Ancient numerical daemons of conceptual hydrological modeling:

1. Fidelity and efficiency of time stepping schemes

Martyn P. Clark¹ and Dmitri Kavetski²

Received 12 November 2009; revised 22 March 2010; accepted 16 April 2010; published 8 October 2010.

[1] A major neglected weakness of many current hydrological models is the numerical method used to solve the governing model equations. This paper thoroughly evaluates several classes of time stepping schemes in terms of numerical reliability and computational efficiency in the context of conceptual hydrological modeling. Numerical experiments are carried out using 8 distinct time stepping algorithms and 6 different conceptual rainfall-runoff models, applied in a densely gauged experimental catchment, as well as in 12 basins with diverse physical and hydroclimatic characteristics. Results show that, over vast regions of the parameter space, the numerical errors of fixed-step explicit schemes commonly used in hydrology routinely dwarf the structural errors of the model conceptualization. This substantially degrades model predictions, but also, disturbingly, generates fortuitously adequate performance for parameter sets where numerical errors compensate for model structural errors. Simply running fixed-step explicit schemes with shorter time steps provides a poor balance between accuracy and efficiency: in some cases daily-step adaptive explicit schemes with moderate error tolerances achieved comparable or higher accuracy than 15 min fixed-step explicit approximations but were nearly 10 times more efficient. From the range of simple time stepping schemes investigated in this work, the fixed-step implicit Euler method and the adaptive explicit Heun method emerge as good practical choices for the majority of simulation scenarios. In combination with the companion paper, where impacts on model analysis, interpretation, and prediction are assessed, this two-part study vividly highlights the impact of numerical errors on critical performance aspects of conceptual hydrological models and provides practical guidelines for robust numerical implementation.

Citation: Clark, M. P., and D. Kavetski (2010), Ancient numerical daemons of conceptual hydrological modeling: 1. Fidelity and efficiency of time stepping schemes, *Water Resour. Res.*, 46, W10510, doi:10.1029/2009WR008894.

1. Introduction

[2] Conceptual hydrological models are popular in operational and research practice because in many applications they can provide adequate and relatively parsimonious descriptions of dominant catchment-scale processes at a low computational cost. Constructing such models generally consists of several steps [e.g., *Beven*, 2001].

[3] 1. Develop a qualitative “perceptual” model that broadly describes “how the system works” and which processes are dominant.

[4] 2. Develop a quantitative “conceptual” model that provides a mathematical description of the perceptual model. Additional simplifications may be hypothesized at this stage (e.g., effective homogeneity of soils, etc). We will refer to these equations as the “governing model equations,” and to

the approximations made at this stage as “model approximations,” which result in “model structural errors.”

[5] 3. Solve the governing model equations, using a numerical algorithm if an exact solution is impossible. This step is the primary focus of this study. Approximations made at this stage are “numerical approximations” that cause “numerical errors.”

[6] 4. Infer the model parameters from observed data.

[7] 5. Evaluate the model predictions and associated uncertainties and scrutinize the assumptions of both the conceptual model and the inference/prediction methodology.

[8] In practice, there may be feedbacks and overlaps between all the steps above. For example, (1) to obtain a mathematically tractable model, the conceptual model may need modification; (2) the numerical implementation may employ some simplifications (e.g., linearization) that could also be viewed as modifications of the governing equations; and (3) posterior evaluation may lead to adjustment of the governing equations.

[9] Paradoxically, although the numerical solution is recognized as a critical step in the model-building process [e.g., *Beven*, 2001]—after all, this is the procedure used to generate the actual model predictions—numerical time stepping schemes for conceptual hydrological models have received

¹National Center for Atmospheric Research, Boulder, Colorado, USA.

²Environmental Engineering, University of Newcastle, Callaghan, New South Wales, Australia.

surprisingly little attention in the literature. For example, most descriptions of conceptual hydrological models outline the governing equations but do not discuss their numerical solution [e.g., *Singh and Frevert*, 2002a, 2002b]. In some cases the complete set of governing equations is not even formulated directly, and the modeler creates a computational algorithm directly from their perceptual understanding. For example, the Sacramento model simply processes individual fluxes in some predetermined order [*Burnash*, 1995]. At least partially, this reflects a tacit assumption that numerical errors are always overwhelmed by uncertainties in the data and governing equations and that attention to numerical robustness is therefore unwarranted. Indeed, why bother with model numerics?

[10] Ideally, as modelers, we prefer to focus on the identification of the dominant hydrological processes and on their adequate representation in the governing equations. However, the belief that numerical errors are insignificant or can be disregarded is unfounded. In particular, *Kavetski et al.* [2003] demonstrated troublesome weaknesses in the time stepping schemes commonly used in many hydrological models, including the popular TOPMODEL [*Beven*, 1997], with large uncontrolled numerical errors occurring over large regions of the parameter space and swamping model conceptualization errors. Subsequent work indicated that many problems widely believed to be endemic to environmental models, including geometrically complex objective functions with morasses of microscale and macroscale local optima [e.g., *Duan et al.*, 1992], are frequently merely artifacts of the numerical implementation [*Kavetski et al.*, 2006; *Kavetski and Kuczera*, 2007].

[11] Despite adaptive time stepping and numerical error control being widely accepted as essential for scientific and engineering computing [e.g., *Kahaner et al.*, 1989; *Shampine*, 1994], poor selection of numerical schemes remains quite widespread in hydrology, even in distributed models. For example, the TopNet model does not solve its state equations directly but instead sequentially adds/subtracts fluxes from model states [*Bandaragoda et al.*, 2004; *Clark et al.*, 2008a]. While gross numerical errors are avoided by using exact solutions for individual fluxes wherever possible, this approach is limited to analytical solutions (quite restrictive given the nonlinearity of catchment dynamics) and leads to complicated code with multiple switches and branches (causing discontinuities in the model predictions). In another recent example, *Blöschl et al.* [2008] implemented a distributed flood forecasting model using a fixed 15 min-step explicit Euler algorithm. While the relatively short step size may constrain numerical errors under most scenarios, the lack of numerical quality control necessarily undermines the confidence in the model predictions in general applications, especially in large-scale studies, or when applying the model in new contexts. Moreover, fixing the step size a priori may lead to inefficiencies when a larger step size is sufficient.

[12] There is a growing, though still insufficient, recognition that numerical robustness is necessary for meaningful environmental modeling. For example, the Sacramento model [*Burnash*, 1995] uses internal substepping (albeit without error control). Other hydrological applications of adaptive time stepping include (1) water quality modeling [e.g., *Whitehead et al.*, 1998]; (2) integrated hydrological models including mass, energy, and momentum balances

[*Reggiani and Rientjes*, 2005]; (3) modeling mean residence time in experimental catchments [*Vache and McDonnell*, 2006]; (4) exploration of regionalization relationships [*Kling and Gupta*, 2009]; and (5) groundwater models, in particular, using Richards equation [e.g., *Miller et al.*, 1998; *Kavetski et al.*, 2002a].

[13] The objective of this two-part study is a thorough evaluation of numerical time stepping schemes in the context of calibration and prediction using conceptual hydrological models. It significantly develops previous work [*Kavetski et al.*, 2003, 2006] by considering more realistic multi-component hydrological models and several classes of time stepping schemes, including explicit, semi-implicit and implicit methods, and first- and second-order methods, both in fixed-step and adaptive implementation.

[14] In this paper we review the time stepping schemes and provide insights into the effect of numerical approximation errors on the model predictions. In addition to comparing the computational efficiency of “standard” implementations, we also evaluate several strategies for boosting the efficiency of implicit time stepping that have not, to our knowledge, been previously examined in the context of hydrological simulations.

[15] To ensure the generality of our analysis and conclusions, we tested all methods in a densely gauged catchment, as well as in 12 other catchments spanning a wide range of physical and hydroclimatic characteristics. The inclusion of a densely gauged catchment controls the influence of data errors and yields, for the first time in hydrological modeling, useful direct insights into the relative magnitudes of model structural errors versus numerical approximation errors (quantified using the concept of “fidelity” of a numerical approximation). We also consider the tradeoffs between forcing data resolution and computational reliability and efficiency.

[16] In the companion paper [*Kavetski and Clark*, 2010], we investigate the impact of the numerical implementation of the hydrological model on sensitivity analysis, parameter optimization using Newton-type and evolutionary methods and, finally, uncertainty assessment using Markov chain Monte Carlo methods. These are all critical stages of model construction, application, and interpretation.

[17] Given our focus on hydrological applications, which remain dominated by simple numerical techniques, the time stepping schemes considered in this study are relatively simple and well established. As well as being easy to implement or retrofit to existing codes, the use of simple algorithms has important didactic benefits for a general hydrological audience. The analysis of more advanced ordinary differential equation (ODE) algorithms available through numerical software libraries [e.g., *Shampine and Reichelt*, 1997] is an important next step, deferred to a separate study.

[18] This paper is organized as follows. Section 2 outlines the state-space formulation of conceptual hydrological models, whereas section 3 reviews the time stepping schemes. Sections 4, 5, and 6 describe, respectively, the models, data, and methodology used in the numerical experiments. Sections 7, 8, and 9 report, respectively, the numerical fidelity, computational cost, and step size dependence of the time stepping schemes, whereas section 10 provides a broad assessment over 12 basins with diverse hydroclimatic regimes. The paper

concludes with section 11, which summarizes the major findings and recommendations in the contexts of numerical reliability and computational efficiency. The impact of numerical errors on model application and interpretation is then explored in the companion paper.

2. Formulation of Governing Model Equations

2.1. Ad Hoc Implementations and Operator-Splitting Schemes

[19] Many hydrological models use ad hoc approaches that add or subtract individual fluxes from the model stores in some predetermined sequence. For example, the popular Sacramento model [Burnash, 1995] is implemented as follows: (1) compute evaporation and subtract it from tension storages, (2) add precipitation to the upper zone tension storage, (3) compute base flow and subtract it from the storage, (4) add excess precipitation (from step 2) to the upper zone free storage, (5) compute drainage and move it to the lower zone storages, and (6) compute surface runoff and update water storage in the additional impervious area. Similar “sequential flux” approaches are used in TOPMODEL [Beven, 1997], Variable Infiltration Capacity (VIC) model [Wood *et al.*, 1992], Precipitation Runoff Modeling System (PRMS) [Leavesley *et al.*, 1983], and other widely used models.

[20] Although seldom recognized, this sequential flux treatment constitutes an “operator-splitting” (OS) approach, a numerical method popular for partial differential equations (PDEs) describing multiple spatially distributed processes with widely different time scales and behavior (e.g., see Geiser [2010] for general theory, Steefel and MacQuarrie [1996] for common geochemical applications, and Kavetski *et al.* [2003] and Schoups *et al.* [2010] for a discussion in rainfall-runoff modeling contexts). At the expense of additional “splitting” error, OS strategies allow the application of specialized numerical approximations to each individual process.

[21] However, the governing equations in conceptual hydrological models are typically much simpler ordinary differential equations (ODEs), usually with 5–10 states at most. Moreover, in hydrological models that use the OS approach, the governing equations are seldom, if ever, formulated directly. Consequently, the model conceptualization is conflated with the numerical implementation. This is a major methodological weakness because modeling errors arising from the physical conceptualization and numerical implementation become difficult to separate, diagnose, and resolve. Furthermore, it becomes difficult to design flexible and modular hydrological model toolkits such as Framework for Understanding Structural Errors (FUSE) [Clark *et al.*, 2008b] and FLEX [Fenicia *et al.*, 2008]. Finally, unless numerical error control is implemented, model predictions computed using the OS approach may depend on the order of processing of the model fluxes, which creates an undesirable arbitrariness in the model construction and behavior.

[22] Given these arguments, this paper does not consider the OS approach and instead focuses on more standard and well-established numerical integration approaches, in particular, explicit, semi-implicit and implicit Runge-Kutta schemes, both with and without adaptive error control. This requires formulating the governing equations in state-space form.

2.2. State-Space Formulation of Hydrological Models

[23] In the majority of conceptual hydrological models, the state equations describe time changes in storages. For example, storage in the unsaturated zone can be described as

$$\frac{dS}{dt} = g(S, t) = f_i(S, t) - f_d(S, t) - f_e(S, t), \quad (1)$$

where $S(t)$ is the storage at time t , and f_i , f_d , and f_e are infiltration, drainage, and evaporation fluxes respectively (which may also depend on rainfall intensity, temperature, and other time-varying forcing).

[24] Hydrological models generally comprise several conceptual stores connected in series and/or in parallel. This corresponds to a system of N_s coupled ODEs,

$$\frac{d\mathbf{S}}{dt} = \mathbf{g}(\mathbf{S}, t) \quad (2)$$

where $\mathbf{S}$ is a state vector and N_s is the number of stores. The flux vector $\mathbf{g}$ is assembled based on the hypothesized connectivity of the reservoirs (e.g., see Figure 1).

[25] In most hydrological applications, the model is applied over a series of discrete time steps determined by the observed data resolution (typically, hourly or daily). In the absence of sufficient information for subscale temporal disaggregation of forcings such as rainfall and potential evapotranspiration, the forcing is usually assumed constant over these “outer” (data-resolution) time steps. This makes ODEs such as (2) autonomous, i.e., independent of t , within each outer step. With little loss of generality, the remainder of the paper hence uses $\mathbf{g}(\mathbf{S})$ rather than $\mathbf{g}(\mathbf{S}, t)$.

[26] Formulating hydrological models in state-space form is beneficial in several respects.

[27] 1. It clearly delineates the model assumptions from the numerical approximations, which is more satisfying scientifically and facilitates the improvement of process representation.

[28] 2. The vast literature on differential equations, including numerical solutions [e.g., Ascher and Petzold, 1998], as well as specialized techniques for parameter estimation [e.g., Cao *et al.*, 2002], can be exploited to control numerical errors, reduce runtimes, etc.

[29] 3. It is useful in the context of data assimilation, in particular, simplifying the application of particle and ensemble Kalman filters [e.g., Weerts and El Serafy, 2006].

[30] 4. It simplifies the design of flexible modular software for hydrological simulations. For example, Clark *et al.* [2008b] implemented elements of TOPMODEL, VIC, PRMS, and Sacramento equations within a unified framework to evaluate individual model components, e.g., the selection of equations representing surface runoff, subsurface flow, etc. Similarly, Fenicia *et al.* [2008] demonstrated the benefits of a flexible model structure in simulating both common and unusual hydroclimatic regimes.

3. Basic Computational Methods for State-Space Formulations

[31] This section outlines simple numerical methods for the state-space model (2). Readers with computational background may wish to skip this section.

Figure 1. Schematics of the FUSE models used in this study (see Appendix C for definitions of state variables, fluxes and parameters, and Table 2 for detailed equations).

3.1. Exact Solutions

[32] The exact solution of ODE system (2) over a step from t^n to $t^{n+1} = t^n + \Delta t$ can be written as

$$\bar{\mathbf{g}}^{n \rightarrow n+1} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{g}(\mathbf{S}(\varsigma), \varsigma) d\varsigma \quad (3)$$

$$\mathbf{S}(t^{n+1}) = \mathbf{S}(t^n) + \Delta t \bar{\mathbf{g}}^{n \rightarrow n+1}, \quad (4)$$

where $\bar{\mathbf{g}}^{n \rightarrow n+1}$ are the average solution derivatives over the time step and ς is a dummy integration variable. When $\mathbf{S}$ represents storages, $\bar{\mathbf{g}}$ corresponds to average fluxes.

[33] Since exact solutions of the implicit integral relation (3)–(4) are analytically intractable except for linear or very simple nonlinear reservoirs, numerical methods based on approximating the flux integral $\bar{\mathbf{g}}^{n \rightarrow n+1}$ are used. It is well known that this approximation controls both quantitative and qualitative properties of the time stepping scheme, including stability, accuracy, smoothness, and efficiency [e.g., Kahaner et al., 1989; Ascher and Petzold, 1998].

[34] The mass balance equations in models with no internal feedbacks (i.e., with no direct or indirect fluxes from downstream to upstream reservoirs) can sometimes be solved sequentially, starting at an upstream reservoir and solving one-store-at-a-time toward the downstream end. This can reduce computational cost (it replaces solving an ODE system by a sequence of solutions of scalar ODEs), but it

introduces additional approximation errors and makes the programming more intricate. Importantly, one-store-at-a-time strategies can be very inaccurate in the presence of internal feedbacks, limiting the flexibility of model configurations. Investigation of this approach is therefore deferred to a separate study.

3.2. Brief Overview of Numerical Time Stepping Schemes

[35] The multitude of numerical ODE integration methods can be broadly classified into several categories [e.g., Hairer and Wanner, 1991; Hairer et al., 1993]. Among the most common practical methods are explicit Runge-Kutta schemes, which are self-starting and easiest to implement with adaptive error control. Common alternatives include linear multistep (and related multistage) methods, such as the (explicit) Adams-Mouton and (implicit) stiff Gear methods. These methods are often more efficient, especially for smooth problems but are more complex and require initialization. Other approaches include (semi-implicit) Rosenbrock schemes, Bulirsch-Stoer methods, etc. [e.g., Press et al., 1992].

[36] Key issues in the selection of a numerical method include its stability, accuracy, and efficiency, as well as suitability and ease of application to the ODE of interest (which may have solution constraints or other special features) [e.g., Shampine, 2007]. To maintain a clear exposition in the context of conceptual hydrological models, here we

limit our attention to low order Runge-Kutta methods with simple embedded error control, which can be more readily retrofitted to existing hydrological model codes. The application of more sophisticated “state-of-the-art” ODE integration algorithms is deferred to a separate study.

3.3. Explicit Runge-Kutta Schemes

[37] The explicit (forward) Euler scheme approximates the average flux by the flux at the start of the step,

$$\bar{\mathbf{g}}_{\text{EE}(1)}^{n \rightarrow n+1} = \mathbf{g}(\mathbf{S}^n) \approx \bar{\mathbf{g}}^{n \rightarrow n+1} + O(\Delta t) \quad (5)$$

and substitutes it into the solution update equation (4).

[38] The notation $O(\Delta t)$ and the subscript EE(1) indicate that the explicit Euler method is first order accurate, i.e., errors in $\mathbf{S}_{\text{EE}(1)}^{n+1}$ arising from approximation (5) are proportional to Δt .

[39] Second-order accuracy can be obtained using a trapezoidal rule for the integral (3). For example, the explicit Heun scheme, which requires an additional function evaluation, is

$$\bar{\mathbf{g}}_{\text{EH}(2)}^{n \rightarrow n+1} = \frac{1}{2} [\mathbf{g}(\mathbf{S}^n) + \mathbf{g}(\mathbf{S}_{\text{EE}(1)}^{n+1})] \approx \bar{\mathbf{g}}^{n \rightarrow n+1} + O(\Delta t^2). \quad (6)$$

While higher order translates into higher accuracy for asymptotically small Δt , low-order methods are usually better behaved when Δt is large [e.g., Wood, 1990]. Approximations (5) and (6) are special cases of the general Runge-Kutta family [e.g., Butcher, 2008], where multiple schemes can be obtained for any specified order of accuracy.

3.4. Implicit Runge-Kutta Schemes

[40] The implicit (backward) Euler scheme uses solely the flux at the end of the step,

$$\bar{\mathbf{g}}_{\text{IE}(1)}^{n \rightarrow n+1} = \mathbf{g}(\mathbf{S}_{\text{IE}(1)}^{n+1}) \approx \bar{\mathbf{g}}^{n \rightarrow n+1} + O(\Delta t). \quad (7)$$

Its truncation error is $O(\Delta t)$; both EE and IE approximate $d\mathbf{S}/dt$ by first-order differences.

[41] An implicit analogue of the Heun scheme (6) is

$$\bar{\mathbf{g}}_{\text{IH}(2)}^{n \rightarrow n+1} = \frac{1}{2} [\mathbf{g}(\mathbf{S}^n) + \mathbf{g}(\mathbf{S}_{\text{IH}(2)}^{n+1})] \approx \bar{\mathbf{g}}^{n \rightarrow n+1} + O(\Delta t^2). \quad (8)$$

In the PDE and engineering literature, it is frequently referred to as the Crank-Nicholson method or the “implicit trapezoidal rule” [e.g., Wood, 1990].

[42] While explicit schemes simply require evaluating the fluxes for given solution values, implicit approximations require the computed flux to be consistent with the estimated solution $\mathbf{S}_{\text{XX}}^{n+1}$, which in general requires iteration, and can hence be expensive (section 3.5).

[43] However, the key advantage of most implicit schemes is unconditional stability: while not necessarily the best methods for general ODEs, implicit methods can be used with much larger time steps than explicit schemes when the ODEs are “stiff,” requiring fewer steps and ultimately lowering the overall computational cost (see Appendix A).

[44] Given its robustness, the implicit Euler scheme is ubiquitous in “industry-standard” engineering software, including the MODFLOW package widely used in ground-water and contaminant transport modeling [Harbaugh,

2005], the ECLIPSE tool used in the petroleum industry for multiphase flow simulations [Schlumberger Information Systems, 2004], geotechnical codes, and many others. It can be readily supplemented with error control [e.g., Kavetski et al., 2002b]; moreover, very efficient noniterative implementations have been reported in geohydrological applications based on Richards equation [e.g., Kavetski et al., 2002a and references therein].

3.5. Newton-Raphson Solution of Implicit Schemes

[45] Implicit approximations result in systems of nonlinear equations, which are typically solved using Newton-Raphson iteration. For an implicit scheme XX, this requires the linear solution

$$\mathbf{J}_{\text{XX}}^{(m)} \Delta \zeta^{(m+1)} = -\mathbf{r}_{\text{XX}}^{(m)} \quad (9)$$

$$\zeta^{(m+1)} = \zeta^{(m)} + \Delta \zeta^{(m+1)}, \quad (10)$$

where the superscript m indexes the iterations (not to be confused with the time steps), $\mathbf{r}_{\text{XX}}$ is the error (mass imbalance over the time step), and $\mathbf{J}_{\text{XX}}$ is its Jacobian matrix,

$$J_{\text{XX}[i,j]}^{(m)} \equiv \left. \frac{\partial r_{\text{XX}[i]}}{\partial \zeta_j} \right|_{\zeta=\zeta^{(m)}}; i = 1..N_s, j = 1..N_s. \quad (11)$$

For the implicit Euler scheme, $\zeta = \mathbf{S}_{\text{IE}(1)}^{n+1}$ and $\mathbf{r}_{\text{IE}(1)}(\zeta) = \zeta - [\mathbf{S}^n + \Delta t \mathbf{g}(\zeta)]$, whereas for the implicit Heun scheme $\zeta = \mathbf{S}_{\text{IH}(2)}^{n+1}$ and $\mathbf{r}_{\text{IH}(2)}(\zeta) = \zeta - [\mathbf{S}^n + \frac{\Delta t}{2} \mathbf{g}(\mathbf{S}^n) + \frac{\Delta t}{2} \mathbf{g}(\zeta)]$.

[46] The Jacobian $\mathbf{J}_{\text{XX}}$ of a discrete approximation can be efficiently computed from the ODE Jacobian $\mathbf{J} = \partial \mathbf{g} / \partial \mathbf{S}$, e.g., $\mathbf{J}_{\text{IE}(1)} = \mathbf{I} - \Delta t \partial \mathbf{g} / \partial \mathbf{S}$ and $\mathbf{J}_{\text{IH}(2)} = \mathbf{I} - \frac{\Delta t}{2} \partial \mathbf{g} / \partial \mathbf{S}$.

[47] In this study $\partial \mathbf{g} / \partial \mathbf{S}$ is approximated using one-sided finite differences, and the linear system (9) is solved using LU decomposition. The iterations (9)–(10) proceed until the max-norm of $\Delta \zeta^{(m)}$ and/or $\mathbf{r}(\zeta^{(m)})$ satisfy a convergence tolerance τ_M (mass balance error tolerance).

[48] To avoid ambiguity, hereafter we use the term “Newton-Raphson solver” to describe the solver (9)–(10) for implicit time stepping methods and “Newton optimizer” to describe the Newton-type optimizer for model parameter calibration (companion paper). Both are based on essentially the same principles [e.g., Nocedal and Wright, 1999].

3.6. Improving the Efficiency of Implicit Schemes

[49] The substantial additional cost-per-step of implicit schemes arises from two distinct sources: (1) multiple evaluations of the Jacobian matrix (11) and (2) multiple solutions of the linear system (9) within each time step. These costs can be reduced using several strategies, as outlined below.

3.6.1. Line Searches

[50] The Newton-Raphson solver can be enhanced with line searches, e.g., backtracking along $\Delta \zeta^{(m+1)}$ at each iteration until the mass balance error $\mathbf{r}$ satisfies a sufficient decrease condition [e.g., Nocedal and Wright, 1999]. Though line searches require additional evaluations of $\mathbf{r}$ at each iteration, they can reduce the total cost by reducing the number of iterations needed for convergence at each time step, and hence the total number of evaluations/factorizations of $\mathbf{J}$. Importantly, they increase numerical robustness by avoiding potential divergence.

3.6.2. Jacobian Refreshment Strategies

[51] Modern ODE integrators keep their Jacobian factorization constant for as long as possible, either over multiple iterations at a single time step or, more aggressively, over many consecutive time steps, refreshing it only when convergence of equations (9)–(10) becomes too slow. While such strategies may require extra iterations because of inaccurate Jacobians, they can dramatically boost the overall efficiency by avoiding repeated evaluation/factorization of $\mathbf{J}$.

[52] Note that, in many hydrological model setups, the Jacobian is independent of the forcing and may actually remain constant over many consecutive time steps. For example, the model $dS/dt = P[1 - A(S)] - kS$ where P is precipitation and k is a parameter, has a constant ODE Jacobian $dg/dS = -k$ for all steps where $P = 0$. Such properties can and should be exploited.

[53] In this study we consider three distinct Jacobian refreshment strategies [e.g., *Smooke*, 1983]: For strategy J-A, refresh $\mathbf{J}$ at each iteration (classic Newton-Raphson algorithm). For strategy J-B, refresh $\mathbf{J}$ if the mass balance errors are not decreasing sufficiently fast,

$$\frac{\|\mathbf{r}^{(m)}\|}{\|\mathbf{r}^{(m-1)}\|} > \tau_{\text{dfunc}}, m = 2, 3, 4, \dots \quad (12)$$

Setting $\tau_{\text{dfunc}} = 0$ yields strategy J-A, whereas setting $\tau_{\text{dfunc}} = 1$ forces a constant Jacobian for all iterations. For strategy J-C, refresh $\mathbf{J}$ at each iteration until the estimates get sufficiently close to the solution (where the Jacobian is likely to remain near-constant)

$$\|\mathbf{r}^{(m)}\| > \tau_{\text{freeze}}, m = 1, 2, 3, 4, \dots \quad (13)$$

The strategies J-B and J-C can reuse Jacobians from previous substeps (or even previous time steps): J-B uses the Jacobian from the previous substep for the first iteration, whereas J-C only refreshes the previous substep Jacobian if condition (13) is met. In our code, we used the max-norm of the errors, $\|\mathbf{r}\|_{\infty}$, in conditions (12)–(13).

3.7. Semi-implicit Schemes

[54] An even more aggressive strategy is to exploit the first-order Taylor linearization

$$\mathbf{g}_L(\mathbf{S}) = \mathbf{g}(\mathbf{S}^n) + \frac{\partial \mathbf{g}(\mathbf{S}^n)}{\partial \mathbf{S}}(\mathbf{S} - \mathbf{S}^n) + O((\mathbf{S} - \mathbf{S}^n)^2) \quad (14)$$

and apply a single Newton-Raphson correction to the solution at the previous time step.

[55] When applied to the implicit Euler scheme, $\mathbf{S} = \mathbf{S}^{n+1}$ is substituted into (14), yielding the (first-order accurate) semi-implicit Euler scheme [Press et al., 1992]

$$\mathbf{S}_{\text{SIE}(1)}^{n+1} = \mathbf{S}^n + \left[\mathbf{I} - \Delta t \frac{\partial \mathbf{g}(\mathbf{S}^n)}{\partial \mathbf{S}} \right]^{-1} \Delta t \mathbf{g}(\mathbf{S}^n). \quad (15)$$

The individual fluxes can be back-calculated using linearization (14). For example, if the net flux $g_i(\mathbf{S})$ for store i comprises N_k individual fluxes $g_i^{(k)}(\mathbf{S})$, i.e., $g_i(\mathbf{S}) = \sum_{k=1}^{N_k} g_i^{(k)}(\mathbf{S})$, then

$$g_i^{(k)}(\mathbf{S}_{\text{SIE}(1)}^{n+1}) \approx g_i^{(k)}(\mathbf{S}^n) + \frac{\partial g_i^{(k)}(\mathbf{S}^n)}{\partial \mathbf{S}}(\mathbf{S}_{\text{SIE}(1)}^{n+1} - \mathbf{S}^n), \quad (16)$$

and the flow volume over the step is $\Delta t g_i^{(k)}(\mathbf{S}_{\text{SIE}(1)}^{n+1})$.

[56] Semi-implicit schemes can be highly efficient for many stiff problems because the use of the Jacobian tends to suppress instabilities (e.g., for linear ODEs, equation (15) simplifies to the implicit Euler scheme given by (5) and is unconditionally stable), whereas the lack of iterations reduces the total cost by large factor.

3.8. Error Control and Adaptive Substepping

[57] It is broadly recognized that time stepping with fixed step size Δt can produce inaccurate (explicit or implicit) or unstable (explicit) solutions. Consequently, good ODE integrators [e.g., *Shampine and Reichelt*, 1997] adjust their step size to satisfy a user-prescribed error tolerance. More sophisticated algorithms also dynamically adjust the order of accuracy to maximize efficiency for the given ODE and error tolerance and may switch between explicit and implicit approximations. However, while error-controlled integration is gradually being adopted across environmental sciences [e.g., *Reggiani and Rientjes*, 2005], it remains rare in conceptual hydrological models, which are mostly implemented using fixed-step schemes with no error control [e.g., *Beven and Kirkby*, 1979; *Leavesley et al.*, 1983; *Wood et al.*, 1992; *Wigmosta et al.*, 1994; *Sivapalan et al.*, 1996; *Lindström et al.*, 1997; *Tague and Band*, 2004; *Blöschl et al.*, 2008].

[58] This paper considers adaptive error-controlled integration based on estimating the truncation error by comparing two different approximations and adjusting the step size to meet a user-prescribed error tolerance (see Appendix B for further background). These error-control approaches are readily applied to the time stepping methods of sections 3.3 and 3.4.

[59] Integration methods in canned ODE packages are based on analogous principles but allow higher order approximations and employ more sophisticated adaptation. For example, the Multiphysics package [COMSOL, 2005] provides a suite of variable-step/variable-order algorithms (e.g., see *Binning and Celia* [2008] for an application in geochemistry). Similarly, the IMSL® library provides variable-order/variable-step (explicit) Adams-Moulton schemes of order up to 12 and (implicit) stiff Gear methods of order up to 5 [Visual Numerics, 2007]. Yet for coarse error tolerances, low-order schemes are competitive and may outperform higher-order methods [e.g., *Wood*, 1990]; indeed, variable-order approaches generally invoke lower order methods as the tolerance is relaxed [Ascher and Petzold, 1998]. A canned ODE algorithm may also be more difficult to adapt to solution constraints (section 3.9), customize to specific problem requirements, or retrofit to existing hydrological model codes. Nevertheless, since high-order methods can be vastly more efficient than low-order approaches when stringent accuracy is required, they will be considered in a future study.

3.9. Solution Constraints

[60] In most hydrological models, the solution must satisfy several physically motivated constraints. For example, catchment storage cannot be negative and cannot exceed a maximum capacity. In most cases, these constraints are automatically satisfied by the exact solution of the governing equations, i.e., the constraint is a natural feature of the model rather than an externally imposed feature. For example, the exact solution of the ODE $dS/dt = P(1 - S/S_{\text{max}}) - kS$ always satisfies $0 < S < S_{\text{max}}$.

[61] However, since numerical schemes yield inexact solutions of the governing equations, they generally do not satisfy such constraints. For example, under most drainage scenarios, fluxes decrease over a time step, and applying the initial flux over the entire time step may result in “over-draining” the catchment, yielding negative storages. In general, even adaptive substepping enforcing an arbitrarily tight error tolerance τ cannot guarantee solution feasibility. For example, a solution estimate S_{XX}^{n+1} in the range $(-\tau, 0)$ violates a nonnegativity constraint but will pass the error test regardless of how tight τ is. While a small negative storage is sometimes tolerable, it will cause computer arithmetic errors for many common flux formulations, e.g., $q_b = S^\alpha$ with $\alpha < 1$.

[62] Ad hoc fixes such as zeroing negative fluxes works in simple cases but are not a satisfactory general solution. For example, in multistate models where the states are coupled via the fluxes, fixing the violation in one state can impact on the feasibility of another. In general, satisfying joint constraints between states, e.g., S_b and S_c , related via nonlinear fluxes, e.g., $B = (S_b - S_c)^\alpha$ where B is known to be always positive, is not straightforward.

[63] Finally, and importantly, ad hoc alterations can interfere with the efficiency of the error control algorithm [Shampine *et al.*, 2005]. Though seemingly mundane, reliable and efficient theoretically based handling of ODE solution constraints is not straightforward; with some notable exceptions such as the Matlab ODE toolkit, recent advances [e.g., Shampine *et al.*, 2005] are yet to be widely implemented. Because it can be difficult to retrofit high-order packaged ODE algorithms with constraint handling (due to code complexity or lack of open source access), we defer the evaluation of advanced algorithms to a later study.

[64] In this paper we test and fix state feasibility violations one state at a time, in a predetermined sequence, with the upper layer states processed before lower layer states and tension storage processed before free storage. The fluxes modified after adjusting the i th state are used to update the $(i + 1)$ th state, and so forth. This processing order, used because most states are connected in series, remains quite arbitrary. Consequently, while empirically it appears adequate for the purposes of this study, especially when adaptive error control is used, more systematic approaches [e.g., Shampine *et al.*, 2005] may be preferable in future work.

[65] Note that the exact solution of the implicit Euler equations satisfies the ODE solution constraints for any step size (it is likely the only simple time stepping scheme that guarantees this, at least “in principle”). However, intermediate iterates may violate solution constraints and, moreover, be slowly convergent for highly nonlinear ODEs. Therefore, an enhanced Newton-Raphson strategy is needed to detect and resolve constraint violations at intermediate iterations. In this study we truncate infeasible Newton-Raphson iterates to satisfy the bound constraints and proceed to the next iteration. An active set method analogous to constrained Newton optimization [Nocedal and Wright, 1999] could further improve efficiency.

4. FUSE Family of Hydrological Models

4.1. General Overview

[66] The numerical experiments in this paper use the Framework for Understanding Structural Errors (FUSE)

hydrological toolkit [Clark *et al.*, 2008b], which provides multiple options when building a hydrological model and selecting numerical approximations. For the specific case of a hydrological model, the fundamental model-building decisions are (1) the choice of state variables in the unsaturated and saturated zones and (2) the choice of flux equations describing surface runoff, interflow, vertical drainage from the unsaturated zone, base flow, and evaporation. The flexibility and modularity of FUSE, largely supported by formulating it in canonical state-space form (2), greatly facilitated the empirical evaluation of all time stepping schemes over a wide range of model configurations and constitutive flux relations.

4.2. Specific FUSE Members

[67] This paper uses 6 FUSE models broadly representative of the wide spectrum of conceptual hydrological models used in research and practice. We include configurations where the upper layer has one or two state variables (e.g., tension and free storage), cases with linear and nonlinear base flow functions, and cases where surface runoff is parameterized as a function of either upper or lower layer storages. Figure 1 depicts these models diagrammatically, whereas Table 1 summarizes their key properties. The state and flux equations are listed in Table 2; the model state variables and fluxes in and the model parameters are summarized in Appendix C. Finally, Appendix D details the formulation of bucket overflow fluxes and the channel time delay following the guidelines of Kavetski and Kuczera [2007] to ensure continuous constitutive relations.

[68] FUSE-070 is analyzed in greater detail for the following reasons: (1) with seven parameters, it is currently the most parsimonious FUSE configuration; (2) it has only 2 state variables, which simplifies the visualization of the model state space, the model response surface and the objective function; and (3) it can be easily coded by an interested reader to test and further explore the ideas of this paper.

5. Calibration Data

5.1. Mahurangi (Mahurangi River Variability Experiment) Data

[69] The detailed simulations in this paper use the hydroclimatic data from the Mahurangi River Variability Experiment (MARVEX) in Northland, New Zealand [see Woods *et al.*, 2001; Ibbitt and Woods, 2004]. Typical annual rainfall over Mahurangi catchment are about 1600 mm (with convective activity in the austral summer and larger frontal systems in winter) and annual pan evaporation is about 1300 mm. Catchment elevations range from sea level to 250 m, and the land use is a mixture of pasture and plantation forestry. Most soils in the catchment are clay loams less than a meter deep.

[70] In this study we used daily basin-average rainfall estimated from spatially averaging 2 min rainfall data from 13 rain gauges, potential evapotranspiration estimated using the Priestly-Taylor method using temperature, humidity, and solar radiation data from a standard network of observing stations [Tait and Woods, 2007] and daily streamflow gauged at the Auckland Regional Council station at Mahurangi at College (drainage area is 46.65 km²). We used the total length of available data (1492 days), with the first 274 days used

Table 1. Summary of the FUSE Models Selected for This Study

	FUSE Model					
	070	060	536	550	092	330
Upper layer architecture						
Single state	✓	✓	✓	✓	–	–
Separate tension and free storage	–	–	–	–	✓	✓
Lower layer architecture						
Single state, fixed storage	–	–	–	✓	–	✓
Single state, unlimited storage, no lower layer evaporation	✓	✓	–	–	–	–
Tension storage plus two parallel base flow reservoirs	–	–	✓	–	✓	–
Evaporation						
Evaporation in upper layer constrained only by storage below field capacity, whereas lower layer evaporation also constrained by the potential evapotranspiration not used up by the upper layer	✓	✓	–	–	✓	✓
Evaporation in each soil layer depends on relative root fraction in the upper and lower soil layers.	–	–	✓	✓	–	–
Drainage						
Nonlinear function of free storage in upper layer	–	✓	–	–	–	✓
Nonlinear function of total storage in upper layer	✓	–	✓	✓	–	–
Linear function of upper layer free storage and non-linear function of total lower layer storage	–	–	–	–	✓	–
Interflow						
No interflow	✓	✓	–	–	✓	–
Linear function of free storage in the upper layer	–	–	✓	✓	–	✓
Base flow						
Single linear reservoir	✓	–	–	–	–	–
Single nonlinear reservoir	–	–	–	✓	–	✓
Single nonlinear reservoir, TOPMODEL parameterization	–	✓	–	–	–	–
Two parallel linear reservoirs	–	–	✓	–	✓	–
Surface Runoff						
Saturated area is related to upper layer storage via a Pareto distribution	✓	–	✓	–	✓	✓
Saturated area is related to lower layer storage via the topographic index	–	✓	–	–	–	–
Saturated area is a linear function of tension storage in the upper layer	–	–	–	✓	–	–

for spin-up to minimize the influence of the unknown initial conditions.

5.2. Model Parameter Estimation Experiment Data

[71] A broader but less detailed numerical assessment was carried out for 12 catchments with diverse hydroclimatic and land surface conditions, using the Model Parameter Estimation Experiment (MOPEX) data set from the eastern United States [Duan *et al.*, 2006]. The MOPEX basins, summarized in Table 3, span dry to wet regimes, croplands to mixed forests, and a range of soil types. Eleven years of daily data were available, with the first year used for spin-up.

[72] Following Clark *et al* [2008b], the original MOPEX precipitation time series were replaced with combined rain plus snowmelt estimates obtained from the SNOW-17 model used for the National Weather Service MOPEX simulations. This avoids the need to setup a snow model and allows us to focus on model simulations of hydrological processes in the soil zone.

6. Methodology

[73] This paper compares the performance of all time stepping schemes described in section 3 for all FUSE configurations listed in section 4. Three major aspects are investigated: (1) numerical performance, via assessments of accuracy and computational cost; (2) influence of forcing data resolution on the numerical performance and efficiency, including numerical accuracy and computational cost; and (3) pertinence of conclusions over a wide range of hydroclimatic and catchment conditions.

6.1. Evaluating Numerical Performance

6.1.1. Fidelity and Its Relation to Accuracy, Stability, and Reliability

[74] Traditionally, numerical analysis focuses strictly on the approximation errors of a numerical scheme XX compared to the exact solution. For example, the absolute numerical error is

$$e_{\text{num(XX)}}^{(\text{abs})} = \|\mathbf{y}_{\text{XX}} - \mathbf{y}_{\text{exact}}\|. \quad (17)$$

where $\mathbf{y}_{\text{XX}}$ is the numerical approximation and $\mathbf{y}_{\text{exact}}$ is the exact solution. However, since practitioners are generally more interested in how well the model predicts the observed data rather than in the numerical errors per se, it is important to quantify numerical errors in the context of overall modeling errors. This is not immediately captured by equation (17).

[75] While alternative approaches are possible, we define the fidelity ϕ of a numerical scheme XX as the additional approximation error with respect to the exact solution of the governing equations given observed data $\tilde{\mathbf{y}}$. In this paper we introduce the fidelity measure

$$\phi_{\text{XX}}^{(\text{abs})} = \|\tilde{\mathbf{y}} - \mathbf{y}_{\text{XX}}\| - \|\tilde{\mathbf{y}} - \mathbf{y}_{\text{exact}}\|. \quad (18)$$

In general, $\|\tilde{\mathbf{y}} - \mathbf{y}_{\text{exact}}\|$ will reflect errors in the input-output data, as well as model structural errors. Densely instrumented catchments, such as the Mahurangi basin, allow reducing (though not eliminating) data errors, so that $\|\tilde{\mathbf{y}} - \mathbf{y}_{\text{exact}}\|$ may primarily reflect model structural error. Note that, given our definition, fidelity is a property of a numerical scheme within

Table 2. State and Flux Equations for the Six Selected FUSE Models

Model	Upper Layer ^a	Lower Layer ^a	Evaporation ^b	Drainage	Interflow	Base flow ^c	Surface Runoff ^d
FUSE-070	$\frac{dS_1}{dt} = (p - q_{ss}) - e_1 - q_{12} - q_{if} - q_{uof}$	$\frac{dS_2}{dt} = q_{12} - q_b$	$e_1 = \text{pet} \min\left(\frac{S_1}{\phi_{\text{ens}} S_{1,\text{max}}}, 1\right)$	$q_{12} = k_u \left(\frac{S_1}{S_{1,\text{max}}}\right)^c$	$q_{if} = 0$	$q_b = \nu S_2$	$q_{ss} = p \left[1 - \left(1 - \frac{S_1}{S_{1,\text{max}}}\right)^b\right]$
FUSE-060	$\frac{dS_1}{dt} = (p - q_{ss}) - e_1 - q_{12} - q_{if} - q_{uof}$	$\frac{dS_2}{dt} = q_{12} - q_b$	$e_1 = \text{pet} \min\left(\frac{S_1}{\phi_{\text{ens}} S_{1,\text{max}}}, 1\right)$	$q_{12} = k_u \left(\frac{S_1}{S_{1,\text{max}}}\right)^c$	$q_{if} = 0$	$q_b = \frac{k_b m}{\lambda_n} \left(\frac{S_2}{m}\right)^n$	$q_{ss} = p \int_{\zeta_{\text{crit}}}^{\infty} f(\zeta) d\zeta$ $\zeta_{\text{crit}} = \lambda_n \left(\frac{S_2}{S_{2,\text{max}}}\right)^{-1}$
FUSE-536	$\frac{dS_1}{dt} = (p - q_{ss}) - e_1 - q_{12} - q_{if} - q_{uof}$	$\frac{dS_2^{\text{Tens}}}{dt} = \kappa q_{12} - e_2 - q_{sof}$ $\frac{dS_2^{\text{FreeA}}}{dt} = \frac{(1 - \kappa)q_{12} + q_{sof}}{2} - q_b^A - q_{sof}^A$ $\frac{dS_2^{\text{FreeB}}}{dt} = \frac{(1 - \kappa)q_{12} + q_{sof}}{2} - q_b^B - q_{sof}^B$	$e_1 = \text{pet} r_1 \min\left(\frac{S_1}{\phi_{\text{ens}} S_{1,\text{max}}}, 1\right)$ $e_2 = \text{pet} r_2 \min\left(\frac{S_2}{\phi_{\text{ens}} S_{2,\text{max}}}, 1\right)$	$q_{12} = k_u \left(\frac{S_1}{S_{1,\text{max}}}\right)^c$	$q_{if} = k_i \left(\frac{S_1^{\text{Free}}}{S_{1,\text{max}}^{\text{Free}}}\right)$	$q_b = \nu_A S_2^{\text{FreeA}} + \nu_B S_2^{\text{FreeB}}$	$q_{ss} = p \left[1 - \left(1 - \frac{S_1}{S_{1,\text{max}}}\right)^b\right]$
FUSE-550	$\frac{dS_1}{dt} = (p - q_{ss}) - e_1 - q_{12} - q_{if} - q_{uof}$	$\frac{dS_2}{dt} = q_{12} - e_2 - q_b - q_{sof}$	$e_1 = \text{pet} r_1 \min\left(\frac{S_1}{\phi_{\text{ens}} S_{1,\text{max}}}, 1\right)$ $e_2 = \text{pet} r_2 \min\left(\frac{S_2}{\phi_{\text{ens}} S_{2,\text{max}}}, 1\right)$	$q_{12} = k_u \left(\frac{S_1}{S_{1,\text{max}}}\right)^c$	$q_{if} = k_i \left(\frac{S_1^{\text{Free}}}{S_{1,\text{max}}^{\text{Free}}}\right)$	$q_b = k_b \left(\frac{S_2}{S_{2,\text{max}}}\right)^n$	$q_{ss} = p \left(\frac{S_1^{\text{Tens}}}{\phi_{\text{ens}} S_{1,\text{max}}^{\text{Tens}}} A_{s,\text{max}}\right)$
FUSE-092	$\frac{dS_1^{\text{Tens}}}{dt} = (p - q_{ss}) - e_1 - q_{uof}$ $\frac{dS_1^{\text{Free}}}{dt} = q_{uof} - q_{12} - q_{if} - q_{uof}$	$\frac{dS_2^{\text{Tens}}}{dt} = \kappa q_{12} - e_2 - q_{sof}$ $\frac{dS_2^{\text{FreeA}}}{dt} = \frac{(1 - \kappa)q_{12} + q_{sof}}{2} - q_b^A - q_{sof}^A$ $\frac{dS_2^{\text{FreeB}}}{dt} = \frac{(1 - \kappa)q_{12} + q_{sof}}{2} - q_b^B - q_{sof}^B$	$e_1 = \text{pet} \min\left(\frac{S_1}{\phi_{\text{ens}} S_{1,\text{max}}}, 1\right)$ $e_2 = (\text{pet} - e_1) \min\left(\frac{S_2}{\phi_{\text{ens}} S_{2,\text{max}}}, 1\right)$	$q_{12} = q_0 d_L \left(\frac{S_1^{\text{Free}}}{S_{1,\text{max}}^{\text{Free}}}\right)^c$ $d_L = 1 + \alpha \left(1 - \frac{S_2}{S_{2,\text{max}}}\right)^w$	$q_{if} = 0$	$q_b = \nu_A S_2^{\text{FreeA}} + \nu_B S_2^{\text{FreeB}}$	$q_{ss} = p \left[1 - \left(1 - \frac{S_1}{S_{1,\text{max}}}\right)^b\right]$
FUSE-330	$\frac{dS_1^{\text{Tens}}}{dt} = (p - q_{ss}) - e_1 - q_{uof}$ $\frac{dS_1^{\text{Free}}}{dt} = q_{uof} - q_{12} - q_{if} - q_{uof}$	$\frac{dS_2}{dt} = q_{12} - e_2 - q_b - q_{sof}$	$e_1 = \text{pet} \min\left(\frac{S_1}{\phi_{\text{ens}} S_{1,\text{max}}}, 1\right)$ $e_2 = (\text{pet} - e_1) \min\left(\frac{S_2}{\phi_{\text{ens}} S_{2,\text{max}}}, 1\right)$	$q_{12} = k_u \left(\frac{S_1^{\text{Free}}}{S_{1,\text{max}}^{\text{Free}}}\right)^c$	$q_{if} = k_i \left(\frac{S_1^{\text{Free}}}{S_{1,\text{max}}^{\text{Free}}}\right)$	$q_b = k_b \left(\frac{S_2}{S_{2,\text{max}}}\right)^n$	$q_{ss} = p \left[1 - \left(1 - \frac{S_1}{S_{1,\text{max}}}\right)^b\right]$

^aThe superscripts Tens and Free denote tension storage and free storage, respectively.^bSmoothed using the methods of Kavetski and Kuczera [2007].^cThe subsurface depth scaling parameter for the base flow parameterization in FUSE-060 is $m = S_{2,\text{max}}/n$.^dThe variable ζ in the surface runoff component of FUSE-060 describes the spatial distribution of the topographic index [Beven and Kirkby, 1979; Beven, 1997].

Table 3. Summary of the Mahurangi and MOPEX Basins^a

Basin ID	River	Latitude	Longitude	Area (km ²)	Soil	Vegetation ^b	Mean Annual Precipitation (mm)	Mean Annual Potential ET (mm)
MAHU	Mahurangi	-36.4166	174.6356	47	Clay loam	Pasture/forest	1600	1300
GUA	Guadalupe	29.8606	-98.3828	3406	Clay	Crop/nat. veg.	765	1528
SAN	San Marcos	29.6650	-97.6497	2170	Clay	Crop/nat. veg.	827	1449
ENG	English	41.4664	-91.7156	1484	Clay loam	Cropland	893	994
SPR	Spring	37.2456	-94.5661	3015	Silt loam/clay loam	Dec. broadleaf	1076	1094
RAP	Rappahannock	38.3222	-77.5181	4134	Clay loam	Mixed forest	1030	920
MON	Monocacy	39.3880	-77.3800	2116	Silt loam	Dec. broadleaf	1041	896
EAS	East Fork White	39.2000	-85.9256	4421	Silt loam/clay loam	Croplands	1015	855
POT	S. Branch Potomac	39.4469	-78.6544	3810	Loam	Dec. broadleaf	1042	761
BLU	Bluestone	37.5439	-81.0106	1020	Silt clay loam/loam	Dec. broadleaf	1018	741
AMI	Amite	30.4639	-90.9903	3315	Silt loam	Ever. Needleleaf	1564	1073
TYG	Tygart Valley	39.1500	-80.0400	2372	Loam	Dec. broadleaf	1166	711
FBR	French Broad	35.6092	-82.5786	2448	Loam	Mixed forest	1383	819

^aFollowing *van Werkhoven et al.* [2008], the basins are ordered from dry to wet based on the wetness index (precipitation/potential evapotranspiration).

^bDec. denotes deciduous, Ever. denotes evergreen, and nat. veg. denotes natural vegetation.

the context of a model and the forcing-response observation system used to evaluate it.

[76] Our adoption of the term “fidelity” parallels the audiovisual industry, where it describes the degree to which an electronic system reproduces the sound or image of its input signal [Parker, 1984]. Fidelity, as defined in equation (18), quantifies the consistency of the numerical approximations with the governing equations in the practical context of fitting observed data. The term “fidelity” is used in preference to “stability” or “accuracy” to emphasize the ability to reproduce the observed data rather than the exact ODE solution. We then reserve the term “stability” strictly for the context of uncontrolled error growth, and “numerical accuracy” strictly for the difference between numerical and exact solutions. Fidelity is also used in preference to “reliability” because the latter is both too broad and has too many other distinct connotations in mathematics, statistics, and engineering. Instead, we informally use the term “reliable” to indicate approximations with high fidelity over a broad range of parameter values and forcing conditions. Note that the antonym of fidelity is “infidelity,” and the corresponding adjectives are “fidelious” and “infidelious.”

[77] The choice of norm in (18) controls the aspects of the errors that are emphasized by the fidelity measure. Here we use the root-mean-square error (RMSE) to quantify discrepancies between any two time series y_1 and y_2 ,

$$\Phi_{\text{RMSE}}[y_1, y_2] = \sqrt{\frac{1}{N_t} \sum_{n=1}^{N_t} (y_1^n - y_2^n)^2}, \quad (19)$$

where the superscript n indexes the time step and N_t is the total number of time steps.

[78] The use of the RMSE (19) within the fidelity definition (18) allows intuitive statements such as “for catchment Z, numerical errors raised the RMSE of hydrological model A from X mm/h to Y mm/h.” Other norms, such as the max-norm, etc., can be used to emphasize other accuracy aspects. More generally, alternative definitions of fidelity can also be designed.

6.1.2. Assessment of Fidelity

[79] The analysis is carried out in two stages. First, we present example time series from the FUSE-070 model for different time stepping schemes. Second, we compare the RMSE of all FUSE models and time stepping schemes, using

10,000 randomly sampled parameter sets. This gauges whether the problems identified during the detailed analysis of FUSE-070 are general issues commonly arising in other model structures and parameter sets.

[80] In the absence of analytical solutions of the FUSE model equations, we use an adaptive implicit Heun approximation with tighter truncation error tolerances, $\tau_R = 0.01\%$ and $\tau_A = 0.0001$ mm, as a proxy for the exact solution. It was verified to be consistent with the other adaptive error-controlled schemes.

6.1.3. Measuring the Computational Cost

[81] The computational cost of all time stepping algorithms is quantified using N_f , the number of right-hand-side (RHS) flux function evaluations of ODE system (2), averaged over all data-resolution time steps of a single run (e.g., daily when the model is forced with daily data). In explicit schemes, N_f depends solely on the number of substeps, whereas for implicit and semi-implicit schemes, it also depends on the number of Jacobian evaluations (each requiring $N_s + 1$ RHS flux evaluations when approximated using one-sided finite differences) and, for iterative implicit schemes, on the number of Newton-Raphson iterations.

[82] This metric of computational cost assumes the latter is dominated by the evaluation of RHS fluxes (rather than, e.g., by internal algorithm bookkeeping, LU decompositions, interprocedural software communication, etc.). This is reasonable in most conceptual hydrological models, where the number of states N_s seldom exceeds 5–10. For reference, typical runtimes on a 2 GHz desktop CPU were of the order of 0.1 s for most FUSE simulations of the 1492 day period in the Mahurangi River basin. For implicit schemes, we also investigate the efficiency gains of the Jacobian refreshment strategies outlined in section 3.6.

6.2. Impact of the Choice of Model Time Step

[83] The numerical assessments of section 6.1 are based on daily step simulations. Since subdaily steps improve numerical accuracy, we evaluate the resulting accuracy-efficiency trade-offs. To this end, we examine the performance of the explicit Euler scheme with data-resolution steps of 3 h, 1 h, and 15 min. Because this study focuses strictly on numerical errors, rather than on structural uncertainty or on effects of data resolution on the model inference, the hydrological data (rainfall, evapotranspiration (ET), streamflow) were always

Figure 2. (top) Precipitation, (middle) upper zone storage, and (bottom) drainage in the Mahurangi basin over June–August 1998, simulated by FUSE-070 implemented using different time stepping schemes. All model simulations are based on the same parameter set, $\phi_{\text{tens}} = 0.5$, $S_{1,\text{max}} = 100$ mm, $k_u = 500$ mm d⁻¹, $c = 10$, $\nu = 0.1$ d⁻¹, $b = 0.2$, and $\mu_\tau = 0.6$ days. The storm event on 25 July 1998 was the largest precipitation event in the calibration period. For the same governing equations and parameter sets, considerable differences in model predictions and internal states are obtained depending on the time stepping scheme.

uniformly disaggregated into the subdaily periods. This methodology ensures that subdaily simulations are directly comparable to the daily simulations and any differences are attributable solely to numerical implementation. It is stressed that while structural and observational uncertainties are beyond the scope of this paper, the improvements in robustness and efficiency advocated in this study will pave the way for a far more meaningful and reliable identification and characterization of these critical errors in future work.

6.3. Impact of the Choice of Basin

[84] Previous numerical experiments in rainfall-runoff hydrology were limited to simple single-state models and a few catchments [e.g., Kavetski *et al.*, 2003]. The detailed analyses of sections 6.1–6.2 are also based solely on the Mahurangi basin. To gauge the generality of our conclusions, we conduct a much broader assessment using the 12 MOPEX basins.

7. Evaluation of Numerical Fidelity

7.1. Numerical Artifacts of Explicit Methods

[85] Figure 2 compares different time stepping solutions of FUSE-070 given a representative parameter set. Substantial differences are immediately clear, with fixed-step explicit

approximations generally much “spikier” than the accurate error-controlled solutions. Troublesome artifacts are evident in the fixed-step explicit Euler and Heun schemes, e.g., the upper zone storage frequently drains in a single time step (middle row of Figure 2). Such behavior is unavoidable in most fixed-step explicit methods when $k \gg S_{\text{max}}/\Delta t$ in the context of solving ODEs such as $dS/dt = P - k(S/S_{\text{max}})$ with $0 < S < S_{\text{max}}$. Indeed, had we not implemented feasibility safeguards as described in section 3.9, the approximations would become oscillatory and possibly unstable.

[86] Note the visible time shift of about one time step between the fixed-step explicit Euler and Heun approximations. In general, the former tends to underpredict drainage at the start of a storm and overpredict at the end and vice versa for the latter. This occurs because the time step average fluxes in the explicit Euler scheme depend solely on the initial states, whereas in the explicit Heun scheme they also depend on the explicit Euler estimates at the end of the step. The daily step size appears exceedingly large for either approximation, and the resulting numerical errors introduce large spurious time shifts.

7.2. Implicit Methods

[87] Figure 2 also exposes some numerical artifacts in the fixed-step implicit Heun and semi-implicit schemes. In par-

Figure 3. Fidelity of fixed-step time stepping schemes, estimated from 10,000 randomly selected parameter sets, assessed by comparing the RMSE of an exact solution of the FUSE equations (x axis) with the RMSE of approximate solutions (y axis). Both RMSEs are computed with respect to the observed Mahurangi streamflows. The exact solution is estimated using the adaptive implicit Heun scheme with tight error tolerances ($\tau_R = 0.01\%$ and $\tau_A = 10^{-4}$ mm). It can be seen that for some models, e.g., FUSE-536 and FUSE-550, fixed-step explicit and semi-implicit time stepping schemes suffer from uncontrollable and excessive numerical errors, and are clearly infidelious.

ticular, the fixed-step implicit Heun scheme is also often excessively dominated by the large flux at the start of the model time step (e.g., consider the storm events in July 1998).

[88] A relatively frequent problem in the implicit Heun method is exceedance of the bucket capacity. This is especially common for parameter sets that correspond to low storage capacity and/or slow drainage rate and requires ad hoc corrections (e.g., “running off” excess inflows to avoid overflowing the bucket). In contrast, violations of nonnegativity constraints were less common because implicit methods are robust when simulating rapid declines in drainage rates over the time step.

[89] Large numerical errors also develop when a highly nonlinear reservoir becomes near-full (Figure 2): averaging fluxes at the start and end of the time step (in the implicit Heun scheme) does not adequately mimic the rapid decay of drainage, as the exact solution itself decays very rapidly and

nonlinearly over the time step. In some rare cases, the initial flux in the implicit Heun scheme is so large that, for a sufficiently large step size Δt , the Newton-Raphson solver fails to converge because the numerical approximation violates solution constraints (in explicit schemes, this scenario would “overdrain” the entire catchment in a single time step).

[90] Figure 2 also shows artifacts in the fixed-step semi-implicit solution. For example, it underestimates drainage during the storms on 13 June and 10 August 1998 by 13 (53%) and 22 mm (84%), respectively (Figure 2). These occur because the semi-implicit linearization (14) is a poor approximation of time step average fluxes when Δt is too large. In extreme cases, the fixed-step semi-implicit solution can become infeasible, e.g., predicting negative fluxes, infeasible drainage, or total ET exceeding the potential value.

[91] Whenever these problems occur for explicit and implicit schemes alike, they represent the effects of large

Figure 4. Fidelity of time stepping schemes across different model structures, represented using equation (18), evaluated using 10,000 randomly selected parameter sets and the Mahurangi basin. Fixed-step explicit approximations have markedly lower fidelity than unconditionally stable schemes (implicit Euler) and/or adaptive solutions (adaptive Heun).

truncation errors arising from excessive step size rather than stability problems. It follows that fixed daily time steps may be exceedingly large for catchment simulation. Simply raising the order of the approximation does not address this problem, e.g., the $O(\Delta t^2)$ explicit Heun method suffers from the same problems as the $O(\Delta t)$ explicit Euler scheme. Indeed, the order of accuracy is an asymptotic concept and becomes meaningless if Δt is exceedingly large.

7.3. Other Model Configurations

[92] To investigate the generality of our conclusions, Figures 3 and 4 compare the RMSE of fixed-step versus adaptive schemes for all six FUSE configurations (Table 1), computed with respect to the observed Mahurangi data. It vividly confirms that macroscale numerical errors are not rare occurrences but represent a ubiquitous problem systematically afflicting fixed-step hydrological models over wide regions of their parameter space. These errors are far from negligible and frequently dwarf structural errors of the governing equations and input-data errors: in many cases, the RMSE of models based on the fixed-step explicit Euler approximation are 2–3 times larger than the RMSE of models based on the exact solutions of the *same* model equations.

[93] Interestingly, models that include the interflow process (FUSE-536, FUSE-550, and FUSE-330) appear particularly prone to numerical errors. Note that here interflow is delivered directly to the stream, and unlike drainage, it is not modulated by the lower soil layer (Figure 1). Numerical problems are relatively less pronounced in FUSE-330 because it has a separate state variable representing storage above field capacity, limiting the amount of interflow that can occur in a single time step. However, these interactions may well be problem specific.

[94] Figures 3 and 4 also show that, of all fixed-step schemes examined in this paper, the implicit Euler and the implicit Heun schemes provide the most faithful representation of the exact solution. These approximations almost

always satisfy solution constraints and hence are less affected by the somewhat arbitrarily derived imposition of solution constraints.

[95] Note that the implicit Euler scheme *always* satisfies solution constraints (section 3.9). While its fixed-step implementation cannot guarantee uniformly tight errors (e.g., Figure 2), it preserves essential solution properties such as nonnegativity, nonoscillatory smooth decay, etc. The resulting qualitative consistency effectively limits solution errors: in most runs shown in Figures 3 and 4, the numerical errors of the implicit Euler scheme represent less than 5% of the RMSE of the exact solution, whereas the numerical errors of the explicit Euler scheme frequently exceeded 100% of this RMSE. Therefore, unlike fixed-step explicit schemes, the fixed-step implicit Euler scheme appears to yield an acceptable accuracy in the context of conceptual hydrological modeling. In all FUSE models, the numerical errors of fixed-step implicit Euler and implicit Heun schemes are below 0.1 mm/d in over 90% of the sampled parameter sets.

7.4. Interaction Between Structural and Numerical Errors of the Implicit Euler Scheme

[96] Interestingly, in a significant number of cases, the fixed-step implicit Euler solutions fit the observed data better than the exact ODE solutions. For example, the RMSE of fixed-step implicit Euler solutions is lower than the RMSE of the exact solution by 0.01 mm in more than 60% of the sampled parameter sets, in all models except FUSE-060. This is consistent with previous studies [e.g., Kavetski *et al.*, 2006, Figure 2]. It could be hypothesized that this occurs whenever the truncation error of the fixed-step implicit Euler scheme fortuitously cancels portions of model error. Given the diffusive (smoothing) properties of the implicit Euler scheme, it could also imply that the exact governing equations are not sufficiently diffusive. Interestingly, numerical errors are indeed sometimes used to reduce model errors, in particular, using “numerical viscosity” [LeVeque, 1990], incidentally,

Figure 5. Trade-off between the truncation error tolerance and computational cost of the three adaptive solutions. The cost of the fixed-step implicit Euler scheme is shown as the horizontal line for reference only; its numerical accuracy is not error controlled. Results represent the average of 100 randomly selected parameter sets (fewer parameter sets were used in this experiment because of the large computational cost of tight error tolerances) for all FUSE models and the Mahurangi basin data. Asymptotically, the error of a first-order scheme decreases linearly as the step size is reduced, whereas for second-order schemes, the dependence is quadratic. As shown here, the theoretically expected relationship with respect to the relative error tolerance breaks down when the absolute error tolerance becomes the limiting factor, when additional costs (Jacobian evaluation) are included, or both. Similar behavior is obtained with respect to the absolute error tolerance.

precisely in implicit schemes. However, this is difficult to implement reliably because it requires detailed insights into the characteristics of both numerical and model errors. Further interpretation of this phenomenon is outside the scope of this paper.

7.5. Why Not Always Adaptive?

[97] Since adaptive substepping generally ensures stability and accuracy, it is widely viewed as essential in scientific computing and is often the only method available in many good numerical ODE software. For example, *Shampine and Reichelt* [1997] expressly do not allow fixed-step integration in their Matlab ODE toolkit. Then why not always use adaptive schemes in conceptual hydrological models? Two

possible reasons could be given (though not necessarily accepted): cost and smoothness. The computational cost is a critical practical performance issue and is discussed next. On the other hand, smoothness can be important in gradient-based parameter optimization and will be considered in the companion paper.

8. Computational Cost

8.1. Adaptive Substepping Versus Fixed-Step Implicit Scheme

[98] The imposition of near-uniform accuracy over the entire simulation makes adaptive substepping computationally more expensive, and much more reliable, than algorithms that do not impose this requirement. Figure 5 shows that the

Figure 6. Average number of flux function evaluations per time step of the fixed-step implicit Euler scheme (black) and the adaptive explicit Heun scheme (gray) with $\tau_R = 1\%$ and $\tau_A = 0.01$ mm. Results obtained using 10,000 randomly selected parameter sets and the Mahurangi basin data. The split box indicates the 25th, 50th, and 75th percentiles, whereas the whiskers indicate the minimum and maximum values. While the adaptive explicit Heun solution is on average cheaper than the fixed-step implicit Euler approximation, it has a high variability in its cost, likely as a consequence of ODE stiffness of the hydrological models for certain forcing conditions and parameter sets.

truncation error tolerance not need be very stringent before the adaptive schemes become more expensive than the fixed-step implicit Euler method with daily steps. For example, the adaptive explicit Heun scheme has a similar cost to the fixed-step implicit Euler scheme when $\tau_R \sim 10^{-3}$ – 10^{-4} in FUSE-070, FUSE-060, FUSE-536, and FUSE-550 but is much more expensive in FUSE-092 and FUSE-330 for all error settings.

[99] The adaptive semi-implicit and adaptive implicit Heun solutions are even more expensive (Figure 5). Note that the explicit and implicit Heun approximations have a similar local truncation error and hence, in the absence of stiffness (Appendix A), require a similar number of substeps for a given truncation tolerance. However, implicit solutions require iterations including Jacobian evaluations, substantially raising their overall cost.

[100] Figure 5 also shows that the semi-implicit scheme and the implicit Heun method have a similar cost at tight error tolerances: for sufficiently small substeps, the Newton-Raphson solver frequently converges in a single iteration. It may also be possible to improve the efficiency of adaptive semi-implicit and implicit methods by keeping the Jacobian constant over many consecutive daily steps, rather than just over substeps (Section 3.6.2).

[101] The computational cost of ensuring uniform accuracy and the trade-off between explicit and implicit schemes are examined next. While the implicit Euler allows errors of up to 0.1 mm/d, which would not pass the truncation error test,

it is arguably tolerable in a conceptual error model with optimal RMSE of the order of 2 mm/d. Figure 6 compares the computational cost of the fixed-step implicit Euler scheme and the adaptive explicit Heun method with moderate truncation error settings ($\tau_R = 1\%$ and $\tau_A = 0.01$ mm). Although the adaptive explicit Heun method can be more expensive for some parameter sets (suggestive of stiffness, see discussion near the end of section 10), it is generally more computationally efficient, and almost always more accurate, than the fixed-step implicit Euler method.

8.2. Improving the Efficiency of Implicit Schemes

[102] The fixed-step implicit Euler method requires significantly more flux function evaluations than the fixed-step explicit Euler method commonly used in hydrological models. For example, in the simulations shown in Figure 2, while the fixed-step implicit Euler scheme required just two Newton-Raphson iterations for convergence to absolute tolerance $\tau_M = 10^{-9}$ mm in 90% of the time steps, it required five or more iterations for convergence during storm events and 10+ iterations on particularly stormy days. Because (one-sided) finite-difference Jacobians require $N_s + 1$ function calls, the model runtime can easily become an order of magnitude higher than for a fixed-step explicit method. Notwithstanding the much better robustness and smoothness of the fixed-step implicit Euler method, this can be discouraging to a practitioner. Consequently, this section investigates the computational savings possible under the Jacobian refreshment strategies described in section 3.6.

[103] Figure 7 compares the computational cost of the fixed-step implicit Euler method enhanced with different Jacobian refreshment strategies, for 1000 randomly selected parameter sets. Overall, the Jacobian strategies were overwhelmingly effective over a wide range of simulation conditions and error settings, the standard strategy of reevaluating the Jacobian at each iteration was, on average, the least efficient approach in all runs. The increases in efficiency can be spectacular. For example, strategies J-B and J-C halved the average computational cost of FUSE-092. Yet in other cases, Jacobian strategies were less effective. For example, for some parameter sets (in particular, corresponding to shallow or fast-draining soils), setting $\tau_{\text{dfunc}} = 1$ (i.e., a constant Jacobian at all iterations within a substep) delays the convergence to the point of foregoing most savings due to avoiding Jacobian evaluation. Current experimentation favors strategy J-B with $\tau_{\text{dfunc}} = 0.2$ or J-C with $\tau_{\text{freeze}} = 0.01$ mm, but strategy J-B is preferred because it directly detects and corrects slow convergence.

9. Impact of Model Step Size

[104] Because many hydrological models are run at sub-daily steps [e.g., Blöschl et al., 2008] and truncation errors are reduced as $\Delta t \rightarrow 0$, it is important to investigate numerical errors for subdaily step models. Figure 8 compares the fidelity of the adaptive explicit Heun (forced with daily data) and fixed-step explicit Euler schemes with forcing data resolution of 1 day, 3 h, 1 h, and 15 min, for the Mahurangi basin. As expected, shorter time steps reduce the large numerical errors evident in the daily step simulations. However, quite remarkably, some models solved using 15 min explicit Euler time steps have similar or lower accuracy than when solved using the adaptive explicit Heun scheme with daily model

Figure 7. Effectiveness of Jacobian refreshment strategies J-B and J-C versus the default approach J-A (“fully variable” Jacobian) in reducing the computational cost of the implicit Euler scheme. Results obtained using 1000 randomly selected parameter sets and the Mahurangi basin data. The split box indicates the 25th, 50th, and 75th percentiles, whereas the whiskers indicate the minimum and maximum values. On average, all Jacobian refreshment strategies lowered the cost compared to the standard approach of recomputing the Jacobian at every time step. However, in some of the runs the refreshment strategies were nonrobust, leading to increased cost variability.

Figure 8. Impact of time step size on the fidelity of the fixed-step explicit Euler scheme. Results obtained using the same 10,000 randomly selected parameter sets as in Figure 4 and the Mahurangi basin data. The fidelity of the adaptive explicit Heun scheme is shown for context. Note that here the x axis is restricted to $(-0.1, 0.1)$, whereas in Figure 4 it ranges from $(-1, 1)$. As expected, fidelity improves for shorter step sizes. Adaptive error control also improves fidelity, e.g., for FUSE-070 and FUSE-330, daily step simulations with adaptive substepping achieved the same fidelity as simulations using 15 min time steps.

steps and a moderate truncation tolerance. Moreover, simulations with small fixed steps incur a substantial cost: the explicit Euler with 15 min steps necessarily requires 96 flux evaluations per day, whereas the adaptive explicit Heun scheme with $\tau_R = 1\%$ and $\tau_A = 0.01$ mm requires (on average) less than 10 flux evaluations per day in the Mahurangi basin simulations examined in this paper. Consequently, the explicit Euler solution with short fixed substeps is warranted only if computational cost is not of immediate concern. Even then, in the absence of direct error control, it would be difficult, if not impossible, to ensure fidelity over all possible values of parameters, forcing, and internal states.

10. Assessment Over General Hydroclimatic and Catchment Conditions

[105] This section provides an evaluation of fidelity (Figure 9) and efficiency (Figure 10) for the 12 MOPEX basins. This broad assessment is necessary to ensure that the conclusions reached in the analysis of FUSE in the experimental Mahurangi basin are pertinent to model applications in different physical and hydroclimatic conditions.

[106] Figure 9 vividly demonstrates large numerical artifacts in the explicit Euler scheme in all 12 MOPEX basins. In 25% of the sampled parameter sets in all 12 basins, additional RMSE due to numerical errors (“numerical RMSE”) exceeded 1 mm/d in FUSE-536 and FUSE-550. Moreover, the largest numerical RMSE in the explicit Heun scheme exceeded 1 mm/d in FUSE-060, FUSE-070, FUSE-092, and FUSE-330 and, rather remarkably, exceeded 10 mm/d in FUSE-536 and FUSE-550 (which, as found in section 7.3, appear especially fragile due to their representation of interflow). Note that these are RMSE-measured errors; maximum

(rather than average) errors within a single model run are even higher. Such gross errors are an indictment of the numerical fragility of the explicit Euler scheme and reinforce that it should never be used without error control.

[107] Conversely, Figure 9 indicates that, as suggested by the MARVEX analysis (Figure 3 and Figure 4), the adaptive explicit Heun and fixed-step implicit Euler methods provide a faithful representation of the exact solution. The MOPEX-wide evaluation confirms that this holds under in diverse hydrological conditions. Interestingly, the additional errors in the implicit Euler method appear positively skewed in the drier basins and negatively skewed in the wetter basins. This feature is most evident in FUSE-070 and FUSE-092 (where the lower quartile for the driest basins is above zero) but is also present in FUSE-536 and FUSE-550 (where the median for the driest basins is above zero). The implicit Euler method is less accurate than the adaptive explicit Heun scheme, but its additional errors are almost always below 0.1 mm/d and are hence negligible compared with total RMSEs, which in general are of the order of 1 mm/d in most simulations.

[108] Figure 10 compares the computational cost of the adaptive explicit Heun method (with $\tau_R = 1\%$ and $\tau_A = 0.01$ mm) and the fixed-step implicit Euler method (with the standard Newton-Raphson method where the Jacobian is recalculated at every iteration). The results are consistent with Figure 6: the fixed-step implicit Euler method requires, on average, more function evaluations than the adaptive explicit Heun method (here, ~ 10 flux evaluations per daily step in the fixed-step implicit Euler scheme versus ~ 3 flux evaluations per daily step in the adaptive explicit Heun algorithm).

[109] However, Figure 10 also indicates that the adaptive explicit Heun method is much more expensive than the fixed-step implicit Euler for certain parameter sets, especially in

Figure 9. Fidelity of the explicit Euler scheme (blue), the implicit Euler scheme (red), and the adaptive explicit Heun scheme with $\tau_R = 1\%$ and $\tau_A = 0.01$ mm (green), for all six FUSE models in all 12 MOPEX basins (ordered from driest to wettest). Fidelity is quantified using equation (18). The split box indicates the 25th, 50th, and 75th percentiles, whereas the whiskers indicate the minimum and maximum values. Results of 10,000 randomly selected parameter sets are shown. On average, using adaptive error control significantly improves the fidelity of the numerical solution. Conversely, using fixed-step methods lowers the fidelity, especially when an explicit approximation is used. The unconditional stability of the implicit Euler scheme contributes to its fidelity even with fixed step size but cannot systematically constrain truncation errors in the same way as adaptive error control.

Figure 10. Computational cost of the implicit Euler scheme (red) and adaptive explicit Heun scheme with $\tau_R = 1\%$ and $\tau_A = 0.01$ mm (blue) for all six FUSE models in all 12 MOPEX basins (ordered from driest to wettest). The split box indicates the 25th, 50th, and 75th percentiles, whereas the whiskers indicate the minimum and maximum values. Results of 10,000 randomly selected parameter sets are shown. The notably lower average cost, but higher cost variability of the adaptive explicit Heun scheme is noted, suggesting that the computational efficiency of this scheme is highly variable.

FUSE-092 and FUSE-330. This could be indicative of ODE stiffness, which forces the step size of adaptive explicit methods (which are conditionally stable with fixed steps), to be constrained by stability rather than by accuracy [e.g., *Kahaner et al.*, 1989]. The variation in stiffness depending on parameter values suggests that adaptive switching between explicit and implicit schemes may improve overall efficiency and robustness, in particular, avoiding gross inefficiencies when an explicit scheme is forced to solve a stiff problem.

11. Conclusions

[110] Despite the widespread use of conceptual hydrological models in research and practice, they remain routinely implemented using numerically unreliable techniques. This paper thoroughly evaluated several classes of time stepping schemes in terms of their numerical fidelity and computational efficiency. To explore the generality of its conclusions, the empirical assessment covered 8 distinct time stepping solutions of 6 different conceptual hydrological models, applied both in a densely gauged catchment with accurate rainfall-runoff data, as well as in 12 basins with diverse physical and hydroclimatic conditions.

[111] The key findings of this study are as follows.

[112] 1. Numerical errors of fixed-step explicit schemes routinely dwarf the conceptualization errors of the governing equations. They yield unacceptably poor model simulations for parameter sets that produce good simulations when solved accurately.

[113] 2. Simply running fixed-step explicit schemes with shorter time steps provides a poor balance between accuracy and efficiency: daily step adaptive explicit schemes with moderate truncation tolerances achieved comparable or higher accuracy than a fixed-step explicit scheme run on 15 min time steps but were nearly 10 times faster. Moreover, it is difficult to a priori determine a “safe” fixed step size adequate for a given application.

[114] 3. From the range of simple methods investigated in this work, the fixed-step *implicit* Euler method and the *adaptive* explicit Heun method emerged as good practical choices for the majority of simulation scenarios. These algorithms are computationally efficient yet highly fideliou: their numerical errors are generally negligible vis-à-vis the structural errors of the governing equations. Conversely, fixed-step explicit Euler estimates are routinely afflicted by numerical errors vastly exceeding errors due to model structure and uncertain data. Although in theory less reliable and efficient due to absence of error control, the fixed-step implicit Euler scheme had less variability in its cost than the adaptive explicit Heun scheme, with the latter being computationally very expensive for some parameter sets and forcing conditions (likely due to stiffness of the model equations). The companion paper elaborates in greater depth on the trade-offs between these methods in the context of hydrological model analysis and prediction.

[115] This paper vividly highlights the fragility of uncontrolled numerical approximations. As further illustrated in the companion paper, the traditional disregard of model numerics by hydrologists is unwarranted and is unnecessarily hampering, confounding, and perhaps prohibiting, the scientific and operational improvement of these models. In addition, similar numerical issues may plague distributed hydrological

models, ecological models, land surface schemes for global climate models and catchment-scale nutrient and contaminant transport models, whenever these are implemented without numerical quality control. Since mature theory, methods, and software are available to tackle numerical errors, allowing them to continue obscuring hydrological model analysis is hardly acceptable.

Appendix A: Stability and “Stiffness”

[116] Stability is a key practical consideration in time stepping schemes. While explicit schemes such as (5) and (6) become unstable and produce meaningless results when

$$\Delta t > c/\text{eigmax}\left[\frac{\partial \mathbf{g}}{\partial \mathbf{S}}\right] \quad (\text{A1})$$

many implicit methods, in particular, (7) and (8) remain stable even as $\Delta t \rightarrow \infty$.

[117] In equation (A1), $\text{eigmax}[\partial \mathbf{g}/\partial \mathbf{S}]$ denotes the largest eigenvalue of the ODE Jacobian matrix $\mathbf{J}$

$$J_{ij} = \left[\frac{\partial \mathbf{g}}{\partial \mathbf{S}}\right]_{ij} := \frac{\partial g_i}{\partial S_j}, i = 1 \dots N_s, j = 1 \dots N_s \quad (\text{A2})$$

and c is a small constant, e.g., $c = 2$ for the explicit Euler and $c = 4$ for the explicit Heun methods [*Kahaner et al.*, 1989].

[118] The linear reservoir case $dS/dt = -kS$ is instructive: the ODE Jacobian is $-k$ and the resulting stability condition $\Delta t < 2/k$ can cause problems for fast-drainage scenarios. For nonlinear reservoirs, e.g., $dS/dt = -kS^\alpha$, the stability constraints on Δt can become even more stringent.

[119] ODEs with large Jacobian eigenvalues are commonly referred to as “stiff” and generally should be solved using implicit schemes (see *Kahaner et al.* [1989] and *Lambert* [1991] for a fuller definition and discussion of stiffness). While usually costlier on a per-step basis, the unconditional stability of implicit schemes permits the use of much larger time steps than explicit approximations, ultimately leading to a lower total cost of the entire simulation. Many stiff problems cannot be feasibly solved using explicit schemes yet are easily and accurately solved using implicit schemes [*Shampine*, 1994].

Appendix B: Adaptive ODE Integration

B1. Numerical Error Monitoring

[120] The error of a numerical approximation can be estimated by comparison to a higher-order solution. For example, for the explicit Euler scheme,

$$\mathbf{e}_{\text{EE}(1)}^{n+1} = \mathbf{S}_{\text{EE}(1)}^{n+1} - \mathbf{S}_{\text{EE}(2)}^{n+1}. \quad (\text{B1})$$

Analogous expressions can be derived for other time stepping approximations. In addition, controlling the error in quantities derived from the state variables may be warranted, e.g., monitoring runoff errors as well as storage errors. Note that monitoring the errors in the runoff alone may be unreliable in hydrological models with lags.

[121] The solution for a given substep is evaluated using a mixed error measure [e.g., *Kavetski et al.*, 2001]. The substep

Table C1. FUSE State Variables and Fluxes

Variable	Description	Units
S_1	Total water content in the upper soil layer	mm
S_1^{Tens}	Tension water content in the upper soil layer	mm
S_1^{Free}	Free water content in the upper soil layer	mm
S_2	Total water content in the lower soil layer	mm
S_2^{Tens}	Tension water content in the lower soil layer	mm
S_2^{FreeA}	Free water content in the primary base flow reservoir	mm
S_2^{FreeB}	Free water content in the secondary base flow reservoir	mm
p	Precipitation	mm d ⁻¹
pet	Potential evapotranspiration	mm d ⁻¹
e_1	Evaporation from the upper soil layer	mm d ⁻¹
e_2	Evaporation from the lower soil layer	mm d ⁻¹
e_1^A	Evaporation from the primary tension store	mm d ⁻¹
e_1^B	Evaporation from the secondary tension store	mm d ⁻¹
q_{sx}	Surface runoff	mm d ⁻¹
q_{12}	Drainage of water from the upper to the lower layer	mm d ⁻¹
q_{if}	Interflow	mm d ⁻¹
q_{b}	Base flow	mm d ⁻¹
q_{b}^A	Base flow from the primary reservoir	mm d ⁻¹
q_{b}^B	Base flow from the secondary reservoir	mm d ⁻¹
q_{uof}	Overflow of water from the primary tension store in the upper soil layer	mm d ⁻¹
q_{utof}	Overflow of water from tension storage in the upper soil layer	mm d ⁻¹
q_{ufof}	Overflow of water from free storage in the upper soil layer	mm d ⁻¹
q_{stof}	Overflow of water from tension storage in the lower soil layer	mm d ⁻¹
q_{sfof}	Overflow of water from free storage in the lower soil layer	mm d ⁻¹
q_{sfofa}	Overflow of water from primary base flow storage in the lower soil layer	mm d ⁻¹
q_{sfofb}	Overflow of water from secondary base flow storage in the lower soil layer	mm d ⁻¹

is accepted if absolute and relative truncation error tolerances, τ_A and τ_R , respectively, are satisfied as follows:

$$\gamma_i^{n+1} = |e_i^{n+1}| - \tau_R |S_i^{n+1}| - \tau_A < 0 \forall i = 1 \dots N_S. \quad (\text{B2})$$

The index iCrit of the largest mixed error $\gamma_{\text{iCrit}}^{n+1}$ is used in step size selection (described in the next section).

[122] With meaningfully specified τ_A and τ_R , the mixed error test (B2) is robust over all solution ranges. As $|S| \rightarrow 0$ the test becomes dominated by the absolute error criterion, while as $|S| \rightarrow \infty$ the test smoothly switches to a relative error check. Setting $\tau_R = 0$ forces a strictly absolute error test, while $\tau_A = 0$ yields the standard relative error test.

B2. Adaptive Step Size Selection

[123] If the current substep is accepted based on equation (B2), the step size for the next substep is calculated using

$$\Delta t_1^{n+1} = \Delta t^n \min \left(s \sqrt{\frac{\tau_R |S_{\text{iCrit}}^n| + \tau_A}{\max(e_{\text{iCrit}}^n, \text{EPS})}}, r_{\text{max}} \right). \quad (\text{B3})$$

Conversely, if the substep is rejected, it is reattempted with a reduced step size based on the latest error estimate

$$\Delta t_{j+1}^{n+1} = \Delta t_j^{n+1} \max \left(s \sqrt{\frac{\tau_R |S_{\text{iCrit}}^{n+1}| + \tau_A}{\max(e_{\text{iCrit}}^{n+1}, \text{EPS})}}, r_{\text{min}} \right), \quad (\text{B4})$$

where j indexes the step size reduction.

[124] The step size multiplier constraints $r_{\text{min}} \cong 0.1$ and $r_{\text{max}} \cong 4.0$, the safety factor $s \cong 0.8$ – 0.9 and the machine constant $\text{EPS} \cong 10^{-10}$ increase the robustness of the algorithm by guarding against spuriously large or small step size changes, as well as against substeps that just fail to meet the error tolerance. These safeguards are necessary because the truncation error estimate (B1) is itself based on asymptotic arguments and contains numerical noise.

[125] The adaptive strategy (B1)–(B4) is standard. The reader is referred to *Shampine* [1994] and other numerical ODE texts for more detailed general derivations; *Kavetski et al.* [2002b] detail an application to the implicit Euler scheme in the context of Richards equation.

B3. Error Control for the Implicit Heun and Semi-Implicit Euler Schemes

[126] An error estimate for the implicit Heun scheme is given by

$$\mathbf{e}_{\text{IH}(2)}^{n+1} = \mathbf{S}_{\text{IH}(2)}^{n+1} - \mathbf{S}_{\text{IH}(1)}^{n+1}, \quad (\text{B5})$$

where an auxiliary first-order implicit Heun solution is

$$\mathbf{S}_{\text{IH}(1)}^{n+1} = \mathbf{S}^n + \Delta t \mathbf{g}(\mathbf{S}_{\text{IH}(2)}^{n+1}). \quad (\text{B6})$$

Estimate (B5) can also be written as the difference between successive derivatives

$$\mathbf{e}_{\text{IH}(2)}^{n+1} = \frac{1}{2} \Delta t [\mathbf{g}(\mathbf{S}_{\text{IH}(2)}^{n+1}) - \mathbf{g}(\mathbf{S}^n)]. \quad (\text{B7})$$

The construction of (B5)–(B7) is analogous to the error control scheme designed by *Kavetski et al.* [2001] and represents a minor departure from the standard strategy in section B1.

[127] The semi-implicit Euler scheme can be supplemented with error control by comparing it to the explicit Euler scheme

$$\mathbf{e}_{\text{SIE}(1)}^{n+1} = \frac{1}{2} \Delta t [\mathbf{g}_L(\mathbf{S}_{\text{SIE}(1)}^{n+1}) - \mathbf{g}(\mathbf{S}^n)]. \quad (\text{B8})$$

or, more efficiently, and in line with the approaches of *Kavetski et al.* [2002a]

$$\mathbf{e}_{\text{SIE}(1)}^{n+1} = \frac{1}{2} \Delta t [\mathbf{g}_L(\mathbf{S}_{\text{SIE}(1)}^{n+1}) - \mathbf{g}_L(\mathbf{S}_{\text{SIE}(1)}^n)]. \quad (\text{B9})$$

This is a further departure from the principles in section B1 because the error estimate is obtained by comparing two

Table C2. FUSE Model Parameters

Parameter	Description	Units	Lower Limit	Upper Limit	FUSE Model					
					070	060	536	550	092	330
$S_{1,\max}$	Maximum storage in the upper layer	mm	25.000	500.000	✓	✓	✓	✓	✓	✓
$S_{2,\max}$	Maximum storage in the lower layer	mm	50.000	5000.000	–	–	✓	✓	✓	✓
ϕ_{tens}	Fraction total storage as tension storage	–	0.050	0.950	✓	✓	✓	✓	✓	✓
ϕ_{rchr}	Fraction of tension storage in primary zone (upper layer)	–	0.050	0.950	–	–	–	–	–	–
ϕ_{base}	Fraction of free storage in primary reservoir (lower layer)	–	0.050	0.950	–	–	✓	–	✓	✓
r_1	Fraction of roots in the upper layer	–	0.050	0.950	–	–	✓	✓	–	–
k_u	Drainage rate	mm d ^{−1}	0.010	1000.000	✓	✓	✓	✓	–	✓
c	Drainage exponent	–	1.000	20.000	✓	✓	✓	✓	–	✓
α	Drainage multiplier for the lower layer	–	1.000	250.000	–	–	–	–	✓	–
ψ	Drainage exponent for the lower layer	–	1.000	5.000	–	–	–	–	✓	–
κ	Fraction of drainage to tension storage in the lower layer	–	0.050	0.950	–	–	✓	–	✓	✓
k_i	Interflow rate	mm d ^{−1}	0.010	1000.000	–	–	✓	✓	–	✓
k_s	Base flow rate	mm d ^{−1}	0.001	1000.000	–	✓	–	✓	–	–
n	Base flow exponent	–	1.000	10.000	–	✓	–	✓	–	–
ν	Base flow depletion rate for single reservoir	d ^{−1}	0.001	0.250	✓	–	–	–	–	–
ν_A	Base flow depletion rate for primary reservoir	d ^{−1}	0.001	0.250	–	–	✓	–	✓	✓
ν_B	Base flow depletion rate for secondary reservoir	d ^{−1}	0.001	0.250	–	–	✓	–	✓	✓
$A_{c,\max}$	Maximum saturated area (fraction)	–	0.050	0.950	–	–	–	✓	–	–
b	ARNO/VIC “b” exponent	–	0.001	3.000	✓	–	✓	–	✓	✓
λ	Mean of the log-transformed topographic index distribution	m	5.000	10.000	–	✓ ^a	–	–	–	–
χ	Shape parameter defining the topographic index distribution	–	2.000	5.000	–	✓ ^a	–	–	–	–
μ_τ	Time delay in runoff	day	0.010	5.000	✓	✓	✓	✓	✓	✓

^aThe probability distribution of the TOPMODEL topographic index [Beven and Kirkby, 1979; Beven, 1997] can be defined using a parametric probability distribution, which requires at least two additional model parameters. The mean and shape parameters used here are for the Gamma distribution [Clark *et al.*, 2008b].

solutions of the same order of accuracy. Empirical analysis suggests that the approach remains reliable, with numerical errors (estimated by comparison to a near-exact solution) uniformly constrained throughout the simulation time period, and with a reduction in the tolerance leading to a corresponding reduction in the numerical errors.

Appendix C: FUSE State Variables, Fluxes, and Parameters

[128] The FUSE state variables and fluxes are listed in Table C1, and the FUSE model parameters are listed in Table C2.

Appendix D: Common FUSE Components

D1. Overflow Fluxes

[129] Bucket overflow fluxes q_{utof} , q_{ufof} , q_{stof} , q_{sfof} , q_{sfofa} , and q_{sfofb} (see Appendix C for definitions) are computed in FUSE using smoothed functions. For example,

$$q_{\text{utof}} = q_{\text{in}} \Upsilon \left(S_1^{\text{TENS}} | S_{1,\max}^{\text{TENS}}, \omega \right), \quad (\text{D1})$$

where q_{utof} is the overflow from the unsaturated storage (described by state variable S_1^{TENS} with upper bound $S_{1,\max}^{\text{TENS}}$), q_{in} is the inflow and $\Upsilon(\cdot, \omega)$ is a smooth function controlling the fraction of inflow that overflows. Note that the infiltration q_{in} into a given store depends both on the model architecture and on the specific selection of flux functions (see Table 2).

[130] A logistic smoothing function $\Upsilon(S | S_{\max}, \omega)$ is used,

$$\Upsilon(S | S_{\max}, \omega) = \left[1 + \exp \left(\frac{S - S_{\max}}{\omega} \right) \right]^{-1}, \quad (\text{D2})$$

where ω is a smoothing parameter. Equation (D2) is further rescaled to range exactly between 0 and $S_{\max}$.

D2. Time Delay

[131] The runoff delay due to basin routing is represented using a Gamma distribution

$$P(\tau | a, \mu_\tau) = \frac{\gamma(a, x)}{\Gamma(a)}, \quad (\text{D3})$$

where τ is the time delay, $\gamma(a, x)$ is the incomplete Gamma function [Press *et al.*, 1992], a is a shape parameter, μ_τ is the mean time delay parameter, and $x = \tau a / \mu_\tau$ is a scaled time variable. Equation (D3) is used to estimate the fraction of runoff in a future time step $t + \tau$. We fix $a = 3$ and only calibrate μ_τ .

[132] **Acknowledgments.** We are grateful to Ross Woods for the MARVEX data and to Yun Duan for the MOPEX data. We also thank Philip Binning, Daniel Collins, James Craig, Richard Ibbitt, Bethanna Jackson, Hilary McMillan, and Scott W. Sloan for useful discussions at various stages of the study, and the three reviewers, including Jasper Vrugt, for their insightful comments and suggestions. This work was funded by the New Zealand Foundation for Research Science and Technology (contract CO1X0812), the National Aeronautic and Space Administration (contract NNG06GH10G), and the National Oceanic and Atmospheric Administration (contract NA06OAR4310065).

References

- Ascher, U. M., and L. Petzold (1998), *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia, Pa.
- Bandaragoda, C., D. G. Tarboton, and R. Woods (2004) Application of TOPNET in the distributed model intercomparison project, *J. Hydrol.*, 298(1–4), 178–201.
- Beven, K. (1997), TOPMODEL: A critique, *Hydrol. Processes*, 11(9), 1069–1085.
- Beven, K. (2001), *Rainfall-Runoff Modelling: The Primer*, 360 pp., John Wiley, Chichester, U. K.

- Beven, K., and M. J. Kirkby (1979), A physically based, variable contributing area model of basin hydrology, *Hydrol. Sci. Bull.*, 24, 43–69.
- Binning, P. J., and M. A. Celia (2008), Pseudokinetics arising from the upscaling of geochemical equilibrium, *Water Resour. Res.*, 44, W07410, doi:10.1029/2007WR006147.
- Blöschl, G., C. Reszler, and J. Komma (2008), A spatially distributed flash flood forecasting model, *Environ. Modell. Software*, 23(4), 464–478.
- Burnash, R. J. C. (1995), The NWS River Forecast System—Catchment modeling, in *Computer Models of Watershed Hydrology*, edited by V. P. Singh, pp. 311–366, Water Resour. Publ., Highlands Ranch, Colo.
- Butcher, J. (2008), *Numerical Methods for Ordinary Differential Equations*, 2nd ed., Wiley.
- Cao, Y., S. Li, and L. Petzold (2002), Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software, *J. Comput. Appl. Math.*, 149, 171–191.
- Clark, M. P., D. E. Rupp, R. A. Woods, X. Zheng, R. P. Ibbitt, A. G. Slater, J. Schmidt, and M. J. Uddstrom (2008a), Hydrological data assimilation with the ensemble Kalman filter: Use of streamflow observations to update states in a distributed hydrological model, *Adv. Water Resour.*, 31(10), 1309–1324.
- Clark, M. P., A. G. Slater, D. E. Rupp, R. A. Woods, J. A. Vrugt, H. V. Gupta, T. Wagener, and L. E. Hay (2008b), Framework for Understanding Structural Errors (FUSE): A modular framework to diagnose differences between hydrological models, *Water Resour. Res.*, 44, W00B02, doi:10.1029/2007WR006735.
- COMSOL (2005), COMSOL multiphysics users guide, Kongens Lyngby, Denmark.
- Duan, Q., et al. (2006), Model Parameter Estimation Experiment (MOPEX): An overview of science strategy and major results from the second and third workshops, *J. Hydrol.*, 320(1–2), 3–17.
- Duan, Q. Y., S. Sorooshian, and V. Gupta (1992), Effective and efficient global optimization for conceptual rainfall-runoff models, *Water Resour. Res.*, 28(4), 1015–1031.
- Fenicia, F., J. J. McDonnell, and H. H. G. Savenije (2008), Learning from model improvement: On the contribution of complementary data to process understanding, *Water Resour. Res.*, 44, W06419, doi:10.1029/2007WR006386.
- Geiser, J. (2010), Consistency of iterative operator-splitting methods: Theory and applications, *Numer. Methods Partial Differ. Equations*, 26, 135–158.
- Hairer, E., and G. Wanner (1991), *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, 2nd revised ed., Springer, New York.
- Hairer, E., S. P. Nørsett, and G. Wanner (1993), *Solving Ordinary Differential Equations I: Nonstiff Problems*, 2nd revised ed., Springer, New York.
- Harbaugh, A. W. (2005), MODFLOW-2005, the U.S. Geological Survey modular ground-water model—The ground-water flow process, *U.S. Geol. Surv. Tech. Methods*, Book 6, Chap. A16.
- Ibbitt, R., and R. Woods (2004), Re-scaling the topographic index to improve the representation of physical processes in catchment models, *J. Hydrol.*, 293(1–4), 205–218.
- Kahaner, D., C. Moler, and S. Nash (1989), *Numerical Methods and Software*, Prentice Hall, Englewood Cliffs, N. J.
- Kavetski, D., and M. P. Clark (2010), The ancient numerical demons of conceptual hydrological modeling: 2. Impact of time stepping scheme on model analysis and prediction, *Water Resour. Res.*, 46, W10511, doi:10.1029/2009WR008896.
- Kavetski, D., and G. Kuczera (2007), Model smoothing strategies to remove microscale discontinuities and spurious secondary optima in objective functions in hydrological calibration, *Water Resour. Res.*, 43, W03411, doi:10.1029/2006WR005195.
- Kavetski, D., P. Binning, and S. W. Sloan (2001), Adaptive time stepping and error control in a mass conservative numerical solution of the mixed form of Richards equation, *Adv. Water Resour.*, 24(6), 595–605.
- Kavetski, D., P. Binning, and S. W. Sloan (2002a), Noniterative time stepping schemes with adaptive truncation error control for the solution of Richards equation, *Water Resour. Res.*, 38(10), 1211, doi:10.1029/2001WR000720.
- Kavetski, D., P. Binning, and S. W. Sloan (2002b), Adaptive backward Euler time stepping with truncation error control for numerical modelling of unsaturated fluid flow, *Int. J. Numer. Methods Eng.*, 53(6), 1301–1322.
- Kavetski, D., G. Kuczera, and S. W. Franks (2003), Semidistributed hydrological modeling: A “saturation path” perspective on TOPMODEL and VIC, *Water Resour. Res.*, 39(9), 1246, doi:10.1029/2003WR002122.
- Kavetski, D., G. Kuczera, and S. W. Franks (2006), Calibration of conceptual hydrological models revisited: 1. Overcoming numerical artefacts, *J. Hydrol.*, 320, 173–186.
- Kling, H., and H. Gupta (2009), On the development of regionalization relationships for lumped watershed models: The impact of ignoring sub-basin scale variability, *J. Hydrol.*, 373, 337–351.
- Lambert, J. D. (1991), *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*, John Wiley, Chichester, U. K.
- Leavesley, G., R. W. Lichty, B. M. Troutman, and L. G. Saindon (1983), Precipitation-runoff modeling system—User’s manual, *U.S. Geol. Surv. Water Resour. Invest. Rep.*, 83-4238, 207 pp.
- LeVeque, R. J. (1990), *Numerical Methods for Conservation Laws*, Birkhauser, Basel.
- Lindström, G., B. Johansson, M. Persson, M. Gardelin, and S. Bergström (1997), Development and test of the distributed HBV-96 hydrological model, *J. Hydrol.*, 201, 272–288.
- Miller, C. T., G. A. Williams, C. T. Kelley, and M. D. Tocci (1998) Robust solution of Richards’ equation for nonuniform porous media, *Water Resour. Res.*, 34, 2599–2610.
- Nocedal, J., and S. J. Wright (1999), *Numerical Optimization*, Springer, New York.
- Parker, S. P. (1984), *McGraw-Hill Dictionary of Scientific and Technical Terms*, 3rd ed., 1781 pp., McGraw-Hill, New York.
- Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (1992), *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, 2nd ed., Cambridge Univ. Press, Cambridge, U. K.
- Reggiani, P., and T. H. M. Rientjes (2005), Flux parameterization in the representative elementary watershed approach: Application to a natural basin, *Water Resour. Res.*, 41, W04013, doi:10.1029/2004WR003693.
- Schlumberger Information Systems (2004), Eclipse technical description, Houston, Tex.
- Schoups, G., J. A. Vrugt, F. Fenicia, and N. C. van de Giesen (2010), Inaccurate numerical implementation of conceptual hydrologic models corrupts accuracy and efficiency of MCMC simulation, *Water Resour. Res.*, doi:10.1029/2009WR008648, in press.
- Shampine, L. F. (1994), *Numerical Solution of Ordinary Differential Equations*, Chapman and Hall, New York.
- Shampine, L. F. (2007), Design of software for ODEs, *J. Comput. Appl. Maths.*, 205(2), 901–911.
- Shampine, L. F., and M. W. Reichelt (1997), The MATLAB ODE suite, *Siam J. Sci. Comput.*, 18(1), 1–22.
- Shampine, L. F., S. Thompson, J. A. Kierzenka, and G. D. Byrne (2005) Non-negative solutions of ODEs, *Appl. Math. Comput.*, 170(1), 556–569.
- Singh, V. P., and D. K. Frevert (2002a), *Mathematical Models of Small Watershed Hydrology and Applications*, 950 pp., Water Resour. Publ., Highlands Ranch, Colo.
- Singh, V. P., and D. K. Frevert (2002b), *Mathematical Models of Large Watershed Hydrology*, 891 pp., Water Resour. Publ., Highlands Ranch, Colo.
- Sivapalan, M., J. K. Ruprecht, and N. R. Viney (1996), Water and salt balance modelling to predict the effects of land-use changes in forested catchments. 1. Small catchment water balance model, *Hydrol. Processes*, 10(3), 393–411.
- Smooke, M. D. (1983), Error estimate for the modified Newton method with applications to the solution of nonlinear, two-point boundary-value problems, *J. Optim. Theory App.*, 39(4), 489–511.
- Steeffel, C. I., and K. T. B. MacQuarrie (1996), Approaches to modeling of reactive transport in porous media, in *Reactive Transport in Porous Media*, p. 181, Mineral. Soc. of Am., Washington, D. C.
- Tague, C. L., and L. E. Band (2004), RHESys: Regional Hydro-Ecologic Simulation System—An object-oriented approach to spatially distributed modeling of carbon, water, and nutrient cycling, *Earth Inter.*, 8(19), 1–42.
- Tait, A., and R. Woods (2007), Spatial interpolation of daily potential evapotranspiration for New Zealand using a spline model, *J. Hydrometeorol.*, 8(3), 430–438.
- Vache, K. B., and J. J. McDonnell (2006), A process-based rejectionist framework for evaluating catchment runoff model structure, *Water Resour. Res.*, 42, W02409, doi:10.1029/2005WR004247.
- van Werkhoven, K., T. Wagener, P. Reed, and Y. Tang (2008), Characterization of watershed model behavior across a hydroclimatic gradient, *Water Resour. Res.*, 44, W01429, doi:10.1029/2007WR006271.
- Visual Numerics (2007), IMSL Fortran Numerical Math Library, version 6., Houston, Tex.

- Weerts, A. H., and G. Y. H. El Serafy (2006), Particle filtering and ensemble Kalman filtering for state updating with hydrological conceptual rainfall-runoff models, *Water Resour. Res.*, 42, W09403, doi:10.1029/2005WR004093.
- Whitehead, P. G., E. J. Wilson, and D. Butterfield (1998), A semi-distributed Integrated Nitrogen model for multiple source assessment in Catchments (INCA): Part I: Model structure and process equations, *Sci. Total Environ.*, 210/211, 547–558.
- Wigmosta, M. S., L. W. Vail, and D. P. Lettenmaier (1994), A distributed hydrology-vegetation model for complex terrain, *Water Resour. Res.*, 30(6), 1665–1679.
- Wood, E. F., D. P. Lettenmaier, and V. G. Zartarian (1992), A land-surface hydrology parameterization with subgrid variability for general circulation models, *J. Geophys. Res.*, 97(D3), 2717–2728.
- Wood, W. L. (1990), *Practical Time-Stepping Schemes*, Oxford Univ. Press, Oxford, U. K.
- Woods, R. A., R. B. Grayson, A. W. Western, M. J. Duncan, D. J. Wilson, R. I. Young, R. P. Ibbitt, R. D. Henderson, and T. A. McMahon (2001), Experimental Design and Initial Results from the Mahurangi River Variability Experiment: MARVEX, in *Observations and Modeling of Land Surface Hydrological Processes*, edited by V. Lakshmi et al., pp. 201–213, AGU, Washington, D. C.
-
- M. P. Clark, National Center for Atmospheric Research, PO Box 3000, Boulder, CO 80307-3000, USA. (mclark@ucar.edu)
- D. Kavetski, Environmental Engineering, University of Newcastle, Callaghan, NSW 2308, Australia.