Advances in Water Resources 53 (2013) 109–117

Contents lists available at SciVerse ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

A terrain-following grid transform and preconditioner for parallel, large-scale, integrated hydrologic modeling

Reed M. Maxwell

Department of Geology and Geological Engineering and Integrated GroundWater Modeling Center, Colorado School of Mines, Golden, CO, USA

ARTICLE INFO

Article history:
Received 14 March 2012
Received in revised form 29 September 2012
Accepted 2 October 2012
Available online 17 October 2012

Keywords:
Integrated hydrologic modeling
Parallel computing
Solvers

ABSTRACT

A terrain-following grid formulation (TFG) is presented for simulation of coupled variably-saturated subsurface and surface water flow. The TFG is introduced into the integrated hydrologic model, ParFlow, which uses an implicit, Newton Krylov solution technique. The analytical Jacobian is also formulated and presented and both the diagonal and non-symmetric terms are used to precondition the Krylov linear system. The new formulation is verified against an orthogonal stencil and is shown to provide increased accuracy at lower lateral spatial discretization for hillslope simulations. Using TFG, efficient scaling to a large number of processors (16,384) and a large domain size (8.1 Billion unknowns) is shown. This demonstrates the applicability of this formulation to high-resolution, large-spatial extent hydrology applications where topographic effects are important. Furthermore, cases where the analytical Jacobian is used for the Newton iteration and as a non-symmetric preconditioner for the linear system are shown to have faster computation times and better scaling. This demonstrates the importance of solver efficiency in parallel scaling through the use of an appropriate preconditioner.

1. Introduction

A compelling hydrologic challenge is to simulate the integrated, hydrologic cycle at large extents and high resolution (e.g., [8,17,28]); to understand scaling of processes and to provide insights into large-scale systems at high resolution. Integrated hydrologic models that solve coupled surface and subsurface flow equations are becoming more commonly available [5,12,15,16,18,21,22,27]; and show great promise in addressing these challenges.

Parallel, or high-performance, computing techniques have significant potential for understanding the scaling of processes using so-called hyperresolution simulations in hydrology [17]. Fully-implicit formulations have several advantages over other approaches. Notably, they solve all governing equations in a global manner over one time step fully capturing nonlinear dynamics and allowing for very finely-resolved domains that do not suffer from a Courant time step limitation. However, these techniques require appropriate preconditioners to provide accurate solution and efficient parallel scaling [2,11] particularly to very large problem size and a large number of compute cores.

The parallel simulation platform, ParFlow [2,11,15,19] exhibits many of the characteristics needed to simulate large extent, high-resolution domains. One limitation is the ability to efficiently and accurately solve complex terrain problems with fine spatial discretization near the ground surface and an optimal number of unknowns. To move toward this goal, certain formulations of the governing equations for integrated surface and subsurface flow have advantages over their orthogonal counterparts.

In this work I develop a new, terrain-following grid (TFG) transform for unsaturated subsurface flow using the Richards' formulation. This is formulated in an implicit solution framework using a Newton-type approach. I develop the analytical Jacobian for this system and discuss the use of the symmetric and non-symmetric terms of this Jacobian as a preconditioner for the linear system. The new formulation is verified against an orthogonal formulation. I demonstrate the improved solver performance using a combination of the analytical Jacobian and the nonsymmetric preconditioner. Finally, I demonstrate efficient parallel scaling of this new formulation out to a large number of processors (16,384) and show that key efficiency gains result from the non-symmetric linear preconditioner developed here.

2. Methodology

Complete details of the approach used to solve for coupled surface-subsurface flow in ParFlow are given in Kollet and Maxwell [15]. The Newton–Krylov formulation for Richards' equation is presented in Jones and Woodward [11]. Newton Krylov approaches to solving nonlinear partial differential equations have been in use for quite some time (e.g., [3]) and are now widely applied across the sciences (e.g., [13]). Here, a brief summary of the equations solved

E-mail address: rmaxwell@mines.edu

0309-1708/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.advwatres.2012.10.001

in regular, orthogonal coordinates is used as the starting point for the terrain-following formulation.

ParFlow solves the Richards equation for variably saturated flow [23] in three spatial dimensions given as:

$$S_s S_w \frac{\partial h}{\partial t} + \phi \frac{\partial S_w(h)}{\partial t} = \nabla \cdot \mathbf{q} + q_r(x,z) \tag{1}$$

where

$$\mathbf{q} = \phi S_w(h)\mathbf{v} = -\mathbf{K}_s(x) k_r(h) \nabla(h+z) \tag{2}$$

In these expressions, h is the pressure-head [L]; z is the elevation with the z-axis specified as upward [L]; $\mathbf{K}_s(\mathbf{x})$ is the saturated hydraulic conductivity tensor [LT^{-1}]; k_r is the relative permeability [–]; S_s is the specific storage [L^{-1}]; ϕ is the porosity [–]; S_w is the relative saturation [–]; and q_r is a general source/sink term that represents transpiration, wells, and other fluxes [LT^{-1}]. The subsurface flow velocity is denoted by $\mathbf{v}$ [LT^{-1}] and the specific volumetric (Darcy) flux is denoted by $\mathbf{q}$ [LT^{-1}]. Note here that we assume that density and viscosity are both constant, though this assumption is not necessarily needed in ParFlow [14].

The van Genuchten [26] relationships are used to describe the relative saturation and permeability functions:

$$S_w(h) = \frac{S_{sat} - S_{res}}{(1 + (\alpha h))^{(1-1/n)}} + S_{res} \tag{3}$$

$$k_r(h) = \frac{\left(1 - \frac{(\alpha h)^{n-1}}{(1+(\alpha h)^n)^{1-1/n}}\right)}{(1 + (\alpha h)^\beta)^{\frac{(1-1/n)}{2}}} \tag{4}$$

where α [L^{-1}] and n [–] are soil parameters, the values S_{sat} [–] is the relative saturated water content (usually set to unity) and the value S_{res} [–] is the relative residual saturation.

Overland flow is represented in ParFlow by the two-dimensional kinematic wave equation included as the overland flow boundary condition resulting from application of continuity conditions for pressure and flux:

$$\mathbf{k}(-\mathbf{K}_s(\mathbf{x}) k_r(h) \nabla(h+z)) = \frac{\partial \|h,0\|}{\partial t} - \nabla \cdot \|h,0\| \mathbf{v}^{sw} + q_r(\mathbf{x}) \tag{5}$$

where $\mathbf{v}^{sw}$ is the two-dimensional, depth-averaged surface water velocity [LT^{-1}]; h is the surface ponding depth [L] equivalent now to h as used in Eq. (1) as follows: if $h > 0$; $q_r(\boldsymbol{x})$ is a general source/sink (e.g., rainfall, ET) rate [LT^{-1}] that is also equivalent to the source/sink term shown in Eq. (1) at the ground surface; and $\mathbf{k}$ is the unit vector in the vertical, again defining positive upward coordinates. Note that $\|h,0\|$ indicates the greater value of the two quantities, that the left hand side of Eq. (5) represents vertical fluxes (e.g., in/exfiltration) across the land surface boundary. The overland flow condition assumes that pressure h represents both surface pressure and the ponding depth at the ground surface under saturated conditions [15] and if h is greater than zero k_r will be one.

3. Topography and the land surface

The upper portion of Fig. 1 demonstrates the standard manner in which ParFlow represents topography or any other non-rectangular domain boundaries. ParFlow uses a structured grid, and the land surface representation (e.g., topography) must be mapped onto this grid. This is done using an octree technique where the topographic surface is mapped onto the orthogonal grid and looping structures that encompass these irregular shapes are constructed. The cells above the land surface are inactive and are not included in the solution (but are still stored in the solution vector). This corresponds to the white, inactive region in Fig. 1. Overland flow is activated at any time in which the pressure head at the land-surface (the top of the active zone cells in Fig. 1) is greater than zero. In Eq. (5), the two terms that represent the storage and continuity of water then become active. This can be due to excess saturation or excess infiltration, corresponding to a saturation of unity in Eq. (3), and allows simultaneous solution of the Richards and kinematic wave equations.

Manning's equation is used to establish a flow depth-discharge relationship, where the $\mathbf{v}^{sw}$ in (5), as follows:

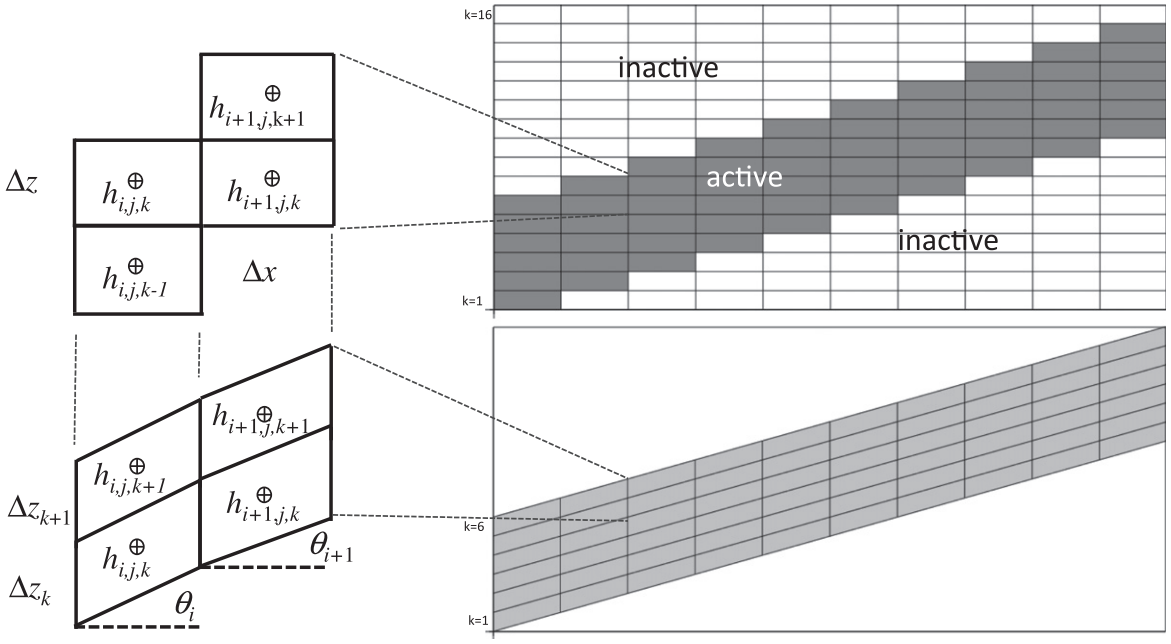

Fig. 1. Depiction of orthogonal (upper) and the terrain following (lower) grid formulations for a topographic land-surface and (left) schematics of the associated finite difference stencils. Note in this figure (and in the formulation presented) that i, j, and k are the x, y, and z cell indices, respectively and that the limits of the domain would be at $i = nx$, $j = ny$ and $k = nz$. Note also that nz is different (greater) in the orthogonal formulations (upper) than in the terrain formulation (lower) due to the difference in inactive cells between formulations.

Table 1
List of parameters for the hillslope verification simulation.

Parameter	Value	Units
Return flow case: common parameters		
Length	100	m
Width	1	m
Thickness	3	m
Δx	10, 5, 1	m
Δy	1	m
Δz	0.05	m
nx	10, 20, 100	
ny	1	
K_{sat}	10	m/h
S_s	1.00E−05	1/m
Δt	0.1	h
Total time	120	h
vanGenuchten α	6	1/m
vanGenuchten n	2	
S_{res}	0.2	
Slope	10, 5	%
Manning's n	1.00E−06	$m^{1/3}/h$
Initial WT depth	2	m
Porosity	0.1	%
Orthogonal parameters		
nz	300, 160	
Terrain parameters		
nz	60	

$$v_x^{sw} = \frac{\sqrt{S_{f,x}}}{n} h^{2/3} \quad \text{and} \quad v_y^{sw} = \frac{\sqrt{S_{f,y}}}{n} h^{2/3} \tag{6}$$

where $S_{f,i}$ [L] is the friction slope, i stands for the x- and y-direction and n [$TL^{-1/3}$] is Manning's coefficient. The friction slope is often characterized as the topographic (or bed) slope, S_0. If the change in pressure in the x and y directions is neglected the friction slope reduces to $S_{f,x} = S_{0,x} + \frac{\partial h}{\partial x}, \frac{\partial h}{\partial x} \approx o$, and Eq. (6) becomes $v_x^{sw} = \frac{\sqrt{S_{0x}}}{n} h^{2/3}$ which is the commonly written form of the kinematic approximation.

Here, the orthogonal formulation is extended to allow for topographic extent without using inactive zones by applying a terrain-following transform. An approach similar to previous formulations (after, e.g., [6,24,4,1,7]) is used to include gravity contributions into the Darcy flux term. In this formulation, the Darcy flux (Eq. (2)) is modified to follow the changes in slope *locally* as depicted in the lower portion of Fig. 1. This transform assumes that the change in slope is uniform over the vertical (that all cells slope the same amount in z) and that there is a continuity of flux between adjacent, sloped, cell faces only. The approach used here neglects non-orthogonal flux contributions and maintains local mass conservation. This slightly more simplified than previous fully-unstructured approaches (which still may maintain local mass conservation depending on approach) that do include these non-orthogonal terms (e.g., [1,7]). This results in a modified Darcy flux as follows

$$\mathbf{q} = -\mathbf{K}_s(\mathbf{x}) k_r(h) [\nabla(h+z)\cos\theta_x + \sin\theta_x] \tag{7}$$

where θ [–] is the local angle of slope, in the x and y directions and may be written as

$$\theta_x = \tan^{-1}(S_{0x}) \quad \text{and} \quad \theta_y = \tan^{-1}(S_{0y})$$

Recognizing that this will change the fluxes in the x and y directions, the flux in x, for example, may be written as

$$\begin{aligned} q_x &= -K_{s,x}(\mathbf{x}) k_r(h) \left[\frac{\partial(h)}{\partial x} \cos\theta_x + \sin\theta_x \right] \\ &= -K_{s,x}(x) k_r(h) \frac{\partial(h)}{\partial x} \cos\theta_x - K_{s,x}(x) k_r(h) \sin\theta_x \end{aligned} \tag{8}$$

where for a flat surface, $S_{0,x} = 0$ and $\theta_x = 0$, Eq. (8) reduces to Eq. (2).

For the Richards flow solution, ParFlow employs an implicit backward Euler scheme in time, and a cell-centered finite-difference scheme in space. The discretization of the new formulation can be written in the context of the function evaluation for the linearization in the Newton iteration

$$\begin{aligned} F(h_{i,j,k}^n) = {} & \Delta x_i \Delta y_i \Delta y_i \Delta z_k S_{s,i,j,k} S_w(h)_{i,j,k}^n (h_{i,j,k}^n - h_{i,j,k}^{n-1}) + \Delta x_i \Delta y_i \Delta z_k \phi_{i,j,k} [S_w(h)_{i,j,k}^n \\ & - S_w(h)_{i,j,k}^{n-1}] - \Delta t^n \Delta x_i \Delta y_j \Delta z_k q_{r,i,j,k}^n \\ & - \Delta t^n \Delta x_i \Delta y_j \Delta z_k \left[\frac{q_{i+1/2,j,k}^x - q_{i-1/2,j,k}^x}{\Delta x_i} + \frac{q_{i,j+1/2,k}^y - q_{i,j-1/2,k}^y}{\Delta y_j} + \frac{q_{i,j,k+1/2}^z - q_{i,j,k-1/2}^z}{\Delta z_k} \right] \end{aligned} \tag{9}$$

where the flux, q, is now written as

$$\begin{aligned} q_{i+1/2,j,k}^x = {} & -[\cos\theta_x K_{s,x} k_r(h)]_{i+1/2,j,k} \left[\frac{h_{i+1,j,k}^n - h_{i,j,k}^n}{\Delta x_{i+1/2}} \right] \\ & - [\sin\theta_x K_{s,x} k_r(h)]_{i+1/2,j,k} \end{aligned} \tag{10}$$

At the cell interfaces, the harmonic averages of the saturated hydraulic conductivities and a one-point upstream weighting of the relative permeabilities are used based upon the direction of total flux as in Eq. (10). For the additional terms introduced in this equation, the sine or cosine of θ at adjacent cells are arithmetically averaged to provide an interface value. This assumes that the slope, θ, is at the cell center. Slopes formulated by other approaches, for example the first-order upwind difference in elevation between two adjacent cells, would directly yield interface values. While this is shown only for the x-direction, the development of these terms in the y-direction is entirely equivalent. Note here that for the vertical fluxes, terrain terms and the gradient of the hydrostatic components appear as is written in Eq. (2) and the terms resulting from the TFG are neglected.

Next, an analytical Jacobian for the new system is developed. The analytical Jacobian can be used in the Newton iteration (in place of a finite-difference approximation of the Jacobian) and may also be used to form a preconditioner for the linear system [11] as described below. The Jacobian for the linear system in Eq. (9) can be written as:

Table 2
List of parameters for the tilted-v catchment verification simulation.

Parameter	Value	Units
Tilted-v case: common parameters		
Length	100	m
Width	110	m
Thickness	5	m
Δx	10, 1	m
Δy	10, 1	m
Δz	0.5	m
nx	11, 110	
ny	11, 110	
K_{sat}	10	m/h
S_s	1.00E−05	1/m
Δt	Variable	h
Total time	120	h
vanGenuchten α	6	1/m
vanGenuchten n	2	
S_{res}	0.2	
Channel slope:	5	%
Hillslope:	±10	%
Manning's n	1.00E−06	$m^{1/3}/h$
Initial WT depth	2	m
Porosity	0.1	%
Orthogonal parameters		
nz	30	
Terrain parameters		
nz	10	

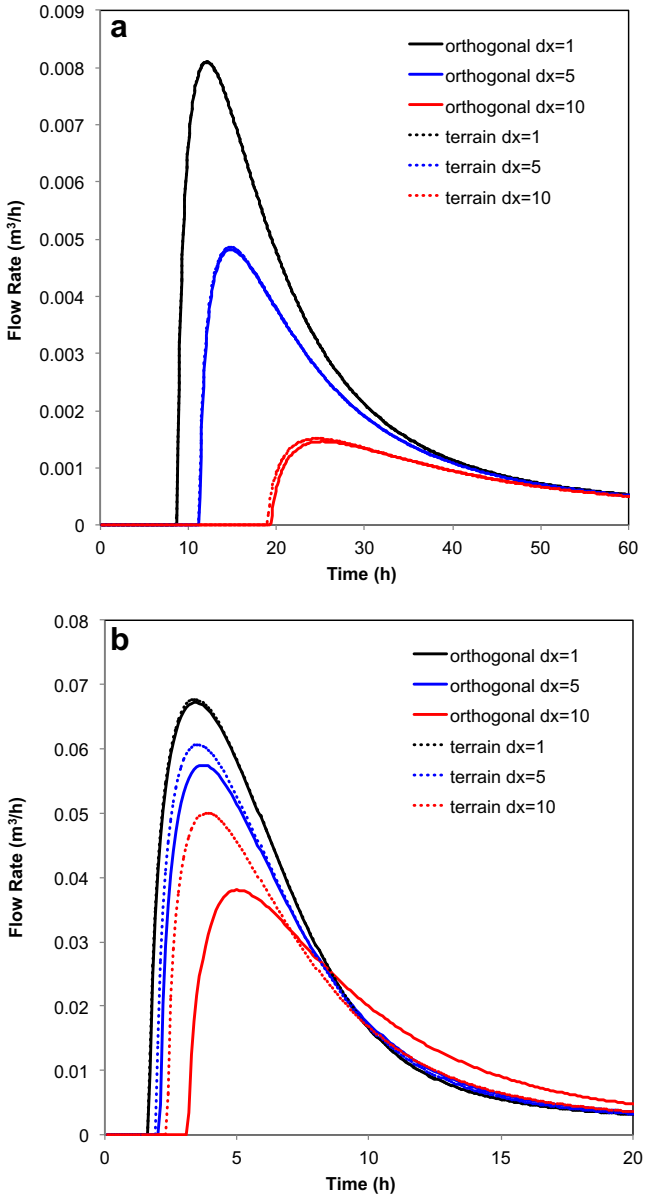

Fig. 2. Plot of outflow for (a) a 5% hillslope and (b) a 10% hillslope for the verification case. Each plot compares the orthogonal (solid lines) and terrain following grids (dashed lines) at three lateral spatial resolutions as shown. Note that each panel has different x and y scales.

$$\frac{\partial F\left(h^n_{i,j,k}\right)}{\partial h^n_{i,j,k}} = \Delta x_i \Delta y_j \Delta z_k S_{s,i,j,k}\left[\frac{\partial S_w(h)^n_{i,j,k}}{\partial h^n_{i,j,k}} h^n_{i,j,k} + S_w(h)^n_{i,j,k}\right] + \Delta x_i \Delta y_j \Delta z_k \phi_{i,j,k}\left[\frac{\partial S_w(h)^n_{i,j,k}}{\partial h^n_{i,j,k}}\right] - \Delta t^n \Delta x_i \Delta y_j \Delta z_k \left[\frac{\frac{\partial q^x_{i-1/2,j,k}}{\partial h^n_{i,j,k}} - \frac{\partial q^x_{i-1/2,j,k}}{\partial h^n_{i,j,k}}}{\Delta x_i} + \frac{\frac{\partial q^y_{i,j+1/2,k}}{\partial h^n_{i,j,k}} - \frac{\partial q^y_{i,j-1/2,k}}{\partial h^n_{i,j,k}}}{\Delta y_j} + \frac{\frac{\partial q^z_{i,j,k+1/2}}{\partial h^n_{i,j,k}} - \frac{\partial q^z_{i,j,k-1/2}}{\partial h^n_{i,j,k}}}{\Delta z_k}\right] \tag{11}$$

where the corresponding derivative of **q** is:

$$\frac{\partial q^x_{i+1/2,j,k}}{\partial h^n_{i,j,k}} = -\frac{(\cos(\theta_x)K_s)_{i+1/2,j,k}}{\Delta x_{i+1/2}}\left[\frac{\partial k_r(h)_{i+1/2,j,k}}{\partial h^n_{i,j,k}}(h^n_{i+1,j,k} - h^n_{i,j,k}) + k_r(h)_{i+1/2,j,k}\right] - \left(\sin(\theta_x)K_s\frac{\partial k_r(h)}{\partial h^n_{i,j,k}}\right)_{i+1/2,j,k} \tag{12}$$

and the derivatives of the saturation, $\frac{\partial S(h)_{i+1/2,j,k}}{\partial h^n_{i,j,k}}$, and relative permeability, $\frac{\partial k_r(h)_{i+1/2,j,k}}{\partial h^n_{i,j,k}}$, are formed by taking the derivatives of Eqs. (3) and (4) with respect to pressure, respectively.

Eqs. (11) and (12) show us that the terrain-following formulation adds only nonsymmetric terms to the Jacobian. The derivative of flux with respect to pressure at the adjacent node yields:

$$\frac{\partial q^x_{i+1/2,j,k}}{\partial h^n_{i+1,j,k}} = -\frac{(\cos(\theta_x)K_s)_{i+1/2,j,k}}{\Delta x_{i+1/2}}\left[-\frac{\partial k_r(h)_{i+1/2,j,k}}{\partial h^n_{i+1,j,k}}(h^n_{i+1,j,k} - h^n_{i,j,k}) + k_r(h)_{i+1/2,j,k}\right] + \left(\sin(\theta_x)K_s\frac{\partial k_r(h)}{\partial h^n_{i,j,k}}\right)_{i+1/2,j,k} \tag{13}$$

which is quite similar to the result of [11], but here with the new terms from the terrain-following formulation. In this equation, while the middle term on the right hand side of Eq. (13) does not change sign compared to (12), the first and last terms do, adding a non-symmetric component to the Jacobian as the derivatives of the fluxes are formulated from different directions.

4. Verification simulations

Complete details on the formulation and verification of the orthogonal formulation for this approach are provided in Kollet and Maxwell [15] and Sulis et al. [25]. Here, I compare the new terrain-following formulation to the previous orthogonal formulation that uses active and inactive regions to define problem geometries and boundaries. I select two return-flow cases for this verification where the water table is initialized a fixed distance below the ground surface of the hillslope and allowed to move toward equilibrium, ponding and generating runoff. In these cases a uniform aquifer thickness is used to simulate topographically-driven flow between the land surface and an impermeable bedrock layer. These verification cases are (1) a simple, two-dimensional hillslope and (2) a tilted-v catchment. While the tilted-v catchment has been widely used previously as an overland flow benchmark case [15,21,25], here it is modified slightly to represent complicated, three-dimensional return-flow. For both cases, the orthogonal formulation, which uses active and inactive zones, is compared to the new terrain-following formulation. Additionally, the lateral discretization is refined systematically to improve accuracy as Δx goes to zero.

The orthogonal formulation will suffer from inaccuracies if the lateral discretization is too coarse to accurately resolve the slope. However, the terrain-following grid transform assumes lateral flux only along the local slope direction, which can also lead to inaccuracies. As the lateral discretization decreases the inaccuracies in both formulations should diminish, and both solutions should converge to the true hillslope solution. Comparing the two solutions as the discretization approaches zero provides a test of the error introduced by neglecting non-orthogonal contributions in the terrain following formulation. Complete problem details for the two-dimensional hillslope cases are given in Table 1 and for the tilted-v catchment cases in Table 2.

Fig. 2 presents the results of the hillslope verification simulation. This figure plots the outflow for the orthogonal and terrain-following formulations for two topographic slopes at three lateral resolutions. At the 5% slope (Fig. 2a) we see excellent agreement between both formulations (i.e. differences between dashed and solid lines) at the finest spatial resolution. Only at the coarsest lateral discretization do we see any significant differences in outflow generated by the two formulations and these differences are approximately 3% at the time of peak flow.

Fig. 2b plots outflow for the 10% slope case simulations. In this figure a more pronounced difference in the solutions is seen as the

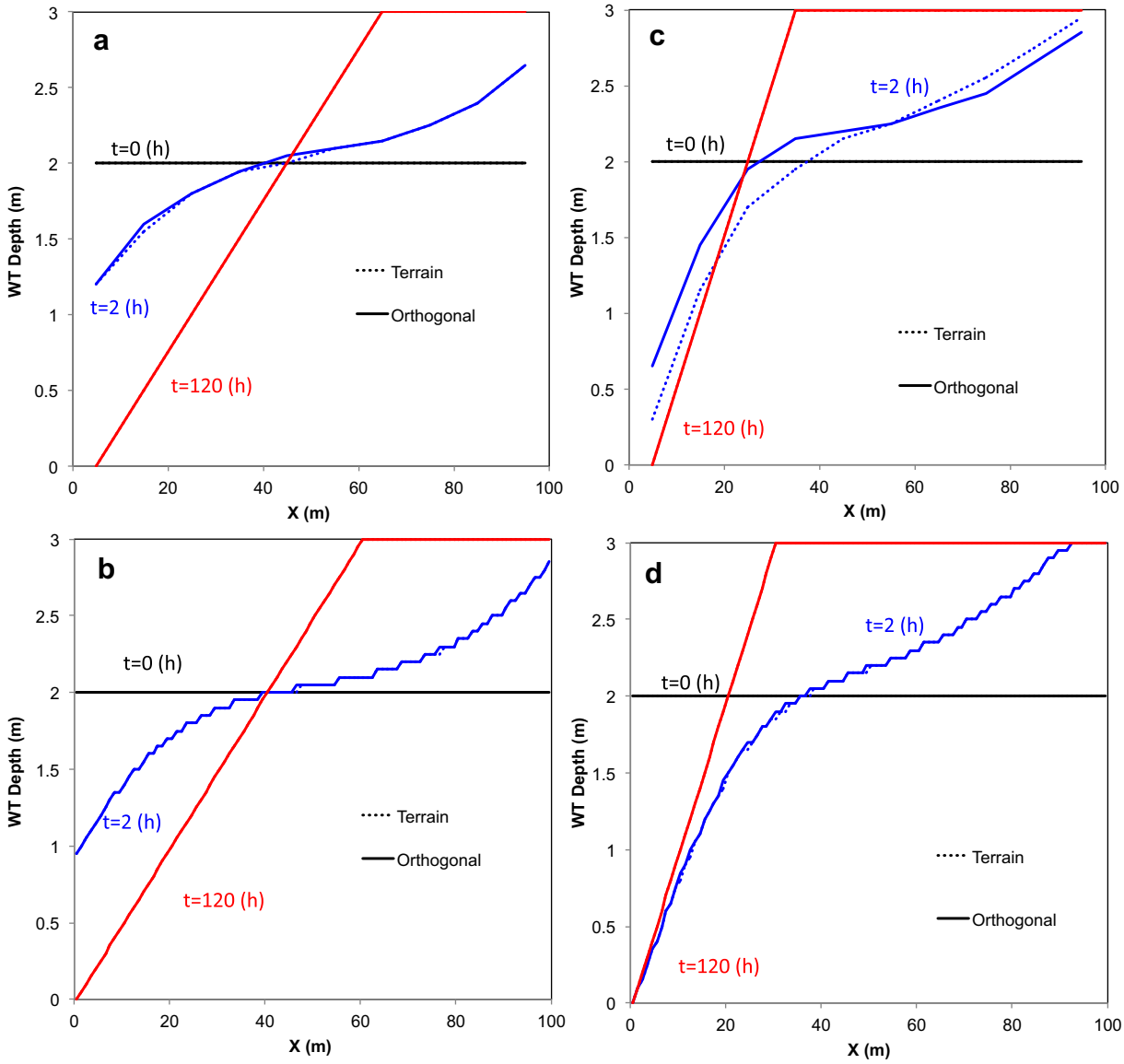

Fig. 3. Plot of water table depth along the *x*-axis for the return flow case. Panels a, b are for the 5% slope and c, d are for the 10% slope. Each plot compares the orthogonal (solid lines) and terrain following grids (dashed lines) at three times as indicated. Plots a, c are for 10 m lateral resolution while b, d are for 1 m lateral resolution. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

lateral discretization is decreased by an order of magnitude. Both formulations still agree closely at the finest spatial resolution, but the errors at coarser resolutions appear to be more significant than seen in Fig. 2a. For this test case, Fig. 2b may be used to make statements regarding the relative contribution of errors in each formulation. If the $\Delta x = 1$ case is assumed to be the most accurate solution for this problem, where both formulations provide close agreement, then this figure shows that the terrain following formulation at coarser lateral discretization more closely approaches the finest scale solution than the orthogonal formulation. This would indicate that, for this problem configuration, the errors in the representation of the problem boundaries at coarse resolutions are greater than the error introduced through the terrain following approximation.

In addition to viewing the convergence of the curves plotted in Fig. 2, the normalized difference in total flow between formulations was also calculated. This normalized difference was −0.004, −0.002 and 0.0007 for the 5% slope cases for $dx = 10$, 5 and 1, respectively and −0.02, −0.009 and 0.00001 for the 10% slope cases for $dx = 10$, 5 and 1, respectively. Decreasing normalized differences with finer spatial discretization further demonstrates the convergence of the two approaches, but also the greater decease in solution error for simulations with larger slopes.

Fig. 3 plots the water table depth evolution over time for this same test problem. Again, we see very good agreement between the orthogonal and terrain-following formulations, particularly at $\Delta x = 1$. The only significant disagreement between the two formulations is again at the coarsest resolution and steepest slope (Fig. 3c). However, it may be noted that the solutions for water table depth agree closely at later times when a quasi-steady behavior is reached and it is only at intermediate times that there are significant differences. Fig. 4 plots a comparison of water table depth at this intermediate time (2 h) for decreases in spatial resolution. Again, if the finest resolution is assumed to be the most accurate solution for the problem, the water table depth predicted by the terrain-following formulation at $\Delta x = 10$ more closely approaches that of $\Delta x = 1$ than the orthogonal case. This would further indicate that the errors incurred in the new

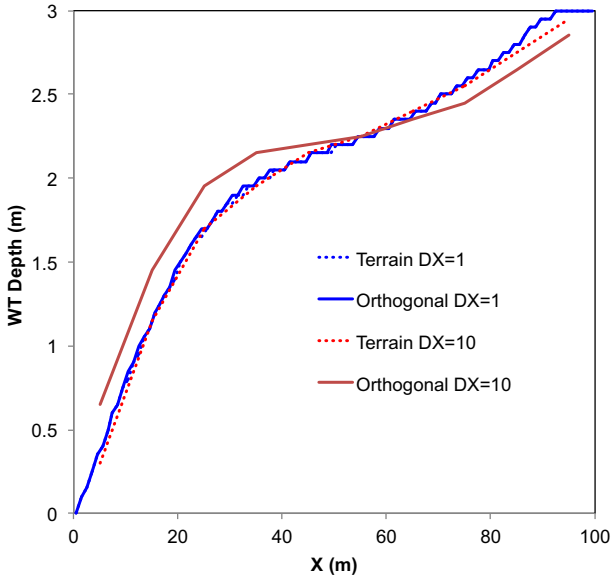

Fig. 4. Comparison of water table depths for two lateral resolutions at 2 h of simulation time for the 10% return flow case.

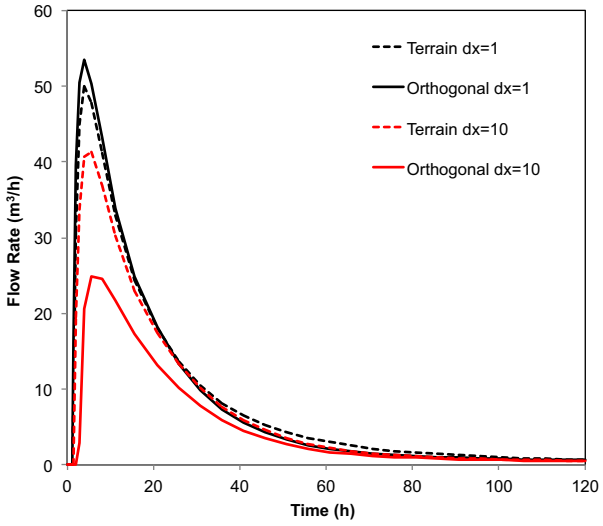

Fig. 5. Plot of outflow for the tilted-v catchment verification case. This plot compares the orthogonal (solid lines) and terrain following grids (dashed lines) at two lateral spatial resolutions as shown.

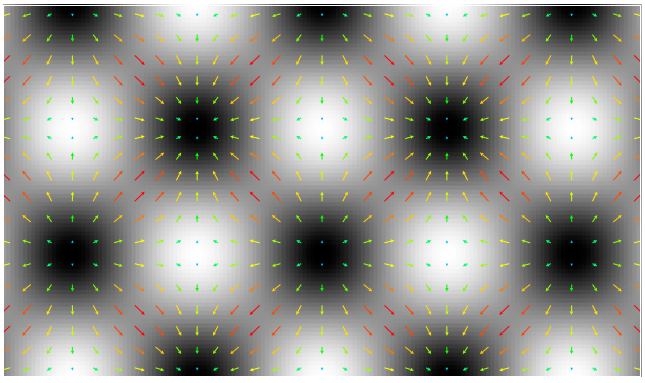

Fig. 6. Schematic of elevation (shaded background, lighter colors are lower elevations) and topographic slope vectors (θ) for the parallel scaling simulation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

formulation are smaller than the errors incurred with a coarse discretization of the topography.

In the interest of providing a more complex test case, the tilted-v catchment was also used to compare the two formulations. This test case has a channel supplied by two hillslopes and generates three-dimensional, coupled, surface and subsurface flow. Fig. 5 plots the results of the tilted-v catchment case. In this figure we again see a disparity between the orthogonal and terrain following representations at coarser resolutions, with agreement at finer lateral resolution. Qualitatively, the results from the tilted-v catchment case are quite similar to the results of the more simple hillslope test cases presented previously.

5. Solver, preconditioner and scaling performance

An advantage of implementing the TFG formulation in the ParFlow simulation platform is the ability to exploit the parallel infrastructure for large-scale simulation. ParFlow is developed for efficient and scalable parallel computing as has been demonstrated previously [2,11,15,17]. Thus it is important to demonstrate that both the new formulation presented here and the solution technique also scale appropriately. Scaled parallel efficiency, E, or a measure of scaled parallel speedup [9] now commonly referred to as *weak scaling* [13] is used to assess the inefficiencies of both the formulation and the approach. Following previous studies [2,11,15,17] a unit problem of size n unknowns is simulated on one processor and both the problem size and the number of processors are increased linearly. Scaled parallel efficiency may then be written as:

$$E = \frac{T(n,1)}{T(p \times n, p)} \tag{14}$$

where T is simulation time, n is the problem size in unknowns and p is the number of processors. Thus for perfect parallel efficiency, the simulation time will remain constant and $E = 1$ for any number of processors. Decreases in parallel efficiency are due to three primary components: hardware, software implementation and algorithmic efficiency. The scaling study presented incorporates all three efficiencies, but here the primary focus will be algorithmic efficiency. If the underlying numerical algorithm does scale linearly with the number of unknowns, then scaled parallel efficiency will suffer, regardless of the implementation as the numerical method will increase the computational time even for a serial implementation. Algorithmic efficiency [13] is limited by the scalability of the numerical approach, often the preconditioner for the linear system (e.g., [2,11,10,20]).

For the scaling study, a regular, topographic case was developed using repeating sine and cosine functions. This problem, shown in Fig. 6, was developed to be realistic, with lateral flow occurring in all directions driven by a wide range of topographic slopes, but also somewhat consistent as the number of unknowns was increased by several orders of magnitude (and the number of repeated sinusoidal patterns also increases). To get an accurate measure of scalability it is important that the problem not become significantly more difficult numerically as differences in scaling might be driven purely by irregularities in problem design. The topographic slopes for this problem were chosen as

$$\theta_x = 0.1 \sin\left(\frac{x}{5}\right) \text{ and } \theta_y = 0.1 \sin\left(\frac{y}{5}\right) \tag{15}$$

where x [L] and y [L] are the dimensional lateral coordinates of the problem, expressed here in fundamental L and T units (denoted by italic font). A unit problem of 500,000 unknowns with a spatial discretization $\Delta x = \Delta y = 1$; $\Delta z = 0.5$ [L] and spatial extent $nx = ny = 100$; $nz = 50$ on a single processor was simulated. The

Table 3
Details for the scaling study.

Case	Jacobian	Preconditioner	Number of processors	Number of cells	Solver time (s)	Kinsol time (s)	Linear iterations	NL iterations	Function Evals	E (solver)	E (kinsol)
1	Analytical	Nonsym metric	1	500,000	914.71	908.48	99	18	23	100.00%	100.00%
2	Analytical	Nonsym metric	4	2,000,000	1032.43	1026.17	111	19	24	88.60%	88.53%
3	Analytical	Nonsymmetric	256	128,000,000	1063.30	1054.91	112	21	26	86.03%	86.12%
4	Analytical	Nonsymmetric	1024	512,000,000	1115.08	1104.20	116	22	27	82.03%	82.27%
5	Analytical	Nonsymmetric	4096	2,048,000,000	1130.50	1113.89	115	22	27	80.91%	81.56%
6	Analytical	Nonsymmetric	16,384	8,192,000,000	1173.71	1146.57	114	22	27	77.93%	79.23%
7	Finite difference	Symmetric	1	500,000	1063.10	1056.84	125	23	176	100.00%	100.00%
8	Finite difference	Symmetric	4	2,000,000	1125.03	1118.78	129	22	178	94.49%	94.46%
9	Finite difference	Symmetric	256	128,000,000	1545.92	1537.53	176	32	245	68.77%	68.74%
10	Finite difference	Symmetric	1024	512,000,000	1748.06	1737.17	195	36	272	60.82%	60.84%
11	Finite difference	Symmetric	4096	2,048,000,000	1821.08	1804.47	200	36	277	58.38%	58.57%
12	Finite difference	Symmetric	16,384	8,192,000,000	2100.81	2073.66	222	40	307	50.60%	50.96%

problem was constructed with a uniform subsurface: saturated hydraulic conductivity $K_s = 0.25$ [L/T], porosity $\phi = 0.25$ [–], specific storage $S_s = 10^{-4}$ [$1/L$], and vanGenuchten parameters $\alpha = 1.0$ [$1/L$] and $n = 3.0$ [–] S_{res}-0.1 [–], and initialized using a hydrostatic pressure distribution with a surface pressure of -10.0 [L]. This resulted in a water table initialized to follow topography, with slightly more than half of the aquifer thickness initially saturated. Note that this water table configuration is out of equilibrium. The problem was simulated until time $t = 10$ [T] with a uniform timestep $\Delta t = 2.0$ [T]. The scaling study was conducted on JUGENE (http://www.fzjuelich.de/jsc/jugene) an IBM BlueGene/P supercomputer with a total of 294,912 processors and 144 TB of memory capable of 0.825 PetaFLOPS (floating point operations per second). The simulation was run from one to 16,384 processors (for 0.5 M to 8.1B problem unknowns) and the results of the scaling study are presented in Table 3 and Fig. 7.

Two complete sets of simulations were undertaken in the scaling study. The first used the analytical Jacobian shown in Eq. (11) two times: within the Newton iteration and as a nonsymmetric preconditioner for the resulting linear system (denoted as case *AJ NSP*). The second used a finite-difference approximation for the Jacobian (so-called *Jacobian Free*) and only the symmetric part of Eq. (11) as the preconditioner for the linear system (denoted as case *FDJ SP*). It should be noted that (1) this study is similar to Jones and Woodward [11] (2) that the computational cost per iteration is greater for the nonsymmetric preconditioner (3) the new formulation introduces only nonsymmetric terms into the Jacobian (as discussed earlier) and that (4) both cases used the PFMG linear preconditioner used and detailed in [11], noting that this is different than MGSemi used in Kollet et al. [17]. Only the terrain-following grid was tested here as the scalability of the orthogonal grid is well-established [17] and the memory requirements for the orthogonal formulation exceeded the capacity of JUGENE for this problem.

Fig. 7 shows that case *AJ NSP* scales to over 16 K processors at 80% scaled parallel efficiency. This figure plots both the total ParFlow solver time (which includes all components of the transient simulation but not initialization) and the nonlinear solver time (KINSOL), which includes the function evaluation, nonlinear and linear iteration time. A slight decrease in efficiency is observed as the problem is increased from 1 to 4 processors but that efficiency is consistent (stable) out to 16,384 processors. The same E is not seen for the *FDJ SP* case, which has a consistent decline in efficiency to 50% at the largest problem size.

This decline in efficiency (and increase in linear and nonlinear iterations) for the *FDJ SP* case is likely due to the nonsymmetric terms introduced by the TFG as demonstrated by Eqs. (12) and (14). These terms introduce off-diagonal forcings that are not included in the symmetric preconditioner but that are included when the full Jacobian is used as in case *AJ NSP*. These results are consistent with those reported previously for improved solver performance using nonsymmetric preconditioners (e.g., [11,20]). The results are particularly consistent with those of Jones and

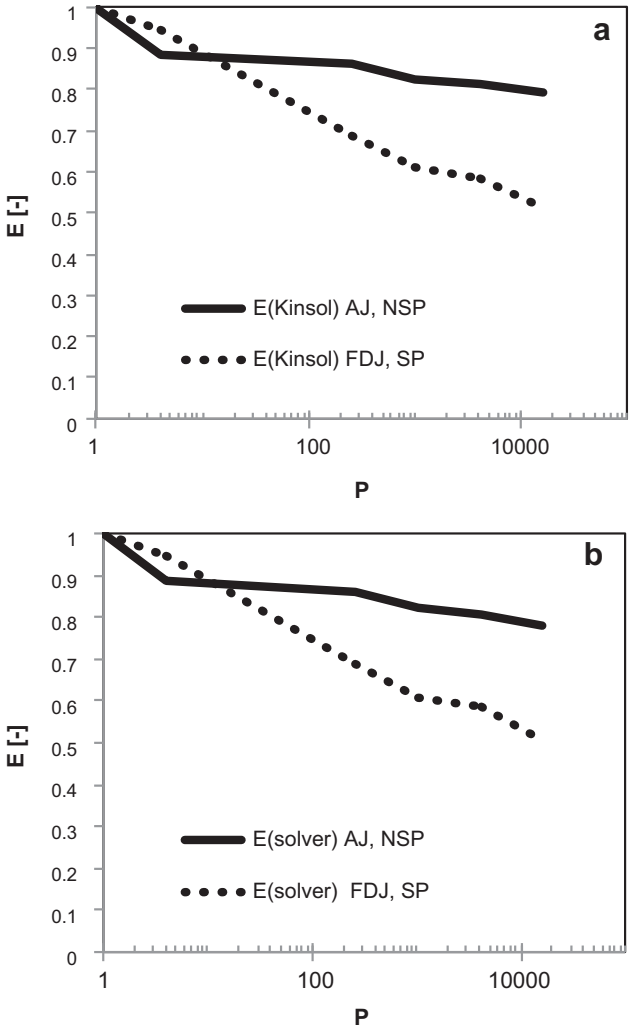

Fig. 7. Plot of scaled parallel efficiency, E, for (a) the KINSOL simulation time and for (b) the entire SOLVER simulation time from one to 16,384 processors, P. Each plot compares two formulations for the nonlinear Jacobian and linear preconditioner. Case *AJ NSP* uses the analytical Jacobian presented in Eq. (11) in the Newton iteration and as a nonsymmetric preconditioner for the linear system. Case *FD SP* uses a finite-difference approximation to the Jacobian and the symmetric part of Eq. (11) as the preconditioner for the linear system.

Woodward [11] where additional preconditioning provided improved solver performance for strongly heterogeneous, tilted-plane type cases. Such cases are very analogous to the scaling problem used here, where the TFG formulation in essence presents a similar problem formulation to the tilted plane locally for large slopes.

Table 3 provides the linear and nonlinear iterations for each simulation in the scaling study. This table shows that the number of nonlinear iterations increases slightly in the *AJ NSP* case from 1 to 4 processors; when the number of processors is greater than 4, the number of nonlinear iterations remains very constant with the same solver work being done for the largest three simulations. Again, this is not the case for the *FDJ SP*, which has a consistent increase in linear and nonlinear iterations throughout the study. While not dramatic, this result indicates that there is a weak nonlinearity in solution time with problem size that impacts the algorithmic efficiency of the method, that is the problem is $O(N^m)$ where m is very close to 1 but not exactly unity.

Since the linear and nonlinear iterations remain constant, the slight decrease in E in the *AJ NSP* case from 256 to 16,384 processors (where the nonlinear and linear iterations remain constant) may be then attributed to other inefficiencies, notably hardware and software. However, Fig. 7 and Table 3 indicate that these amount to approximately 3%; indicating that effort spent to improve the solution techniques even for only very slight improvements in linearity of solution time with problem size, has much more of an impact on E.

6. Conclusions

A new, terrain following transform is developed and implemented in the parallel simulation platform ParFlow. The function evaluation for the new formulation and the corresponding analytical Jacobian are also developed. This formulation is verified against the orthogonal formulation for two cases (hillslope and tilted-v catchment) and the two approaches are shown to be in good agreement at fine spatial discretization. This verification study demonstrates that errors incurred by neglecting off-slope fluxes are smaller than the errors incurred through a coarse representation of the hillslope terrain. Finally, a scaling study is conducted which demonstrates excellent scaled parallel efficiency of the new approach out to 16,384 processors and 8.1B unknowns. It is shown, however, that good scalability depends heavily on algorithmic performance. The use of the analytical Jacobian and nonsymmetric preconditioner developed here are necessary for good scaling results.

The verification results show that the TFG formulation may improve accuracy for steep slopes and coarse lateral resolution. This improves accuracy in complex terrain for fewer unknowns, while still demonstrating good solver performance, linearity of solution time with problem size and efficient parallel scaling. The parallel scaling study also emphasizes that the terrain-following approximation provides a direct approach to solving very large domains (8.1 billion unknowns), over large areas (128 M [L^2]), at very high lateral resolution ($\Delta x = \Delta y = 1$ [L]) for fully-integrated, topographically-driven hydrologic problems. The scaling problem at the largest number of processors is presented in a non-dimensional manner and may be used as a response to the hydrologic hyperresolution "Grand Challenge" presented by Wood et al [28]. Extension of this problem may be made to dimensional problems by applying length and time multipliers to problem dimensions. For example, this problem could represent a high-resolution regional scale by choosing a discretization of $\Delta x = \Delta y = 1$ [m] with $nx = ny = 12{,}800$, resulting in a 12.8 $\times$ 12.8 km domain or a 163 [km^2] extent, similar to that demonstrated in Kollet et al. [17] but with complex terrain. By this same logic, $\Delta x = \Delta y = 100$ [m] yields a 1.63 M [km^2] domain, which is at the continental-scale, again with complex terrain. This formulation, as implemented in a parallel simulation platform, provides a direct path to solving one of hydrology's grand challenges

Acknowledgements

The author is grateful to FZ-Jülich for the compute time grant on JUGENE used for the scaling study in this work. The author is also grateful to E. Siirila and L. Condon, who reviewed a draft of this manuscript prior to peer review and to the three anonymous reviewers who contributed to the clarity and consistency of this work.

References

[1] Arobogast T, Dawson CN, Keenan PT, Wheeler MF, Yotov I. Enhanced cell-centered finite differences for elliptic equations on general geometry. SIAM J Sci Comput 1998;19(2):404–25.
[2] Ashby SF, Falgout RD. A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. Nucl Sci Eng 1996;124(1):145–59.
[3] Brown P, Saad Y. Hybrid Krylov methods for nonlinear systems of equations. SIAM J Sci Stat Comput 1990;11:450–81.
[4] Brutsaert W. The unit response of groundwater outflow from a hillslope. Water Resour Res 1994;30(10):2759–63. http://dx.doi.org/10.1029/94wr01396.
[5] Camporese M, Paniconi C, Putti M, Orlandini S. Surface-subsurface flow modeling with path-based runoff routing, boundary condition-based coupling, and assimilation of multisource observation data. Water Resour Res 2010;46(2):W02512. http://dx.doi.org/10.1029/2008wr007536.
[6] Childs EC. Drainage of groundwater resting on a sloping bed. Water Resour Res 1971;7(5):1256–63. http://dx.doi.org/10.1029/WR007i005p01256.
[7] Farthing MW, Kees CE, Miller CT. Mixed finite element methods and higher-order temporal approximations. Adv Water Resour 2002;25(1):85–101. http://dx.doi.org/10.1016/s0309-1708(01)_00022-7.
[8] Freeze RA, Harlan RL. Blueprint for a physically-based, digitally-simulated hydrologic response model. J Hydrol 1969;9:237–58.
[9] Gustafson JL, Montry GR, Benner RE. Development of parallel methods for a 1024-processor hypercube. SIAM J Sci Stat Comput 1988;9(4):609–38.
[10] Herbst M, Gottschalk S, Reigel M, Hardelauf H, Kasteel R, Javaux M, Vanderborght J, Vereecken H. On preconditioning for a parallel solution of the Richards equation. Comput Geosci 2008;34(12):1958–63. http://dx.doi.org/10.1016/j.cageo.2008.02.020.
[11] Jones JE, Woodward CS. Newton–Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems. Adv Water Resour 2001;24(7):763–74.
[12] Jones JP, Sudicky EA, Brookfield AE, Park YJ. An assessment of the tracer-based approach to quantifying groundwater contributions to streamflow. Water Resour Res 2006;42. http://dx.doi.org/10.1029/2005wr004130.
[13] Knoll DA, Keyes DE. Jacobian-free Newton–Krylov methods: A survey of approaches and applications. J Comput Phys 2004;193(2):357–97. http://dx.doi.org/10.1016/j.jcp. 2003.08.010.
[14] Kollet SJ, Cvijanovic I, Schüttemeyer D, Maxwell RM, Moene AF, Bayer P. The influence of rain sensible heat, subsurface heat convection and the lower temperature boundary condition on the energy balance at the land surface. Vadose Zone J 2009;8(4):12. http://dx.doi.org/10.2136/vzj2009.0005.
[15] Kollet SJ, Maxwell RM. Integrated surface-groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model. Adv Water Resour 2006;29(7):945–58. http://dx.doi.org/10.1029/2007WR006004.
[16] Kollet SJ, Maxwell RM. Capturing the influence of groundwater dynamics on land surface processes using an integrated, distributed watershed model. Water Resour Res 2008;44(W02402):18. http://dx.doi.org/10.1016/j.advwatres.2005.08.006.
[17] Kollet SJ, Maxwell RM, Woodward CS, Smith S, Vanderborght J, Vereecken H, Simmer C. Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources. Water Resour Res 2010;46:W04201. http://dx.doi.org/10.1029/2009wr008730.
[18] Kumar M, Duffy CJ, Salvage KM. A second order accurate, finite volume based, integrated hydrologic modeling (FIHM) framework for simulation of surface and subsurface flow. Vadose Zone J 2009;8:873–90.
[19] Maxwell RM, Miller NL. Development of a coupled land surface and groundwater model. J Hydrometeorol 2005;6(3):233–47.
[20] Morway ED, Niswonger RG, Langevin CD, Bailey RT, Healy RW. Modeling variably saturated subsurface solute transport with MODFLOW-UZF and MT3DMS. Ground Water 2012. http://dx.doi.org/10.1111/j.1745-6584.2012.00971.x.
[21] Panday S, Huyakorn PS. A fully coupled physically-based spatially-distributed model for evaluating surface/subsurface flow. Adv Water Resour 2004;27(4):361–82.

[22] Qu Y, Duffy CJ. A semidiscrete finite volume formulation for multiprocess watershed simulation. Water Resour Res 2007;43(W08419):18. doi:10.1029/2006WR005752.
[23] Richards LA. Capillary conduction of liquids in porous mediums. Physics 1931;1:318–33.
[24] Sloan PG, Moore ID. Modeling subsurface stormflow on steeply sloping forested watersheds. Water Resour Res 1984;20(12):1815–22. http://dx.doi.org/10.1029/WR020i012p01815.
[25] Sulis M, Meyerhoff SB, Paniconi C, Maxwell RM, Putti M, Kollet SJ. A comparison of two physics-based numerical models for simulating surface water-groundwater interactions. Adv Water Resour 2010;33(4):456–67. http://dx.doi.org/10.1016/J.Advwatres.2010.01.010.
[26] van Genuchten MT. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Sci Soc Am J 1980;44(5):892–8.
[27] VanderKwaak JE, Loague K. Hydrologic-response simulations for the r-5 catchment with a comprehensive physics-based model. Water Resour Res 2001;37(4):999–1013.
[28] Wood EF, Roundy JK, Troy TJ, van Beek LPH, Bierkens MFP, Blyth E, de Roo A, Döll P, Ek M, Famiglietti J, Gochis D, van de Giesen N, Houser P, Jaffè PR, Kollet S, Lehner B, Lettenmaier DP, Peters-Lidard C, Sivapalan M, Sheffield J, Wade A, Whitehead P. Hyperresolution global land surface modeling: meeting a grand challenge for monitoring earth's terrestrial water. Water Resour Res 2011;47(5):W05301. http://dx.doi.org/10.1029/2010wr010090.