\}$ for unsteady solution of eqs. (1) and (2) whereas SSM needs a first guess only for A in the solution of eq. (6). As the Q solution by the graph traversal method for eq. (5) does not require any starting conditions, it follows that for the PTM the best choice for $Q(t_0)$ is the same Q developed by simple graph traversal approach used for SSM (i.e. Alg. 2). To obtain starting conditions for A in SSM or $A(t_0)$ in PTM, a reasonable guess is the A associated with the "normal depth", denoted A_n for the starting Q. The normal depth is obtained from the Chezy-Manning equation solved for normal flow conditions, i.e.

$$Q = \frac{1}{\tilde{n}} A_n R_h^{2/3} S_0^{1/2} \tag{11}$$

The hydraulic radius at normal depth requires the area at normal depth and the wetted perimeter at normal depth, $R_{h(n)} = A_n/P_n$, which implies Chezy-Manning can be written as:

$$A_n = \left(\frac{\tilde{n}Q}{S_0^{1/2}} \right)^{3/5} P_n^{2/5} \tag{12}$$

Thus an initial guess for A can be computed for known Q, $\tilde{n}$, and S_0, where $P = P(A)$ is a known piece-wise continuous function based on river bathymetric data. Since $P(A)$ is a nonlinear function, eq. (12) must be solved with a nonlinear solution method. A simple bi-section method can be used following Algorithm 4 with the residual function $r(A)$ defined as:

$$r(A) = \left(\frac{\tilde{n}Q}{S_0^{1/2}} \right)^{3/5} P(A)^{2/5} - A = 0 \tag{13}$$

Note that failure to converge for Algorithm 4 is not necessarily fatal; unconverged results are likely to be adequate as they are simply the initial guess for iterative solution by PTM or SSM. As a further simplification, it seems likely that the $P(A)$ in eq. (12) could be approximated using a simple rectangular cross-section, $P(A) = W + 2A/W$, where known channel widths (W) are used. This simplification is valuable in continental-scale river network simulations, where adequate river geometric data throughout a network cannot be guaranteed (Hodges, 2013).

References

Abbott, M., Bathurst, J., Cunge, J., O'connell, P., and Rasmussen, J. (1986). An introduction to the european hydrological system—systeme hydrologique europeen,"she", 2: Structure of a physically-based, distributed modelling system. Journal of hydrology, 87(1):61–77.

Ajami, H., McCabe, M. F., Evans, J. P., and Stisen, S. (2014). Assessing the impact of model spin-up on surface water-groundwater interactions using an integrated hydrologic model. Water Resources Research, 50(3):2636–2656.

Beighley, R., Eggert, K., Dunne, T., He, Y., Gummadi, V., and Verdin, K. (2009). Simulating hydrologic and hydraulic processes throughout the amazon river basin. Hydrological Processes, 23:1221–1235.

Caleffi, V., Valiani, A., and Zanni, A. (2003). Finite volume method for simulating extreme flood events in natural channels. Journal of Hydraulic Research, 41(2):167–177.

Chau, K. W. and Lee, J. H. (1991). Mathematical modelling of shing mun river network. Advances in Water Resources, 14(3):106–112.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., et al. (2001). Introduction to algorithms, volume 2. MIT press Cambridge.

Cunge, J. (1974). Evaluation problem of storm water routing mathematical models. Water Research, 8(12):1083 – 1087.

Cunge, J. A., Holly, F. M., and Verwey, A. (1980). Practical aspects of computational river hydraulics. Pitman Publishing.

David, C. H., Maidment, D. R., Niu, G.-Y., Yang, Z.-L., Habets, F., and Eijkhout, V. (2011). River network routing on the NHDPlus dataset. Journal of Hydrometeorology, 12:913–934.

de Saint-Venant, A. B. (1871). Th\\'{e} orie du mouvement non permanent des eaux, avec application aux crues des rivi\{e} res et\{a} l\'introduction des mar\\'{e} es dans leurs lits. Comptes Rendus des séances de l\'Académie des Sciences, 73:237–240.

Hodges, B. R. (2013). Challenges in continental river dynamics. Environmental Modelling & Software, 50:16–20.

Hodges, B. R. and Liu, F. (2014). Rivers and electrical networks: Crossing disciplines in modeling and simulation. Foundations and Trends in Electronic Design Automation, 8(1):1–116.

Kazolea, M. and Delis, A. (2013). A well-balanced shock-capturing hybrid finite volume–finite difference numerical scheme for extended 1d boussinesq models. Applied Numerical Mathematics, 67:167–186.

Kvočka, D., Falconer, R. A., and Bray, M. (2015). Appropriate model use for predicting elevations and inundation extent for extreme flood events. Natural Hazards, 79(3):1791–1808.

Lax, P. D. (2006). Gibbs phenomena. Journal of Scientific Computing, 28(2-3):445–449.

Liang, D., Falconer, R. A., and Lin, B. (2006). Comparison between tvd-maccormack and adi-type solvers of the shallow water equations. Advances in water resources, 29(12):1833–1845.

Liu, F. (2014). SPRNT: a river dynamics simulator. `https://github.com/frank-y-liu/SPRNT`.

Liu, F. and Hodges, B. R. (2014). Applying microprocessor analysis methods to river network modeling. Environmental Modelling & Software, 52:234–252.

Nobre, A., Cuartas, L., Hodnett, M., Rennó, C., Rodrigues, G., Silveira, A., Waterloo, M., and Saleska, S. (2011). Height above the nearest drainage – a hydrologically relevant new terrain model. Journal of Hydrology, 404(1–2):13 – 29.

Paiva, R. C. D., Collischonn, W., Bonnet, M. P., and de Goncalves, L. G. G. (2012). On the sources of hydrological prediction uncertainty in the amazon. Hydrology and Earth System Sciences, 16(9):3127–3137.

Ponce, V. M., Simons, D. B., and Li, R.-M. (1978). Applicability of kinematic and diffusion models. Journal of the Hydraulics Division, 104(3):353–360.

Pramanik, N., Panda, R. K., and Sen, D. (2010). One dimensional hydrodynamic modeling of river flow using dem extracted river cross-sections. Water Resources Management, 24(5):835–852.

Preissmann, A. (1961). Propagation des intumescences dans les canaux et rivieres. In Premier Congres De L'Association Francaise de Calcul AFCAL, Grenoble 14 - 15 - 16, 1960, pages 433–442. Gauthier-Villars, Paris.

Preissmann, A. and Cunge, J. (1961). Calcul du mascaret sur machine électronique. La Houille Blanche, 16(5):588–596.

Rahman, M. M. and Lu, M. (2015). Model spin-up behavior for wet and dry basins: A case study using the xinanjiang model. Water, 7(8):4256–4273.

Rhodes, D. D. (1977). The b-f-m diagram; graphical representation and interpretation of at-a-station hydraulic geometry. American Journal of Science, 277(1):73–96.

Sanders, B. F. (2001). High-resolution and non-oscillatory solution of the St. Venant equations in non-rectangular and non-prismatic channels. Journal of Hydraulic Research, 39(3):321–330.

Santibanez, A. R. H. (2015). Evaluating river cross section geometry for a hydraulic river routing model : Guadalupe and San Antonio river basins. Master's thesis, The University of Texas at Austin, the United State.

Seck, A., Welty, C., and Maxwell, R. M. (2015). Spin-up behavior and effects of initial conditions for an integrated hydrologic model. Water Resources Research, 51(4):2188–2210.

Western, A. W., Finlayson, B. L., McMahon, T. A., and O'Neill, I. C. (1997). A method for characterising longitudinal irregularity in river channels. Geomorphology, 21(1):39 – 51.

Yang, L., Hals, J., and Moan, T. (2012). Comparative study of bond graph models for hydraulic transmission lines with transient flow dynamics. Journal of Dynamic Systems, Measurement, and Control, 134(3):031005.

Zhao, D., Shen, H., Lai, J., and III, G. T. (1996). Approximate riemann solvers in fvm for 2d hydraulic shock wave modeling. Journal of Hydraulic Engineering, 122(12):692–702.

Zheng, X. (2016). Texas river hydraulic properties. HydroShare, `http://www.hydroshare.org/resource/40d4dfa1afb04cf9a64831c3419e7443`.

Algorithm 4 Bi-Section Method

procedure BISECTION(Q, n,S_0,N_{max}, ϵ) ▷ ϵ: tolerance
  $A_u \leftarrow 0.01$ ▷ Search for upper bound
  repeat
    $A_u \leftarrow 2A_u$
    Evaluate $r(A)$ in eq. (13)
  until $r > 0$
  $A_l \leftarrow 0$
  for $i = 1$ to N_{max} **do** ▷ Bisection method
    $r_l \leftarrow r(A_l)$
    $r_u \leftarrow r(A_u)$
    $A_m \leftarrow (A_u + A_l)/2$
    if $|r_u - r_l| < \epsilon$ **then return** A_m
    else
      $r_m \leftarrow r(A_m)$
      if $r_m > 0$ **then**
        $A_u \leftarrow A_m$
      else
        $A_l \leftarrow A_m$
      end if
    end if
  end for
  return A_m
end procedure