C-SWAT: The Soil and Water Assessment Tool with consolidated input files in alleviating computational burden of recursive simulations

Haw Yen \( ^a,^c \star \), Mehdi Ahmadi \( ^b \), Michael J. White \( ^c \), Xiuying Wang \( ^a \), Jeffrey G. Arnold \( ^c \)

\( ^a \) Blackland Research & Extension Center, Texas A\&M AgriLife Research, 720 East Blackland Road, Temple, TX 76504, USA
\( ^b \) Spatial Science Laboratory, Texas A\&M University, 1508 Research Plaza, College Station, TX 77843, USA
\( ^c \) Grassland, Soil & Water Research Laboratory, USDA-ARS, 808 East Blackland Road, Temple, TX 76502, USA

**A R T I C L E   I N F O**

Article history:
Received 17 March 2014
Received in revised form
15 May 2014
Accepted 31 July 2014
Available online 8 August 2014

**Keywords:**
SWAT
Auto-calibration
Consolidated inputs
Parallel processing
Shuffled complex evolution

**A B S T R A C T**

The temptation to include model parameters and high resolution input data together with the availability of powerful optimization and uncertainty analysis algorithms has significantly enhanced the complexity of hydrologic and water quality modeling. However, the ability to take advantage of sophisticated models is hindered in those models that need a large number of input files, such as the Soil and Water Assessment Tool (SWAT). The process of reading large amount of input files containing spatial and computational units used in SWAT is cumbersome and time-consuming. In this study, the Consolidated SWAT (C-SWAT) was developed to consolidate 13 groups of SWAT input files from subbasin and Hydrologic Response Unit (HRU) levels into a single file for each category. The utility of the consolidated inputs of model is exhibited for auto-calibration of the Little Washita River Basin (611 km\(^2\)). The results of this study show that the runtime of the SWAT model could be reduced considerably with consolidating input files. The advantage of the proposed method was further promoted with application of the optimization method using a parallel computing technique. The concept is transferrable to other models that store input data in hundreds or thousands of files.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

The Soil and Water Assessment Tool (SWAT) (Arnold et al., 1993) is a continuous-time, daily-based and semi-distributed watershed simulation model developed in prediction of hydrologic and water quality processes. The SWAT model formulates the natural watershed system by constructing the stream network, categorizing soil types, and dividing land cover into groups by applying the concept of Hydrologic Response Units (HRUs) (Neitsch et al., 2011). SWAT is being used extensively in a wide range of water use and water quality applications in the United States and around the world. These include total maximum daily loads (TMDL) analysis (Borah et al., 2006; Harmel et al., 2010; Moriasi et al., 2012), assessment of conservation practices (Liew et al., 2007; Osorio et al., 2014), and impact analysis of climate and land use changes (Chiang et al., 2010; Daggupati et al., 2011). In addition, spatial scale of the SWAT model applications ranges from

---

\( \star \) Software availability: Haw Yen, scientist & model developer at the Blackland Research and Extension Center, Texas A\&M AgriLife Research, completed the development in FORTRAN. The open-source code is available from Haw Yen, 720 E. Blackland Research Road, Temple, TX 76502, USA. Windows 7 or XP and any FORTRAN compiler are required.

* Corresponding author. Tel.: +1 254 774 6004.
E-mail address: hyen@brc.tamus.edu (H. Yen).

http://dx.doi.org/10.1016/j.cageo.2014.07.017
0098-3004/© 2014 Elsevier Ltd. All rights reserved.
majority of files in the TxtInOut are associated with the smallest computational units of the SWAT model, the HRUs. Physical and geochemical characteristics of each HRU are included in several separate input files, thus when the watershed model is delineated in a finer resolution, the number of HRUs and, accordingly, the number of files increases proportionately. The consecutive process of the opening–reading–closing of files is a time-consuming procedure during the parameter estimation, sensitivity and/or uncertainty analysis, and SWAT simulations. In addition, there is a certain amount of overhead to locate each file within the computers file system, and to reposition the mechanical components within a hard disk to read that file. This overhead time increases considerably as the number of HRUs increases. Therefore, a SWAT model with a large spatial extent and fine simulation units can be very large and may contain many files. Bosch et al. (2011) reported 94,000 SWAT input files for simulating the Western Lake Erie Basin for flow and nutrient predictions. National scale SWAT models, e.g., the Conservation Effect Assessment Project (CEAP)-National Cropland Assessment (Duriancik et al., 2008; USDA-NRCS, 2009), have employed more than 1 million files. Combined with the use of iterative calibration techniques, which may require hundreds or thousands of simulations, the file storage paradigm can be unmanageable. In order to conduct auto-calibration for a 10,000 HRUs model, which requires approximately 2 hours per simulation, with 1000 iterations would take more than 80 days. Moreover, it is likely that the auto-calibration may not be able to reach completion/convergence within 1000 iterations. Therefore any iterative algorithms, which require a large number of simulations, are considered impractical for spatially high resolution SWAT model applications. Joseph and Guillaume (2013) pointed out that the ability to use advanced approaches, such as the Bayesian Markov chain Monte Carlo (MCMC) framework, is hindered in models with a large number of input files, such as SWAT. A number of research efforts have sought to reduce computational time requirements associated with large scale environmental modeling (Duan et al., 1992; Tolson and Shoemaker, 2007; Vrugt et al., 2009; Yen, 2012). Despite the fact that commonly used distributed models have had two decades of development efforts, most efforts to reduce computational time have either been directed toward using metamodels (Johnson, 2013) or running models in parallel (Vrugt et al., 2009). The trained artificial neural networks (ANNs) and a support vector machine (SVM) were developed as surrogate models to approximate SWAT predictions, saved 20–50% in computational time. Other efforts in application of parallel processing for calibration and uncertainty analysis to the SWAT model include (Rouholahnejad et al., 2012; Whittaker, 2004; Yalew et al., 2013; Zhang et al., 2013). In application of the parallel computing, iterative process is split into multiple sub-processes, which can run simultaneously on different computational nodes, greatly reducing computational time. Growth of parallel computing systems and programming techniques sparked research on efficiently solving complex, high dimensional computational problems (Brazier et al., 2000; Freer et al., 2004).

The core software architecture and inputs structure used in SWAT has largely remained unchanged in the last decade. Although, the availability of highly detailed input data has dramatically increased, spurring the development of very large SWAT models. Researchers are seeking to utilize these detailed data with the applications of SWAT in conjunction with auto-calibration optimization techniques under more affordable computational effort (Yen, 2012; Yen et al., 2014a). Therefore, the primary goal of this paper is to present a conceptually simple, robust method for enhancing the computational speed of the SWAT model, using tools to which modelers have immediate access. With the proposed work presented in this study, we expect to reduce barriers to the use of potentially prohibitive algorithms/analyses for model applications with spatially high resolution details. Specifically, the objectives of this study are to (1) modify SWAT input structure so that all HRU and subbasin level input files were consolidated into a single file for each category (e.g. chemical, ground water, soil) for the modified SWAT, referred to as C-SWAT hereafter, and (2) test the utility of re-structured inputs in combination with a parallelized Shuffled Complex Evolution (SCE) optimization algorithm. The parallelized SCE algorithm was also coded for the original SWAT model and the efficiency of C-SWAT was compared with the original SWAT through a computationally intensive auto-calibration of streamflow for the Little Washita River Basin (LWRB) in Oklahoma, USA.

2. Methods and materials

2.1. Watershed model description: Soil and Water Assessment Tool

The Soil and Water Assessment Tool (SWAT) (Arnold et al., 2012) is a process based, distributed parameter, continuous time, and long-term watershed model that runs on a daily time step. It subdivides a watershed into subbasins connected by a stream network, and further delineates HRUs consisting of unique combinations of land cover and soils in each sub-basin. Watershed processes simulated by SWAT include snow accumulation, snow melt, evapotranspiration, infiltration, percolation losses, surface runoff, and groundwater flows (Arnold et al., 2012). SWAT can simulate major nutrient processes within a watershed. Nutrients and sediment are introduced into the main channel through surface runoff and lateral flow and transported downstream with channel flow. Nutrient transformations in the stream are controlled by the in-stream water quality component of the model that is adapted from the QUAL2E in-stream water quality model (Brown and Barnwell, 1987). More detailed description of the hydrologic sediment and nutrient components of SWAT can be found in Arnold et al. (2012).

2.2. Consolidating HRU and subbasin level input Files

SWAT users normally use ArcSWAT (Winchell et al., 2008), the ArcGIS-based user interface, to automate generation of the required inputs of SWAT model. ArcSWAT utilizes ArcGIS (ESRI, Redlands, California) to calculate all the GIS-based input data such as soil, land use, and topographic characteristics of the study watershed. These data together with user-defined management and other inputs are stored in the Access database file Watershed.mdb. The Watershed.mdb file generated by ArcSWAT contains all essential SWAT input data, which are eventually written into SWAT required input files. In Table 1, seven categories (*.chm, *.gw, *.hru, *.sep, *.ops, and *.sol) for each HRU and six categories (*.pnd, *.rte, *.sub*, .swq, *.wgn, and *.wus) are attributed for each subbasin.

To represent the multitude of HRUs in the subbasins, the 13 categories in HRU and subbasin levels may result in enormous number of input files for the original SWAT model. In this study, the Consolidated SWAT, referred to as C-SWAT hereafter, was developed to reduce hundreds or thousands of input files corresponding in each HRU or subbasin levels. The cumbersome process is simplified in C-SWAT, where the original SWAT code was modified to read only 13 consolidated files at the HRU and subbasin level. Therefore, the adjustment at the HRU and subbasin level only needs to handle a maximum of 13 files during auto-calibration. The 13 files can be easily created directly from the Watershed.mdb file described above by simply copying and pasting data (see Appendix A). In addition, MATLAB (Mathworks, 2013) codes was also developed to automatically convert inputs from original SWAT format to the C-SWAT structure (available upon request). The source code applied in this study was based on...
SWAT2009 (rev477); however it can be integrated into the latest SWAT2012 code (rev610) with no major issues because the basic structure in opening/reading/closing major input files remained the same. In this study, four major structures of TxtInOut can be found in Appendix A.

2.2.1. Structure I

Structure I is associated with ten consolidated files (*.gw, *.hru, *.sep, *.sol, *.pnd, *.rte, *.sub, *.swq, *.wgn, and *.mgt files except the operation management section). As shown in Fig. 1 (illustrated from a *.hru file), the format of Structure I is that each row of an input file contains only one representative parameter value. For parameters in some HRU and subbasin level files, a matrix for the specific parameter in the original source code is a one-dimension matrix since at least one parameter value will be assigned to a single file.

2.2.2. Structure II

Structure II can be found in six consolidated files (*.chm, *.sol, *.rte, *.sub, *.wgn, and *.wus). As shown in Fig. 2 (illustrated from a *.chm file), the format of Structure II is that each row of an input file contains more than one representative parameter values. For parameters in HRU level files, a matrix for the specific parameter in the source code is a two-dimension matrix.

2.2.3. Structure III

Structure III can be only found in the lower part (operations) of *.mgt files where the upper part of *.mgt files is in Structure I. As shown in Fig. 3 (illustrated from a *.mgt file), the format of Structure III is that each location has specific variables assigned, and the file has no fixed length. More detailed format of *.mgt files is shown in Table 2. The operations section of the *.mgt uses the first four columns to schedule and define operations, the 5th to 13th columns share 37 parameters in these nine columns to define parameters associated with differing operation types (see Appendix A). In the Watershed.mdb file, these 37 parameters for management practices are stored in 37 columns respectively. However, the parameter value will be zero or null if a specific parameter is not required for the row defining each operation.

2.2.4. Structure IV

Structure IV can be only found only in *.ops files. As shown in Fig. 4 (illustrated from a *.ops file), the format of Structure IV is that each location has a specific variable assigned. The detail format of *.ops files is shown in Table 3. The first four columns are month (MON), day (DAY), year (IYEAR), and management operation number (MGT_OP). The 5–11th columns share 28 parameters in some HRU and subbasin level files, a matrix for the specific parameter in the original source code is a one-dimension matrix since at least one parameter value will be assigned to a single file.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>SWAT input files in HRU and subbasin levels.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SWAT input files</td>
</tr>
<tr>
<td></td>
<td>Structure Group</td>
</tr>
<tr>
<td></td>
<td>HRU level file</td>
</tr>
<tr>
<td></td>
<td>*.chm II</td>
</tr>
<tr>
<td></td>
<td>*.gw I</td>
</tr>
<tr>
<td></td>
<td>*.hru I</td>
</tr>
<tr>
<td></td>
<td>*.mgt I/III</td>
</tr>
<tr>
<td></td>
<td>*.ops IV</td>
</tr>
<tr>
<td></td>
<td>*.sep I</td>
</tr>
<tr>
<td></td>
<td>*.sol I/II</td>
</tr>
<tr>
<td></td>
<td>Subbasin level file</td>
</tr>
<tr>
<td></td>
<td>*.pnd I</td>
</tr>
<tr>
<td></td>
<td>*.rte I/II</td>
</tr>
<tr>
<td></td>
<td>*.sub I/II</td>
</tr>
<tr>
<td></td>
<td>*.swq I</td>
</tr>
<tr>
<td></td>
<td>*.wgn I/II</td>
</tr>
<tr>
<td></td>
<td>*.wus II</td>
</tr>
</tbody>
</table>

It is possible that a single SWAT input file with more than one structure.

![Fig. 1. Structure I: each row contains only one parameter value.](image1)

![Fig. 2. Structure II: each row contains more than one parameter values.](image2)
parameters in these seven locations (see Appendix A). In the Watershed.mdb file, 28 parameters for management practices are available in 28 columns respectively.

2.3. Case study

The Little Washita River Basin (LWRB) (611 km²) is located in southwestern Oklahoma (Fig. 5). The elevation of the basin ranges between about 300–500 m. The bedrock exposed in the watershed consists of Permian age sedimentary rocks and soil textures range from fine sand to silty loam. The main land cover in LWRB is range, pasture, forest and cropland. The climate is characterized as moist and sub-humid with a spatially average, annual precipitation of 760 mm and temperature of 16°C.

A SWAT model was developed using a 10-m resolution digital elevation model (DEM) (USGS, 2013), Cropland Data Layer 2012.

Table 2

<table>
<thead>
<tr>
<th>MON DAY</th>
<th>HUSC</th>
<th>MGT_OP</th>
<th>mgt1</th>
<th>mgt2</th>
<th>mgt3</th>
<th>mgt4</th>
<th>mgt5</th>
<th>mgt6</th>
<th>mgt7</th>
<th>mgt8</th>
<th>mgt9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plant/begin growing season</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>PLANT_ID</td>
<td>-</td>
<td>CURYR_MAT</td>
<td>HEAT_UNITS</td>
<td>LAL_INIT</td>
<td>BIO_INIT</td>
<td>HI_TARG</td>
<td>BIO_TARG</td>
</tr>
<tr>
<td>Irrigate</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>IRR_AMT</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Fertilizer application</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>FERT_ID</td>
<td>-</td>
<td>-</td>
<td>FRT_KG</td>
<td>FRT_SURFACE</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Pesticide application</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>PEST_ID</td>
<td>-</td>
<td>-</td>
<td>PST_KG</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Harvest/kill operation</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>CNOP</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Tillage operation</td>
<td>-</td>
<td>-</td>
<td>6</td>
<td>TILL_ID</td>
<td>-</td>
<td>-</td>
<td>CNOP</td>
<td>HARV_EFF</td>
<td>HI_OVR</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Harvest operation</td>
<td>-</td>
<td>-</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kill/end growing season</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grazing</td>
<td>-</td>
<td>-</td>
<td>9</td>
<td>GRZ_DAYS</td>
<td>MANUR_E_ID</td>
<td>-</td>
<td>BIO_EAT</td>
<td>BIO_TRMP</td>
<td>MANUR_E_KG</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Auto irrigation</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>WSTRS_ID</td>
<td>-</td>
<td>-</td>
<td>AUTO_WSTRS</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Auto fertilization</td>
<td>-</td>
<td>-</td>
<td>11</td>
<td>AFERT_ID</td>
<td>-</td>
<td>-</td>
<td>AUTO_NSTRS</td>
<td>AUTO_NPYR</td>
<td>AUTO_EFF</td>
<td>AUTO_EFF</td>
<td>MANUR_E_KG</td>
</tr>
<tr>
<td>Sweep operation</td>
<td>-</td>
<td>-</td>
<td>12</td>
<td>-</td>
<td>IMP_TRIG</td>
<td>-</td>
<td>-</td>
<td>SWEEPEFF</td>
<td>FR_CURB</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Release/impound</td>
<td>-</td>
<td>-</td>
<td>13</td>
<td>-</td>
<td>FERT_DAYS</td>
<td>CFRID</td>
<td>-</td>
<td>FRT_FREQ</td>
<td>CFRID</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Continuous fertilization</td>
<td>-</td>
<td>-</td>
<td>14</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3. Structure III: MGT files.
and Soil Survey Geographic database (SSURGO) (USDA NRCS, 2012). The watershed was subdivided into 75 sub-basins and a total of 1109 hydrologic HRUs. Temperature and precipitation data from 20 weather data stations were used. Overall, 20 SWAT parameters were selected that are relevant in regard to their ability to affect hydrologic fluxes and are typically used in model calibration of the SWAT models (Table 4). Daily streamflow data was available at the watershed outlet from a USGS gaging station (USGS gauge no. 07327550). In case study, simulations were done for daily streamflow calibration starting from 2006 through 2010 (5 years), including 1 year of warm-up.

### 2.3.1. Parameter estimation technique

The Shuffled Complex Evolution (SCE) method (Duan et al., 1992, 1994) was used for parameter estimation for this C-SWAT model. SCE is a global optimization algorithm that synthesizes the concepts of deterministic and probabilistic, controlled random search, competitive evolution, and complex shuffling approaches. SCE has been successfully used for calibration of the SWAT model (Sharma et al., 2006; Zhang et al., 2009). A general description of the steps for SCE method with $n$ parameters, $p$ complexes, and $m$ points in each complex is given as follows (Duan et al., 1992; Duan et al., 1994): (1) generate sample: randomly sample $s (= pm)$ initial population of parameter sets $\theta$ from the feasible parameter space $\Theta$ and compute the criterion value at each point, (2) rank points: sort the $s$ points with increasing criterion value, (3) partition into complexes: partition the $s$ points into $p$ complexes; each containing $m$ points, (4) complex evolution: evolve each complex according to the competitive complex evolution (CCE), (5) shuffle complexes: combine the points in the evolved complexes into a single sample population, sort the sample population in order of increasing criterion value and shuffle the sample population into $p$ complexes according to the procedure specified in Step (3), and

![Fig. 4. Structure IV: OPS files.](image)

![Fig. 5. Location of the Little Washita River Basin.](image)

Table 3  
Structure IV: fixed locations for multiple parameters in *.ops files (Neitsch et al., 2011).

<table>
<thead>
<tr>
<th>MON</th>
<th>DAY</th>
<th>IYEAR</th>
<th>MGT_OP</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terracing operation</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>TERR_P</td>
<td>TERR_CN</td>
<td>TERR_SL</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Tile drainage operation</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>2</td>
<td>DRAIN_D</td>
<td>DRAIN_T</td>
<td>DRAIN_G</td>
<td>DRAIN_IDEP</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Contouring operation</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>3</td>
<td>CONT_CN</td>
<td>CONT_P</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Filter strip operation</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>4</td>
<td>FILTER_J</td>
<td>FILTER_RATIO</td>
<td>FILTER_CON</td>
<td>FILTER_CH</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Strip cropping operation</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>5</td>
<td>STRIP_N</td>
<td>STRIP_CN</td>
<td>STRIP_C</td>
<td>STRIP_P</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Fire operation</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>6</td>
<td>FIRE_CN</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Grassed waterways operation</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>7</td>
<td>GWATI</td>
<td>GWATN</td>
<td>GWATSPCON</td>
<td>GWATD</td>
<td>GWATW</td>
<td>GWATL</td>
</tr>
<tr>
<td>Plant parameter update</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>8</td>
<td>CROPNO_UPD</td>
<td>HI_UPD</td>
<td>LAIMX_UPD</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
Check convergence: if any of the pre-specified convergence criteria are satisfied, stop the process; otherwise, return to Step (4). Given the current developments of parallel and distributed computing technologies, overall runtime can be reduced considerably for optimization methods that can utilize individual parallel processing units. The SCE method is inherently well suited for parallelization within each individual complex.

Schematic diagram of the calibration procedure of SWAT model is depicted in Fig. 6. For consistency, 5000 model parameter sets were evaluated for all optimization configurations.

2.3.2. Objective functions

The Nash–Sutcliff Efficiency (NSE) (Nash and Sutcliffe, 1970) was used as the objective function during auto-calibration where it measures the relative magnitude of the residual variance compared to the observed data variance. It indicates how well the plot of observed versus simulated values fits a 1:1 line. The coefficient can range from \(-\infty\) to a perfect match of +1:

\[
\text{NSE} = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}
\]

where \(\hat{y}_i\) and \(y_i\) denote, respectively, the simulated and observed responses for time step \(i\), \(n\) is the number of observations, and \(\bar{y}\) denotes the mean of observed responses. NSE has been recommended for use by the American Society of Civil Engineers (ASCE) (ASCE, 1993) and is a commonly used statistical measure for calibration of watershed models. Servat and Dezetter (1991) also found NSE to be the best objective function for reflecting the overall fit of a hydrograph.

2.3.3. Implementation of the C-SWAT and parallel SCE

In case study, four major steps are conducted to implement C-SWAT. Firstly, the SWAT project was created through ArcSWAT, which subdivided the watershed into 75 subbasins and a total of 394 HRUs. This resulted in a total of 3148 (394 \times 7 + 65 \times 6 = 3148) files at the HRU and subbasin level for the original SWAT model saved in the TxtInOut project directory.

Secondly, the source data file watershed.mdb is opened and created input files for C-SWAT. This results in a total of 7 files at the

<table>
<thead>
<tr>
<th>Parameter number</th>
<th>Parameter name</th>
<th>Input file</th>
<th>Description</th>
<th>Unit</th>
<th>Lower bound</th>
<th>Upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ALPHA_BF .gw</td>
<td>Base flow alpha factor for recession constant</td>
<td>1/days</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CH_KII .rte</td>
<td>Fraction change in hydraulic conductivity in the main channel</td>
<td>mm/h</td>
<td>–0.01</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CH_NII .rte</td>
<td>Manning's n value for the main channels</td>
<td>–</td>
<td>0.01</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CH_SII .rte</td>
<td>Average slope of main channel along the channel length</td>
<td>%</td>
<td>–5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CH_F .mgt</td>
<td>Fraction change in SCS runoff curve number</td>
<td>%</td>
<td>–10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>FILTERW .mgt</td>
<td>Width of edge-of-field filter strip</td>
<td>m</td>
<td>0</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>GW_DELAY .gw</td>
<td>Groundwater delay time</td>
<td>day</td>
<td>0</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>OW_N .hru</td>
<td>Manning's n value for overland flow</td>
<td>–</td>
<td>0.01</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>SFTMP .bsn</td>
<td>Snow melt base temperature</td>
<td>°C</td>
<td>–5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>SLOPE .hru</td>
<td>Average slope steepness</td>
<td>%</td>
<td>–10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>SLSUBBSN .hru</td>
<td>Average slope length</td>
<td>m</td>
<td>10</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>SMFMN .bsn</td>
<td>Minimum melt rate for snow</td>
<td>mm/°C-day</td>
<td>0</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>SMFMX .bsn</td>
<td>Maximum melt rate for snow</td>
<td>mm/°C-day</td>
<td>0</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>SMTMP .bsn</td>
<td>Snow melt base temperature</td>
<td>°C</td>
<td>–5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>SNOCOV .bsn</td>
<td>Snow water equivalent that corresponds to 50% snow cover</td>
<td>mm</td>
<td>0</td>
<td>650</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>SNOCOVMX .bsn</td>
<td>Minimum snow water content that corresponds to 100% snow cover</td>
<td>mm</td>
<td>0</td>
<td>650</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>SOL_K .sol</td>
<td>Fraction change in saturated hydraulic conductivity</td>
<td>%</td>
<td>–50</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>SURLAG .bsn</td>
<td>Surface runoff lag time</td>
<td>day</td>
<td>1</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>TIMP .bsn</td>
<td>Snow pack temperature lag factor</td>
<td>–</td>
<td>0.01</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>DEP_IMP .hru</td>
<td>Depth to impervious layer</td>
<td>mm</td>
<td>1500</td>
<td>2500</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 6. Schematic chart of calibration processes.
processors resulted in increased idle time for processor cycles as they hard disk drive I/O workload limitation. In addition, adding more speedup may be attributed to the limited shared memory and limited 26% lower than the ideal speedup line, respectively. This loss of processes resulted in 6.8 and 8.9 times speedup, which are 14% and deviate from ideal line. For example, calibration with 8 and 12 parallel processes resulted in 30% faster than calibration using original SWAT. Fig. 7 depicts the speedup obtained by a combined use of C-SWAT and parallel computing for calibration of the LWRB model. In all single and 4 parallel processes. The highest speedup rate was seen for single core of CPU in computer) using both original and C-SWAT models to compare their computational efficiencies. The single program multiple data (SPMD) construct was used to implement parallel computation of the SCE calibration algorithm. The SPMD construct defines a block of code that runs in parallel on all the CPU cores (Mathworks, 2013). Speedup of each configuration was computed based on its overall runtime and runtime of the calibration on a single core of original SWAT model. Each of the parallel calibration runs was repeated two times on the same machine (a total of 2 × 7 = 14 trials) and the average values were reported in this study. For consistency, the runtime for a total number of 5000 simulations was evaluated for each of the original SWAT and C-SWAT optimization configuration. Note that global optimization algorithms, such as SCE, may require a very high number of simulations for convergence and our purpose here was not to pursue calibration results rather than to compare the computational speed of C-SWAT vs. original SWAT.

2.4. Hardware configuration and implementation

The parallel algorithm implementation for SCE was configured on a multicore server with two 6-core, Intel® Xeon® ES645 CPUs with overall 24 threads and 24 GB of RAM. The server was running a 64 bit Windows Server 2008 R2 Enterprise. Both the original SWAT (rev477) and C-SWAT models were run in parallel with the SCE optimization. The source code applied in this study can be integrated into the latest SWAT code with no major issues. In addition, the parallelization can be used on a single machine with multiple threads or processors. Therefore, anyone having a desktop with multiple threads or processors has opportunities to conduct otherwise prohibitive high number of simulations by using C-SWAT in parallel. The basic structure in opening–reading–closing major input files remained the same.

3. Results and discussion

In average, calibration of the LWRB using C-SWAT model was approximately 30% faster than calibration using original SWAT. Fig. 7 depicts the speedup obtained by a combined use of C-SWAT and parallel computing for calibration of the LWRB model. In all single and parallel configurations, C-SWAT improved overall runtime of the calibration by 28–35%. As the number of parallel processors increased, calibration speedup also increased (Fig. 7); however, the observed speedup was not linear. Comparison of calibration speedup using original SWAT model with 1:1 line; that is the ideal speedup shows that calibration runs with more than 4 parallel processes tend to deviate from ideal line. For example, calibration with 8 and 12 processes resulted in 6.8 and 8.9 times speedup, which are 14% and 26% lower than the ideal speedup line, respectively. This loss of speedup may be attributed to the limited shared memory and limited hard disk drive I/O workload limitation. In addition, adding more processors resulted in increased idle time for processor cycles as they wait for other parallel runs within each generation. Increasing processors will excessively degrade the parallel runs from its expected ideal speedup.

However, use of C-SWAT compensated for the loss of speedup. Observed speedup in all C-SWAT parallel runs was on or above the ideal speedup line. Similar to the application of original SWAT, speedup of the calibration degraded from the ideal with more than 4 parallel processes. The highest speedup rate was seen for single serial run, with 34% speedup. The highest deviation from ideal speed line was observed for calibration with 4 parallel processors. Using 12 parallel processors, the actual performance of C-SWAT was compared with the ideal performance of standard SWAT.

Profiling a single simulation of original and C-SWAT showed that consolidated inputs reduced runtime for three major stages of calibration, including (i) file copying, (ii) modifying parameters, and (iii) running SWAT model, by more than 36% (Fig. 8). The highest reduction was seen in modifying parameter values in consolidated inputs where runtime was reduced by 76% from 22 s to less than 5.5 s. SWAT runtime was also reduced by less than 7%. Copying files during each iteration was also faster in C-SWAT by 60%. Faster copying and parameter alteration was directly related to the consolidated inputs that reduced the number of files for copying and reading/writing during the calibration process. Faster C-SWAT run was also due to the limited number of inputs. Depending on the size of the project and simulation period, these results can be different.
4. Conclusion and future development of SWAT

Inputs of the SWAT model were consolidated in C-SWAT and model structure was modified to reduce number of HRU and subbasin level files to only 13 files. Improvement of the modified structure was presented in calibration of the streamflow data in the Little Washita River Basin, Oklahoma, USA, using a parallelized Shuffled Complex Evolutionary algorithm. The calibration was examined on different parallel processes ranging from 1 to 12 pools. The results showed that a single simulation using C-SWAT was 30% faster than the original SWAT model in this specific case study. However, the computation effort can be further reduced in larger case studies with higher number of input files. Application of C-SWAT in a parallel computing platform decreased the runtime of the calibration considerably. In addition, the proposed changes are potentially beneficial to large scale studies, web-based technologies and iterative computation in reducing the computational burden.

The SWAT model is currently being restructured into a more modular format to improve maintainability and to allow development by multiple scientists. The main data module used to transfer all data between all subroutines has been replaced by individual modules for each spatial object including subbasins, HRUs, channels, and reservoirs. The only data passed between spatial object modules are data structures for: (1) flow, sediment, and related transport constituents, (2) weather, and (3) time. A major change in model structure is the separation of the HRU into aquifer and soil/plant modules. In addition to the new spatial module structure, inputs are being transformed into a relational database format. To define HRUs, a new HRU data file points to weather, soils, and management information that defines the HRU. For example, all soils are appended to a single file and each HRU points to the appropriate soil within that file. We are also restructing the process routines into subroutines with all input and output variables defined as call arguments. This will facilitate multiple developers working together in the same library of subroutines and facilitate parallelization.

Acknowledgments

The development of C-SWAT started from an independent research work conducted at the Colorado State University by Dr. Haw Yen. By knowing the motivation of Dr. Yen’s idea, Dr. Larry A Roesner, the founder of the SWMM model has supported the Compaq Visual FORTRAN 6.6 license with no hesitation. In addition, contributions were made by Dr. John W Labadie and Dr. Mazdak Arabi with accommodating suggestions. Without all the helping hands above, the manuscript will never be published. Later on, the development of C-SWAT was further supported by the funding from the USDA-NRCS through the Conservation Effects Assessment Project (CEAP). This study was also jointly supported by the International S&T Cooperation Program from the Ministry of Science and Technology of China (2012DFA91530), the National Natural Science Foundation of China (41661140353, 91325302), the First Youth Excellent Talents Program of the Organization Department of the Central Committee of the CPC, and the Fundamental Research Funds for the Central Universities (TD-JC-2013-2). The authors gratefully appreciate the constructive comments from the reviewers and associate editor that improved the presentation of the study. Thanks a lot!

Appendix A

Parameters included in each SWAT input files mentioned in Table 1 are listed as follows.

1. [HRU Level] There are seven major HRU level files in the TxtInOut directory (see Table 1). The structure of open/read/ close in the FORTRAN source code is very similar in *.chm, *.gw, *.hru, *.sep, and *.sol. On the other hand, the coding routine is quite different in *.mgt and *.ops where the operation practices require specific loops in order to proceed the simulation processes.

2. [HRU Level] CHM Files (*.chm) – Structure II

3. [Subroutine Added] – allocate_allsysystem.f/readallchm.f

4. [Subroutine Modified] – main.f/hruallo.f/modparm.f/readsub.f/ readchm.f

1. (Physical group)

   SOL_NO3(1–10), SOL_ORGN(1–10), SOL_SOLP(1–10), SOL_ ORGP(1–10)

2. (Chemical group)

   PESTNUM(1–10), PPLSTP(1–10), SOLSTP(1–10), PSTENR(1–10)

Parameters in *.chm files can be categorized into two groups: physical and chemical characteristics (Arnold et al., 2011). There are 40 inputs (4 parameters with 10 layers each) in the first group and 40 (4 parameters with 10 different pesticide if assigned) in the second. In the first group, each parameter has 10 values according to the layer number (soil layer from 1 to 10). In the second group, each parameter may have more than one value according to the pesticide applied (the maximum is ten types of pesticide). In this case, total 80 parameters (see Appendix A) in the *.chm files are allocated in the “chm” table of the Watershed.mdb file by the order of HRU number. In C-SWAT, parameter information of CHM files is read from a single file named allchm.txt with inclusion of all information from the “chm” table of the Watershed.mdb. Parameter information will be first read by subroutine readallchm.f and saved in a single matrix. Next, parameter information originally read in the subroutine readchm.f will be substituted by passing values from the newly generated matrix instead of open/read/close each *.chm file.

2. [HRU Level] GW files (*.gw) – Structure I

   SOL_NO3(1–10), SOL_ORGN(1–10), SOL_SOLP(1–10), SOL ORGP(1–10)

   Parameters in *.gw can be categorized into unconfined and confined aquifer (Arnold et al., 2011). A total of 13 parameters (see Appendix A) located in the *.gw files are allocated in the “gw” table of the Watershed.mdb file by the order of HRU number. In C-SWAT, parameter information of GW files is read from a single file named allgw.txt with inclusion of all information from the “gw” table of the Watershed.mdb. Parameter information will be first read by subroutine readallgw.f saved in a single matrix. Next, parameter information originally read in the subroutine readgw.f will be substituted from passing values from the newly generated matrix instead of open/read/close each *.gw file.

3. [HRU Level] HRU files (*.hru) – Structure I

   SOL_NO3(1–10), SOL_ORGN(1–10), SOL_SOLP(1–10), SOL ORGP(1–10)

   Parameters in *.hru files can be categorized into five groups: topographic characteristics, water flow, erosion, land cover, and digressional storage areas (Arnold et al., 2011). A total of 24 parameters (see Appendix A) located in the “hru” files are allocated in the “hru” table of the Watershed.mdb file by the order of HRU number. In C-SWAT, parameter information for the HRU files is read from a single file named allhru.txt with
inclusion of all information from the “hru” table of the Watershed.mdb. Parameter information will be first read by subroutine readallhruf.saved in a single matrix. Next, parameter information originally read in the subroutine readhruf will be substituted by passing values from the newly generated matrix instead of open/read/close each *.hru file.

4. [HRU Level] MGT files (*.mgt) – Structure I & III
   [Subroutine Added] – allocate_allsystem.f/readallmgt01.f/readallmgt02.f
   [Subroutine Modified] – main.f/hruallo.f/modparm.f/readsub.f/readmt.f

(1) Initialization inputs
   NMGT, IGRO, PLANT_ID, LAI_INIT, BIO_INIT, PHU_PLT, BIOMIX, CN2, USLE_P, BIOMIN, FILTERW, IURBAN, URBLU, IRRSC, IRMNO, FLOWMIN, DIVMAX, FLOWFR, DDRAIN, TDRAIN, GDRAIN, NROT

(2) Management operation schedules
   (i) Basic information
       MONTH, DAY, HUSC, MGT_OP
   (ii) Operational practices
       PLANT_ID, FERT_ID, PEST_ID, TILL_ID, GRZ_DAYS, WSTSTRS_ID, AFERT_ID, IMP_TRIG, FERT_DAYS, MANURE_ID, CFRT_ID, CURR_YR_MAT, IFRT_FREQ, HEAT_UNITS, IRR_AMT, FRT_KG, PST_KG, CNOP, HARVEFF, BIO_EAT, AUTO_WSTRS, AUTO_NSTRS, SWEEPEFF, CFRT_KG, LAI_INIT, FRT_SURFACE, HI_OVR, BIO_TRMP, AUTO_NAPP, FR_CURB, BIO_INIT, MANURE_KG, AUTO_NYR, HI_TARG, AUTO_EFF, BIO_TARG, AFRT_SURFACE

Parameters in *.mgt files can be categorized into two groups: initialization inputs and management operation schedules (Arnold et al., 2011). In the first group, each HRU has only one input item (e.g. CN2: 83.00). A total of 22 parameters (see Appendix A) located in the *.mgt files are allocated in the “mgt1” table of the Watershed.mdb file by the order of HRU number. In the second group, each HRU has only one input item (e.g. soil physical and chemical characteristics (Arnold et al., 2011). In the second group, each parameter is specifically for single input (MONTH, DAY, IYEAR, MGT_OP). However, there are 37 parameters sharing 9 locations from location number 5 to 13. Therefore, total (4 + 37) parameters (see Appendix A) in the *.mgt files are allocated in the “mgt2” table of the Watershed.mdb file by the order of HRU number. In addition, more rows will be added in the “mgt2” Table if the number of rotation years is more than one. In C-SWAT, parameter information of MGT files is read from two separate files named allmgt01.txt and allmgt02.txt with inclusion of all information from the “mgt1” and “mgt2” tables of the Watershed.mdb. Parameter information will be first read by subroutine readallmgt1.f and readallmgt2.f, and then saved in two matrices. Next, parameter information originally read in the subroutine readmt.f will be substituted by passing values from the newly generated matrix instead of open/read/close each *.mgt file.

5. [HRU Level] OPS files (*.ops) – Structure IV
   [Subroutine Added] – allocate_allsystem.f/readallops.f
   [Subroutine Modified] – modparm.f/readsub.f/readops.f

(1) Basic information
   MONTH, DAY, IYEAR, MGT_OP

(2) Operational practices
   Terracing operation: TERR_P, TERR_CN, TERR_SL.
   Drainage operation: DRAIN_D, DRAIN_T, DRAIN_G, DRAIN_N_IDEP.
   Contouring operation: CONT_CN, CONT_P.
   Filter strip operation: FILTER_I, FILTER_RATIO, FILTER_CON, FILTER.CH.
   Strip cropping operation: STRIP_N, STRIP_CN, STRIP_C, STRIP_P.
   Fire operation: FIRE_CN.

Grassed waterways operation: GWATI, GWATN, GWATSPCON, GWATD, GWATW, GWATL, GWATS.

Parameter information update: CROPNO_UPD, HL_UPD, LAIMX_UPD.

The scheduled management operations (*.ops) are developed to simulated eight types of structural practices which include terracing operation, tile drainage, contouring, filter strip, strip cropping, fire, grassed waterways, and plant parameter update (Arnold et al., 2011). There are 11 fixed length locations for 11 parameter inputs. The first 4 locations are specifically for single input (MONTH, DAY, IYEAR, MGT_OP). The following 7 locations are for different operation practices and not all locations are filled in every operation (e.g. 7 parameters are required for grass waterways operation but only one for fire operation). Therefore, total (4 + 28) parameters (see Appendix A) in the *.ops files are allocated in the “ops” Table of the Watershed.mdb file by the order of HRU number. In addition, more rows will be added in the “ops” Table if the number of operation practices is more than one. In C-SWAT, parameter information of OPS files is read from a single file named allops.txt with inclusion of all information from the “ops” table of the Watershed.mdb. Parameter information will be first read by subroutine readallops.f, and then saved in a single matrix. Next, parameter information originally read in the subroutine readops.f will be substituted by passing values from the newly generated matrix instead of open/read/close each *.ops file.

6. [HRU Level] SEP files (*.sep) – Structure I
   [Subroutine Added] – allocate_allsystem.f/readallslep.f
   [Subroutine Modified] – modparm.f/readsub.f/readsepticbz.f/ipop_sep, isep_typ, bZ_z, bZ_THK, bZ_AREA, bIO BD, COEFF_BOD DC, COEFF_BOD_CONV, COEFF_FEC, COEFF_FC1, COEFF_FC2, COEFF_FEC. 
   Parameters in *.sep files can be categorized into four groups: septic system types, biozone geometry, biomass characteristics, and coefficients of bio-physical reaction (Arnold et al., 2011). A total of 18 parameters (see Appendix A) located in the *.sep files are allocated in the “sep” table of the Watershed.mdb file by the order of HRU number. In C-SWAT, parameter information of SEP files is read from a single file named allsep.txt with inclusion of all information from the “sep” table of the Watershed.mdb. Parameter information will be first read by subroutine readallslep.f, and saved in a single matrix. Next, parameter information originally read in the subroutine readsep.f will be substituted by passing values from the newly generated matrix instead of open/read/close each *.sep file.

7. [HRU Level] SOL files (*.sol) – Structures I and II
   [Subroutine Added] – allocate_allsystem.f/readallsof.f
   [Subroutine Modified] – main.f/hruallo.f/modparm.f/readsub.f/readsol.f

(1) Physical group
   SNAM, HYDGRP, SOL_ZMX, ANION_EXCL, SOL_CRK, TEXTURE.

(2) Chemical group
   SOL_Z, SOL BD, SOL_AWC, SOL K, SOL CBN, SOL CLAY, SOL Silt, SOL SAND, SOL ROCK, SOL ALB, USLE K, SOL EC.
   Parameters in *.sol files can be categorized into two groups: physical and chemical characteristics (Arnold et al., 2011). There are 6 inputs in the first group and 12 in the second. In the first group, each HRU has only one input item (e.g. soil hydrologic group: B). In the second group, each parameter may have more than one value according to how many layers are existed in the soil body (e.g. SOL Z:mm: 430.00 710.00 2030.00). In this case, the following 11 parameters are also carrying the same number of values depending on the layer. Assume there are N soil layers, total (6 + 12 x N) parameters (see...
Appendix A) in the *.hru files are allocated in the “sol” table of the Watershed.mdb file by the order of HRU number. In C-SWAT, parameter information of SOL files is read from a single file named alloSoltxt with inclusion of all information from the “sol” table of the Watershed.mdb. Parameter information will be first read by subroutine readallsol.f and saved in a single matrix. Next, parameter information originally read in the subroutine readso1.f will be substituted by passing values from the newly generated matrix instead of open/read/close each *.sol file.

II. [Subbasin Level Files]. In addition to HRU level files, there are six major subbasin level files in the TxtInout directory (see Table 1). The structure of open/read/close in the FORTRAN source code is very similar in *.chl, *.gw, *.hru, *.sep, and *.sol. On the other hand, the coding routine it is quite different in *.mgt and *.ops where the operation practices require specific loops in order to proceed the simulation processes.

1. [Subbasin Level] PND files (*.pnd) – Structure I
   [Subroutine Added] – allocate_allsystem.f/readallpnd.f
   [Subroutine Modified] – modparm.f/readsub.f/readpnd.f

   (1) Pond group
   PND_FR, PND_PSA, PND_PVOL, PND_ESA, PND_EVOL, PND_VOL, PND_SED, PND_NSED, PND_K, IFLOD1, IFLOD2, NDTARG, PSETLPI1, PSETLPI2, PSETLP2, CHLAP, SECCIP, PND_NO3, PND_SOLP, PND_ORGN, PND_ORGP, PND_D50, IPND1, IPND2.

   (2) Wetland group
   WET_FR, WET_NSA, WET_NVOL, WET_MXSA, WET_MXVOL, WET_VOL, WET_SED, WET_NSED, WET_K, PSETLW1, PSETLW2, NSETLW1, NSETLW2, CHLAW, SECCIW, WET_NO3, WET_SOLP, WET_ORGN, WET_ORGP, PNDEVCOEFF, WETCOEFF.

   Parameters in *.pnd files can be categorized into two groups: pond and wetland (Arnold et al., 2011). A total of (25+21) parameters (see Appendix A) locates in the *.pnd files are allocated in the “pnd” table of the Watershed.mdb file by the order of HRU number. In C-SWAT, parameter information of PND files is read from a single file named allpnd.txt with inclusion of all information from the “pnd” table of the Watershed.mdb. Parameter information will be first read by subroutine readallpnd.f and saved in a single matrix. Next, parameter information originally read in the subroutine readpnd.f will be substituted by passing values from the newly generated matrix instead of open/read/close each *.pnd file.

2. [Subbasin Level] RTE Files (*.rte) – Structures I and II
   [Subroutine Added] – allocate_allsystem.f/readallrte.f
   [Subroutine Modified] – modparm.f/readsub.f/readrte.f

CHW2, CHD, CH_S2, CH_L2, CH_N2, CH_K2, CH_COV1, CH_COV2, CH_WDR, ALPHABNK, ICANAL, CH_ONCO, CH_OPCO, CH_SIDE, CH_BNK_BD, CH_BED_BD, CH_BNK_KD, CH_BED_KD, CH_BNK_D50, CH_BED_D50, CH_BNK_TC, CH_BED_TC, CH_ERODMO(1–10), CH_EQN.

Parameters in *.rte files can be categorized into three groups: sediment, nutrient, and pesticide (Arnold et al., 2011). A total of (23+10) parameters (see Appendix A) located in the *.rte files are allocated in the “rte” table of the Watershed.mdb file by the order of HRU number. In C-SWAT, parameter information of RTE files are read from a single file named allrte.txt with inclusion of all information from the “rte” table of the Watershed.mdb. Parameter information will be first read by subroutine readallrte.f and saved in a single matrix. Next, parameter information originally read in the subroutine readrte.f will be substituted by passing values from the newly generated matrix instead of open/read/close each *.rte file.

3. [Subbasin Level] SUB Files (*.sub) – Structures I and II
   [Subroutine Added] – allocate_allsystem.f/readallsub.f
   [Subroutine Modified] – modparm.f/readsub.f/readsub.f

(1) Climate in subbasin
   SUB_KM, SUB_LAT, SUB_ELEV, IRGAGE, ITGAGE, ISGAGE, IHGAGE, IWGAGE, FCST_REG.

(2) Elevation bands
   ELEV(B(1–10), ELEV_B(FR(1–10), SNOEB(1–10).

(3) Precipitation/temperature and channels
   PLAPS, TLAWS, SNO_SUB, CH_L1, CH_S1, CH_W1, CH_K1, CH_N1, CO2.

(4) Climate change information
   RFINC(1–12), TMPRFINC(1–12), RADDRFINC(1–12), HUMRFINC(1–12).

(5) Total number of HRUs modeled in subbasin
   HRUTOT

Parameters in *.sub files can be categorized into six groups: subbasin size/location, climate data specified in subbasin, topographic relief within subbasin, tributary details within subbasin, climate change variables, names of HRU input files (Arnold et al., 2011). There are five parts in the *.sub file: a total of 9 parameters in the first part, 30 inputs (3 parameters with 10 values of each) in the second part, 9 parameters in the third part, 48 inputs (4 parameters with 12 values of each) in the fourth part and one parameter in the last part. A total of 96 parameters (see Appendix A) locates in the *.sub files are allocated in the “sub” table of the Watershed.mdb file by the order of HRU number. In C-SWAT, parameter information of SUB files is read from a single file named allsub.txt with inclusion of all information from the “sub” table of the Watershed.mdb. Parameter information will be first read by subroutine readallsub.f and saved in a single matrix. Next, parameter information originally read in the subroutine readsub.f will be substituted by passing values from the newly generated matrix instead of open/read/close each *.sub file.

At the end of *.sub one can find a number of file names such as *.hru, *.gw, *.pnd, and *.mg. These file names do not have to be included in the C-SWAT input files because there is no need to designate individual file names anymore. All 13 HRU and subbasin level input files are being read from consolidate files such as allhru.txt, allosltxt, and allchm.txt.

4. [Subbasin Level] SWQ Files (*.swq) – Structure I
   [Subroutine Added] – allocate_allsystem.f/readallsqw.f
   [Subroutine Modified] – modparm.f/readsub.f/readswq.f

(1) Nutrient group
   RS1, RS2, RS3, RS4, RS5, RS6, RS7, RK1, RK2, RK3, RK4, RK5, RK6, BC1, BC2, BC3, BC4.

(2) Pesticide group
   CHPST_REA, CHPST_VOL, CHPST_KOC, CHPST_STL, CHPST_RSP, CHPST_MIX, SEDPST_CONC, SEDPST_REA, SEDPST_BR, SEDPST_ST_ACT.

Parameters in *.swq files can be categorized into three groups: nutrient and pesticide for in-stream water quality processes (Arnold et al., 2011). A total of (17+10) parameters (see Appendix A) located in the *.swq files are allocated in the “swq” table of the Watershed.mdb file by the order of HRU number. In C-SWAT, parameter information of SWQ files is read from a single file named allswq.txt with inclusion of all information from the “swq” table of the Watershed.mdb. Parameter information will be first read by subroutine readallsqw.f and saved in a single matrix. Next, parameter information originally read in the subroutine readswq.f will be substituted by passing values from the newly generated matrix instead of open/read/close each *.swq file.

5. [Subbasin Level] WGN Files (*.wgn) – Structures I and II
   [Subroutine Added] – allocate_allsystem.f/readallwgn.f
   [Subroutine Modified] – modparm.f/readsub.f/readwgn.f

(1) Basic information
   WLATTITUDE, WLONGITUDE, WELEV, RAIN_YRS.
Appendix B. Supporting information

Supplementary data associated with this article can be found in the online version at [http://dx.doi.org/10.1016/j.cageo.2014.07.017](http://dx.doi.org/10.1016/j.cageo.2014.07.017).

References


