**(1) What kinds of hydrodynamic models require very good initial conditions to avoid numerical issues? Do simplified models as finite volumes using inertial approximation (Bates et al. 2010) as Lisflood-FP(Neal et al., 2012), MGB-IPH (Pontes et al., 2017), Camaflood (Yamazaki et al., 2013) require all this effort for very precise and consistent initial conditions? Please discuss this issue.**

In general, stability issues are associated with SVE models and are not necessarily a problem for reduced physics models. We have clarified this with new text on lines 40-43 which state: "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."

**(2) *Why good initial states are necessary? What is more important? Having no numerical issues by using a consistent set of initial states? Or having good initial states to get correct unsteady simulations? Please discuss and clarify it.**

We have addressed this issue in two places. First, on lines 32-40 in Section 1.1 we introduce the idea that time-marching model results should only be considered after spin-up, when the effects of the initial conditions have been washed out. Second, we have substantially re-written Section 1.2 (lines 63 – 111) to discuss the difference between using observed vs. synthetic initial conditions and the concept of consistency. This is based on our interpretation of the reviewer's comment as asking us to discuss the difference between "good" initial states that are closer to observed values versus synthetic initial states that are smoother but farther from observations. We disagree with the idea that having "good initial states" leads to "correct unsteady simulations." Using the definition of spin-up time, the initial conditions *cannot* affect the model results and thus cannot make an unsteady model either more or less correct. Both "good" and "bad" initial conditions that are convergent will result in exactly the same unsteady behavior after spin-up (else spin-up has not been reached). The only way to decide if an initial state is "good" is the computational cost of obtaining the initial condition and the computational cost of spin-up. As a caveat, we recognize that there are nonlinear systems with extremely long memory where the initial conditions can cause bifurcations of the system state. However, this is not generally the case for river networks driven by a runoff model.

**(3) 115-121. Add references.**

Reference have been added.

**(4) Why simply solving backwater equations is not a good option for cold start???? Please discuss it.**

The proposed SSM method is essentially the same as a backwater (gradually varying flow) computation for spatially-varying geometry and inflows. A discussion of the correspondence between the methods has been added in Lines 238-241

**(5) Can you also have numerical issues on PTM depending on how you set the Q0 and A0? If cold start is to bad, I guess that you can have numerical issues on PTM. Please discuss it.**

Yes, choices of the starting conditions with PTM can affect the length of spin-up time and cause numerical problems. We have added this point in the presentation of the method (Lines 210 – 215) and pointed out its occurrence in Results (Lines 404 – 411). A discussion of how this affects repeatability is provided in the Discussion (Lines 441-444).

**(6) One alternative to PTM is to start the model with flat water levels (h*), as a reservoir, and them run the model with downstream boundary conditions as water levels decreasing slowly from h* to h0. Please discuss this option. Please discuss it.**

The reviewer's proposed method can be considered a form of cold start with a pseudo-time boundary condition adjustment. We do not think it appropriate for us to include a method that has been thought up by a reviewer. However, we have expanded the discussion of cold start methods in general (Lines 127-139) to note that there are a large number of possible approaches to the cold start method, which all suffer from inconsistencies between the flows/depths and the hydrological inflows. As an aside, the reviewer's approach with a flat water level is likely impractical for any sufficiently large network where the bottom of an upstream reach is higher than the highest practical water level at the downstream outlet; e.g. to provide a flat water surface in the San Antonio and Guadalupe basin from the 1st order stream reaches would require an artificial depth of about 200 m at the outlet. On the other hand, if we interpret the reviewer's comment as implying a series of stepped flat water levels through the network, then the intersection points become infinite slopes (shocks) that destabilize the solution.

**(7) What are the difficulties of SSM? Does it work for looped river networks? What about situations with hydraulic structures, dams, or hydraulic controls? Please discuss it.**

We have added Section 5.3, Lines 496-511 to discuss the limitations of the method. The DFS approach for simple junctions can be extended to any connectivity of river system, but requires additional information about Q partitioning in multiple downstream reaches. Reservoirs should be considered as places to subdivide the method (assuming reservoir operating behaviors are known). Hydraulic structures should be handled as in any unsteady SVE model – the major difference is the Q must pass through for steady state.

**(8) How convergence time of the unsteady solver compares to SSM time to solve steady flow equations? Please discuss it.**

We could interpret this question in two ways. First, as a request for direct comparison of the unsteady solver as used in the PTM method to obtain the same steady-flow solution as the SSM method. The convergence time comparisons of PTM to SSM are provided in Tables 3 and 4. Solver iterations required for convergence are compared in Table 2. All the results show the unsteady solver does poorly in obtaining a steady solution relative to the SSM. A second interpretation could be that the reviewer is interested in the relative performance of the unsteady solver on unsteady equations to the steady solver on the steady equations – i.e. does the PTM perform poorly because the unsteady solver is inefficient? To address this issue, we reorganized the Discussion section so that the Model Performance subsection comes after the Spin-up subsection. We then discuss (Lines 488-494) the relationship between the unsteady model performance during spin-up from SSM and the CPU time to compute the SSM. The SSM requires 3.8 seconds of CPU time to compute steady conditions, and the SPRNT model requires about 5 minutes of CPU time to compute 150 hours of real-world time for the San Antonio and Guadalupe river system. That is, the SPRNT unsteady model was running about 1800x faster than real time and the SSM computation is essentially irrelevant to the overall time-marching simulation. Another way to think of this is that in the time it took to compute the SSM, the unsteady solver computed 114 minutes of real-world simulation. Thus, the unsteady solver is extremely fast, except when it is given a set of inconsistent initial conditions and is iterating to remove the inconsistency (as in the PTM).