Adaptation and validation of the ParSWMS numerical code for simulation of water flow and solute transport in soilless greenhouse substrates

Mohammad R. Gohardoust a, Jirka Šimůnek b, Horst Hardelauf c, Markus Tuller a,∗

a Department of Environmental Science, The University of Arizona, Tucson, AZ, USA
b Department of Environmental Sciences, University of California, Riverside, CA, USA
c Institute of Bio-and Geosciences, IBG-3 Agrosphere, Jülich, Germany

1. Introduction

Because of the urgent need to secure and sustain the food and water supply for an ever-growing human population, especially in underdeveloped arid and semiarid regions of the world, and an increasing demand for out-of-season fruits, vegetables, and ornamentals in the industrial world, there is a momentous incentive to shift from soil-based crop production to more resource-efficient containerized soilless production systems (Raviv et al., 2019). Soilless substrates exhibit major advantages over soils. Besides the alleviated risk for spreading soilborne pathogens, their physical and hydraulic properties (i.e., plant water availability after irrigation and the aeration characteristics) are superior to those of arable soils (Savvas, 2003; Savvas and Gruda, 2018). Furthermore, the nutrient availability to plant roots can be better controlled in soilless substrates, which commonly leads to healthier plants and higher yields than in soil-based production (Raviv et al., 2019). It is even possible to tailor (i.e., engineer) substrates that exhibit specific physical and hydraulic properties beneficial for the growth of different crops. Though the same physical principles apply to both soilless substrates and soils, their physical and hydraulic properties are vastly different.

In addition, there are fundamental differences with regard to...
dynamic water, air, and nutrient distribution processes and root growth and development between spatially confined growth containers and unconfined field soils. While an impermeable container bottom (with drainage holes) restricts water flow and nutrient transport in growth containers, water drains and redistributes to much deeper layers in drainage holes) restricts water flow and nutrient transport in growth module design and irrigation and fertilization management, thereby preventing costly mistrials.

Today, numerical computer codes that solve the governing 3D water flow and solute transport equations are indispensable for advancing the understanding of complex porous media processes. However, intricate hydraulic properties and boundary conditions demand high-resolution discretization of the spatial and temporal domains, often leading to several million nodes requiring numerical evaluation (Siminek et al., 2016). Even the current computing prowess of workstations is not adequate to tackle such simulations within a reasonable timeframe, especially when numerous realizations are desired to optimize the geometry, soilless substrate properties, and irrigation and fertilization management of plant growth modules. Hence, parallelization of the numerical codes and utilization of high-performance computing (HPC) clusters are vital.

Hardelauf et al. (2007) were among the first to develop a parallelized code for 3D simulation of water flow and solute transport that they termed ParSWMS. The code is based on the SWMS-3D model of Siminek et al. (1995), which numerically solves the Richards equation for saturated/unsaturated water flow and the convection–dispersion equation (CDE) for solute transport using a Galerkin-type linear finite element scheme. The resulting nonlinear partial differential equation (PDE) for water flow and the linear PDE for the CDE (i.e., assuming linear adsorption isotherms) are solved in SWMS-3D with preconditioned conjugate gradient and Orthomin (Vinsome, 1976) methods, respectively. In ParSWMS, Hardelauf et al. (2007) employed ParMETIS, an open source MPI-based library, to distribute the simulation to multiple processing nodes, and the Portable, Extensive Toolkit for Scientific Computation (PETSc) library (https://www.mcs.anl.gov/petsc/) for preconditioning and solving the resulting system of linear equations.

Driven by the need to execute a vast number of 3D flow and transport simulations to aid with the design and management of containerized plant growth modules, we adopted ParSWMS and modified the code to: (1) permit the simultaneous application of time-variable flux and free boundary conditions (BCs) and allow for the time-variable flux BC to be treated as an atmospheric BC, and (2) enable multi-solute transport simulations with nonlinear adsorption. These modifications are essential for the simulation of surface drip irrigation and to account for the nonlinear adsorption behavior of phosphorus and ammonium.

Using HYDRUS (2D/3D) version 3.02.0560 (Siminek et al., 2018) as a reference, we first meticulously verified the modified ParSWMS code for a simple, yet realistic, soilless growth module setup. Thereafter, we focused on computational efficiency and evaluated a vast number of preconditioner/solver combinations for both water flow and solute transport based on a real growth module used in production-scale greenhouse experiments. Subsequently, we benchmarked the soilless substrate flow and transport simulation results against HYDRUS (2D/3D) on a workstation with both Windows and Linux OS installed. As a final step, we executed the modified ParSWMS code on two HPC clusters to demonstrate the significant computational speedup relative to the HYDRUS (2D/3D) simulations performed on the workstation. Please note that for brevity we refer to HYDRUS instead of HYDRUS (2D/3D) in the following.

2. Modifications to ParSWMS

2.1. Boundary conditions

In the ParSWMS water flow simulation each node is assigned an integer code ranging from –6 to 6. While positive values are allocated to nodes with prescribed pressure heads, zero and negative values represent nodes with prescribed volumetric water fluxes. In the original ParSWMS code (Hardelauf et al., 2007) both time-variable flux and free drainage boundary nodes are assigned –3. Hence, these BCs cannot be used simultaneously. To resolve this issue, the free drainage boundary condition code was set to –6, similar as in HYDRUS, which was not allocated in the original ParSWMS. The discharge rate at a free drainage node n is determined as q(n) = –width(n) × K(h), where width(n) is the surface area associated with node n, and K(h) is the hydraulic conductivity as a function of the pressure head h.

To account for stage-I (i.e., potential evaporation rate controlled by atmospheric demand) and stage-II (i.e., falling evaporation rate limited by the ability of the substrate to transmit water to the surface) evaporation, it is necessary to treat the variable flux BC imposed for simulating surface drip irrigation as an atmospheric BC. The atmospheric BC that is implemented in both HYDRUS and ParSWMS is expressed as (Neuman et al., 1974):

\[
\begin{aligned}
K' & \left( K' \frac{d h}{d x} + K'' \right) n_c \leq E \\
\text{and} \quad h_{\text{CritA}} \leq h \leq h_{\text{CritS}}
\end{aligned}
\]

where K is the hydraulic conductivity, K' are the components of the anisotropy tensor K', x(i = 1,2,3) are the spatial coordinates, n_c are the components of the outward unit vector normal to the atmospheric boundary, E is the maximum potential evaporation rate, and h_{\text{CritA}} and h_{\text{CritS}} are two limiting values for the surface pressure head h. The Einstein summation convention is applied to Eq. (1), which implies summation over all possible values of the index in algebraic terms with repeated indices. While the h_{\text{CritS}} value specifies the maximum allowed surface pressure head defined based on equilibrium conditions between soil water and atmospheric vapor. The modified ParSWMS code now automatically switches between Dirichlet and Neumann boundary conditions for nodes with a variable flux BC if one of these limiting points is reached.

2.2. Multi-solute transport with nonlinear equilibrium adsorption

To allow realistic simulations of phosphorus and ammonium transport processes in containerized soilless substrates, an option for nonlinear equilibrium solute adsorption was added to the modified ParSWMS code. The underlying CDE in ParSWMS is given as:

\[
\frac{\partial c}{\partial t} + \frac{\partial s}{\partial t} + \frac{\partial \left( \theta \frac{\partial c}{\partial x_i} \right)}{\partial x_i} - \mu_s \frac{\partial c}{\partial x_i} + \mu_s ps + \gamma_s \theta + \gamma_s \rho - Sc = 0
\]

where \( \theta \) is the volumetric substrate water content [L^3 L^{-3}], c is the solution concentration [M L^{-3}], s is the concentration adsorbed to the solid phase [M L^{-3}], \( \rho \) is the bulk density of the substrate [M L^{-3}], D_p is the dispersion coefficient tensor [L^2 T^{-1}], q_i is the i-th component of the volumetric flux [LT^{-1}], \( \mu_s \) and \( \mu_p \) are first-order rate constants for solutes in the liquid and solid phases [T^{-1}], respectively, \( \gamma_s \) and \( \gamma_p \) are zero-order rate constants for the liquid [M L^{-3} T^{-1}] and solid [T^{-1}] phases, respectively, S is the sink term in the water flow equation [T^{-1}], and c is the concentration of the sink term [M L^{-3}] with potential \( i \) and \( j \) values of...
1, 2, 3. The equilibrium adsorption isotherm in its general form can be written as:

\[ s = \frac{k_i c^\beta}{1 + q c^\eta} \]  

(3)

where \( k_i \) \([M^3M^{-1}]\) is the distribution coefficient, and \( \beta \) \([-\] and \( \eta \) \([L^3M^{-1}]\) are coefficients of the nonlinear Freundlich and Langmuir adsorption isotherm models, respectively. For linear adsorption \( \beta = 1 \) and \( \eta = 0 \) with dimensions of \([L^3M^{-1}]\), for nonlinear Langmuir adsorption \( \beta = 1 \), and for nonlinear Freundlich adsorption \( \eta = 0 \).

Substituting Eq. (3) into Eq. (2) yields a nonlinear differential equation that can be solved with nonlinear iterative schemes. Picard iteration was chosen for the nonlinear CDE in the same fashion as for the Richards equation. Note that numerical solvers and preconditioners specific for nonsymmetric matrices are required for the discretized CDE (see Section 3 below). To increase the computational efficiency, simultaneous multi-solute transport simulations (i.e., phosphorus and ammonium) were enabled in the modified ParSWMS code – again, similar as in the current version of HYDRUS.

3. Hydraulic and chemical soilless substrate properties

The selection of the three soilless substrates that were considered for the simulations was guided by production scale greenhouse tomato growth experiments at the Volcani Center in Israel. The substrates included perlite, volcanic tuff, and a 70/30 vol-% volcanic tuff/coconut coir mixture.

Horticultural perlite (Fig. 1a) is a natural amorphous volcanic glass that is formed through the hydration of obsidian. Perlite is commonly heated to 1000°C, which causes structural water to evaporate and the volume to expand to about 4 to 20 times of its original size when rehydrated (Bar-Tal et al., 2019). The sieved perlite aggregates are light weight, inert, and pathogen free (Noland et al., 1992).

Tuff (Fig. 1b) is a pyroclastic volcanic material with high porosity and surface area. The mineral composition and weathering stage, in conjunction with physical alterations (i.e., grinding and sieving), determine its physicochemical properties (Silber et al., 1994). Tuff possesses a high buffering capacity and may adsorb and release nutrients throughout the growing season, especially phosphorus (Silber et al., 1999; Silber and Raviv, 1996). It exhibits bulk densities between 0.8 and 1.5 g cm^{-3} and a total porosity ranging from 60 to 80%.

Coconut coir consists of short and medium-length fibers of the mesocarp of Cocos nucifera L. that are left from various industrial applications. It exhibits remarkable physical and chemical characteristics such as high water holding capacity, good drainage and aeration properties, and high cation exchange capacity (Evans et al., 1996; Abad et al., 2005). It is commonly used as a surrogate for peat moss and mixed with mineral substrates such as tuff (Fig. 1c).

The substrate water characteristics (SWC) were measured with Tempe cells (Soilmoisture Equipment Corp., Santa Barbara, CA) connected to a pressure manifold with a high-resolution pressure/vacuum regulator and gauge. Initially, saturated samples were sequentially desaturated via application of increasing pressures after each equilibration phase. The saturated hydraulic conductivities (\( K_{sat} \)) were measured with an automated constant head permeameter. Details about the SWC and \( K_{sat} \) measurements are provided in Gohardoust et al. (2020).

After thoroughly testing several SWC models comprising the van Genuchten (1980) model, a modified version of the van Genuchten model (Vogel and Cislnerova, 1988; Vogel et al., 2000), and the Brooks and Corey (1966) model, we found that the latter was best suited for mitigation of numerical instability issues caused by the extreme nonlinearity of the hydraulic conductivity function near saturation for all considered soilless substrates (Fig. 1). The Brooks and Corey (BC) SWC model is given as:

\[
\begin{aligned}
\theta(h) &= \theta_s + (\theta_r - \theta_s) \left( \frac{h}{h_b} \right)^{\lambda} & h < h_b \\
\theta(h) &= \theta_r & h \geq h_b \\
K(h) &= K_{sat} \left( \frac{h}{h_b} \right)^{-2-\lambda} 
\end{aligned}
\]  

(4)

where \( h \) is the pressure head, \( h_b \) is the air entry pressure (i.e., pressure threshold related to the onset of drainage of the largest pore in the system – a transition from fully to partially saturated conditions), \( \theta \) is the water content expressed as a function of \( h \), \( \theta_s \) is the saturated water content, \( \theta_r \) is the residual water content, \( \lambda \) is an empirical shape parameter, and \( K \) is the hydraulic conductivity expressed as a function of \( h \).

The parameters of the BC model were determined via least-square fitting to the measured \( \theta - h \) data pairs (Fig. 2), with SWC parameters and \( K_{sat} \) listed in Table 1.

The phosphorus and ammonium adsorption isotherms were measured with a combination of calorimetric spectrometry and inductively coupled plasma mass spectrometry. Details are provided in Gohardoust et al. (2020). The derived isotherm parameters are listed in Table 2.

4. Verification of the modified ParSWMS code

Using HYDRUS version 3.02.0560 (Šimůnek et al., 2018) as a reference, we meticulously verified the modified ParSWMS code using a simple domain representative of a containerized growth module (Fig. 3). We note that HYDRUS has been extensively tested for specific flow or transport problems based on analytical solutions and measured data (e.g., Cook et al., 2006; Kandelous and Šimůnek, 2010; Karlsson et al., 2015; Lassabatere et al., 2014; Luo and Sophocleous, 2010; Neumann et al., 2011; Vanderborght et al., 2005), and code-to-code validation is a scientifically accepted approach (e.g., Greenwald, 2010; Hardelauf et al., 2007; Kačur and Minár, 2013; Orgogozo et al., 2014).

Fig. 1. Soilless substrates considered for flow and transport simulations. (a) perlite, (b) tuff, and (c) 70/30 vol-% volcanic tuff/coconut coir mixture.
Table 1  
Parameters of the Brooks & Corey SWC model and measured $K_{sat}$.

<table>
<thead>
<tr>
<th>Substrate</th>
<th>$\theta_s$ (cm$^3$ cm$^{-3}$)</th>
<th>$\theta_r$ (cm$^3$ cm$^{-3}$)</th>
<th>$\theta_{sat}$ (cm$^3$ cm$^{-3}$)</th>
<th>$\lambda$ (cm)</th>
<th>$K_{sat}$ (cm h$^{-1}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perlite</td>
<td>0.818</td>
<td>0.001</td>
<td>0.458</td>
<td>0.149</td>
<td>305.1</td>
</tr>
<tr>
<td>Tuff</td>
<td>0.483</td>
<td>0.014</td>
<td>0.382</td>
<td>0.247</td>
<td>304.2</td>
</tr>
<tr>
<td>Tuff/Coconut Coir</td>
<td>0.549</td>
<td>0.014</td>
<td>1.102</td>
<td>0.219</td>
<td>110.7</td>
</tr>
</tbody>
</table>

Table 2  
Nonlinear adsorption isotherm parameters for phosphorus and ammonium.

<table>
<thead>
<tr>
<th>Substrate</th>
<th>Phosphorus $k_a$ (cm$^3$ g$^{-1}$)</th>
<th>Phosphorus $b$ (cm$^3$ g$^{-1}$)</th>
<th>Ammonium $k_a$ (cm$^3$ g$^{-1}$)</th>
<th>Ammonium $b$ (cm$^3$ g$^{-1}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perlite</td>
<td>17.71</td>
<td>984</td>
<td>147.19</td>
<td>3376</td>
</tr>
<tr>
<td>Tuff</td>
<td>17.86</td>
<td>66</td>
<td>58.43</td>
<td>135</td>
</tr>
<tr>
<td>Tuff/Coconut Coir</td>
<td>24.66</td>
<td>102</td>
<td>42.94</td>
<td>83</td>
</tr>
</tbody>
</table>

Fig. 2. The Brooks and Corey (1964) SWC model fitted to the measured data for tuff (left), the 70/30 vol-% tuff/coconut coir mixture (center), and perlite (right).

Fig. 3. Flow domain geometry with water flow (top) and solute transport (bottom) BCs. The Cauchy-1 BC is used for the solute concentrations at the top atmospheric and bottom free drainage boundaries. The Cauchy-2 BC is used for the solute concentrations at the drip irrigation boundaries.

To mimic irrigation and fertigation with drip emitters, water, phosphorus, ammonium, and nitrate were applied to the surface within two circles with variable flux BCs. When no water or solutes were applied, the variable flux BC was treated as an atmospheric BC to conform with the remaining evaporating substrate surface. The 70/30 vol-% tuff/coconut coir mixture (see Table 1 for hydraulic parameters), which was dry (i.e., $\approx 5000$ cm pressure head) at the start of the simulation, was used for validation. While nonlinear Langmuir adsorption was considered for ammonium ($NH_4^+$) and phosphorus ($H_2PO_4^-$) (see adsorption isotherm parameters in Table 2), nitrate ($NO_3^-$) did not interact (i.e., adsorb) with the solid phase. The applied concentrations for $H_2PO_4^-$, $NH_4^+$, and $NO_3^-$ were 20 mg l$^{-1}$, 20 mg l$^{-1}$, and 80 mg l$^{-1}$, respectively. Water was applied every 8 h at a rate of 1.0 l h$^{-1}$ for 4 min with one-fourth allowed to evaporate. The longitudinal and transverse dispersivities were assumed to be 2.0 and 0.2 cm, respectively. The diffusion coefficients for $H_2PO_4^-$, $NH_4^+$, and $NO_3^-$ in water were assumed as 0.032, 0.063, and 0.068 cm$^2$ h$^{-1}$, respectively. Similar values have been reported in Buffle et al. (2007) and Hashitani and Tanaka (1983). The water content tolerance was set to $2 \times 10^{-5}$ cm$^3$ cm$^{-3}$, and the absolute and relative concentration tolerances were set to $10^{-6}$ mmol cm$^{-3}$ and $10^{-3}$, respectively. The flow and transport domains were discretized with finite elements of 1.4-cm in the horizontal direction and 0.7–cm in the vertical direction. Mesh refinements for the top 1-cm layer (i.e., using half of the finite element size) led to a total of 25,264 spatial nodes. The temporal discretization was dynamically calculated throughout the simulation based on the specified initial time step of 2.4E–7 hr. The total simulation time was ten days.

Input files for the modified ParSWMS code were generated with the “Export to ParSWMS” function in HYDRUS. To extract and analyze ParSWMS simulation results, dedicated MATLAB® – Version R2019b (MathWorks, Natick, MA, USA) scripts were created. The simulations were performed on a workstation with two 2.40 GHz Intel® Xeon® E5-2630 v3 processors. The hard drive was partitioned with Windows 10 Pro (Version 1909, OS Build 18363.1016) as well as Ubuntu 18.04.5 LTS installed to run the HYDRUS and modified ParSWMS codes, respectively.

The modified ParSWMS and HYDRUS water flow and solute transport simulation results were compared at the end of the simulations based on the normalized root mean squared error (NRMSE) given as:

$$NRMSE = \sqrt{\sum_{i=1}^{N} \frac{(C_{ParSWMS,i} - C_{HYDRUS,i})^2}{C_{HYDRUS,max} - C_{HYDRUS,min}}}$$

where $C$ is the specific output quantity to be compared, and $N$ is the number of spatial nodes.

The comparison of the modified ParSWMS and HYDRUS volumetric water content ($\theta$) and $H_2PO_4^-$, $NH_4^+$, and $NO_3^-$ concentration simulation results after 10 days are depicted in Fig. 4. The coefficients of determination for $\theta$ and $H_2PO_4^-$, $NH_4^+$, and $NO_3^-$ concentrations exceeded 0.999 with associated NRMSEs of less than 1%, which indicates excellent agreement between HYDRUS and the modified ParSWMS code. This is further solidified by the very low mass balance errors of less than 0.3%
for water and less than 2.5% for the solutes (Table 3). Notably, the mass balance errors of the modified ParSWMS code are smaller than the errors of HYDRUS.

5. Evaluation of preconditioners and solvers

To simulate water flow and solute transport, separate sets of linear equations need to be solved. To assure computational efficiency, it is imperative to apply the optimal preconditioner/solver combinations, which differ for flow and transport. The discretization of the associated partial differential equations via finite element or finite difference methods yields linear systems of the form:

$$Ax = b$$

with $A \in \mathbb{R}^{N \times N}$ is the coefficient matrix, $b \in \mathbb{R}^{N}$ is the right-hand side vector, and $x$ is the vector of unknowns.

Depending on the extent of the problem, direct or iterative solvers may be employed to solve Eq. (7). To enhance the numerical robustness and stability of ill-posed problems with very high condition numbers, sparse direct solvers are preferred despite their high memory usage and extended computation times, especially for large three-dimensional problems (Kwack et al., 2016; Duff and Scott, 2004; Benzi, 2002). On the other hand, sparse iterative solvers such as preconditioned Krylov subspace solvers can most efficiently handle massive linear equation systems (Dongarra and Sullivan, 2000). To resolve Eq. (7) iteratively, the solution is approximated via constructing a sequence of $x_n$, starting with an initial guess $x_0$, by moving in an affine subspace $x_0 + \mathcal{K} \subset \mathbb{R}^N$. In order to create suitable subspaces $\mathcal{K}$, methods containing efficient operations such as matrix–vector products (e.g., Krylov subspaces) are of great value (Del Corso et al., 2015). Assuming the residual of the $n$th iteration $r_n = b - Ax_n$, Krylov subspace projection methods can be separated into three main categories (Golub and van der Vorst, 1997): (1) Ritz-Galerkin methods in which $x_n$ is constructed such that the residual is orthogonal to the current subspace – examples are conjugate gradients (CG), the full orthogonalization method (FOM) (Saad, 1981), and generalized conjugate gradients (GENCG) (Eisenstat et al., 1983); (2) minimum residual methods in which $x_n$ minimizes the second norm of the residual over $\mathcal{K}^n$ – the generalized minimal residual (GMRES) (Saad and Schultz, 1986), and the minimum residual (MINRES), ORTHOMIN (Vinsome, 1976), and ORTHODIR (Young and Jea, 1980) methods fail

Table 3

<table>
<thead>
<tr>
<th>Irrigation (cm$^3$)</th>
<th>$E_{sv}$ (cm$^3$)</th>
<th>$E_{sv}^*$ (cm$^3$)</th>
<th>$FD$ (cm$^3$)</th>
<th>$\Delta S_w$ (cm$^3$)</th>
<th>Mass Balance Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParSWMS</td>
<td>4001.86</td>
<td>902.8</td>
<td>49.13</td>
<td>912.5</td>
<td>2147.41</td>
</tr>
<tr>
<td>HYDRUS</td>
<td>4001.88</td>
<td>902.7</td>
<td>48.88</td>
<td>912.9</td>
<td>2147.41</td>
</tr>
</tbody>
</table>

* Water evaporation from areas with an atmospheric BC.
* Water evaporation from areas with a variable flux BC when not irrigated.
* Water loss from the free drainage boundaries.
* Change in water storage.

---

Fig. 4. Comparison of the modified ParSWMS and HYDRUS volumetric water content ($\theta$) (a) and $H_2PO_4^-$ (b), $NH_4^+$ (c), and $NO_3^-$ (d) concentration simulation results for the 25,264 spatial nodes after 10 days.
within this category; and (3) Petrov-Galerkin methods in which \( x_n \) is determined so that the residual is orthogonal to some other \( n \)-dimensional subspace – the biconjugate gradient (Bi-CG) (Fletcher, 1975) and quasi-minimal residual (QMR) methods (Freund and Nachtigal, 1991) are part of this category. In addition, there are hybrids of the above categories such as the conjugate gradient squared (CGS) (Sonneveld, 1989), biconjugate gradient stabilized (Bi-CGSTAB) (van der Vorst, 1992), transpose-free quasi-minimal residual (TFQMR) (Freund, 1993), and flexible generalized minimal residual (FGMRES) (Saad, 1993) methods.

In brief, preconditioning is applied to convert a given problem into a form that is more amenable for numerical solution (Herbst et al., 2008). This means that Eq. (7) is transformed to the form:

\[
M_1^{-1}A M_2^{-1} y = M_1^{-1} b \quad y = M_2 x
\]

where \( M_1 \) and \( M_2 \) are nonsingular matrices with inverses that can be rapidly and accurately solved. This leads to faster convergence and requires much less memory (Simoncini and Szyld, 2007).

To find optimal preconditioner/solver combinations with regard to both robustness and computational efficiency, we consider a realistic flow and transport simulation applicable for soilless culture growth modules. Note that the optimal preconditioner/solver combinations differ for water flow and solute transport simulations as they exhibit symmetric and asymmetric coefficient matrices, respectively. Hence, they are evaluated separately.

5.1. Preconditioners and solvers for water flow

Computational efficiency is essential when simulating flow and transport processes in complex porous systems, where intricate hydraulic properties and boundary conditions demand high-resolution discretization of the spatial and temporal domains, often leading to several million nodes requiring numerical evaluation. This holds true for soilless plant growth modules, especially when numerous realizations are desired to optimize container geometry, soilless substrate properties, and irrigation and fertigation management.

To evaluate potential preconditioner/solver combinations for water flow, we considered a typical growth module for greenhouse tomato production at the Agricultural Research Organization (ARO) Volcani Center in Israel (Fig. 5a), where water and nutrients are applied via surface drip emitters and the solution drains through circular openings in the bottom of the concrete containers (Fig. 5b). The 70/30 vol-% tuff/coconut coir mixture with its hydraulic properties listed in Table 1 was used as the substrate for the test case. The applied water flow and solute transport BCs are consistent with the flow domain depicted in Fig. 3. For water flow, this means an atmospheric BC at the top with ten circular inclusions with variable flux BCs to mimic surface drip irrigation (i.e., two angle arrow drippers per tomato plant) and seven openings at the bottom to allow for free drainage. For root water uptake, a simple root distribution model (Vrugt et al., 2001a, 2001b) was considered for each of the five tomato plants (Siminek et al., 2018). While uptake conforms with the Feddes et al. (1978) water stress response function, the potential uptake rate was constrained at 13% of the evapotranspiration rate. Nutrients were applied together with irrigation water (see Section 4 for concentrations). Drip irrigation occurred every two hours for 2.75 min at a rate of 1.0 L per hour per angle arrow dripper (i.e., a total of 5.5 L were applied per day).

To consider heterogeneity, independent scaling factors for the pressure head and hydraulic conductivity function were introduced. The scaling factors were generated via sequential Gaussian simulation with the VISIM program (Hansen and Mosigaard, 2008) in the open source mGstat MATLAB® toolbox (Hansen, 2020) with a variance of 0.05, a mean value of 1.0, and correlation lengths of 4 and 2 cm in the horizontal and vertical directions, respectively. The spatial distribution of the hydraulic conductivity scaling factors within the flow domain is depicted in Fig. 6.

The flow domain was discretized with finite elements of 1.0-cm in the horizontal direction and 0.5-cm in the vertical direction. Mesh refinements for the top 1-cm layer, and in areas with variable flux and free drainage BCs, led to a total of 408,848 nodes and 2,315,763 3D mesh elements.

To solve Eq. (7) with its symmetric sparse coefficient matrix for water flow, a total of 15 preconditioners and 13 solvers were considered (Table 4). All preconditioners and solvers, but two, were extracted from the Portable, Extensible Toolkit for Scientific Computation (PETSc) library (https://www.mcs.anl.gov/petsc/). Two preconditioners were incorporated from the High Performance Preconditioners (HYPRE) library (http://www.llnl.gov/CASC/hypre/) via the PETSc interface. The sparsity pattern of the coefficient matrix associated with the water flow simulation is depicted in Fig. 7. Because of the large number of considered preconditioner/solver combinations, the total time for each

![Fig. 5. Greenhouse tomato growth trial at the ARO Volcani Center in Israel (a). Rendering of a concrete growth module used in the greenhouse trial (b).](image-url)
combinations together with the ideal speedup line that is defined as the setup and application time, as well as the time for solver application, solver combinations, the wall clock time, which includes preconditioner and 94, respectively.

The number of cores (NC) utilized were 12, 24, 48, respectively. The simulation was limited to 10 h. The simulations were performed on the University of Arizona Puma HPC cluster with system specifications provided in Table S1. The number of cores (NC) utilized were 12, 24, 48, and 94, respectively.

To compare the computational efficiency of various preconditioner/solver combinations, the wall clock time, which includes preconditioner setup and application time, as well as the time for solver application, was used as a metric. Fig. 8 depicts the ten fastest preconditioner/solver combinations together with the ideal speedup line that is defined as the ratio of utilized cores and the number of cores used for the base case (i.e. 12):

\[ T_p = \frac{p_{\text{base}}}{p} T_{\text{base}} \]  

where \( T_p \) is the simulation time when \( p \) processors are utilized, and \( p_{\text{base}} \) and \( T_{\text{base}} \) are the number of processors used for the base case and the associated simulation time, respectively. The inset in Fig. 8 shows all successful combinations. As apparent, the choice of the preconditioner/solver combination significantly impacts computational efficiency, as the determined wall clock times vary by more than one order of magnitude. Note that some of the simulations with the GAMG preconditioner did not converge when 94 cores were utilized. The reported wall clock times represent the median of 5 separate simulation runs. The simulation speedup follows the ideal speedup line up to 48 cores, beyond which the performance starts to degrade. This is attributable to the increase of the communication to computation ratio (Hammond et al., 2014). For the executed simulations, the average number of nodes for each core was about 8500 and 4400 when utilizing 48 and 94 cores, respectively. This suggests that for optimal scalability, the minimum number of nodes per processor should fall within this range.

Because the order of the most efficient combinations is dependent on the number of cores (NC), the wall clock times were normalized by the smallest value for each NC group (i.e., 12, 24, 48, and 94) and then averaged. The five fastest combinations and their associated normalized times are listed in Table 5. Although the DGMRES solver applied in conjunction with the BJACOBI preconditioner performed better than other widely used solvers such as CG, the differences are minor between the five computationally most efficient combinations (Table 5). This leads to the conclusion that BJACOBI, which performs the incomplete lower-upper factorization with zero level fill.

**Table 4** Preconditioners and solvers considered for the water flow simulations.

<table>
<thead>
<tr>
<th>Preconditioner</th>
<th>Abbr.</th>
<th>Solver</th>
<th>Abbr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JACOBI (diagonal scaling)</td>
<td>JACOBI</td>
<td>Conjugate Gradient</td>
<td>CG</td>
</tr>
<tr>
<td>Block JACOBI†</td>
<td>BJACOBI</td>
<td>Conjugate Gradient</td>
<td>CGS</td>
</tr>
<tr>
<td>Additive Schwarz (Restrict)</td>
<td>ASM-R</td>
<td>Flexible Conjugate Gradient</td>
<td>FCG</td>
</tr>
<tr>
<td>Additive Schwarz (Interpolate)</td>
<td>ASM-I</td>
<td>Generalized Minimal Residual</td>
<td>GMRES</td>
</tr>
<tr>
<td>Additive Schwarz (Basic)</td>
<td>ASM-B</td>
<td>Pipelined FCG</td>
<td>PIPEFCG</td>
</tr>
<tr>
<td>Additive Schwarz (None)</td>
<td>ASM-N</td>
<td>Deflated GMRES</td>
<td>DGMRES</td>
</tr>
<tr>
<td>Shared Blocks ASM (Restrict)</td>
<td>GASM-R</td>
<td>Loose GMRES</td>
<td>LGMRES</td>
</tr>
<tr>
<td>Shared Blocks ASM (Interpolate)</td>
<td>GASM-I</td>
<td>Biconjugate Gradient</td>
<td>BCG</td>
</tr>
<tr>
<td>Shared Blocks ASM (Basic)</td>
<td>GASM-B</td>
<td>Biconjugate Gradient</td>
<td>BCG</td>
</tr>
<tr>
<td>Shared Blocks ASM (None)</td>
<td>GASM-N</td>
<td>Improved Stabilized BICGSTAB</td>
<td>IBCG</td>
</tr>
<tr>
<td>Successive Over Relaxation</td>
<td>SOR</td>
<td>Enhanced BICGSTAB</td>
<td>BCGSL</td>
</tr>
<tr>
<td>SymmetricSOR</td>
<td>EISEN</td>
<td>Minimum Residual</td>
<td>MINRES</td>
</tr>
<tr>
<td>Classical Algebraic Multigrid</td>
<td>BOOMERAMG</td>
<td>Chebyshev Iterative Method</td>
<td>CHEBYSHEV</td>
</tr>
<tr>
<td>Sparse Approximate Inverse</td>
<td>PARASAIL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Geometric Algebraic Multigrid</td>
<td>GAMG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

† Inner preconditioner for each block is ILU(0), i.e., incomplete lower-upper factorization with zero level fill.

† Equivalent to the block Jacobi with SOL on each block.

‡ From the HYPRE library. BOOMERAMG with “HIMIS” coarsen type, “ext + I” interpolation type, zero number of levels of aggressive, and 0.5 threshold value for being strongly connected.

§ GAMG with type ‘aggregate’ and one smoothing step.

‖ With restart parameter 60.

![Fig. 6. Spatial distribution of the scaling factor (\(a_k\)) for the hydraulic conductivity function within the flow domain.](image)

![Fig. 7. Sparsity pattern of the symmetric coefficient matrix for water flow simulations.](image)
simulations, which is most likely because of the more complex water distributions within our flow domain due to the implementation of surface drip irrigation, root water uptake, and free drainage from the bottom boundary.

5.2. Preconditioners and solvers for solute transport

The test case for solute transport is identical to the test case in Section 5.1 with regard to water flow. In addition, $H_2PO_4^-$, $NH_4^+$, and $NO_3^-$ were applied with the irrigation water at concentrations of 20 mg l$^{-1}$, 20 mg l$^{-1}$, and 80 mg l$^{-1}$, respectively. While nonlinear Langmuir adsorption was considered for ammonium ($NH_4^+$) and phosphorus ($H_2PO_4^-$), nitrate ($NO_3^-$) did not interact with the solid phase. Fifteen preconditioners and 12 solvers were evaluated (Table 6).

The ten computationally most efficient preconditioner/solver combinations for solute transport are depicted in Fig. 9 together with the ideal speedup line. The inset shows all successful combinations, which vary over two orders of magnitude. As for water flow, the simulation speedup follows the ideal speedup line up to 48 cores and then the performance starts degrading. Combinations with the GAMG preconditioner were the least efficient. Although the BOOMERAMG preconditioner performed better for solute transport, it is not among the top-ranked. These results differ from Sbai and Larabi (2021), who simulated field scale chemical transport in groundwater. The deviations can be attributed to the consideration of adsorbing solutes in our simulations as well as the generally better performance of the algebraic multigrid preconditioner for large transport domains (Shai and Larabi, 2020). The most efficient combinations consisted of preconditioners of the successive over relaxation category (i.e., SOR and EISENSTAT) and variations of the GMRES solver. It should be noted that the parallel SOR and EISENSTAT preconditioners are equivalent to BJACOBI with SOR on each block and are therefore not true parallel preconditioners. As evident from Fig. 9 and Table 7, the SOR/GMRES combination is best suited for our application.

![Wall clock times for the ten most efficient preconditioner/solver combinations for simulation of water flow. The five most efficient combinations are represented by solid lines and symbols, the remaining five by dashed lines. The inset depicts all successful preconditioner/solver combinations.](image)

![Wall clock times for the ten most efficient preconditioner/solver combinations for simulation of water flow. The five most efficient combinations are represented by solid lines and symbols, the remaining five by dashed lines. The inset depicts all successful preconditioner/solver combinations.](image)

### Table 5
The most efficient preconditioner/solver combinations for the water flow simulations.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Preconditioner/Solver Combination</th>
<th>Normalized Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BJACOBI/DGMRES</td>
<td>1.0128</td>
</tr>
<tr>
<td>2</td>
<td>BJACOBI/GMRES</td>
<td>1.0276</td>
</tr>
<tr>
<td>3</td>
<td>BJACOBI/LGMRES</td>
<td>1.0450</td>
</tr>
<tr>
<td>4</td>
<td>BJACOBI/CG</td>
<td>1.0704</td>
</tr>
<tr>
<td>5</td>
<td>SOR/DGMRES</td>
<td>1.1407</td>
</tr>
</tbody>
</table>

### Table 6
Preconditioners and solvers considered for the solute transport simulations.

<table>
<thead>
<tr>
<th>Preconditioner</th>
<th>Abbr.</th>
<th>Solver</th>
<th>Abbr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JACOBI (diagonal scaling)</td>
<td>JACOBI</td>
<td>Transpose Free QMR</td>
<td>TFQMR</td>
</tr>
<tr>
<td>Block JACOBI</td>
<td>BJACOBI</td>
<td>A Variant of QMR</td>
<td>TCQMR</td>
</tr>
<tr>
<td>Additive Schwarz (Restrict)</td>
<td>ASM-I</td>
<td>Flexible GMRES</td>
<td>FGMRES</td>
</tr>
<tr>
<td>Additive Schwarz (Interpolate)</td>
<td>ASM-B</td>
<td>Deflated GMRES</td>
<td>DGMRES</td>
</tr>
<tr>
<td>Additive Schwarz (Basic)</td>
<td>ASM-N</td>
<td>Pipelined GMRES</td>
<td>PGMRES</td>
</tr>
<tr>
<td>Shared Blocks ASM (Restrict)</td>
<td>GASM-I</td>
<td>GMRES</td>
<td>LGMRES</td>
</tr>
<tr>
<td>Shared Blocks ASM (Interpolate)</td>
<td>GASM-B</td>
<td>Conjugate Gradient Squared</td>
<td>CGS</td>
</tr>
<tr>
<td>Shared Blocks ASM (None)</td>
<td>GASM-N</td>
<td>Stabilized Bicgstab</td>
<td>BCGS</td>
</tr>
<tr>
<td>Successive Over Relaxation (Eisenstat)</td>
<td>EISEN</td>
<td>Improved Stabilized BICG</td>
<td>IBCG</td>
</tr>
<tr>
<td>Symmetric SOR (Eisenstat)</td>
<td>BOOMERAMG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classical Algebraic Multigrid</td>
<td>PARASAIL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sparse Approximate Inverse</td>
<td>GAMG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Geometric Algebraic Multigrid</td>
<td>GAMG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Inner preconditioner for each block is $ILU(0)$, i.e., incomplete lower–upper factorization with zero level fill.
* Equivalent to BJACOBI with SOR on each block.
* From the HYPRE library. The BOOMERAMG with “HMIS” coarsen type, “ext + i” interpolation type, zero number of levels of aggressive, and 0.5 threshold value for being strongly connected.
* GAMG with type ‘aggregate’ and zero smoothing steps.
* QMR: Quasi Minimal Residual method.
* With restart parameter 60.
respectively. Equation (7) is preconditioned with the incomplete lower 18.04.5 LTS) installed on a partitioned hard drive. A workstation with water flow and solute transport with the CG and ORTHOMIN solvers, computing and the computing prowess of the latest multicore worksta tions, HYDRUS relies on the HYdrus PARallelized (HYPAR) module and the Microsoft Parallel Patterns Library to numerically solve Eq. (7) for either directly on the finite element mesh or on geometric objects in materials, initial conditions, boundary conditions, and domain properties to a total of 132,570 nodes and 726,264 3D mesh elements. Note that the hydraulic characteristics are calculated directly from the hydraulic functions by setting the lower and upper limits of the internal interpo lation tables to zero in both HYDRUS and the modified ParSWMS code. The BJACOBI/DGMRES and SOR/FGMRES preconditioner/solver com binations were used for ParSWMS water flow and solute transport simulations, respectively.

Fig. 9. Wall clock times for the ten most efficient preconditioner/solver combinations for simulation of solute transport. The five most efficient combinations are represented by solid lines and symbols, the remaining five by dashed lines. The inset depicts all successful preconditioner/solver combinations.

### Table 7
The most efficient preconditioner/solver combinations for the solute transport.

<table>
<thead>
<tr>
<th>Ranking</th>
<th>Preconditioner/Solver Combination</th>
<th>Normalized Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOR/FGMRES</td>
<td>1.0056</td>
</tr>
<tr>
<td>2</td>
<td>EISENSTAT/DGMRES</td>
<td>1.0186</td>
</tr>
<tr>
<td>3</td>
<td>EISENSTAT/GMRES</td>
<td>1.0312</td>
</tr>
<tr>
<td>4</td>
<td>EISENSTAT/LGMRES</td>
<td>1.0421</td>
</tr>
<tr>
<td>5</td>
<td>JACOBI/FGMRES</td>
<td>1.0810</td>
</tr>
</tbody>
</table>

### 6. Computational efficiency and stability of HYDRUS and the modified ParSWMS

As already indicated in Section 5, computational efficiency and stability are essential when simulating flow and transport processes in soilless plant growth modules, especially when a vast number of realizations is required. Hence, it is of interest to compare the Windows OS based HYDRUS with the Linux OS based modified ParSWMS. Because workstations and HPC clusters are not intercomparable, such compar ison is only viable on a workstation, with both Windows (i.e., Windows 10 Pro Version 1909, OS Build 18363.1016) and Linux (i.e., Ubuntu 18.04.5 LTS) installed on a partitioned hard drive. A workstation with two 2.40 GHz Intel® Xeon® E5-2630 v3 Dual 8-Core processors and 32 GB RAM was used for the comparison.

HYDRUS benefits from a powerful GUI with numerous advanced features such as the ability to create or import complex domain geometries or to specify various spatially variable properties such as mate rials, initial conditions, boundary conditions, and domain properties either directly on the finite element mesh or on geometric objects independent of the mesh (Siminek et al., 2016). To utilize parallel computing and the computing prowess of the latest multicore workstations, HYDRUS relies on the HYdrus PARallelized (HYPAR) module and the Microsoft Parallel Patterns Library to numerically solve Eq. (7) for water flow and solute transport with the CG and ORTHOMIN solvers, respectively. Equation (7) is preconditioned with the incomplete lower upper factorization method. Both the solvers and the preconditioner are from the ORTHOFEM library (Mendoza et al., 1991).

The test case for the stability and efficiency comparison conforms with the descriptions in Sections 5.1 and 5.2, except that the SWC and hydraulic conductivity functions were not scaled for the considered tuff, tuff/coconut coir, and perlite substrates (see hydraulic properties in Table 1). The simulation duration was 14 days (336 hrs). The flow domain was discretized with finite elements of 1.90-cm in the horizontal direction and 0.95-cm in the vertical direction. Mesh refinements for the top 1-cm layer and in areas with variable flux and free drainage BCs, led to a total of 132,570 nodes and 726,264 3D mesh elements. Notably, the ParSWMS solute mass balance errors are significantly lower than the HYDRUS errors.
7. Conclusions

With the goal of performing a vast number of flow and transport simulations on HPC clusters to aid the design and management of soil-less culture growth modules, we modified the open source 3D ParSWMS code to not only enable nonlinear equilibrium solute adsorption and multi-solute transport simulations, but also the application of boundary conditions to realistically represent typical soilless culture systems. The modified ParSWMS code was thoroughly tested using HYDRUS as a reference. To optimize the computational efficiency of the modified ParSWMS, numerous preconditioner/numerical solver combinations were tested for both the water flow and solute transport equations.

Fig. 10. Temporal mass balance error evolutions for water flow (left) and $H_2PO_4^-$, $NH_4^+$, and $NO_3^-$ transport (right) simulated with both HYDRUS and the modified ParSWMS code.

Fig. 11. Water content and $NO_3^-$ concentration distributions at the end of the HYDRUS and modified ParSWMS simulations. Note the localized very high $NO_3^-$ concentrations depicted in the bottom right graph for HYDRUS.
While the BJACOBI/DGMRES preconditioner/solver combination was the most efficient for water flow, SOR/FGMRES worked best for solute transport. In general, GMRs-type and CG solvers applied in conjunction with the BJACOBI preconditioner were very efficient for water flow simulations. The most efficient combinations for solute transport consisted of preconditioners of the successive over relaxation category (i.e., SOR and EISENSTAT) and variations of the GMRES solver. To compare the stability and computational efficiency of modified ParSWMS with HYDRUS, a real growth module from a production scale greenhouse experiment at the ARO Volcani Center in Israel was simulated concerning three soilless substrates and $\text{H}_2\text{PO}_4^-$, $\text{NH}_4^+$, and $\text{NO}_3^-$ transport. The simulations were performed on a workstation and two HPC clusters. The results revealed that the modified ParSWMS was about 22% more efficient than HYDRUS when the simulations on the workstation were compared. Simulations on the HPC clusters were up to 94% more efficient when 94 cores were utilized. If the full computational power of the HPC clusters (i.e., no competing projects and up to 23,616 available cores) was to be utilized, simulation times would again be drastically reduced to a very small fraction of the HYDRUS wall clock time. While we experienced some stability issues for tuff and $\text{NO}_3^-$ with HYDRUS, leading to an unrealistically high mass balance error, all modified ParSWMS simulations were stable on both the workstation and HPC clusters. In general, all water and solute mass balance errors generated with the modified ParSWMS where the mass balance errors of HYDRUS.

The availability of the modified and tested ParSWMS code opens new avenues for streamlining 3D water flow and solute transport simulations for complex porous media via the utilization of HPC environments. Simulations that took days to complete can now be resolved within a matter of a few hours, allowing for a vast number of realizations within a short period of time.

8. Availability of software

All computer codes and Matlab scripts created for this project are available from the corresponding author upon reasonable request.

CRediT authorship contribution statement

Mohammad R. Gohardoust: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - original draft, Writing - review & editing. Jirka Simůnek: Conceptualization, Methodology, Writing - review & editing. Horst Hardelauf: Conceptualization, Software, Writing - review & editing. Markus Tuller: Conceptualization, Methodology, Formal analysis, Writing - original draft, Writing - review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by The United States - Israel Binational Agricultural Research and Development Fund (BARD), grant number US-4764-14 R and by the United States Department of Agriculture (USDA) National Institute of Food and Agriculture (NIFA). Hatch/Multi-State project number ARZT-1370600-R21-189. We thank Dr. Asher Bar-Tal from the ARO Volcani Center in Israel for providing details about the design and management of the tomato growth experiment that was simulated in this study.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jhydrol.2021.126053.

References


