Environmental Modelling and Software 139 (2021) 105003

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: http://www.elsevier.com/locate/envsoft

A dual-layer MPI continuous large-scale hydrological model including Human Systems

Diego Avesani [a,b,*], Andrea Galletti [b], Sebastiano Piccolroaz [a,b,c], Alberto Bellin [b], Bruno Majone [b]

[a] Faculty of Sciences and Technologies, Free University of Bolzano-Bozen, Piazza Universitá-Universitätsplatz, 5 39100, Bolzano-Bozen, Italy
[b] Department of Civil, Environmental and Mechanical Engineering, University of Trento, via Mesiano, 77 38123, Trento, Italy
[c] Physics of Aquatic Systems Laboratory (APHYS) – Margaretha Kamprad Chair, École Polytechnique F édérale de Lausanne, CH-1014, Lausanne, Switzerland

ARTICLE INFO

Keywords:
Water resources management
Large-scale hydrological modeling
Parallel computing
High performance computing
Water infrastructures
Hydropower
HYPERstream routing scheme

ABSTRACT

Large-scale hydrological models are demanding both in term of memory allocation and CPU time, particularly when assessment of modeling uncertainty is required. High Performance Computing offers the opportunity to reach resolutions not achievable with standard serial coding. However, the advantages may be offset by poor scalability of the model due to components that have to be executed in series, such as to simulate the presence of hydraulic infrastructures.

Driven by this motivation, we developed HYPERstreamHS, a model that adopts a holistic approach to simulate hydrological processes in large river basins with streamflow altered by hydraulic infrastructures. The model adopts a dual-layer parallelization strategy, where the paralleled version of the hydrological kernel is the first-layer, with the second layer taking care of inverse modeling.

The results show that the processors should be carefully organized and grouped in order to achieve the best overall performance and suggests that this subdivision is problem specific.

1. Introduction

Distributed hydrological modeling is widely applied as a tool to inform water resources management and climate change impact assessment studies (see e.g., Kundzewicz et al., 2007). In assessing water resources and devising management strategies, modeling is often extended over long time periods at high spatial resolution, particularly when the objective is to evaluate the likely impact of climate change on water resources (Todd et al., 2011). As a consequence, distributed hydrological simulations on medium to large watersheds are demanding both in term of memory requirements and computational time needed to simulate the relevant processes over such high-resolved domains (Vivoni et al., 2011; Liu et al., 2013). Nonetheless, high resolution offers a better representation of spatio-temporal dynamics of state variables, such as soil water storage (Rojas et al., 2008) and snow accumulation (Scipión et al., 2013).

Simulation of spatially (and temporally) variable hydrological fluxes is affected by uncertainty, stemming from: i) an imperfect conceptual model (i.e., epistemic uncertainty), ii) parameters that should be inferred from observational data because not directly measurable (parametric uncertainty), and iii) input data uncertainty (i.e., measurement errors in meteorological forcing data as well as their spatialization). Quantification of these uncertainties is a difficult task that attracted a wealth of interest in hydrology as well as in the wider environmental modeling community (see e.g., Montanari et al., 2009; Beven and Binley, 1992; Beven, 1993). The curse of dimensionality increases dramatically the number of forward simulations needed to explore the parameters space in inverse modeling procedures, as the number of parameters increases, thereby rocketing the already high computational cost of high resolution.

Parallelization techniques such as Message Passing Interface (MPI, MPI Forum, 1994) and Open Multi-Processing (OpenMP, Dagum and Menon, 1998) have been applied to distributed hydrological models in order to reduce the computational time and manage efficiently large amount of data. For example, Wu et al. (2013) and Li et al. (2011) implemented the MPI standard into both SWAT and DWM hydrological

* Corresponding author. Faculty of Sciences and Technologies, Free University of Bolzano-Bozen, Piazza Universitá-Universitätsplatz, 5 39100, Bolzano-Bozen, Italy.
E-mail address: diego.avesani@unitn.it (D. Avesani).

https://doi.org/10.1016/j.envsoft.2021.105003
Accepted 16 February 2021
Available online 23 February 2021

models by separating the watershed in units attributed to different processors. Burstedde et al. (2018) employed MPI paradigm in order to enhance computational performances of ParFlow model through parallel adaptive mesh refinement. Recently, WRF-hydro modeling system (Gochis D.J., 2020) has been equipped with MPI-based libraries in order to manage spatial domain decomposition as well as to improve communication among processors (Senatore et al., 2015; Fersch et al., 2020). GPU-based (Graphics Processing Unit) parallel computing has also been used to reduce the computational time of hydrological modules dealing with the extraction of the digital drainage network of large river basins (Ortega and Rueda, 2010) and parallelization of streamflow accumulation (Qin and Zhan, 2012; Rueda et al., 2016).

All these models apply single layer parallel computing with the objective of reducing the computational time of the forward simulations. However, a limitation in scalability, which measures the system's capacity to reduce computational time in proportion to the number of processors, emerges when the number of processors grow above a given threshold (Amdahl, 1967), which depends on the model adopted, the computational domain and hardware. This limitation is due to the following factors: i) tasks inter-dependency; ii) time spent to transfer information between parallel threads, which increases with the number of processors adopted; and iii) load imbalance, i.e., the uneven computational work distribution between processors. Although the last limitation can be alleviated through careful coding or employing parallel automatic balancing procedures (e.g. Mendicino et al., 2006; Giordano et al., 2020), the first two factors are endemic to parallel computing. In addition, most of the existing hydrological models contain modules that should be run in series, thereby reducing the theoretical speed-up regardless the number of processors used (Amdahl, 1967). For example, routing schemes based on the solution of the de Saint-Venant equation or on cell-by-cell mass conservation coupled with lumped streamflow river storage relationships suffer from these limitations (Yamazaki et al., 2011; de Paiva et al., 2013).

When multiple model runs are required in order to explore the parameters space, a parallelization scheme dispatching to multiple processors independent forward simulations has been also envisioned (Pitman, 1973; Rouholahnejad et al., 2012; Tristram et al., 2014; Kan et al., 2018; Fersch et al., 2020). However, the efficiency of such single layer parameter domain decomposition scheme is hampered by memory allocation problems arising when managing large amounts of data, as typically occurs in large-scale hydrological modeling applications.

The aforementioned limitations can be alleviated by using layered parallelization techniques that use model specific strategies to reduce serial dependencies and improve the gain. Examples are layering the model's units according to flow direction (Liu et al., 2014), or assigning sub-basins to separate cluster nodes with the processors, within each node, performing the computations at the level of the sub-basin (Liu et al., 2016). In another work, Zhang et al. (2016) introduced a dual-layer parallelization system for the calibration of DYRIM hydrological model, in which the higher layer (second layer) handled multiple parallel forward simulations (first layer) by means of a standard scheduler approach. Layered parallelization is therefore appealing for distributed modeling, though applications in hydrology are still limited in number.

An important aspect in hydrological modeling for water resources management is the implementation of Human Systems (HS) dealing with water transfer and storage for irrigation, energy production and other water uses (see e.g., Nazemi and Wheater, 2015a, b, for a review on the challenges posed by the incorporation of water resources management modules into Earth System Models). Hydraulic infrastructures for water uses introduce changes in the network topology and streamflow timing, which have to be taken into account in modeling (Gregory, 2006). Their effect is also relevant in developing realistic future scenarios and water management plans, including the assessment of the impact of new infrastructures. Reservoirs, for example, alter streamflow timing, while diversion channels change network topology and require the introduction of additional water budget equations (see e.g., Bellin et al., 2016). The simulation of these artificial systems therefore poses additional limitations to scalability because of the dependencies in upstream-downstream water transfer they introduce, which can be alleviated by parallel computing layering.

We contribute to this ongoing effort by developing a new modeling framework, we coined HYPERstreamHS, which enhances the capabilities of the HYPERstream routing scheme recently introduced by Piccolroaz et al. (2016). HYPERstream is a multi-scale streamflow routing scheme, based on the travel time approach, which has been specifically designed for reproducing accurately horizontal hydrological fluxes, while avoiding grid refinement beyond what needed for the accurate reproduction of vertical fluxes. Furthermore, it can be easily made parallel, due to linearity of the routing scheme. The development of improved routing schemes to adequately resolve horizontal fluxes with an acceptable computational effort has been indicated as one of the priorities for the improvement of existing large-scale hydrological models (see e.g., the review by Clark et al., 2015). By adopting a holistic approach, we coupled HYPERstream with continuous modules for surface and subsurface flow generation and added specific modules dealing with the alterations introduced by water infrastructures, such as reservoirs and diversions. Furthermore, HYPERstreamHS adopts a dual-layer parallelization strategy, based on the MPI standard, in order to fully exploit both spatial domain and parameters domain decomposition. MPI was preferred due to its flexibility and portability (i.e. it can be used with different platforms, Musiał et al. (2008)). Dual-layering is applied by subdividing the processors in groups, each one performing forward modeling with a set of parameters, and within a group partitioning the computational domain among processors. The number of processors assigned to each group is below the threshold over which scalability deteriorates, such as to optimize the overall performance. Scalability of HYPERstreamHS is here presented and tested with reference to the Adige river basin case study (south-eastern Alps, Italy). In addition, the capabilities of the model to reproduce observed streamflows in locations impacted by the presence of human infrastructures are presented with reference to one of the gauging stations available in the case study.

Sect. 2 presents the structure of HYPERstreamHS, the adopted parameterization of physical processes (2.1), and of water transfer and storage infrastructures (2.2) as well as data requirements (2.3). Parameters identification procedures are presented in Sect. 3, followed, in Sect. 4, by the description of the dual-layer parallelization strategy and in Sect. 5 by an example of application to the Adige river basin. Finally, concluding remarks are drawn in Sect. 6.

2. HYPERstreamHS hydrological model

2.1. Natural hydrological system

Large-scale hydrological simulations are performed by coupling the HYPERstream routing scheme (Piccolroaz et al., 2016) with a grid-based runoff generation model composed by the SCS-CN runoff model (Michel et al., 2005) combined with a bucket-type soil moisture model (Majone et al., 2010). Routing is performed by means of the Instantaneous Unit Hydrograph (IUH) applied to each cell-node pair. The nodes are selected positions along the river network where the water discharge is computed. According to the IUH theory (see, e.g., Gupta et al., 1980; Rinaldo and Rodriguez-Iturbe, 1996) runoff generated at the hillslope is transferred along the river network by means of a transfer function coinciding with the probability density function (pdf) of the particle residence time, which in turn is given by the pdf of the travel distance from the cell to the node multiplied by the celerity of the flow signal (Rodríguez-Iturbe and Rinaldo, 1997). The pdf of the travel distance is extracted from the Digital Elevation Map (DEM) of the catchment at a finer resolution with respect to the grid used to resolve the vertical fluxes. This allows to use a relatively coarse grid, of the order of a few kilometers, without compromising routing accuracy (Piccolroaz et al.,

2016).

Vertical fluxes are represented as follows. Firstly, precipitation falling on a given cell is classified as rainfall or snow according to a given threshold temperature and the latter is accumulated in the snowpack compartment. Mass balance is applied to the snowpack with water produced by snowmelting computed by the degree-day model (Rango and Martinec, 1995; Majone et al., 2010). The resulting water volume, composed by the sum of rainfall and snowmelting, is partitioned into infiltration and surface runoff by means of the generalization of the SCS-CN model proposed by Michel et al. (2005). In the absence of snowpack and if the precipitation is classified as rainfall, the entire precipitation is transferred to the SCS-CN module. Soil water dynamics is modelled by a nonlinear reservoir (Majone et al., 2010) with infiltration as input flux and evapotranspiration, evaluated by means of the Hargreaves and Samani (1982) model, subsurface flow and deep infiltration as output fluxes. Deep infiltration enters a linear reservoir representing the groundwater compartment with return flow as output (see Piccolroaz et al., 2015; Bellin et al., 2016; Majone et al., 2016, for applications of this modeling framework in Alpine catchments.).

A schematic of the vertical water fluxes separation and routing scheme is depicted in Fig. 1, and additional details are provided in Appendix A. The modeling framework relies on the following key geometric objects (see Fig. 1): i) *macrocells* (i.e., grid cells), where the meteorological forcing is assigned and the vertical water fluxes are evaluated (Fig. 1a), and ii) *nodes*, located along the stream network where water discharge is computed. The drainage directions and the stream network are extracted from the DEM according to one of the several criteria available for the identification of the hillslope-channel separation (e.g., Tarboton et al., 1991; Lazzaro, 2009). The information derived from the DEM is used to build the probability density function (pdf) of the flow path lengths (i.e., width function) connecting the hillslope-channel transition sites of a macrocell to the first downstream node. A width function is evaluated for each macrocell-node pair (Fig. 1b), and then it is rescaled through a constant stream velocity to obtain the Instantaneous Unit Hydrograph used to transfer the flux to the downstream node (see Appendix A for details). The assumption of constant stream velocity makes the transfer process linear and hence the routing model highly parallelizable. It has been shown that using a constant (uniform) velocity along the river network is not a limitation (see e.g., Rinaldo et al., 1991; Rodríguez-Iturbe and Rinaldo, 1997), while it provides great flexibility in parallelization.

2.2. Human Systems module

The effects of water uses is simulated by including hydraulic infrastructures, represented by nodes, edges and reservoirs into the natural network. This leads to an augmented network. Fig. 2 shows an example of inclusion of a hydropower system composed by a reservoir, a diversion channel, a penstock and a power station. In addition to the reservoir water is taken from natural channels intersecting the diversion channel (intakes) and is returned to the natural river network at the powerhouse node. The main elements (types) of HS module and their associated water balance equations are presented below.

2.2.1. Type-reservoir

Reservoirs represent special nodes where water storage may occur. As shown in Fig. 3, each reservoir is subdivided into three volume pools: flood control, active, and inactive volumes. Reservoir functioning is

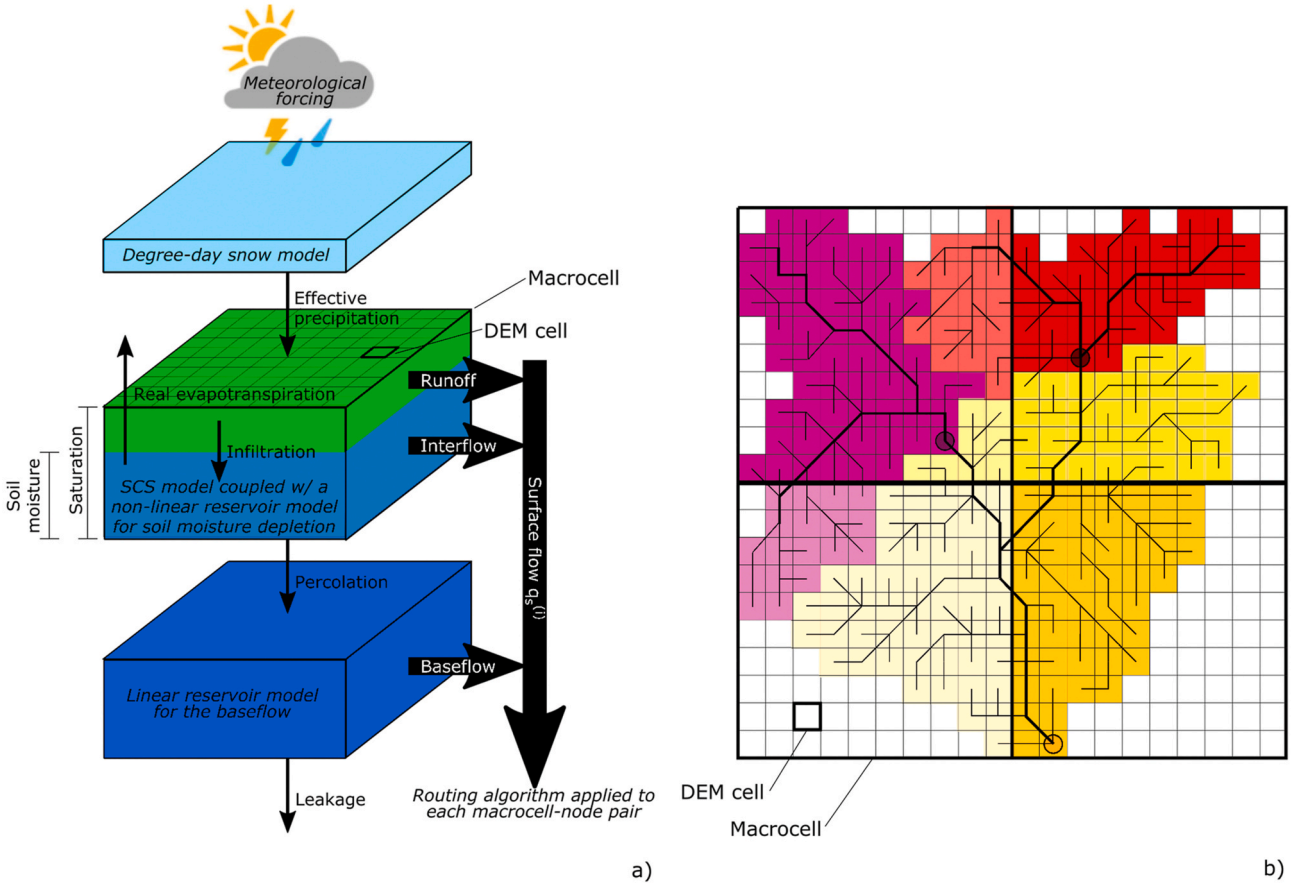

Fig. 1. Sketch of the natural component of the hydrological system; a) compartments used to represent the vertical fluxes, b) HYPERstream routing scheme. See appendix A for the description of the three compartments used to represent the vertical fluxes. Panel b) represents four macro-cells and three nodes; colors are used to highlight the portion of the macrocell contributing to each node.

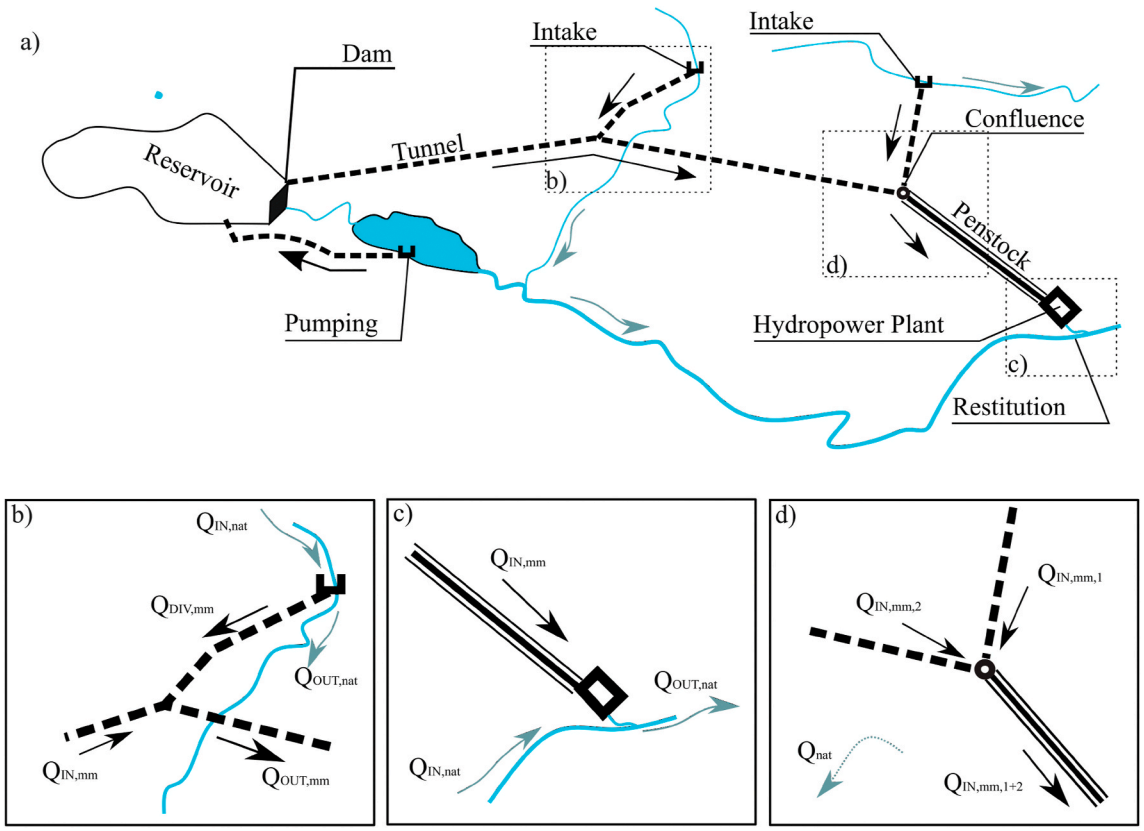

Fig. 2. (a) Schematic of the infrastructures simulated in the Human System module of HYPERstreamHS. Insets in the lower panel detail a particular structure that can be simulated with the object *type-Intake*: diversion (b), restitution point (c) and confluence between tunnels (d).

described by the curve volume versus elevation and the schedule of water use, both entering into the mass balance equation:

$$\Sigma Q_{IN}(t) - \Sigma Q_{U}(t) - Q_{EF}(t) - Q_{spill}(t) = \frac{dV}{dt} \quad (1)$$

where $\sum Q_{IN}$ (t) is the inflow to the reservoir, $\sum Q_U$ is the flow transferred through the inlet gate to the uses, Q_{EF} is the flow released to the downstream river network according to ecological flow rules, Q_{spill} is the flow released by the spillways and diversion gates, and finally V is the volume of water contained in the reservoir and t is the time. The summation in the first two terms of Eq. (1) indicates that the reservoir may receive water from catchments other than the upstream catchment (connected catchments) and deliver water to multiple uses.

Mass balance described by Eq. (1) is applied by considering the following priority of releases:

a. *Ecological Flow,* Q_{EF}: local environmental regulations determine the minimum amount of water that must be released to the downstream river network by reservoirs and intake points to preserve freshwater ecosystem functioning.
b. *Spillways,* Q_{spill}: it is the sum of the flux discharged downstream through the spillways and the bottom outlet gates. The former operates only when the water level is above the maximum regulation level ($h_{max.reg}$), while the latter can be operated when the water level is above the minimum regulation level ($h_{min.reg}$) and can be used to empty the reservoir or to create the volume needed to contain an incoming flood. Reservoirs may be equipped with one or more spillways, with outflow regulated by a gate or not regulated. In HYPERstreamHS we used an equivalent free (non regulated) spillway with the following characteristics: it starts to operate when the water level in the reservoir is at $h_{max.reg}$ and its length L_{eq} is

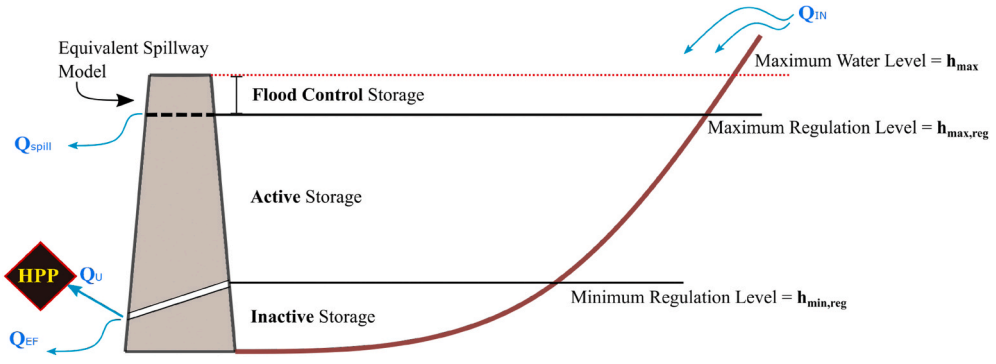

Fig. 3. Sketch of the partitioning of the volume of a reservoir, and of the corresponding operational water levels and the water fluxes entering into the mass balance equation (1). The power generated at a given hydropower plant (HPP) is calculated according to equation (7).

computed by imposing that the water discharge is equal to the maximum outflow ($Q_{spill,max}$) when $h = h_{max}$:

Therefore, the water discharge from the equivalent spillway is given by

$$Q_{spill}(t) = L_{eq} C_q \sqrt{2g} \left[h(t) - h_{max,reg}\right]^{3/2} \quad (2)$$

where $C_q \simeq 0.49$ is the coefficient of discharge, and L_{eq} is computed according to the following equation:

$$L_{eq} = \frac{Q_{spill,max}}{C_q \sqrt{2g} \Delta h_{fc}^{3/2}} \quad (3)$$

In Eq. (3), $\Delta h_{fc} = h_{max} - h_{max,reg}$ is the maximum hydraulic head. The time step adopted in the solution of Eq. (1) is reduced during flood events in such a way to reduce the numerical error. For example, in our simulations the adopted time step is of 1 h for the hydrological kernel of HYPERstreamHS, which is reduced to 1 min in the solution of Eq. (1) during flood events.

c. *Utilization schedule*, Q_U: for $h > h_{min,reg}$, a water use schedule, i.e. a time series of the values assigned to Q_U is provided externally by the user.

2.2.2. Type-intake

Intakes are nodes where water is diverted to satisfy water demand. The water can be sent directly to the final uses or stored in a reservoir for successive utilization (see Fig. 2). Intake nodes are also used to model restitution points and confluences between channels or conduits.

An intake node is located along the river network and is connected by a channel to the man-made network or to a reservoir (see Fig. 2a–b). At the intake the following mass balance equation is applied (Fig. 2b):

$$Q_{IN,nat}(t) - Q_{DIV,mm}(t) - Q_{OUT,nat}(t) = 0 \quad (4)$$

where $Q_{IN,nat}$ and $Q_{OUT,nat}$ are the river water discharge immediately upstream and downstream the intake, respectively, and $Q_{DIV,mm}$ is the diverted water discharge, which is bounded by the maximum water discharge that the diversion channel can convey.

The diversion channel or conduit may discharge in another channel, in a reservoir, or directly to the final use. In the former case a second mass balance equation should be applied:

$$Q_{IN,mm}(t) - Q_{OUT,mm}(t) + Q_{DIV,mm}(t) = 0 \quad (5)$$

the first two terms have a similar meaning as in Eq. (4) but referred to the man-made channel network. If the final node of the diversion is connected to a reservoir Eq. (5) is not included and $Q_{IN,mm}$ constitutes one of the terms of ΣQ_{IN} in Eq. (1).

Concerning the restitution points (Fig. 2c), water flux arriving from the man-made network ($Q_{IN,mm}$) is added to the incoming flow ($Q_{IN,nat}$) to obtain the flux downstream the junction:

$$Q_{OUT,nat}(t) = Q_{IN,nat}(t) + Q_{IN,mm}(t) \quad (6)$$

Confluences in the man made network (see Fig. 2d) are treated in a similar way by adding the flux of the merging channel to the incoming flux in the receiving network.

2.2.3. Type-hydropower plant

Power plants are treated similarly to restitution points illustrated above. The water derived from a node is returned to the river at the restitution point and the generated power is computed as follows:

$$HPP(t) = \eta \sum_{i=1}^{N} \gamma \, Q_{TURB,i} \Delta H_i(t) \quad (7)$$

where $Q_{TURB,i}$ and ΔH_i are respectively the turbined water discharge and the hydraulic head of the $i-$th power unit, and N is the number of power units (turbine plus alternator) of the power plant. The hydraulic head ΔH_i is given by the difference between water elevations of the reservoir and the turbine or of the downstream impoundment, depending if an action or a reaction turbine is used. In run-off-the river power plants the head is assumed constant in time. Finally, γ is the specific weight of water and η is the turbine efficiency, which is assumed constant. In the example of application presented in Sect. (5) we set $\eta = 0.8$, corresponding to the mean of the efficiency of 12 out of 22 reservoir hydropower plants belonging to the study area, for which this value was known.

2.2.4. Type-diversion channel

Diversion channels divert water from the natural stream network by means of an intake and route it downstream along the man-made channel network. These objects are fully characterized by the channel length and water celerity, with the latter used to compute the delay with which the water is conveyed downstream, and is set by the user prior to the simulation. In the example of application presented in Sect. (5) we assumed a value for the celerity equal to 2.0 m/s (Kumar and Singhal, 2015).

2.3. Data requirements

HYPERstreamHS is conceived as a parsimonious model. Temperature and rainfall gridded dataset are the meteorological forcing required to run the hydrological kernel. DEM, land-use and land cover maps are also needed (see Appendix A.1 for additional details).

Human Systems module requires detailed information of the hydraulic infrastructures of the river basin: geometry and position of storage reservoirs, head and penstock capacity of hydropower plants, location of intakes and restitution points, and capacity of diversion channels. To show that HYPERStreamHS can be succesfully applied in most practical situations in the present work we used only publicly available data. Reservoir information was retrieved from the Italian Registry of Dams (RID, available at http://dgdighe.mit.gov.it), while information on hydropower systems were primarily extracted from public reports such as those provided by ENEL (https://www.enel.it), the former public operator that managed Italian hydropower production from 1962 to 2004, when the power market was liberalized. Furthermore, information on plants' installed power and penstock capacity was retrieved directly from operating companies online databases or from informative leaflets.

3. Inverse modeling and parameter identification

As customary in hydrological simulations the 12 parameters of HYPERstreamHS are set such as to provide the best possible representation of observational data, typically under the form of water discharge measurements. With this objective in mind, inverse modeling is therefore applied by searching the parameters hyperspace with the objective of updating the prior parameters distribution (Bayesian updating) or identifying the set of parameters that maximizes a given objective function. Therefore, HYPERstreamHS is provided with two searching strategies: the Particle Swarm Optimizer (PSO) (Kennedy and Eberhart, 1995; Robinson and Rahmat-Samii, 2004; Castagna and Bellin, 2009; Majone et al., 2010; Piccolroaz et al., 2015), and the stratified Latin Hypercube Sampling (LHS) (McKay et al., 1979; Piccolroaz et al., 2016). The former is a genetic searching algorithm which has been shown to optimize the searching path (Kennedy and Eberhart, 1995; Robinson and Rahmat-Samii, 2004) down to the optimal point, while the latter is often used to efficiently sample the entire parameters space.

In both cases we used the Nash-Sutcliffe efficiency (NSE) index (Nash and Sutcliffe, 1970) as objective function for streamflow time series evaluated at given nodes of the river network:

$$NS = 1 - \frac{\sigma_e^2}{\sigma_o^2}\ , \tag{8}$$

where σ_o^2 is the variance of the observed streamflow time series and σ_e^2 is the variance of the residuals, i.e. the difference between observed and simulated streamflow.

Since parameters inference is a computationally demanding step, particularly when the number of parameters is large, our objective is to alleviate this burden by means of the dual parallelization strategy described below.

4. Dual-layer parallelization strategy

Dual-layer parallelization is organized in such a way that a given number of forward simulations is run in parallel (second layer) each with a different set of parameters chosen by the searching algorithm. Each forward run is performed in parallel as well by partitioning the hydrological kernel among a subset of processors (first layer).

This framework is achieved by exploiting the master-slaves paradigm according to the MPI standard. For illustration purpose, Fig. 4 shows an example with $n_p = 16$ processors (named as $p_0, \ldots..p_{15}$), hypothetically available in a HPC system. As shown in Fig. 4a, before parallelization all the processors p_k, $k = 0, \ldots, n_p - 1$ belong to the same group, named as *MPI_GROUP_WORLD*, and are associated to its corresponding communicator, termed according to the MPI terminology, as *MPI_COMM_WORLD*. For the sake of clarity, here we remind that the concept of *MPI group* corresponds to the mathematical concept of *set*, i.e. a well-defined collection of distinct objects (i.e., processors) with their own identifiers (i.e., processor *id*). On the other hand, a *communicator* is an object connecting all the processors belonging to the same set and giving them independent identifiers (p_k, $k = 0, \ldots, n_p - 1$) according to an ordered topology. Details about MPI groups, communicators and topology can be found in MPI Forum (1994) and in Gropp et al. (1996). Afterwards, the global group and communicator are split into n_g sub-sets, with each sub-set composed by $n_s = n_p/n_g$ processors.

Each sub-set constitutes a univocal set of processors and defines the first layer of the MPI parallelization where independent simulation runs are performed. For the sake of clarity it is useful to define the following topological operator $p_{i,j} = p_{\delta(k)}$ which links global processor topology p_k ($k \in [0, n_p - 1]$) with the local topology, where $i \in [0, n_g - 1]$ identifies the group (i.e., the sub-set) to which the processor belongs, j uniquely defines the processor *id* inside the group, $k \in [0, n_p - 1]$ is the processor *id* according the default global communicator, and δ is the topology operator which associates the local communicator (*MPI_LOCAL_COMM*) to the global communicator (*MPI_COMM_WORLD*). In the illustrative example (see Fig. 4b) the total number of $n_p = 16$ processors is divided into 4 sub-sets, i.e. $n_g = 4$, with each set containing 4 processors (see dashed thick lines in Fig. 4b). The Figure also highlights that each sub-set has its own local communicator, *MPI_LOCAL_COMM*, and local identifier $p_{i,j}$ for the processor *id*. In addition, in each sub-set a *master* processor is defined (identified with $j = 0$, i.e. $p_{i,0}$), which controls the other processors within the sub-set. These latter processors are usually termed as *slaves* and are identified as $p_{i,j}$, with $j \in [1, n_s - 1]$. Notice that each master also acts as slave within its own sub-set. Furthermore, the second layer of parallelization is organized creating a new group, *MPI_GROUP_MASTER* (see continuous thick line in Fig. 4b), which gathers the master processor of each sub-set together with its own communicator, *MPI_MASTER_COMM*.

For the sake of completeness, Listing (1) shows the MPI Fortran code used to split the processors into groups and to set up the local communicator, while Listing (2) illustrates the instructions applied to set-up the communicator between the masters and to exclude the slave processors. In the ensuing sub-Sections, the two layers of parallelization are described in details with reference to the implementation within the HYPERstreamHS hydrological model.

```
CALL MPI_COMM_SPLIT(MPI_COMM_WORLD,colorl,MPIworld%rank,MPI_LOCAL_COMM,MPIworld%iErr)
CALL MPI_COMM_RANK(MPI_LOCAL_COMM, MPIlocal%rank,MPIlocal%iErr)
CALL MPI_COMM_SIZE(MPI_LOCAL_COMM, MPIlocal%nCPU,MPIlocal%iErr)
```

Listing 1. MPI command to create processor groups.

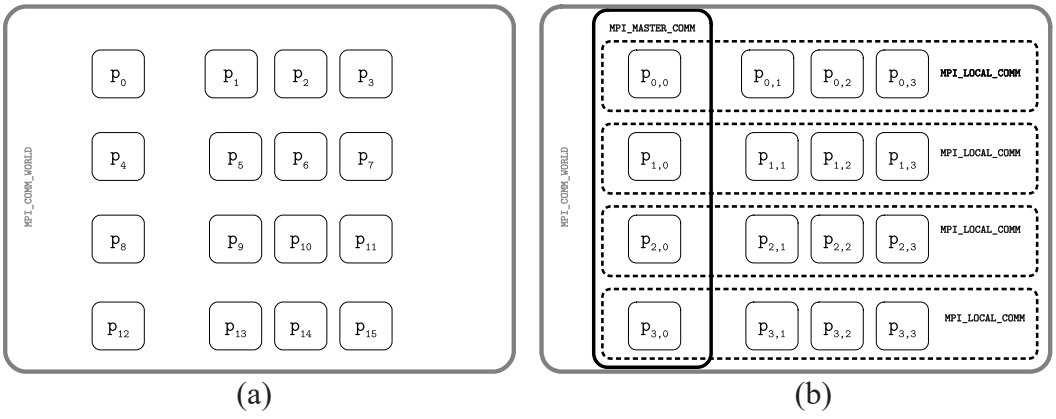

Fig. 4. Illustrative example of the MPI dual-layer parallelization considering 16 processors. (a) The standard MPI set of processors (grey box) with its own communicator *MPI_COMM_WORLD*. Each processor has a unique identifier p_k with $k \in [0, 15]$. (b) The sub-sets derived from the original MPI set: 4 local sets of processors (dashed line) with the local communicator *MPI_LOCAL_COMM*. Each processor has its local identifier $p_{i,j}$, where $i \in [0, 3]$ identifies the group and $j \in [0, 3]$ represents the local numbering within the sub-set. Notice that $j = 0$ identifies the master of each set. The additional set composed by masters of each group with its communicator *MPI_MASTER_COMM* is also presented as continuous thick line.

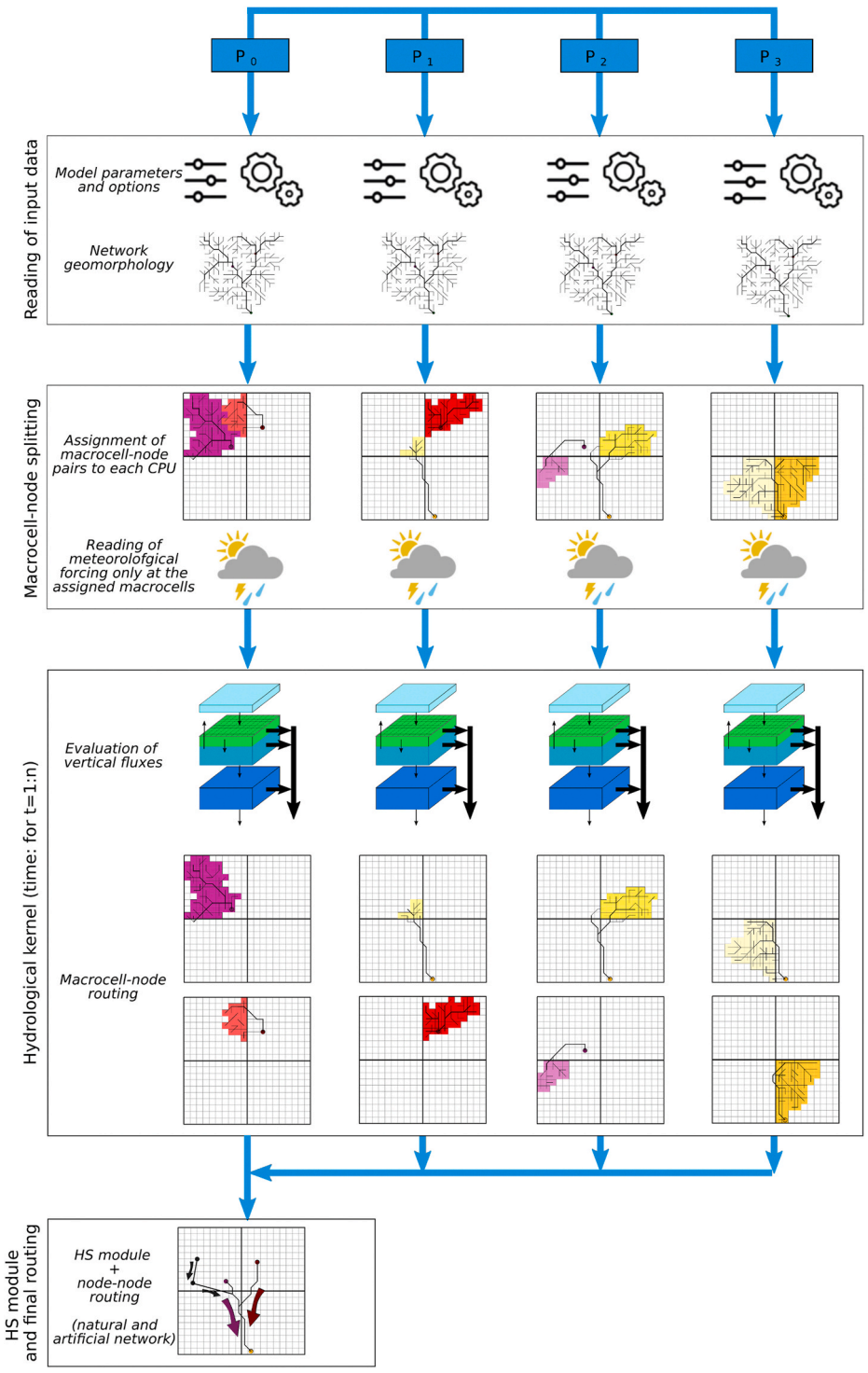

Fig. 5. Illustrative example of the first parallelization level with 4 processors. All processors read river topology and model parameters (upper frame); macrocell-node pairs are assigned to processors which read the proper meteorological data (second frame); each processor evaluates vertical fluxes of the assigned macrocells and compute the macrocell-node routing (third frame); a single processor (the master) collects and aggregates streamflow at the node level, solves for the water budget at the nodes of the hydraulic infrastructures and performs the final routing along the global (natural and man-made) network (lower frame).

```
!Create master group
CALL MPI_GROUP_INCL(MPI_GROUP_WORLD, nBranch, MRANKS, MPI_GROUP_MASTER,iErr)
!Create a new communicator based on the master group
CALL MPI_COMM_CREATE_GROUP(MPI_COMM_WORLD,MPI_GROUP_MASTER,0,MPI_MASTER_COMM,iErr)
!
!Assign id to processor in the master groups
!and count the processor in the group
IF(MPI_COMM_NULL .NE. MPI_MASTER_COMM)THEN
  CALL MPI_COMM_RANK(MPI_MASTER_COMM, MPImaster%rank,MPIlocal%iErr)
  CALL MPI_COMM_SIZE(MPI_MASTER_COMM, MPImaster%nCPU,MPIlocal%iErr)
ELSE
  MPImaster%rank = MPI_PROC_NULL
ENDIF
```

Listing 2. MPI command to create the group of masters.

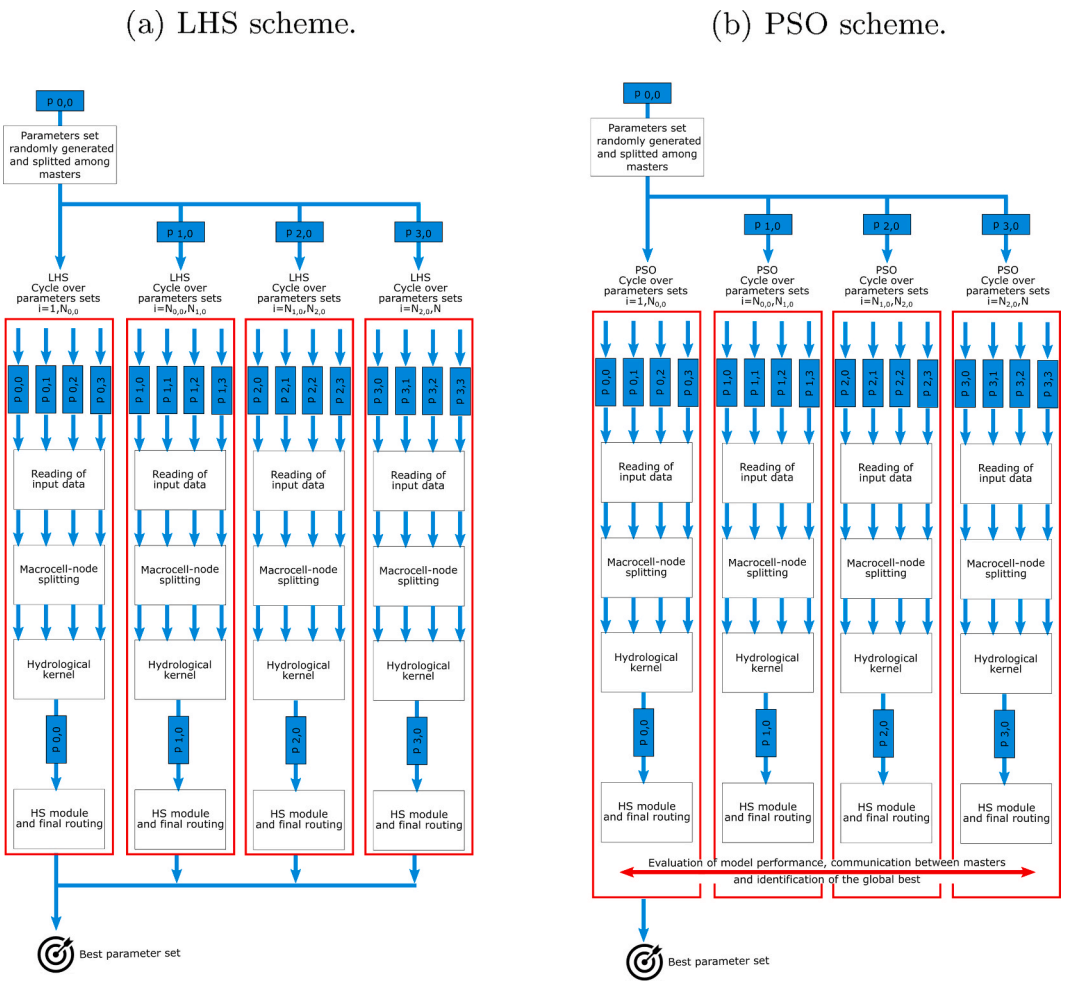

Fig. 6. Dual-layer parallelization scheme implemented with reference to LHS (a) and PSO (b) algorithms, respectively. In both schemes the master of the masters ($p_{0,0}$) computes the parameters and divides them equally among the other masters. Each master runs the parallel hydrological model with its own slaves. In the LHS case (a), there is no communication between masters; in the PSO case (b) at each iteration step the masters communicate in order to identify the best global solution (red horizontal arrow). In both cases, the cycling over the parameter sub-set assigned to each master is depicted with a red box.

4.1. First parallelization layer

As discussed in Sect. 2, HYPERstreamHS is suitable for parallelization given that: i) the computational domain is divided into independent macrocells each one with its own meteorological forcing; ii) vertical water fluxes of a macrocell are independent from those of the other macrocells; and iii) runoff is transferred from the macrocell to downstream node by means of the linear IUH. Therefore, the runoff produced by the macrocells contributing to a given node can be transferred separately and then summed up.

To take advantage of these characteristics, at the first layer tasks are assigned according to the number of macrocell-node pairs identified by the pre-processor (see Sect. 2.1 and Appendix A.1). Fig. 5 illustrates the parallelization strategy adopted in the first MPI layer. As a first step all processors store the model parameters and the network topology (see the horizontal frame *Reading of input data*). Then, the macrocell-node pairs are equally subdivided among all the available processors which load the corresponding macrocell-node width functions, as well as the meteorological forcing pertaining only to the macrocells identified in the pairs selection (see the horizontal frame *Macrocell-node splitting*). Memory is allocated to the processors according to the number of associated macro-cells and width functions. The following step is performed in parallel with each processor that computes the runoffs of the assigned macrocells and transfers them to the connected downstream node (see the horizontal frame *Hydrological Kernel* in Fig. 5). Streamflow is then aggregated at each network node by summing up all the contributions pertaining to the macrocells-node pairs. Notice that this operation refers only to the streamflow contributions originating from the natural component of the hydrological model. Once streamflow is aggregated at each node level a single processor is designed to perform the serial computations needed to i) perform the water budgets at the nodes of the hydraulic infrastructures (see Sect. 2.2); and ii) routing along the natural and man-made network of water fluxes simulated at the nodes (see horizontal frame *HS module and final Routing* in Figure (5).

Notice that, among the possible parallelization strategies for balancing the computational load, the equally subdivision of macrocell-node pairs over the available processors presents the key advantage that the interaction between processors is limited to streamflow aggregation at the network nodes, thereby avoiding the exchange of information on vertical fluxes. As a consequence, each processor performs independent macrocell-node convolution without waiting processed data from macrocells belonging to other processors. An overall reduction of computational times is then achieved as well as an improved load balance between processors which perform approximately the same number of convolutions. As a side effect, a limited number of macrocells may be assigned to two or more processors (see second horizontal panel in Fig. 5), which leads to replicate computations of the same vertical fluxes in these cells. However, the associated computational burden is less than the gain achieved by using this strategy.

Further improvement of load balancing can be envisioned by implementing a subdivision of macrocell-node pairs between processors based on the macrocell-node convolution length (which is a proxy of the computational time required to transfer runoff from the macrocell to the pertaining node). In this respect, profiling of the code and implementation of automatic load balancing techniques such as scatter-type decomposition (e.g. Mendicino et al., 2006) or Graph Partitioning (e.g. Karypis and Kumar, 2009) will be considered as possible further code optimization.

4.2. Second parallelization layer

A second parallelization layer can be introduced when the application includes inverse modeling (i.e., parameter identification). In HYPERstreamHS we implemented the Latin Hypercube Sampling (LHS) and the Particle Swarm Optimization (PSO) algorithms, but other procedures such as Markov Chain Monte Carlo (see e.g. Vrugt, 2016) can be implemented as well. Fig. 6a and b illustrates a schematic of the implemented dual-layer framework for LHS and PSO, respectively. The n_p processors that will be used are subdivided into $n_{masters}$ masters, each one having n_{slaves} slaves, such that $n_{masters} \times n_{slaves} + n_{masters} = n_p$. In the examples shown in Fig. 6 the masters are 4, each one with 3 slaves for a total of $n_p = 16$ processors. The master of masters processor, indicated

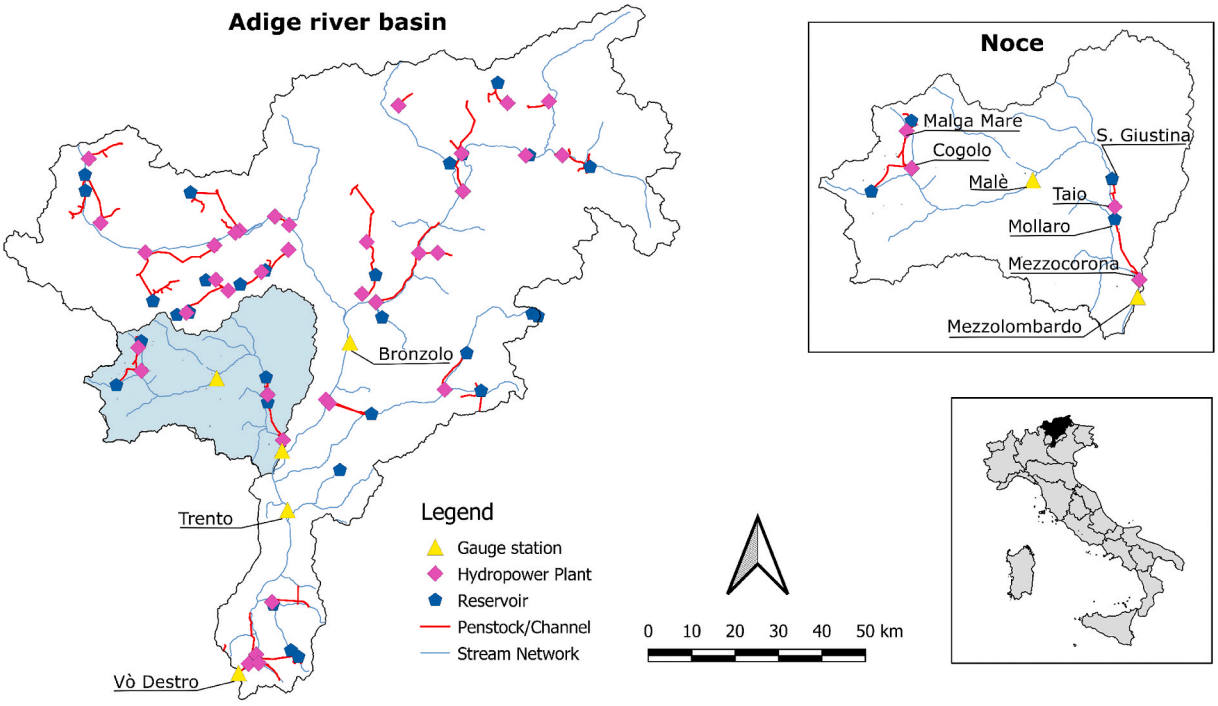

Fig. 7. Map of the Adige river basin with indicated the hydraulic infrastructures. The stream gauging stations adopted for model calibration and validation as well as the Vò Destro outlet are also displayed (yellow dots). Upper right panel provides a detailed view of the Noce catchment, while lower right inset shows the geographic location of the Adige river basin within the Italian territory.

with $p_{0,0}$ in Fig. 6, equally subdivide the sets of parameters to the $n_{masters}$ masters and each master performs the forward simulations with the assigned parameters by using $n_{slaves}+1$ processors in parallel (we recall that a master also acts as slave within its own sub-set), as described in Sect. (4.1). This procedure is iterated until the space of parameters is explored as desired.

With the LHS algorithm the parameters space is explored visiting the locations of a pre-defined grid. The sequence of the locations that are visited is established by the master of masters processor $p_{0,0}$. Therefore, the masters do not communicate among them until the end of the procedure when the efficiency metric at all the explored grid locations is collected and further elaborated according to the objectives of the simulation (see Fig. 6a). For example, a simple sorting will be performed if the objective is to identify the set of parameters providing the best efficiency metric, while more complex elaborations will be performed in case of Bayesian updating. PSO, on the other hand, requires that at the end of each forward simulation the efficiency metric of each particle is compared with its (personal) best and with the overall best (the best among all the particles) in order to update all the personal bests and the global best as well (see Fig. 6b). This is needed because the personal bests and the global best are used in the PSO algorithm to move the particles in the space of parameters at each iteration step. This procedure thus requires inter-master communication since all the forward simulations controlled by the masters need to be completed before updating the position of the particles. If more than one particle is assigned to a master (this may happen when the number of particles is larger than the number of masters) the above control and the movement of the particles can be performed only after all the forward simulations have been completed. The cycling over the parameter sets assigned to each master is illustrated with red boxes in Fig. 6 for both LHS and PSO, while inter-master communication occurring at every PSO iteration is represented with a red horizontal arrow (see Fig. 6b).

5. Example of application

Computational scalability of HYPERstreamHS and its capability to reproduce observed streamflows in locations impacted by the presence of human infrastructures are discussed here with reference to the Adige river basin closed at the gauging station of Vò Destro ($45^{\circ}44^{'}6.5^{''}N$ $10^{\circ}57^{'}26.9^{''}W$) with a contributing area of 10600 km^2 (see Fig. 7). The Adige is a typical Alpine river basin with a complex morphology characterized by deep valleys (131 m a.s.l. at Vò Destro) and high mountain crests (maximum elevation of 3905 m a.s.l. at Cima Ortles, in the northwestern part of the basin). The annual average precipitation ranges from 500 mm in the North-West to 1600 mm in the Southern portion of the basin (Laiti et al., 2018; Diamantini et al., 2018), while streamflow has a typical alpine regime with two seasonal maxima, one occurring in spring-summer due to snow and glacier melt, and the other in autumn triggered by cyclonic storms (Chiogna et al., 2016; Di Marco et al., 2021; Lutz et al., 2016; Mallucci et al., 2019). Fig. 7 shows the natural river network, the superimposed man-made network (red edges), the hydropower plants and the reservoirs. Though fragmented, the man-made network is distributed over the entire river system and exerts a significant impact on streamflow (Zolezzi et al., 2009; Bellin et al., 2016; Majone et al., 2016; Pérez Ciria et al., 2019).

5.1. Model setup

The characteristics of HYPERstreamHS are illustrated by means of two applications: i) streamflow simulation in the Noce river, which is heavily impacted by hydropower, particularly in its lower reaches (see the upper right panel of Fig. 7), and ii) analysis of the computational performances with reference to the entire Adige.

All the simulations were conducted at a daily time step in the period 1989–2006 by using a grid size of 5 km. Streamflow time series collected at the Trento, Bronzolo, Malè and Mezzolombardo gauging stations (see Fig. 7) were used as observations in the inverse modeling experiments. Furthermore, for all configurations the first two years of simulations were used as a spin-up and thus were excluded from the computation of the NSE efficiency index. Precipitation and temperature were extracted from the ADIGE dataset (Mallucci et al., 2019), while land use and land cover, considered invariant in time, were extracted fromthe CORINE 2006 product (http://www.eea.europa.eu/publications/COR0-landcover). The width functions used in the routing scheme were obtained by an offline morphological analysis of the 30 m resolution DEM provided by EUDEM project (https://www.eea.europa.eu/data-and-maps/data/eu-dem). The characteristics of the hydraulic infrastructures were retrieved from the Italian Registry of Dams (RID, available at http://dgdighe.mit.gov.it), and from public reports and informative leaflets of the hydropower companies.

Parameters inference for the Noce experiment was performed adopting the Malè gauging station as calibration node, and by including in the simulations the presence of the hydraulic infrastructures (see the inset of Fig. 7). Management rules of the reservoirs were inherited by the work of Bellin et al. (2016) and are at the monthly time scale. PSO was adopted as calibration algorithm. Validation of the model was performed at the downstream Mezzolombardo gauging station, which streamflow is dominated by the release of the Mezzocorona hydropower plant, which is fed by the water derived from the Mollaro reservoir, which in turn receives the water turbined by the Taio phydropower plant fed by the Santa Giustina reservoir (see the upper right panel of Fig. 7). Results of the simulation are shown in Section 5.2.

Concerning speed-up analyses, a few different levels of detail in representing the hydraulic infrastructures were implemented in order to investigate the computational load of including them in the simulation. In a first setup all the hydraulic infrastructures in the Adige were included: 30 storage reservoirs, 41 intake points and 40 hydropower plants, for a total (including all available streamflow gauging stations) of 138 nodes. Of the 27 gauging stations included into the river basin, the two stations located at Trento and Bronzolo were used for parameters inference. Furthermore, a simplified version of the full setup was adopted, this time only considering 5 out of the 138 nodes: 2 storage reservoirs plus three gauging stations, Trento and Bronzolo used for inference of the model parameters, and Vò Destro as closing section. The two setups differ in the number of macrocells-node pairs that need to be considered in the parellelization of the HYPERstreamHS model; 569 and 1167 macrocells-node pairs for the 5-nodes and 138-nodes cases, respectively.

According to the parallelization strategy described in Sect. 4, the computational performance of HYPERstreamHS and its scalability are tested considering both a single and a dual-layer parallelization scheme (see Sects. 5.3 and 5.4, respectively). In the forward simulations of the one-layer scalability experiment the parameters characterizing the natural hydrological component (see Appendix A) were fixed equal to the values obtained during one of the optimization experiments described in Sect. 5.4. In the case of the dual-layer implementation we considered 500 runs for LHS and 10,000 for PSO (specifically 100 iterations with 100 particles). It is worth to clarify that here we are not interested either in evaluating the convergence to the global optimum with PSO scheme or in a full uncertainty assessment of the posterior *pdf* of the model parameters, but rather to evaluate how HYPERstreamHS scalability is influenced by the adopted setups.

As a quantitative dimensionless metric of scalability we adopted the speed-up, S (e.g. Mendicino et al., 2006):

$$S=\frac{T_s}{T_{np}}, \tag{9}$$

where T_s is the computational time of a serial simulation (i.e., when only a processor is used) and T_{np} is the computational time of the same simulation conducted using np processors. Under ideal scaling condi-

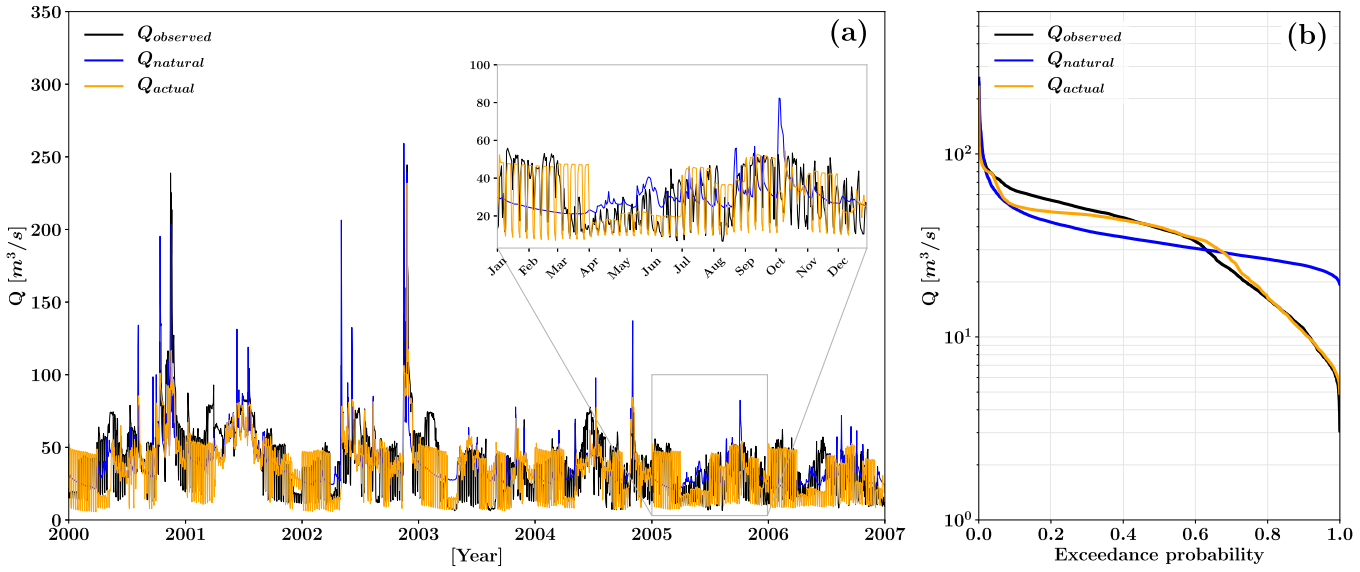

Fig. 8. Comparison of observed (black) and simulated streamflows at the Mezzolombardo gauging station. Simulated streamflows refer to the *actual* and *natural* setup as detailed in Sect. 5.2. a) Daily streamflow time series in the period 2000–2006 with the inset showing the year 2005 to better visualize the significant alteration of the natural flow introduced by the hydropower systems; b) flow duration curves during the entire period 1991–2006.

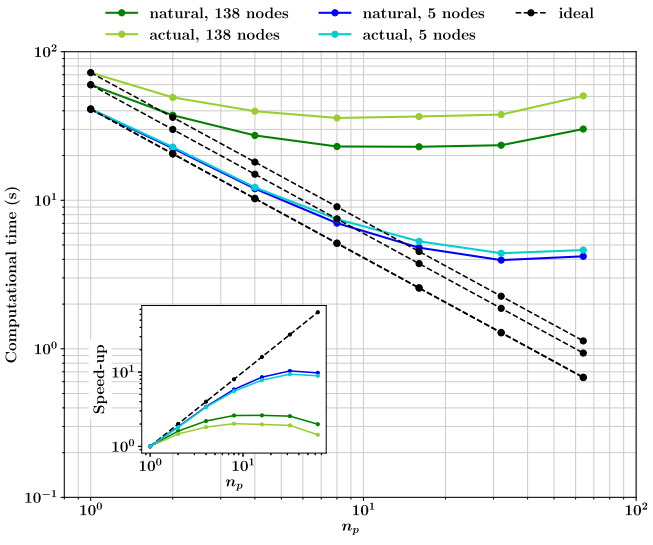

Fig. 9. Computational time of first MPI layer considering 5 and 138 network nodes under both natural and actual (i.e., considering the presence of hydraulic infrastructures) conditions, respectively. The presence of hydraulic infrastructures. The inset in the lower-left corner shows the corresponding speed-up. Theoretical (ideal) computational time and speed-up are represented with dashed lines.

tions speed-up is linear, meaning that a reduction in the computational time of *np* times with respect to the serial simulation is achieved when *np* processor are used. We remark that ideal speed-up, however, is a theoretical condition due both to communication times between processors and to portions of the program that can not be parallelized.

5.2. Hydrological modeling performances in the presence of Human Systems

In the middle course of the Noce river streamflow is stored into the large reservoir of Santa Giustina (storage capacity of about 182 Mm^3) to be used in the Taio hydropower plant (maximum penstock capacity of 66 m^3s^{-1}). The water turbined in the Taio powerplant is then collected by the Mollaro reservoir, which has a small yet appreciable regulation capacity (0.86 Mm^3), and then transferred to the Mezzocorona hydropower plant (maximum penstock capacity of 60 m^3s^{-1}). Mezzocorona plant is located along the Noce main course 2.5 *km* upstream of Mezzolombardo, thus exerting a strong control on streamflow alterations observed at the gauging station (see the inset of Fig. 8).

Calibration of HYPERstreamHS to the 1991–2006 observed streamflow at Malè gauging station provided the satisfactory NSE index of 0.77. The resulting model's parameters have been used to compute streamflow at Mezzolombardo either including or not the hydropower systems of Taio (fed by the Santa Giustina reservoir) and Mezzocorona (fed by the Mollaro reservoir). Hereafter these two cases are referred to as actual and natural conditions, respectively. Fig. 8a compares the time series of both simulations with observations at Mezzolombardo, while the corresponding comparison in terms of flow duration curves is depicted in Fig. 8b. NSE efficiencies were 0.27 and 0.43 in natural and actual conditions, respectively. Though in absolute terms these NSE values indicate that streamflow is reproduced partially, the gain in adding the available, though incomplete, information on hydraulic infrastructures is remarkable. This considering that operational schedule of the reservoir is known only in average and at monthly time scale, which is a limitation in the case at hand, given that hydropower companies typically operate reservoirs at finer time scales, weekly/daily up to the hourly scale, in order to maximize the revenue. Nonetheless, the modeling framework is able to generally reproduce the observed streamflow pattern in terms of both timing and magnitude, with biases occurring solely as a consequence of the non-perfect knowledge of high-resolved in time reservoir operating rules (see the inset of Fig. 8a for easiness of visualization). On the other hand the inclusion of reservoir regulation by means of the Human Systems module allows a very good reproduction of the peak flows.This is not the case of the *natural* simulation which completely fails in reproducing the observed streamflow.

These differences are even more evident when considering the flow duration curves depicted in Fig. 8b. Both *natural* and *actual* cases are able to reproduce fairly well high flows (i.e., 80 m^3s^{-1} and higher),

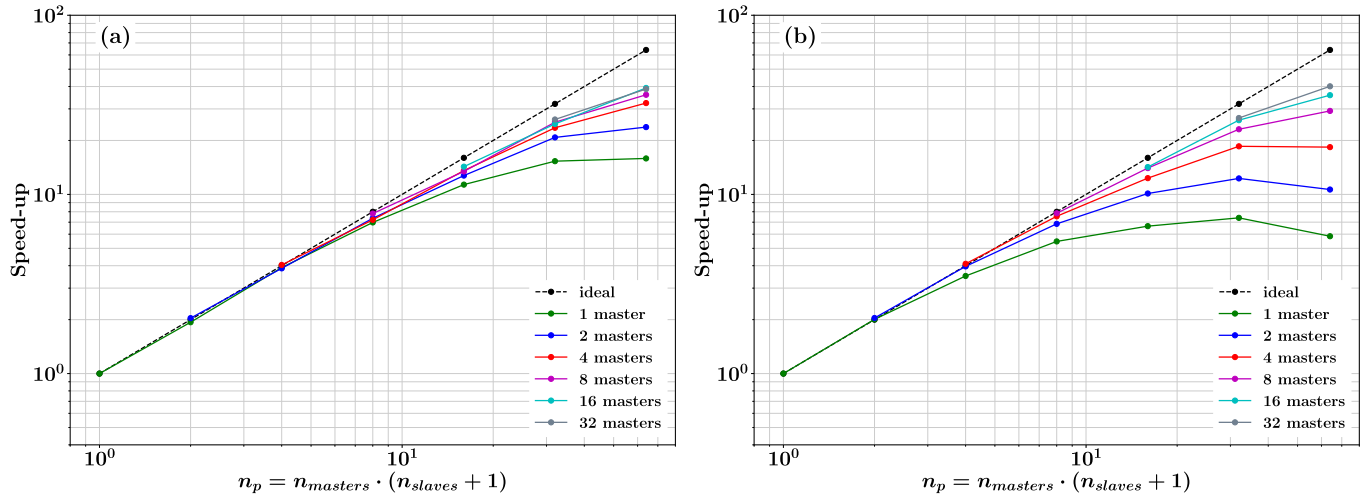

Fig. 10. MPI dual-layer speed-up of parallel LHS scheme for 5 (a) and 138 network nodes (b), respectively. *np* represents the total number of processors as given by different combinations of masters ($n_{masters}$) and slaves (n_{slaves}) used in the simulation. Theoretical (ideal) speed-up is represented with dashed lines.

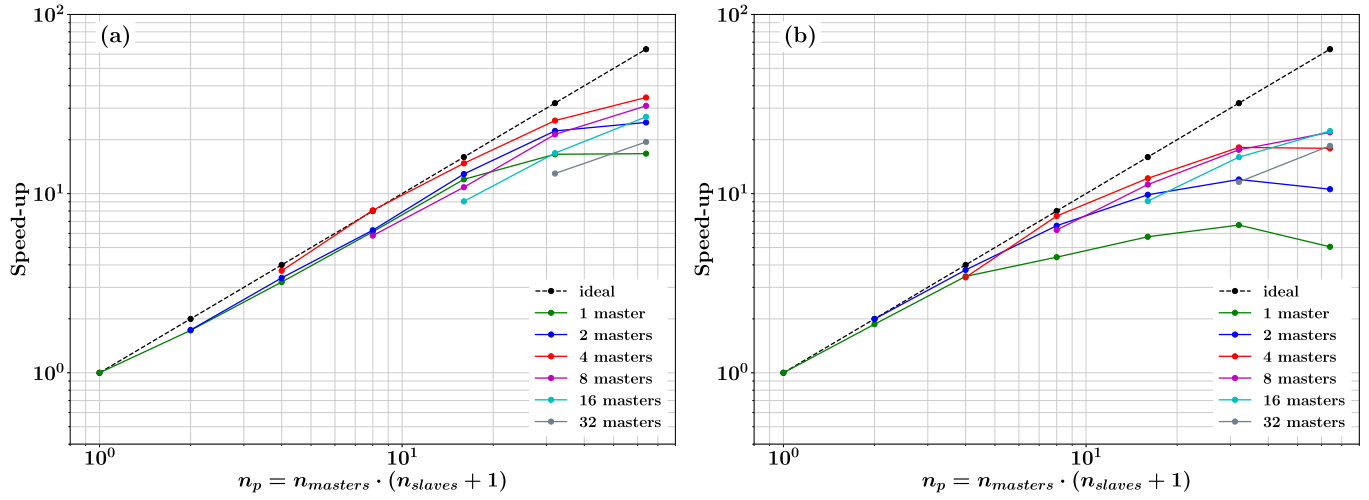

Fig. 11. MPI dual-layer speed-up of parallel PSO scheme for 5 (a) and 138 network nodes (b), respectively. *np* represents the total number of processors as given by different combinations of masters ($n_{masters}$) and slaves (n_{slaves}) used in the simulation. Theoretical (ideal) speed-up is represented with dashed lines.

which are indeed associated to the hydrological response of the watershed. On the contrary, the mismatch widens moving from intermediate to low flows (i.e. 40 m^3s^{-1} and lower), where the natural simulation misses completely the observed streamflow, while the inclusion of the hydraulic infrastructures allows to accurately reproduce the observed flow duration curve.

5.3. First layer parallel performance

In a first set of experiments, aimed at assessing computational performance, HYPERstreamHS was applied to the entire Adige catchment closed at Vò Destro. It was run by choosing 5 and 138 nodes where to compute streamflow and with two different setups: a natural scenario neglecting the impact of hydraulic infrastructures and a second scenario in which the infrastructures are included as described above. Hence, a total of four configurations were investigated. In general, both the number of macrocell-node width functions and the number of interprocessors communications increase with the number of nodes, leading to a larger computational burden, which is increased further when the hydraulic infrastructures are added.

Fig. 9 shows the computational time against the number n_p of processors in log-log scale, while the inset illustrates the corresponding speed-up. Computations have been performed by using 1,2,4,8,16, 32, 64 processors. Theoretical computational time and speed-up, corresponding to 100% efficiency in parallel computing, are shown with dashed lines. As expected, there is a strong dependency on the number of nodes: the test case with 5 nodes shows a better speed-up than the case with 138 nodes and exhibits small differences between the natural and the actual scenario. The low incidence of computations that should be performed in series is the reason of the small difference between the natural and actual scenario for the 5 nodes case. The difference widens when 138 nodes are used, which loss speed-up rather soon (see inset of Fig. 9). For $n_p > 8$ the curve of the speed-up flattens and there is no longer an advantage in increasing the number of processors. This limit is extended to $n_p = 32$ for 5 nodes, though the gain in term of speed-up is significantly lower than the theoretical one when more than 8 processors are used. Both cases show an increase in computational time for $n_p > 32$. This shows also that the fraction of computations with respect to the total that should be performed in series increases with the number of nodes, thereby reducing the gain achieved by using a larger number of processors. In addition, the large number of reservoirs (30) in the case with 138 nodes explains the difference in the computational time

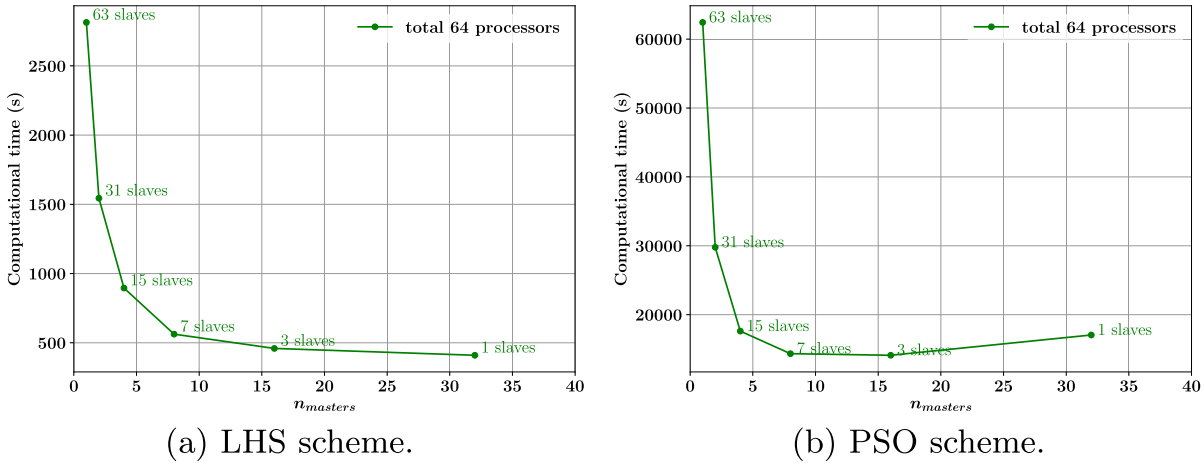

Fig. 12. Computational time of the dual-layer system applied to HYPERstreamHS in the LHS scheme (a) and PSO scheme (b) with a fixed number of 64 processors and different masters and slaves configurations applied to the 138 nodes case.

between natural and actual scenarios (green lines in Fig. 9). In fact, when the number of infrastructures increases, a progressively larger share of time should be spent in performing the water budgets needed to evaluate the effect of the hydraulic infrastructures, as described in Sect. 2.2. In the specific case with 138 nodes the difference between the computational time between the natural and actual scenarios increased from 17.15%, when only one processor is used, up to 35.74%, when 64 processors are employed.

Loss of speed-up due to serial computations involved in streamflow routing is a well-known drawback in parallel computing typical of single layer decomposition schemes (see e.g. Wang et al. (2012) and Zhang et al. (2016)). Nevertheless, a first layer of parallelization of the hydrological model reduces the computational time with respect to a single processor run and, more importantly, allows to handle a larger amounts of data overcoming allocation problems typical of shared memory codes. According to Lai et al. (2018), this is a mandatory feature in order to cope with the large amount of data that should be managed in large-scale simulations. On the light of dual-layer approach we adopted in the present work, the analysis of scalability of the first MPI level represents a preliminary step in order to define the optimal number of processors to be used as slaves in the final dual-layer MPI implementation.

5.4. Second layer parallel performance

In this Section we analyze several combinations of master-slave processors in a two-layer parallelization with the objective to identify possible optimal conditions. These combinations are intermediate between two extreme cases: a single processor at one extreme (the master acts also as a slave), and 32 masters, each one with its own slave for a total of 64 processors, on the other extreme. The former case is assumed as reference because it represents the standard situation of sequential computation. All the analyses are performed with both 5 and 138 network nodes and with the presence of hydraulic infrastructures. Speed-up of the investigated cases, as a function of the number of processors (both masters and slaves, is shown in Figs. 10 and 11 for LHS and PSO, respectively.

Fig. 10a and b shows the results of the simulations conducted with LHS with 5 and 138 network nodes, respectively. For the 5 node case, when a single master is used the speed-up curve flattens as the number of processors increases. On the other hand, speed-up decreases for $n_p > 32$ in the 138 nodes case when 1 or 2 masters are used. This is in accordance with the results presented in Fig. 9, given that for a single master configuration the second MPI layer is deactivated. A similar behavior is observed for 2 and 4 masters in the case of 138 nodes (Fig. 10a), also when inverse modeling is performed with PSO (Fig. 11). However, as the number of masters increases the speed-up for PSO diverges from the theoretical behavior more than for the LHS case, in both the 5 and 138 network nodes configurations (Fig. 11a and b).

The different behavior characterizing the dual-layer scheme with LHS and PSO is further investigated in Fig. 12a and b, respectively. Computations have been performed with 64 processors subdivided in masters (shown in the abscissa) and slaves, indicated above the bullets in the graph. As expected in both cases one master results in the largest computational time, and while for the LHS scheme the minimum is obtained with 32 masters, each one with a slave, for PSO the minimum is obtained with 8 masters with 7 slaves each (see Fig. 12a and 12b). In both cases the computational time reduces quickly as the number of masters increases from 1 to 8 and the marginal gain achieved by increasing the number of masters beyond 8 is small for LHS, while for PSO the computational time increases. This is the consequence of the communication between the masters (only active within the PSO algorithm) that becomes progressively more relevant as the number of masters ($n_{masters}$) increases. Consequently, the optimal configuration of the dual-layer setup depends not only on the number of available processors but also on the number of macrocell-node pairs and the network topology.

Results presented in this Section also highlight that, for a given total number of processors, the dual-layer strategy applied to LHS scheme presents the best performances as the number of masters is increased at the expenses of the slaves, with the only limit being represented by memory availability. On the contrary, PSO requires a preliminary scalability analysis of the forward simulation **to** identify the optimal number of masters dealing with the inverse modeling procedure (i.e., the second MPI layer).

6. Conclusions

We presented a new dual-layer MPI parallel hydrological model, named HYPERstreamHS, specifically developed to deal with large-scale simulations in river basins containing hydraulic infrastructures. The model allows inverse modeling with the Nash-Sutcliffe efficiency metric (but other metrics could be easily implemented) and the space of parameters explored by using either the Latin Hypercube Sampling or Particle Swarm Optimizer.

HYPERstreamHS is designed such as to facilitate parallel computing

being composed by i) a runoff generation module that solves for the vertical water fluxes on the independent macrocells in which the computational domain is subdivided, ii) a streamflow routing scheme based on the Width Function IUH theory with the assumption of constant stream velocity, which makes the transfer process linear. Taking advantage of these characteristics, we built a dual-layer parallelization strategy and designed the inclusion of hydraulic infrastructures (for example reservoirs and diversion channels), in such a way to reduce the computational time through an optimal use of the available processors.

The main characteristics of HYPERstreamHS are as follows:

i) an optimal integration of natural hydrological components with water transfer and accumulation due to the management of hydraulic infrastructures such as reservoirs, diversion channels, intakes and hydropower systems. This results in a holistic modeling framework for hydrological simulations at multiple scales in river basins with streamflow altered by water uses;
ii) the adoption of a dual-layer parallelization strategy where the parallelized version of the hydrological kernel, including the modules dealing with hydraulic infrastructures, is the first-level, with the second level taking care of inverse modeling aimed at the identification of model's parameters, including their uncertainty;
iii) the optimal configuration of the dual-layer setup (masters and slaves partitioning) depends not only on the number of available processors but also on the inverse modeling algorithm adopted;
iv) finally, the dual-layer structure adopts the MPI standard which ensures high portability and optimal exploitation of HPC resources.

HYPERstreamHS has been tested in the Adige river basin (Italy), showing that the adopted dual-layer parallelization scheme optimizes both the speed-up of simulations and the allocation of HPC resources. In particular, the results showed the clear advantages of considering a dual-layer parallelization strategy, which compensates for the sub-optimal gain of the hydrological kernel when increasing the number of processors, especially in more regulated configurations. This suggests to use a portion of the processors for the second layer parallelization dealing with the inversion procedure. Furthermore, the capabilities of the model to reproduce observed streamflows in locations heavily impacted by the presence of human infrastructures have been presented and discussed. Overall, the model showed high potential for applications aimed at supporting water resources management and optimization issues in regulated river basins.

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 received financial support by the Energy oriented Centre of Excellence (EoCoE-II), GA number 824158, funded within the Horizon2020 framework of the European Union, and by the project "Seasonal Hydrological-Econometric forecasting for hydropower optimization (SHE)" funded within the Call for projects "Research Südtirol/Alto Adige" 2019 Autonomous Province of Bozen/Bolzano – South Tyrol. This research has been also supported by the Italian Ministry of Education, University and Research (MIUR) under the Departments of Excellence, grant L.232/2016. Streamflow data were provided by the Hydrographic Office of the Autonomous Province of Bolzano (www.provincia.bz.it/hydro) and by the Ufficio Dighe of the Autonomous Province of Trento (www.floods.it).

A. Natural Hydrological conceptual model

In this Appendix, the vertical water flux generation module adopted in HYPERstremHS model is presented, together with the main concepts of HYPERstream routing scheme (Piccolroaz et al., 2016).

A.1. Computational grid

The spatial domain is partitioned into M macrocells of equal shape and size (Fig. 1b). Furthermore, N nodes are identified, where streamflow is computed (Fig. 1a). Macrocells can be defined such that the hydrological model shares the same grid of an overlaying climate model or of a gridded dataset providing the input meteorological forcing. The nodes are arbitrarily distributed along the river network, and typically are located in correspondence of existing gauging stations where streamflow observations are available (needed for model calibration and validation), or at relevant nodes of hydraulic infrastructures that need to be simulated by the human system module.

A one-time and offline pre-processing step is run to prepare the geometrical information needed to implement the streamflow routing scheme. The DEM is analyzed in order to extract the river network, the drainage characteristics of the study area, and to derive the corresponding geomorphological width functions for each macrocell-node pair (see Fig. 1b). Other properties useful for the evaluation of the water fluxes (e.g., average elevation, soil use and type, crop coefficient etc.) are computed for each macrocell based on the analysis of the available DEM and land-use/land-cover spatial maps. A detailed example of macrocell-discretization and width functions derivations can be found in Piccolroaz et al. (2016).

A.1.2. Vertical water flux module

The vertical water flux module adopted in HYPERstreamHS employs the formulation proposed by Laiti et al. (2018). It relies on the coupling of the continuous soil-moisture accounting SCS-CN scheme for surface flow generation (Michel et al., 2005) with a non-linear bucket model for soil moisture dynamics (Majone et al., 2010), and a linear reservoir model to simulate the base-flow component. In addition, a simple degree-day model is used for the simulation of snow dynamics while Hargreaves and Samani (1982) formulation is employed for the computation of potential evapotranspiration (ET_p). A schematic of the water flux generation module is presented in Fig. 1a, while each component is presented below.

The degree-day model for snow accumulation and melting dynamics is based on the following water balance equation (e.g., Rango and Martinec, 1995; Hock, 2003; Majone et al., 2010):

$$\frac{dh_s}{dt} = p_s - p_m \,, \tag{A.1}$$

where h_s [L] is the snowpack water equivalent, p_s [LT^{-1}] is the solid precipitation intensity, and p_m [LT^{-1}] is the snowmelt intensity. p_s and p_m are quantified according to the following equations depending on two air temperature thresholds:

$$p_s = \begin{cases} p, & T_a \le T_s \\ 0, & T_a > T_s \end{cases} \tag{A.2}$$

$$p_m = \begin{cases} 0, & T_a \le T_m \\ c_m(T_a - T_m), & T_a > T_m \end{cases} \tag{A.3}$$

where, T_a [$^\circ C$] is the air temperature and T_s [$^\circ C$] and T_m [$^\circ C$] are two threshold temperatures. The former is the temperature below which precipitation is assumed as solid, and the latter the temperature above which the snowpack melts. Furthermore, c_m [$L\ T^{-1}\ ^\circ C^{-1}$] is the melting factor providing the amount of snow melted per unit of time and temperature. Notice that when $T_s < T_a \le T_m$ precipitation is liquid, but the energy input to the snowpack is not sufficient for triggering melting. The effective liquid precipitation p_e [LT^{-1}] entering the soil can be then computed as follows:

$$p_e = \begin{cases} 0, & T_a \le T_s \\ p, & T_s < T_a \le T_m \\ p + p_m, & T_a > T_m \end{cases}$$

The degree-day approach used here requires the mean daily air temperature T_a as input, thus provides mean daily values of p_m. If the time step integration of the HYPERstreamHS model is less than one day, the melt water contribution is evenly distributed during the day.

Soil moisture is controlled by the following mass conservation equation:

$$\frac{dSM}{dt} = p_e - q_r - q_p - ET_r \,, \tag{A.4}$$

where SM [L] is the soil moisture, q_r [LT^{-1}] is the surface runoff rate, q_p [LT^{-1}] is the leakage flux (i.e., the water flux from the top soil layer towards groundwater), and ET_r[LT^{-1}] is the real evapotranspiration. The surface runoff rate q_r is evaluated according to the procedure proposed by Michel et al. (2005), which is an extension of the well-known SCS-CN approach (U.S. Soil Conservation Service, 1964) accounting for a time varying soil moisture in the active soil. In particular, q_r is evaluated as follows:

$$q_r = \begin{cases} p_e \dfrac{SM - S_a}{S^*}\left(2 - \dfrac{SM - S_a}{S^*}\right), & SM \ge S_a \\ 0, & SM < S_a \end{cases}$$

where S^* [L] is the maximum potential soil infiltration, and S_a [L] is the threshold above which runoff is generated. S^* is given by the product of the maximum potential infiltration estimated from the land use and lithological characteristics of the soil (S [L]) and a scaling coefficient c_s [-], such that:

$$S^* = c_s S \,. \tag{A.5}$$

This correction allows to account for possible uncertainties in the identification of S. S_a is the soil moisture at the beginning of a precipitation event plus the initial abstraction, and is assumed proportional to S^* through the following relationship:

$$S_a = c_a S^* \,, \tag{A.6}$$

where c_a [-] is a scaling coefficient. Notice that $p_e - q_r$ is the infiltration rate into the soil layer and that its saturation level is given by $S^* + S_a$.

Daily reference evapotranspiration ET_0 [LT^{-1}] is estimated through the equation proposed by Hargreaves and Samani (1982), based on mean, minimum, and maximum daily air temperature T_a. Following Allen et al. (1998), ET_0 is multiplied by a monthly varying crop coefficient K_c [-] in order to estimate potential evapotranspiration ET_p, thus accounting for the presence of specific crop or natural vegetation and their seasonal vegetative conditions:

$$ET_p = K_c 0.408 \left[0.0023 R_a (T_a + 17.8)\sqrt{T_a^{max} - T_a^{min}}\right] \,, \tag{A.7}$$

where R_a is the extraterrestrial radiation [$MJm^{-2}d^{-1}$]. Finally, real evapotranspiration ET_r is computed taking into account that evapotranspiration reaches its potential (upper) limit only when soil moisture is larger than the field capacity SM_{fc} [-], and that ET_r tends to zero as SM approaches its residual limit SM_r:

$$ET_r = \begin{cases} 0, & SM \le SM_r \\ ET_p \dfrac{SM - SM_r}{SM_{fc} - SM_r)} & SM_r < SM \le SM_{fc} \\ ET_p, & SM > SM_{fc} \end{cases}$$

where

$$SM_r = (S^* + S_a)\mathbf{c_r} \,,$$

$$SM_{fc} = (S^* + S_a)\mathbf{c_{fc}} \,,$$

and c_{fc} and c_r are calibration coefficients smaller than one. In particular, for values of SM between SM_r and SM_{fc}, ET_r is assumed to vary linearly from zero to ET_p. Similarly to the snowmelting, ET_r is evaluated as daily average, and if the computational time step of HYPERstreamHS is smaller than one day, ET_r is evenly distributed during the day.

The leakage flux from the active soil q_p is evaluated through the following exponential law

$$q_p = q_{ref}\left[e^{\left(\frac{SM-SM_r}{\mu}\right)} - 1\right] \,, \tag{A.8}$$

where q_{ref} $[LT^{-1}]$ and μ $[L]$ are calibration parameters controlling the maximum and rate of variation of the leakage flux, respectively. We notice that both q_p and ET_r tends to zero as SM approaches SM_r, thus avoiding SM to drop below its lower physical bound SM_r in absence of infiltration (i.e., when $p_e - q_r = 0$). Similar exponential relationships for storage-discharge dynamics coupled with the surface runoff model by Michel et al. (2005) have been successfully applied in previous applications conducted in Alpine and Mediterranean catchments (Piccolroaz et al., 2015; Bellin et al., 2016; Majone et al., 2012, 2016). The leakage flux q_p is then divided into two components through a partition parameter α. The first component contributes to the runoff as interflow q_i $[LT^{-1}]$, whilst the latter constitutes deep percolation to groundwater q_{dp} $[LT^{-1}]$:

$$q_i = q_p\boldsymbol{\alpha} \,,$$

$$q_{dp} = q_p(1-\boldsymbol{\alpha}) \,.$$

Finally, q_{dp} feeds a linear reservoir model described by the following continuity equation:

$$k\frac{dq_b}{dt} = q_{dp} - q_b \,, \tag{A.9}$$

where q_b $[LT^{-1}]$ is the baseflow contributing to the streamflow, and k is a calibration parameter.

The vertical water flux generation model described above is applied to all macrocells separately, using as input the meteorological forcing pertaining to each of them. At each macrocell, the total runoff per unit area q_s $[LT^{-1}]$ is evaluated by summing together the surface runoff rate q_r, the interflow q_i, and the baseflow contribution q_b.

A.1.3. routing algorithm

Following Piccolroaz et al. (2016), the streamflow $Q_k^{(i)}(t)$ $[L^3T^{-1}]$ generated by macrocell i and contributing to the node k at time t reads as follow:

$$Q_k^{(i)}(t) = A_k^{(i)}\int_0^t q_s^{(i)}(t-\tau)f_k^{(i)}(\tau)d\tau = A_k^{(i)}\left(q_s^{(i)}(t) * f_k^{(i)}(t)\right) \,, \tag{A.10}$$

where $A_k^{(i)}$ $[L^2]$ is the fraction (area) of macrocell i contributing to node k, $q_s^{(i)}$ $[LT^{-1}]$ is the vertical water flux per unit area produced by the macrocell i, $f_k^{(i)}$ is the pdf of the travel times of macrocell i relative to node k obtained by rescaling the width function (built in the pre-processing step) by the stream velocity, and the asterisk denotes convolution. In doing this, it is assumed that $q_s^{(i)}$ is constant through the macrocell and is evaluated according to the vertical flux generation module described above.

The total streamflow at node k ($Q_k(t)$) is computed as the sum of the contribution of each macrocell to that node, plus the streamflow transferred from the nodes upstream of k:

$$Q_k(t) = \sum_{i=1}^{M_k^{con}} Q_k^{(i)}(t) + \sum_{j=1}^{N_k^{up}} Q_j\left(t-\tau_{jk}\right) \,, \tag{A.11}$$

where $\tau_{jk} = D_{jk}/V_c$ $[T]$ is the travel time from node j, located upstream of k, to node k, D_{jk} $[L]$ is the distance between the two nodes, V_c $[LT^{-1}]$ is the stream velocity, M_k^{con} is the number of macrocells contributing to node k, and N_k^{up} is the number of nodes upstream of k.

A.4 Model's parameters

The model requires that 12 parameters are set, 11 pertaining to the vertical water flux generation module and 1 to the HYPERstream routing scheme. Spatial heterogeneity of evapotranspiration and infiltration is reproduced by setting the parameters K_c and S as spatially variable with their (deterministic) values chosen from infiltration capacity and soil use maps. All the remaining parameters are assumed as spatially uniform, but uncertain. The routing scheme requires the definition of a single parameter, the stream velocity V_c, which is assumed constant, thus making the model linear and easily parallelizable. The list of the 12 calibration parameters, with their units and range of variation is presented in Table A1.

Table A.1
List of the calibration parameters with their range of variation.

Parameter	Description	Range of variation	Unit
T_s	temperature threshold for snow precipitation	$-2:6$	$[^{\circ}C]$
T_m	temperature threshold for snow melting	$-2:6$	$[^{\circ}C]$
c_m	snow melting factor	$0:10$	$[mm^{\circ}C^{-1}d^{-1}]$
c_s	parameter of the rainfall excess model	$0.1:10$	[-]
c_a	parameter of the rainfall excess model	$0.01:1$	[-]
q_{ref}	parameter of the nonlinear bucket	$10^{-7}:10^{-3}$	$[mm\ s^{-1}]$
μ	parameter of the nonlinear bucket	$0.5:300$	$[mm]$
c_{fc}	coefficient for field capacity	$0:1$	[-]
c_r	coefficient for residual soil moisture	$0:0.25$	[-]
k	mean residence time for baseflow linear reservoir	$200:1000$	$[day]$
α	partition coefficient for leakage flux	$0:1$	[-]
V_c	stream velocity	$0.2:4$	$[m\ s^{-1}]$

References

Allen, R., Allen, R., Food, of the United Nations, A. O., 1998. Crop evapotranspiration: guidelines for computing crop water requirements. No. nos. 56-57 in FAO irrigation and drainage paper. Food and Agriculture Organization of the United Nations. URL. https://books.google.it/books?id=42hRAAAAMAAJ.

Amdahl, G.M., 1967. Validity of the single processor approach to achieving large scale computing capabilities. In: Proceedings of the April 18-20, 1967, Spring Joint Computer Conference. AFIPS '67 (Spring). ACM, New York, NY, USA, pp. 483–485. https://doi.org/10.1145/1465482.1465560. URL.

Bellin, A., Majone, B., Cainelli, O., Alberici, D., Villa, F., 2016. A continuous coupled hydrological and water resources management model. Environ. Model. Software 75, 176–192. https://doi.org/10.1016/j.envsoft.2015.10.013. URL. http://www.sciencedirect.com/science/article/pii/S1364815215300712.

Beven, K., 1993. Prophecy, reality and uncertainty in distributed hydrological modelling. Adv. Water Resour. 16 (1), 41–51. https://doi.org/10.1016/0309-1708(93)90028-E research Perspectives in Hydrology. URL. http://www.sciencedirect.com/science/article/pii/030917089390028E.

Beven, K., Binley, A., 1992. The future of distributed models: model calibration and uncertainty prediction. Hydrol. Process. 6 (3), 279–298. https://doi.org/10.1002/hyp.3360060305. URL. https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.3360060305.

Burstedde, C., Fonseca, J.A., Kollet, S., Feb, 2018. Enhancing speed and scalability of the parflow simulation code. Comput. Geosci. 22 (1), 347–361. https://doi.org/10.1007/s10596-017-9696-2. URL.

Castagna, M., Bellin, A., 2009. A bayesian approach for inversion of hydraulic tomographic data. Water Resour. Res. 45 (4) https://doi.org/10.1029/2008WR007078. URL. https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008WR007078.

Chiogna, G., Majone, B., Paoli, K.C., Diamantini, E., Stella, E., Mallucci, S., Lencioni, V., Zandonai, F., Bellin, A., 2016. A review of hydrological and chemical stressors in the adige catchment and its ecological status. Sci. Total Environ. 540, 429–443. https://doi.org/10.1016/j.scitotenv.2015.06.149, 5th Special Issue SCARCE: River Conservation under Multiple stressors: Integration of ecological status, pollution and hydrological variability. URL. http://www.sciencedirect.com/science/article/pii/S0048969715303430.

Clark, M., Fan, Y., Lawrence, D., Adam, J., Bolster, D., Gochis, D., Hooper, R., Kumar, M., Leung, L., Mackay, D., Maxwell, R., Shen, C., Swenson, S., Zeng, X., 2015. Improving the representation of hydrologic processes in earth system models. Water Resour. Res. 51 (8), 5929–5956. https://doi.org/10.1002/2015WR017096, 8.

Dagum, L., Menon, R., 1998. Openmp: an industry standard api for shared-memory programming. Computational Science & Engineering, IEEE 5 (1), 46–55.

de Paiva, R.C.D., Buarque, D.C., Collischonn, W., Bonnet, M.-P., Frappart, F., Calmant, S., Bulhões Mendes, C.A., 2013. Large-scale hydrologic and hydrodynamic modeling of the amazon river basin. Water Resour. Res. 49 (3), 1226–1243. https://doi.org/10.1002/wrcr.20067. URL. https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/wrcr.20067.

Di Marco, Nicola, Avesani, Diego, Righetti, Maurizio, Zaramella, Mattia, Majone, Bruno, Borga, Marco, 2021. Reducing hydrological modelling uncertainty by using MODIS snow cover data and a topography-based distribution function snowmelt model. Journal of Hydrology. https://doi.org/10.1016/j.jhydrol.2021.126020.

Diamantini, E., Lutz, S.R., Mallucci, S., Majone, B., Merz, R., Bellin, A., 2018. Driver detection of water quality trends in three large european river basins. Sci. Total Environ. 612, 49–62. https://doi.org/10.1016/j.scitotenv.2017.08.172. URL. http://www.sciencedirect.com/science/article/pii/S004896971732171X.

Fersch, B., Senatore, A., Adler, B., Arnault, J., Mauder, M., Schneider, K., Völksch, I., Kunstmann, H., 2020. High-resolution fully coupled atmospheric–hydrological modeling: a cross-compartment regional water and energy cycle evaluation. Hydrol. Earth Syst. Sci. 24 (5), 2457–2481. https://doi.org/10.5194/hess-24-2457-2020. URL. https://hess.copernicus.org/articles/24/2457/2020/.

Giordano, A., Rango, A.D., Rongo, R., D'Ambrosio, D., Spataro, W., 2020. A dynamic load balancing technique for parallel execution of structured grid models. Lecture Notes in Computer Science Numerical Computations: Theory and Algorithms 11973 278–290. https://doi.org/10.1007/978-3-030-39081-5_25.

Gochis, D.J., Barlage, M., C. R. C. M. D. A. F. K. M. M. M. J. R. A. R. L. S. K. Y. D. Z. Y., 2020. The WRF-Hydro modeling system technical description. NCAR Technical Note. https://doi.org/10.5281/zenodo.3625238 (version 5.1.1) Edition. https://ral.ucar.edu/sites/default/files/public/WRFHydroV511TechnicalDescription.pdf.

Gregory, K., 2006. The human role in changing river channels. Geomorphology 79 (3), 172–191. https://doi.org/10.1016/j.geomorph.2006.06.018, 37th Binghamton Geomorphology Symposium. URL. http://www.sciencedirect.com/science/article/pii/S0169555X06002509.

Gropp, W., Lusk, E., Doss, N., Skjellum, A., 1996. A high-performance, portable implementation of the mpi message passing interface standard. Parallel Comput. 22 (6), 789–828. https://doi.org/10.1016/0167-8191(96)00024-5. URL. http://www.sciencedirect.com/science/article/pii/0167819196000245.

Gupta, V.K., Waymire, E., Wang, C.T., 1980. A representation of an instantaneous unit hydrograph from geomorphology. Water Resour. Res. 16 (5), 855–862. https://doi.org/10.1029/WR016i005p00855.

Hargreaves, G., Samani, Z., 1982. Estimating potential evapotranspiration. Journal of the Irrigation & Drainage Division - ASCE 108, 225–230, 01.

Hock, R., 2003. Temperature index melt modelling in mountain areas. J. Hydrol. 282 (1), 104–115. https://doi.org/10.1016/S0022-1694(03)00257-9 mountain Hydrology and Water Resources. URL. http://www.sciencedirect.com/science/article/pii/S0022169403002579.

Kan, G., He, X., Ding, L., Li, J., Hong, Y., Zuo, D., Ren, M., Lei, T., Liang, K., 2018. Fast hydrological model calibration based on the heterogeneous parallel computing accelerated shuffled complex evolution method. Eng. Optim. 50 (1), 106–119. https://doi.org/10.1080/0305215X.2017.1303053. URL.

Karypis, G., Kumar, V., 2009. MeTis: unstructured Graph partitioning and sparse matrix ordering system, version 4.0. http://www.cs.umn.edu/~metis.

Kennedy, J., Eberhart, R., Nov, 1995. Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, pp. 1942–1948. https://doi.org/10.1109/ICNN.1995.488968 vol. 4.

Kumar, A., Singhal, M., 2015. Optimum design of penstock for hydro projects. Int. J. Energy Power Eng. 4, 216–226. https://doi.org/10.11648/j.ijepe.20150404.14, 08.

Kundzewicz, Z., Mata, L., Arnell, N., Doell, P., Kabat, P., Jiménez, B., Miller, K., Oki, T., Şen, Z., Shiklomanov, I., 2007. Freshwater Resources and Their Management, pp. 173–210, 01.

Lai, C., Shi, X., Huang, M., 2018. Efficient utilization of multi-core processors and many-core co-processors on supercomputer beacon for scalable geocomputation and geo-simulation over big earth data. Big Earth Data 2 (1), 65–85. https://doi.org/10.1080/20964471.2018.1434265.

Laiti, L., Mallucci, S., Piccolroaz, S., Bellin, A., Zardi, D., Fiori, A., Nikulin, G., Majone, B., 2018. Testing the hydrological coherence of high-resolution gridded precipitation and temperature data sets. Water Resour. Res. 54 (3), 1999–2016. https://doi.org/10.1002/2017WR021633.

Lazzaro, M.D., 2009. Regional analysis of storm hydrographs in the rescaled width function framework. J. Hydrol. 373 (3), 352–365. https://doi.org/10.1016/j.jhydrol.2009.04.027.

Li, T., Wang, G., Chen, J., Wang, H., 2011. Dynamic parallelization of hydrological model simulations. Environ. Model. Software 26 (12), 1736–1746. https://doi.org/10.1016/j.envsoft.2011.07.015. URL. http://www.sciencedirect.com/science/article/pii/S1364815211001769.

Liu, J., Zhu, A.-X., Liu, Y., Zhu, T., Qin, C.-Z., 2014. A layered approach to parallel computing for spatially distributed hydrological modeling. Environ. Model. Software 51, 221–227. https://doi.org/10.1016/j.envsoft.2011.07.015. URL. http://www.sciencedirect.com/science/article/pii/S1364815213002363.

Liu, J., Zhu, A.-X., Qin, C.-Z., 2013. Estimation of theoretical maximum speedup ratio for parallel computing of grid-based distributed hydrological models. Comput. Geosci. 60, 58–62. https://doi.org/10.1016/j.cageo.2013.04.030. URL. http://www.sciencedirect.com/science/article/pii/S0098300413001568.

Liu, J., Zhu, A.-X., Qin, C.-Z., Wu, H., Jiang, J., 2016. A two-level parallelization method for distributed hydrological models. Environ. Model. Software 80, 175–184. https://

doi.org/10.1016/j.envsoft.2016.02.032. URL. http://www.sciencedirect.com/science/article/pii/S1364815216300524.
Lutz, S.R., Mallucci, S., Diamantini, E., Majone, B., Bellin, A., Merz, R., 2016. Hydroclimatic and water quality trends across three mediterranean river basins. Sci. Total Environ. 571, 1392–1406. https://doi.org/10.1016/j.scitotenv.2016.07.102. URL. http://www.sciencedirect.com/science/article/pii/S0048969716315480.
Majone, B., Bertagnoli, A., Bellin, A., 2010. A non-linear runoff generation model in small alpine catchments. J. Hydrol. 385 (1), 300–312. https://doi.org/10.1016/j.jhydrol.2010.02.033. URL. http://www.sciencedirect.com/science/article/pii/S0022169410001228.
Majone, B., Bovolo, C.I., Bellin, A., Blenkinsop, S., Fowler, H.J., 2012. Modeling the impacts of future climate change on water resources for the gállego river basin (Spain). Water Resour. Res. 48 (1) https://doi.org/10.1029/2011WR010985.
Majone, B., Villa, F., Deidda, R., Bellin, A., 2016. Impact of climate change and water use policies on hydropower potential in the south-eastern alpine region. Sci. Total Environ. 543, 965–980. https://doi.org/10.1016/j.scitotenv.2015.05.009 special Issue on Climate Change, Water and Security in the Mediterranean. URL. http://www.sciencedirect.com/science/article/pii/S004896971530067X.
Mallucci, S., Majone, B., Bellin, A., 2019. Detection and attribution of hydrological changes in a large alpine river basin. J. Hydrol. 575, 1214–1229. https://doi.org/10.1016/j.jhydrol.2019.06.020. URL. http://www.sciencedirect.com/science/article/pii/S0022169419305712.
McKay, M.D., Beckman, R.J., Conover, W.J., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21 (2), 239–245. https://doi.org/10.2307/1268522. URL. http://www.jstor.org/stable/1268522.
Mendicino, G., Senatore, A., Spezzano, G., Straface, S., 2006. Three-dimensional unsaturated flow modeling using cellular automata. Water Resour. Res. 42 (11) https://doi.org/10.1029/2005WR004472. URL. https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2005WR004472.
Michel, C., Andréassian, V., Perrin, C., 2005. Soil conservation service curve number method: how to mend a wrong soil moisture accounting procedure? Water Resour. Res. 41 (2) https://doi.org/10.1029/2004WR003191.
Montanari, A., Shoemaker, C.A., van de Giesen, N., 2009. Introduction to special section on uncertainty assessment in surface and subsurface hydrology: an overview of issues and challenges. Water Resour. Res. 45 (12) https://doi.org/10.1029/2009WR008471. URL. https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2009WR008471.
Mpi Forum, M.P., 1994. Mpi: a message-passing interface standard. Tech. rep. (Knoxville, TN, USA).
Musiał, G., Dbski, L., Jeziorek-Knioła, D., Gołab, K., 2008. A self-scheduling scheme for parallel processing in heterogeneous environment: simulations of the Monte Carlo type. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (Eds.), Parallel Processing and Applied Mathematics. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 429–438.
Nash, J., Sutcliffe, J., 1970. River flow forecasting through conceptual models part i - a discussion of principles. J. Hydrol. 10 (3), 282–290. https://doi.org/10.1016/0022-1694(70)90255-6. URL. http://www.sciencedirect.com/science/article/pii/0022169470902556.
Nazemi, A., Wheater, H.S., 2015a. On inclusion of water resource management in earth system models; part 1: problem definition and representation of water demand. Hydrol. Earth Syst. Sci. 19 (1), 33–61. URL. https://www.hydrol-earth-syst-sci.net/19/33/2015/.
Nazemi, A., Wheater, H.S., 2015b. On inclusion of water resource management in earth system models; part 2: representation of water supply and allocation and opportunities for improved modeling. Hydrol. Earth Syst. Sci. 19 (1), 63–90. URL. https://www.hydrol-earth-syst-sci.net/19/63/2015/.
Ortega, L., Rueda, A., 2010. Parallel drainage network computation on cuda. Comput. Geosci. 36 (2), 171–178. https://doi.org/10.1016/j.cageo.2009.07.005. URL. http://www.sciencedirect.com/science/article/pii/S0098300409002970.
Pérez Ciria, T., Labat, D., Chiogna, G., 2019. Detection and interpretation of recent and historical streamflow alterations caused by river damming and hydropower production in the adige and inn river basins using continuous, discrete and multiresolution wavelet analysis. J. Hydrol. 578 https://doi.org/10.1016/j.jhydrol.2019.124021.
Piccolroaz, S., Di Lazzaro, M., Zarlenga, A., Majone, B., Bellin, A., Fiori, A., 2016. Hyperstream: a multi-scale framework for streamflow routing in large-scale hydrological model. Hydrol. Earth Syst. Sci. 20 (5), 2047–2061. https://doi.org/10.5194/hess-20-2047-2016. URL. https://www.hydrol-earth-syst-sci.net/20/2047/2016/.
Piccolroaz, S., Majone, B., Palmieri, F., Cassiani, G., Bellin, A., 2015. On the use of spatially distributed, time-lapse microgravity surveys to inform hydrological modeling. Water Resour. Res. 51 (9), 7270–7288. https://doi.org/10.1002/2015WR016994.
Pitman, W.V., 1973. A mathematical model for generating monthly river flows from meteorological data in South Africa. Tech. Rep. 2/73 (Johannesburg : University of the Witwatersrand, Dept. of Civil Engineering).
Qin, C.-Z., Zhan, L., 2012. Parallelizing flow-accumulation calculations on graphics processing units-from iterative dem preprocessing algorithm to recursive multiple-flow-direction algorithm. Comput. Geosci. 43, 7–16. https://doi.org/10.1016/j.cageo.2012.02.022. URL. http://www.sciencedirect.com/science/article/pii/S0098300412000787.
Rango, A., Martinec, J., 1995. Revisiting the degree-day method for snowmelt computations. JAWRA Journal of the American Water Resources Association 31 (4), 657–669. https://doi.org/10.1111/j.1752-1688.1995.tb03392.x.
Rinaldo, A., Marani, A., Rigon, R., 1991. Geomorphological dispersion. Water Resour. Res. 27 (4), 513–525. https://doi.org/10.1029/90WR02501.
Rinaldo, A., Rodriguez-Iturbe, I., 1996. Geomorphological theory of the hydrological response. Hydrol. Process. 10 (6), 803–829. https://doi.org/10.1002/(SICI)1099-1085(199606)10:6<803::AID-HYP373>3.0.CO;2-N.
Robinson, J., Rahmat-Samii, Y., 2004. Particle swarm optimization in electromagnetics. IEEE Trans. Antenn. Propag. 52 (2), 397–407. https://doi.org/10.1109/TAP.2004.823969.
Rodríguez-Iturbe, I., Rinaldo, A., 1997. Fractal River Basins –- Chance and Self-Organization. Cambridge University Press, Cambridge.
Rojas, R., Velleux, M., Julien, P.Y., Johnson, B.E., 2008. Grid scale effects on watershed soil erosion models. J. Hydrol. Eng. 13 (9), 793–802. https://doi.org/10.1061/(ASCE)1084-0699(2008)13:9(793).
Rouholahnejad, E., Abbaspour, K., Vejdani, M., Srinivasan, R., Schulin, R., Lehmann, A., 2012. A parallelization framework for calibration of hydrological models. Environ. Model. Software 31, 28–36. https://doi.org/10.1016/j.envsoft.2011.12.001. URL. http://www.sciencedirect.com/science/article/pii/S1364815211002829.
Rueda, A.J., Noguera, J.M., Luque, A., 2016. A comparison of native gpu computing versus openacc for implementing flow-routing algorithms in hydrological applications. Comput. Geosci. 87, 91–100. https://doi.org/10.1016/j.cageo.2015.12.004. URL. http://www.sciencedirect.com/science/article/pii/S0098300415300959.
Scipión, D.E., Mott, R., Lehning, M., Schneebeli, M., Berne, A., 2013. Seasonal small-scale spatial variability in alpine snowfall and snow accumulation. Water Resour. Res. 49 (3), 1446–1457. https://doi.org/10.1002/wrcr.20135. URL. https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/wrcr.20135.
Senatore, A., Mendicino, G., Gochis, D.J., Yu, W., Yates, D.N., Kunstmann, H., 2015. Fully coupled atmosphere-hydrology simulations for the central mediterranean: impact of enhanced hydrological parameterization for short and long time scales. J. Adv. Model. Earth Syst. 7 (4), 1693–1715. https://doi.org/10.1002/2015MS000510.
Tarboton, D.G., Bras, R.L., Rodriguez-Iturbe, I., 1991. On the extraction of channel networks from digital elevation data. Hydrol. Process. 5 (1), 81–100. https://doi.org/10.1002/hyp.3360050107.
Todd, M.C., Taylor, R.G., Osborn, T.J., Kingston, D.G., Arnell, N.W., Gosling, S.N., 2011. Uncertainty in climate change impacts on basin-scale freshwater resources – preface to the special issue: the quest-gsi methodology and synthesis of results. Hydrol. Earth Syst. Sci. 15 (3), 1035–1046. https://doi.org/10.5194/hess-15-1035-2011. URL. https://www.hydrol-earth-syst-sci.net/15/1035/2011/.
Tristram, D., Hughes, D., Bradshaw, K., 2014. Accelerating a hydrological uncertainty ensemble model using graphics processing units (gpus). Comput. Geosci. 62, 178–186. https://doi.org/10.1016/j.cageo.2013.07.011. URL. http://www.sciencedirect.com/science/article/pii/S0098300413002008.
U.S. Soil Conservation Service, 1964. SCS National Engineering Handbook. Washington.
Vivoni, E.R., Mascaro, G., Mniszewski, S., Fasel, P., Springer, E.P., Ivanov, V.Y., Bras, R.L., 2011. Real-world hydrologic assessment of a fully-distributed hydrological model in a parallel computing environment. J. Hydrol. 409 (1), 483–496. https://doi.org/10.1016/j.jhydrol.2011.08.053. URL. http://www.sciencedirect.com/science/article/pii/S0022169411006093.
Vrugt, J.A., 2016. Markov chain Monte Carlo simulation using the dream software package: theory, concepts, and matlab implementation. Environ. Model. Software 75, 273–316. https://doi.org/10.1016/j.envsoft.2015.08.013.
Wang, H., Zhou, Y., Fu, X., Gao, J., Wang, G., 2012. Maximum speedup ratio curve (msc) in parallel computing of the binary-tree-based drainage network. Comput. Geosci. 38 (1), 127–135. https://doi.org/10.1016/j.cageo.2011.05.015. URL. http://www.sciencedirect.com/science/article/pii/S0098300411001919.
Wu, Y., Li, T., Sun, L., Chen, J., 2013. Parallelization of a hydrological model using the message passing interface. Environ. Model. Software 43, 124–132. https://doi.org/10.1016/j.envsoft.2013.02.002. URL. http://www.sciencedirect.com/science/article/pii/S1364815213000327.
Yamazaki, D., Kanae, S., Kim, H., Oki, T., 2011. A physically based description of floodplain inundation dynamics in a global river routing model. Water Resour. Res. 47 (4) https://doi.org/10.1029/2010WR009726. URL. https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2010WR009726.
Zhang, A., Li, T., Si, Y., Liu, R., Shi, H., Li, X., Li, J., Wu, X., 2016. Double-layer parallelization for hydrological model calibration on hpc systems. J. Hydrol. 535, 737–747. https://doi.org/10.1016/j.jhydrol.2016.01.024. URL. http://www.sciencedirect.com/science/article/pii/S002216941600041X.
Zolezzi, G., Bellin, A., Bruno, M.C., Maiolini, B., Siviglia, A., 2009. Assessing hydrological alterations at multiple temporal scales: adige river, Italy. Water Resour. Res. 45 (12) https://doi.org/10.1029/2008WR007266. URL. https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008WR007266.