Geoscientific
Model Development

TerrSysMP–PDAF (version 1.0): a modular high-performance data assimilation framework for an integrated land surface–subsurface model

Wolfgang Kurtz[1,2], **Guowei He**[1,2], **Stefan J. Kollet**[1,2], **Reed M. Maxwell**[3], **Harry Vereecken**[1,2], and **Harrie-Jan Hendricks Franssen**[1,2]

[1]Forschungszentrum Jülich GmbH, Institute of Bio- and Geosciences, IBG-3 (Agrosphere), Jülich, Germany
[2]Centre for High-Performance Scientific Computing in Terrestrial Systems (HPSC-TerrSys), Geoverbund ABC/J, Jülich, Germany
[3]Department of Geology and Geological Engineering and Integrated Groundwater Modeling Center, Colorado School of Mines, Golden, CO, USA

Correspondence to: Wolfgang Kurtz (w.kurtz@fz-juelich.de)

Abstract. Modelling of terrestrial systems is continuously moving towards more integrated modelling approaches, where different terrestrial compartment models are combined in order to realise a more sophisticated physical description of water, energy and carbon fluxes across compartment boundaries and to provide a more integrated view on terrestrial processes. While such models can effectively reduce certain parameterisation errors of single compartment models, model predictions are still prone to uncertainties regarding model input variables. The resulting uncertainties of model predictions can be effectively tackled by data assimilation techniques, which allow one to correct model predictions with observations taking into account both the model and measurement uncertainties. The steadily increasing availability of computational resources makes it now increasingly possible to perform data assimilation also for computationally highly demanding integrated terrestrial system models. However, as the computational burden for integrated models as well as data assimilation techniques is quite large, there is an increasing need to provide computationally efficient data assimilation frameworks for integrated models that allow one to run on and to make efficient use of massively parallel computational resources. In this paper we present a data assimilation framework for the land surface–subsurface part of the Terrestrial System Modelling Platform (TerrSysMP). TerrSysMP is connected via a memory-based coupling approach with the pre-existing parallel data assimilation library PDAF (Parallel Data Assimilation Framework). This framework provides a fully parallel modular environment for performing data assimilation for the land surface and the subsurface compartment. A simple synthetic case study for a land surface–subsurface system (0.8 million unknowns) is used to demonstrate the effects of data assimilation in the integrated model TerrSysMP and to assess the scaling behaviour of the data assimilation system. Results show that data assimilation effectively corrects model states and parameters of the integrated model towards the reference values. Scaling tests provide evidence that the data assimilation system for TerrSysMP can make efficient use of parallel computational resources for > 30k processors. Simulations with a large problem size (20 million unknowns) for the forward model were also efficiently handled by the data assimilation system. The proposed data assimilation framework is useful in simulating and estimating uncertainties in predicted states and fluxes of the terrestrial system over large spatial scales at high resolution utilising integrated models.

1 Introduction

Terrestrial system model predictions are often associated with a considerable degree of uncertainty. These uncertainties stem from a limited knowledge of the governing model

forcing terms and model parameters, which are, for example, related to the spatial and temporal variability of certain model input like precipitation, soil hydraulic properties or vegetation parameters. In addition, the determination of adequate initial conditions for terrestrial system simulations is often highly uncertain. Ensemble-based data assimilation (DA) techniques are gaining increasing attention in the geoscientific community as a tool to merge such uncertain model predictions with uncertain observation data. These techniques follow a Monte Carlo approach in which an ensemble of different model realisations is integrated forward in time. The different model realisations can include various uncertainties in the model input, which then allows one to approximate the variability of model predictions for different model state variables given the different uncertainty sources. These uncertain model predictions are then sequentially conditioned to available observation data where predictions and data are optimally combined according to their uncertainties. This results in updated model states, which are merged closer towards the measurements and provide an improved model forecast for the following time steps. Data assimilation has already been applied to a wide variety of models in different compartments of the terrestrial system, including atmosphere, land surface and groundwater using various kinds of observation data. In this overview we focus on the water and energy cycles of the terrestrial system. The most commonly applied data assimilation algorithm in such systems is the ensemble Kalman filter (EnKF) (Evensen, 1994; Burgers et al., 1998) and its deterministic variants (e.g. Anderson, 2001; Bishop et al., 2001; Tippett et al., 2003). In groundwater hydrology, usually pressure head data are assimilated (Chen and Zhang, 2006; Hendricks Franssen and Kinzelbach, 2008; Nowak, 2009) and to a lesser extend also transport-related data, like solute concentrations (Liu et al., 2008; Li et al., 2012), molar fractions of chemical constituents (Gharamti et al., 2014) or groundwater temperatures (Kurtz et al., 2014). Data assimilation has also been applied for variably saturated conditions in synthetic model set-ups (e.g. Erdal et al., 2014; Song et al., 2014; L. Shi et al., 2015) as well as for real-world data (e.g. Li and Ren, 2011; Wu and Margulis, 2011, 2013). Typically in these cases, point measurements of pressure or soil moisture are assimilated. Data assimilation techniques were also used in the context of coupled surface–subsurface flow (Camporese et al., 2009; Bailey and Baù, 2012; Rasmussen et al., 2015) where the focus is mostly on the assimilation of pressure head and discharge data. In land surface data assimilation the most commonly assimilated data types are remotely sensed soil moisture products or brightness temperatures (Crow and Wood, 2003; De Lannoy et al., 2007; Han et al., 2013) but also land surface temperature (Kumar and Kaleita, 2003; Ghent et al., 2010; Reichle et al., 2010; Han et al., 2013), snow cover data (Andreadis and Lettenmaier, 2006; Su et al., 2010; Xu and Shu, 2014) or leaf area index (Sabater et al., 2008; Ford and Quiring, 2013; Barbu et al., 2014). The assimilation of such observation data into

either land surface or subsurface models usually leads to an improvement of the predictive capability of the respective model. Besides the correction of model state variables, it has become common especially in subsurface and land surface data assimilation to also correct model parameters jointly with model states. The reason is that the parametric uncertainty in such models is rather high compared to other compartments, like the atmosphere, where initial value problems dominate the uncertainty. Examples for the joint correction of model states and parameters in land surface and subsurface models are the correction of hydraulic subsurface parameters like hydraulic conductivity (Chen and Zhang, 2006; Hendricks Franssen and Kinzelbach, 2008; Rasmussen et al., 2015; Pasetto et al., 2015), porosity (Li et al., 2012, 2015), leakage coefficients (Kurtz et al., 2013; Rasmussen et al., 2015), van Genuchten parameters (Li and Ren, 2011; Montzka et al., 2011; Y. Shi et al., 2014; L. Shi et al., 2015), dispersion parameters (Li et al., 2015) or textural information (Han et al., 2014). Other examples in the context of land surface modelling include the estimation of vegetation parameters like stomatal resistance and canopy water storage (Y. Shi et al., 2015) or the estimation of parameters related to land surface flux partitioning (Bateni and Entekhabi, 2012). In most cases, this joint updating of model states and model parameters leads to better simulation results than a correction of model states alone because the uncertainties coming from a wrong parameterisation are reduced.

Data assimilation in the above-mentioned examples is often performed within an individual, isolated geoscientific compartment, e.g. either for the land surface, the subsurface compartment or the atmosphere. However, there is a growing number of publications that emphasise the dynamic feedbacks between different geoscientific compartments. For example, Kollet and Maxwell (2008) demonstrated the sensitivity of land surface fluxes on the depth of the groundwater table. They found a critical water table depth of 1–5 m where the influence on the energy balance is most pronounced and where the influence also depends on soil heterogeneity and land use type. Similar effects of water table depth on land surface fluxes have been found by Rihani et al. (2010), who used an idealised simulation set-up to infer the effects of topography, land cover, atmospheric forcings and subsurface heterogeneity on land surface fluxes. Ferguson and Maxwell (2010) investigated the feedback between groundwater table and energy fluxes under changing climate conditions and found that such interactions depend on the prevailing hydrological conditions (energy limited vs. water limited). Tian et al. (2012) also found a significant influence of water table depth on land energy fluxes for simulations of the Heihe catchment (China). Williams and Maxwell (2011) investigated the propagation of heterogeneity of subsurface parameters and the corresponding soil moisture distribution into the atmosphere and found a strong dependency of land surface fluxes and wind fields on uncertainty in subsurface parameters. Butts et al. (2014) showed that the two-way coupling of a ground-

water and a regional climate model leads to different precipitation and evapotranspiration estimates compared to the stand alone regional climate model especially during summer time. Maxwell et al. (2011), Shrestha et al. (2014) and Rahman et al. (2015) provided further examples how subsurface dynamics affect the development of the atmospheric boundary layer.

Due to these various feedbacks, there is a growing number of modelling platforms that integrate different compartment models for subsurface, land surface and atmosphere, e.g. ParFlow-CLM (Maxwell and Miller, 2005; Kollet and Maxwell, 2008), ParFlow-WRF (Maxwell et al., 2011), COSMO-CLM^2 (Davin et al., 2011), AquiferFlow-SiB2 (Tian et al., 2012), Terrestrial System Modelling Platform (TerrSysMP) (Shrestha et al., 2014) or HIRHAM-MIKESHE (Butts et al., 2014). Such models allow for a more integrated view of the terrestrial system and water cycle in particular and the coupling leads to a physically more consistent description of processes across compartment scales. However, while such integrated modelling approaches provide a better description of model physics, which effectively reduces model structural errors that often occur in single compartment models through the parameterisation of lower or upper boundary conditions, the parameter and forcing uncertainty still remains in such models. Therefore, data assimilation methods may also help to quantify the uncertainties of integrated modelling approaches and to improve their forecast capability through the merging with observation data. Integrated models are usually computationally expensive and often need to be run on a high-performance computational infrastructure. Therefore, there is a need to establish data assimilation frameworks that can efficiently cope with the high computational burden of integrated terrestrial system models. This is especially relevant when simulations are performed at a high spatial resolution and when a relatively high number of model realisations are needed, which is typically the case for ensemble-based data assimilation with land surface and subsurface models.

A number of frameworks exist that can be used to perform data assimilation for specific Earth system components. Land surface examples include the Canadian Land Data Assimilation System (CaLDAS) (Carrera et al., 2015) or the Global Land Data Assimilation System (GLDAS) (Rodell et al., 2004). An example for an atmospheric data assimilation system is provided by Barker et al. (2012) who developed this system for the numerical weather prediction model WRF (WRFDA). Ridler et al. (2014) developed an assimilation system for the hydrological model MIKE SHE. However, these data assimilation systems usually rely on a simplified representation of groundwater dynamics because the process description in most land surface models does not include lateral flows and surface water–groundwater interactions. Additionally, most data assimilation frameworks are unable to perform joint state–parameter estimation, which has been shown to be important in the context of subsurface

and land surface data assimilation. An exception is the data assimilation system for the groundwater model MIKE SHE, which includes lateral groundwater flow and surface water–groundwater exchange and also allows for a joint update of states and model parameters. However, unsaturated flow in MIKE SHE is still only calculated in 1-D.

Besides the above-mentioned data assimilation systems for certain Earth system compartments, there are also a number of generic data assimilation frameworks, which are not tailored to a specific simulation model. Examples of such generic data assimilation frameworks are the Data Assimilation Research Testbed (DART) (Anderson et al., 2009), the Parallel Data Assimilation Framework (PDAF) (Nerger and Hiller, 2013) or the OpenDA framework (OpenDA, 2013). These different frameworks provide various data assimilation algorithms and the necessary computational infrastructure to operate with any kind of simulation model. Ridler et al. (2014) demonstrated the use of the OpenDA framework to establish a data assimilation system for the hydrological model MIKE SHE. This was achieved by connecting both components with the Open Modelling Interface (OpenMI) software. This kind of interfacing is based on Java and .NET technology and can also be used for other OpenMI compliant models. However, the utilised communication approach between model and data assimilation may not be efficient enough to be applied for large data assimilation problems.

In this paper, we present a data assimilation system for the terrestrial system modelling platform TerrSysMP (Shrestha et al., 2014). This modelling platform consists of three component models for the subsurface (ParFlow), the land surface (Community Land Model) and the atmosphere (COSMO-DE). It includes a comprehensive process description of surface and subsurface water flow (3-D variably saturated flow equation, integrated surface–subsurface flow, surface water routing), land surface processes (radiative transfer, energy flux partitioning, photosynthesis, biogeochemical fluxes) and atmospheric processes (convection-permitting formulation of flow equations). In addition, it provides a scale-consistent two-way coupling between the different Earth system compartments. The data assimilation for TerrSysMP is established with the PDAF library (Nerger and Hiller, 2013). Currently, the data assimilation system is restricted to the land surface–subsurface part of TerrSysMP (CLM and ParFlow) and provides the possibility to perform state as well as joint state–parameter updates. The data assimilation framework uses a memory-based communication between model and data assimilation routines and avoids frequent re-initialisations of the model, which is beneficial for the scalability and the application to large-scale hydrological systems. The paper is structured as follows: in Sects. 2 and 3 we first introduce the modelling platform TerrSysMP and the data assimilation software PDAF. In Sect. 4 the technical implementation of the data assimilation system with TerrSysMP and PDAF is described in detail. Section 5 provides an illustrative example of the designed data assimi-

lation framework for a simple land surface–subsurface set-up with a focus on hydrological model states and fluxes. In this section also the scaling behaviour of the data assimilation framework and its applicability for high-resolution modelling problems is tested and presented in detail. Finally, Sect. 6 provides conclusions and an outlook on possible further developments.

2 Terrestrial System Modelling Platform

The recently developed TerrSysMP (Shrestha et al., 2014) is a modular scale-consistent terrestrial system model consisting of three already well-established models for the atmosphere, the land surface and the subsurface (see Fig. 1).

Atmospheric processes are simulated with COSMO-DE (Baldauf et al., 2011), which is the operational forecast model of the German weather service. COSMO-DE is convection permitting and utilises a terrain-following coordinate system with variable vertical layer thickness. For more details on the model physics see Shrestha et al. (2014).

The land surface part of TerrSysMP consists of the CLM version 3.5 (Oleson et al., 2004, 2008). CLM calculates the transfer of energy, momentum and carbon between the subsurface, vegetation and the atmosphere. In CLM, the subsurface is represented with 10 soil layers of variable thickness with a total extent of 3 m. Soil water and soil temperature dynamics are calculated only in a vertical direction; i.e. there is no lateral exchange between grid cells. Snow accumulation is represented with up to five snow layers on top of the soil layer. Vegetation is parameterised with up to 16 plant functional types providing the plant physiological parameters that are used to calculate the contribution of vegetation to radiative transfer, land surface fluxes and carbon dynamics. CLM provides prognostic variables for the subsurface (soil moisture, soil temperature, groundwater storage), surface water routing, land surface fluxes (evaporation from ground and vegetation, transpiration from vegetation, sensible heat fluxes from ground and vegetation), radiative transfer (adsorption/transmittance of solar radiation, adsorption/emission of short-wave radiation) and carbon fluxes.

The subsurface part of TerrSysMP consists of the variably saturated finite-difference groundwater model ParFlow (Ashby and Falgout, 1996; Jones and Woodward, 2001; Kollet and Maxwell, 2006; Maxwell, 2013). ParFlow solves the 3-D Richards equation and includes a surface water routing scheme, which is based on the kinematic wave approximation of overland flow coupling subsurface and overland flow in an integrated fashion (Kollet and Maxwell, 2006). The system of partial differential equations is solved with a Newton–Krylow method (Jones and Woodward, 2001). Additionally, ParFlow provides a terrain-following grid transform with variable vertical discretisation (Maxwell, 2013), which allows it to solve groundwater problems with high topographic gradients.
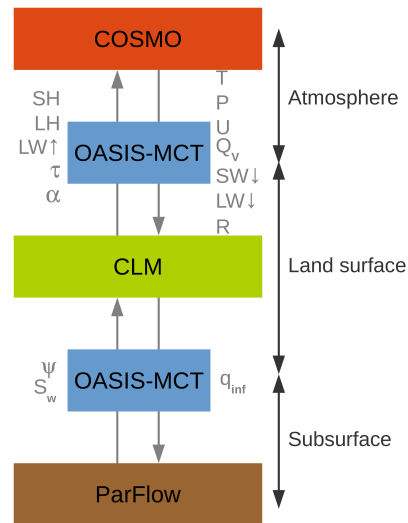
Figure 1. Coupling of the TerrSysMP component models ParFlow (subsurface), CLM (land surface) and COSMO-DE (atmosphere) by OASIS-MCT. The exchanged fluxes and state variables are: ψ (subsurface pressure), S_w (subsurface saturation), q_{inf} (net infiltration flux), SH (sensible heat flux), LH (latent heat flux), LW $\uparrow$ (outgoing long-wave radiation), τ (momentum flux), α (albedo), P (air pressure), T (air temperature), U (wind velocity), SW $\downarrow$ (incoming short-wave radiation), LW $\downarrow$ (incoming long-wave radiation), Q_V (specific humidity) and R (precipitation).

The coupling of the three component models of TerrSysMP is accomplished with the coupling software OASIS-MCT (Ocean-Atmosphere-Sea-Ice-Soil coupler – Model Coupling Toolkit) (Valcke, 2013; Valcke et al., 2013; Gasper et al., 2014). The OASIS-MCT coupler is a library that provides a generic interface to exchange information between two models. OASIS-MCT routines are called during the initialisation stage of each component model to define the model variables that should be exchanged between the component models and to establish the parallel communication between the coupled models. The exchange of variables then takes place during the runtime of the models by calling OASIS-MCT routines at explicitly defined time intervals. During this exchange of data between models, it is also possible to define interpolation and scaling operations for the respective variables. The coupled models within TerrSysMP are run in a multiple program multiple data (MPMD) fashion; i.e. the different program executables are started independently in the same parallel environment and share the same global communicator (`MPI_COMM_WORLD`). This global communicator is utilised by the OASIS-MCT library functions to establish the data transfer between the different component models. Note that the data assimilation framework for TerrSysMP presented in this study does not follow this MPMD program execution mode any more (see Sect. 4).

The data that are exchanged in TerrSysMP via OASIS-MCT are schematically shown in Fig. 1. ParFlow provides

CLM with its calculated subsurface pressure (ψ) and saturation (S_w) values for the first 10 subsurface layers and in return CLM provides the upper boundary condition for ParFlow, consisting of the recharge values (q_{inf}) that are calculated based on the land surface fluxes of CLM (precipitation, interception, total evaporation, total transpiration). In the land surface–atmosphere part of TerrSysMP, CLM provides land surface fluxes (sensible heat flux SH and latent heat flux LH), outgoing long-wave radiation (LW $\uparrow$), momentum flux (τ) and albedo (α) as a lower boundary condition to COSMO-DE. In turn, COSMO-DE provides forcing data to CLM including air pressure (P), air temperature (T), wind velocity (U), incoming short-wave (SW) and long-wave (LW$\downarrow$) radiation, specific humidity (Q_V) and precipitation (R).

The advantages of this integrated modelling approach with TerrSysMP are twofold:

1. The coupling of the different component models improves the physical representation especially at the interfaces of the different geoscientific compartments. For example, ParFlow replaces the simplified soil hydrology (1-D only) and surface water routing (uncoupled) schemes in CLM by a fully integrated 3-D variably saturated surface–subsurface flow model. In COSMO the simplified land surface scheme TERRA is replaced with the more sophisticated land surface scheme of CLM, for example, concerning the representation of vegetation.

2. This modelling approach allows for an integrated view of the terrestrial water, energy and carbon cycles because the dynamic feedbacks of the different geoscientific compartments are explicitly taken into account.

Another interesting feature of TerrSysMP is its modularity: apart from the fully coupled system (ParFlow, CLM and COSMO-DE) it is also possible to compile and run only the land surface–subsurface part (CLM and ParFlow) or the land surface–atmosphere part (CLM and COSMO-DE) or each of the component models individually. Regarding the parallel performance, TerrSysMP has already shown to be highly scalable on the massively parallel supercomputing environment JUQUEEN (Jülich BlueGene/Q) (Gasper et al., 2014).

3 Parallel Data Assimilation Framework

The PDAF library (Nerger and Hiller, 2013) provides a generic framework for applying data assimilation with any kind of geoscientific model. PDAF provides parallel algorithms of already well-established data assimilation methods like the ensemble Kalman filter (Evensen, 1994; Burgers et al., 1998) or the local ensemble transform Kalman filter (LETKF) (Hunt et al., 2007). Furthermore, PDAF provides the user with generic routines to interface the model with the data assimilation algorithms and it includes methods for establishing the parallel communication for the model and

the data assimilation algorithms. The data transfer (coupling) between the model and the data assimilation module can be handled in two ways:

– offline coupling: data exchange via the input/output files of the model;

– online coupling: data exchange via main memory.

The first method (offline coupling) is more ad hoc and also applicable when the source code of the model is not available. In this case, the user needs to take care of the execution of the model forward runs to the next assimilation cycle. An additional executable containing calls to PDAF routines is then used to perform the data assimilation. Within this executable, PDAF reads the state vector from the model output files, performs the assimilation and writes out the assimilation results in the form of input files for the next model integration. One drawback is that this coupling method produces a lot of I/O overhead because a huge number of files has to be read and written at each assimilation step. Another drawback of the offline coupling is that the model needs to be re-initialised after each assimilation step. In the second variant (online coupling), PDAF is directly integrated into the model source code. This enables a direct data transfer between the model and the data assimilation algorithms of PDAF via main memory. Additionally, the model only needs to be initialised once because the model integration is only paused for the data assimilation with PDAF within the time stepping loop of the model. This coupling variant is significantly faster in terms of CPU time but requires more programming effort and the availability of the model source code.

The model coupling for both coupling variants (offline and online) is defined by the user through the aforementioned generic interface routines that are provided by PDAF. These routines include

– The definition of the state vector for PDAF, which has to be provided by the model either from the model output files or via exchange in main memory.

– The definition of the measurement vector and the corresponding measurement uncertainties and error covariances (usually via observation files).

– Rules on how the updated state vector is transferred back to the model.

– Pre- and post-processing routines, e.g. printing out diagnostic information on the assimilation process.

These interface routines partly depend on the filter algorithm that is used for data assimilation. For example, local filter variants like LETKF need special routines to infer the position of each element of the state vector in the model domain in order to perform the localisation. Also, the definition of the error covariance matrix can vary depending on

the application and several interface routines are provided by PDAF to construct this matrix.

At the very beginning of the initialisation phase of the model, a PDAF routine needs to be called that establishes the parallel communication within the model and the data assimilation algorithms. This is especially important for fully parallel models like the ones in TerrSysMP. In this phase, PDAF creates three parallel communicators: the model communicator, the coupling communicator and the filter communicator. The general layout of these communicators is depicted in Fig. 2 for a model set-up with three ensemble members and four processors per model realisation. The model communicator is created for each ensemble member separately and in the case of a parallel model it is equal to the models' internal communicator (i.e. a replacement of `MPI_COMM_WORLD`). The filter communicator is used to perform the filter algorithms that are only applied on the processors of the first ensemble member (marked in red colour in Fig. 2), while the other processors remain idle during the filter update. Within the filter communicator, the processors exchange information about the simulation results at observation points and global ensemble statistics (ensemble mean and variance). The coupling communicator is the communicator for exchanging data between the processors in the filter communicator and the remaining ensemble members before and after the assimilation step. As noted by the arrows in Fig. 2, this data exchange is according to the processor ranks in the model communicator; i.e. the data exchange only takes place for each subdomain of the model and not on a global level. A global vector of model states is never used in this scheme.

4 TerrSysMP–PDAF

This section describes the implementation and usage of the data assimilation system for TerrSysMP, which is referred to as TerrSysMP–PDAF in the following.

4.1 Technical implementation

In order to establish a data assimilation system for the land surface–subsurface part of TerrSysMP (CLM and ParFlow) with the data assimilation framework PDAF, an interface between the model and the data assimilation framework was created. As the forward model is already computationally very demanding and the source code of the model is readily available, we decided to follow the online coupling approach (data exchange via main memory and not running the model as a single executable) in order to avoid frequent re-initialisations of the model and a significant overhead in I/O operations both of which degrade the performance of the constructed data assimilation system for TerrSysMP. In order to accomplish this, several changes in the source code and the building script of TerrSysMP had to be undertaken. First, the main program sections of the two component mod-
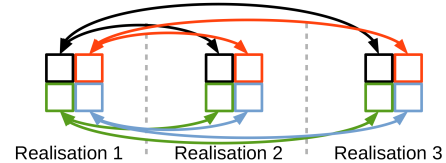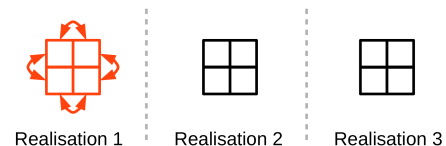

Figure 2. Communicators in PDAF for a parallel set-up with three ensemble members and four processors per ensemble member. Colours indicate the membership of the respective processors and arrows exemplify the parallel communication between the different processors.

els were split into separate callable routines for initialising, advancing and finalizing the respective model. With these routines the models were packed into (pseudo-)libraries making them callable from within the main program section of TerrSysMP–PDAF. This step was undertaken to keep all the necessary model data in main memory and thus avoid a frequent re-initialisation of the model, which, for example, would be present if TerrSysMP was called as a binary program. Second, several changes in OASIS-MCT were necessary to allow for the propagation of all ensemble members at once. One problem that arose here was that OASIS-MCT implicitly used the global communicator `MPI_COMM_WORLD` to establish the exchange of state and flux variables between the component models of TerrSysMP. As only one `MPI_COMM_WORLD` can be present within a MPI job this prevented the ensemble propagation. Therefore, this implicit declaration of the data exchange communicator was replaced in several OASIS-MCT routines and the data exchange communicator was replaced with the model communicator that is provided by PDAF. Third, several CLM and OASIS-MCT output files have a fixed naming convention. In order to avoid an overwriting of these files by the different ensemble members it was necessary to rename these output files. This was done by adding the realisation number of the respective ensemble member to these file names. With these changes in

the model source code (and building procedure), it was possible to combine the model libraries for CLM and ParFlow (including OASIS-MCT) with the data assimilation libraries provided by PDAF in one main program. Figure 3 sketches the different components of the TerrSysMP–PDAF framework. The TerrSysMP–PDAF driver (i.e. the main program) controls the whole framework. This includes the initialisation and finalisation of MPI, TerrSysMP and PDAF as well as the time stepping control for the model forward integration and the data assimilation. The TerrSysMP wrapper is used to interface the driver program with the individual model libraries (libclm and libparflow coupled via OASIS-MCT). The PDAF user(-defined) functions are specifically adapted to TerrSysMP and the desired assimilation scheme (EnKF in this case) and include, for example, the definition of the state vector, the observation vector and the observation error covariance matrix. These data are either provided by the model directly (e.g. state vector) or are read from files or command line options (e.g. observations and observation errors). The PDAF core functions provide the algorithms for different filtering methods. This part of PDAF is not modified for the implementation of TerrSysMP–PDAF because the input for the PDAF core functions (e.g. state vector, observation vector, observation error covariance matrix) is already provided by the PDAF user functions.

The TerrSysMP–PDAF driver program proceeds in the following steps:

1. initialisation of MPI;

2. initialisation of the parallel communication by PDAF;

3. model initialisation for CLM and ParFlow;

4. initialisation of data structures in PDAF (state vector, measurement vector, etc.);

5. time loop over measurement time steps:

    a. advance CLM and ParFlow to the next assimilation time step;

    b. filter step by PDAF;

    c. update of the relevant model variables in CLM and ParFlow;

6. finalisation of PDAF, CLM and ParFlow.

In steps 1 and 2, the global MPI communicator as well as the PDAF communicators (see Sect. 3 and Fig. 2) are initialised. In step 3, all processors first read a common input file, which holds information about specific settings for the data assimilation run. This includes the number of processors for CLM and ParFlow for each model realisation, the number of ensemble members, time stepping information, specification of the observation data and the model variables that should be updated as well as settings for the model output. Then, within each realisation (model communicator) each
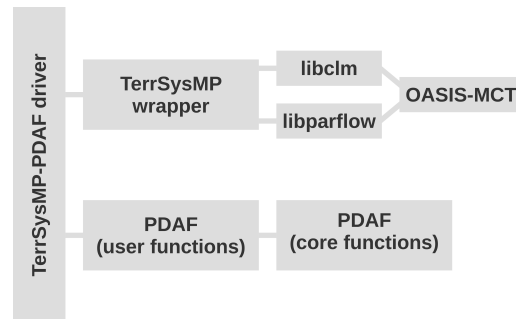
Figure 3. Components of TerrSysMP–PDAF.

processor is assigned either to CLM or ParFlow depending on the processor rank within the model communicator. An example of this model assignment is given in Fig. 4. The first processors within a model communicator are always assigned to CLM and the rest to ParFlow summing up to the total number of processors for each model realisation. Afterwards, each of the component models is initialised with the initialisation function of the corresponding model library (see above). In this step also the model communicators of the respective model realisation are handed over to OASIS-MCT. For the model initialisation, each realisation of CLM and ParFlow reads a different model-specific input file (see Sect. 4.2), which holds information about the specific initial conditions, forcings or parameters of the corresponding realisation. Furthermore, in this step some data structures, which hold the model-specific state vector, are created. In step 4, the data structures for the data assimilation in PDAF are created. This, for example, includes the size of the state and measurement vector, the matrices for model and measurement covariances, etc. After the initialisation phase of TerrSysMP–PDAF (steps 1 to 4) the time loop over the assimilation cycles takes place. Note that this loop only refers to the updating cycles and that CLM and ParFlow can run at a smaller time step; i.e. the updating cycle is a multiple of the model time steps. For each assimilation cycle, first TerrSysMP is advanced to the next observation time in step 5a. At the end of this step the data structure holding the state vector for the respective model component is filled with the corresponding model variable. The model variables that form the state vector are described further below. Next, in step 5b, the data assimilation algorithm in PDAF is called. In this step the model state vectors are collected on the filter communicator with the help of the coupling communicator (see Fig. 2). Then the observation data are read from netCDF files, which hold the measurement values, the corresponding measurement errors and information on their spatial location. The spatial location of observations has to be provided in the form of model grid cell indices; i.e. the user needs to determine the grid cells that match the observation locations. The grid cell indices provided in the observation files are then handed over to PDAF, which will use these indices to extract the simulation
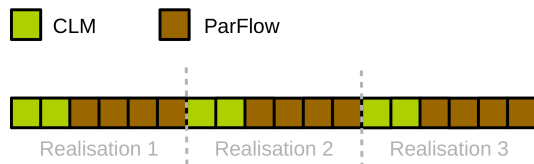
Figure 4. Example of the processor layout of TerrSysMP–PDAF for three model realisations where each realisations is simulated with two processors for CLM and four processors for ParFlow.

results at observation locations from the state vector. Note that by using grid cell indices, no interpolation or other kind of measurement operation is performed because the observations are simply clipped to the nearest model grid cell. The measurement covariance matrix in the current implementation is always diagonal (i.e. the measurement errors for the different observations are uncorrelated) and the measurement error can be different for the individual observations. Afterwards, the filter update is performed. The choice of the data assimilation algorithm for TerrSysMP is currently restricted to the ensemble Kalman filter. After the filtering step, the updated state vector is transferred back to the corresponding model variables in step 5c and TerrSysMP–PDAF proceeds to the next assimilation cycle. When all assimilation cycles are finished, the data structures of the individual components of TerrSysMP–PDAF are deallocated in step 6 and the program is shut down.

Time stepping for the TerrSysMP component models as well as in the data assimilation loop is static; i.e. there is a constant time step for the model integration of TerrSysMP and a constant frequency for the assimilation steps, which is a multiple of the TerrSysMP time step.

As TerrSysMP is designed in a modular fashion, the same approach was also chosen for the data assimilation system for the land surface–subsurface part of TerrSysMP. That is, the data assimilation system can run only with ParFlow, only with CLM or for CLM and ParFlow coupled with OASIS-MCT. In the case of using a single model of TerrSysMP, the aforementioned changes of OASIS-MCT communicators for allowing an ensemble propagation are not applicable any more. Instead, the model communicator of PDAF is directly transferred to the internal model communicators of CLM or ParFlow.

In the coupled (ParFlow + CLM) and uncoupled (ParFlow stand alone) TerrSysMP configuration, measurements of pressure or soil moisture can be assimilated in ParFlow. The assimilation of both measurement types involves an update of pressure values in ParFlow because this is ParFlow's prognostic variable and soil moisture (or saturation) is a derived quantity, which is not directly used as a state variable for the next time step. For the assimilation of pressure data, simulated pressure values in ParFlow are directly modified by the pressure observations. For the assimilation of soil moisture data, two options are implemented in TerrSysMP–PDAF

to update pressure values with soil moisture observations: (1) the state vector consists of soil moisture and pressure values of ParFlow. In this case, pressure values in ParFlow are indirectly corrected with the incoming soil moisture measurements through the correlations between soil moisture and pressure. (2) The state vector solely contains soil moisture values and the updated soil moisture values are transformed back to pressure values via the "inverse" van Genuchten function before the next time step. Apart from the state update, it is also possible to include permeability values or Mannings coefficients of ParFlow in the state vector (both log-transformed) and thus to correct these model parameters with incoming pressure or soil moisture measurements. In case the data assimilation framework is only applied with the CLM component, the state vector is constructed with the soil moisture provided by CLM, which can be corrected with incoming soil moisture measurements.

4.2 Installing and running TerrSysMP–PDAF

TerrSysMP–PDAF can be seen as an add-on to a regular TerrSysMP installation. A patch script is provided that applies the necessary code changes in OASIS-MCT, ParFlow and the build script of TerrSysMP (see Sect. 4.1). All other routines (e.g. initialisation of parallel communication with PDAF, user specified routines to create the state vector for PDAF, wrapper functions for TerrSysMP) are additional components on top of a regular TerrSysMP installation. In order to run TerrSysMP–PDAF, the user needs to provide separate model input files for CLM and/or ParFlow for each single ensemble member following a certain naming convention (⟨problemname_xxxxx⟩), where xxxxx is the number of the realisation preceded by zeros. In each of these input files the user can specify a different input for initial conditions, forcing terms or parameters, which are used to approximate the variability of the model prediction in the data assimilation framework. These files have the same structure as the standard model input files for ParFlow and CLM except for the special naming convention. Additionally, an input file for the control of the data assimilation has to be provided, which includes information on the number of model realisations, the number of processors that are used for each model component, the timing information, information on the desired updating scheme (e.g. kind of observation data and additional parameter update) and settings for the output profile of TerrSysMP–PDAF.

5 Illustrative example

In order to illustrate the TerrSysMP–PDAF framework and investigate its scaling properties, we provide a simple synthetic data assimilation exercise (twin experiment) in the following section. This example deals with the assimilation of soil moisture data into a virtual catchment, which
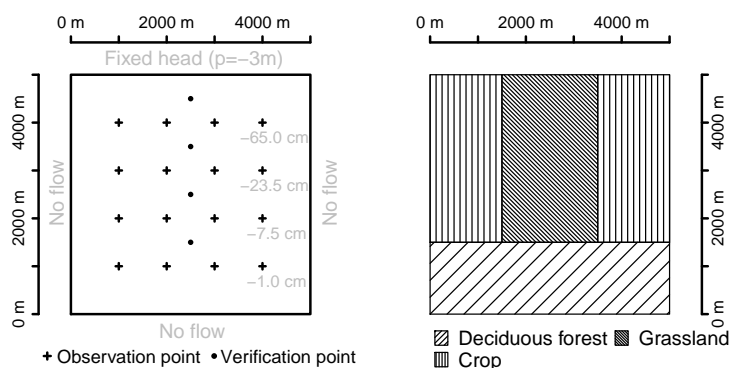
Figure 5. Synthetic set-up for the twin experiment. The left hand figure shows boundary conditions of the subsurface model (ParFlow) and the location of observation (crosses) and verification (filled circles) points. Grey numbers indicate the depth of observation and verification nodes, which are constant in west–east direction. The right hand figure shows the spatial distribution of plant functional types used in the land surface model (CLM).

is set up for the land surface–subsurface part of TerrSysMP (CLM + ParFlow). A synthetic reference run with predefined subsurface parameters (spatially distributed field of saturated hydraulic conductivity K_s) provides virtual measurement data of soil moisture content for several observation locations. The ensemble for the data assimilation experiment consists of different realisations for spatially distributed saturated hydraulic conductivity and different precipitation rates. The synthetic soil moisture observation data are used to jointly update soil water content and saturated hydraulic conductivity with EnKF on a daily basis.

5.1 Experimental set-up

The domain of the virtual catchment has a horizontal extension of $5000\,\text{m} \times 5000\,\text{m}$ and is discretised into 200×200 grid cells with a grid cell size of $25\,\text{m} \times 25\,\text{m}$ (see Fig. 5). The subsurface domain (modelled with ParFlow) has a vertical extension of 13 m, which is discretised into 20 cells with variable thickness. The uppermost 10 subsurface layers in ParFlow have an exponentially increasing profile with depth, corresponding to the soil layer thicknesses in CLM. These 10 layers sum up to a total thickness of 3 m. Note that for these 10 layers, pressure, saturation and land surface fluxes are exchanged between CLM and ParFlow (see Sect. 2). The 10 remaining subsurface layers have a constant thickness of 1 m. The topography of the model domain is flat, which means that there is no topographically driven overland flow within the domain. The porosity is set to a value of $0.4\,\text{m}^3\,\text{m}^{-3}$ and the specific storage to $1 \times 10^{-4}\,\text{m}^{-1}$ and both are spatially constant throughout the subsurface domain. Variably saturated flow was parameterised with the van Genuchten model (van Genuchten, 1980). The van Genuchten parameters α and n were both set to a spatially constant value of $2.0\,\text{m}^{-1}$ and 2.0, respectively.

The land surface is covered by three vegetation types (i.e. plant functional types) in this example: deciduous forest,

cropland and grassland (see Fig. 5). The meteorological forcings (Fig. 6) are taken from reanalysis data of the German Weather Service (DWD) for the year 2013 for a grid cell close to Jülich (Germany) and the assigned meteorological forcings are spatially homogeneous within the virtual catchment. The boundary conditions for the subsurface domain are no flow in the southern, eastern and western faces and a constant head boundary condition (water table depth of -3 m) at the northern face. The initial groundwater table in all simulations is linearly decreasing from -2 m at the southern boundary to -3 m at the northern boundary.

For the data assimilation experiments a synthetic reference run was created with the model mentioned above. The synthetic reference field of $\log_{10}(K_s)$ was generated with two dimensional unconditioned sequential Gaussian simulation using the gstat package (Pebesma, 2004) in the statistical software R (R Core Team, 2015). A spherical variogram with a nugget of $0.0\,\log_{10}(\text{m}\,\text{h}^{-1})$, a sill of $0.1\,\log_{10}(\text{m}^2\,\text{h}^{-2})$ and a range of 70 model grid cells (1750 m) was used for the simulations. A constant value of $-3\,\log_{10}(\text{m}\,\text{h}^{-1})$ was added to the generated $\log_{10}(K_s)$ field and the final field was assigned to each model layer. The synthetic reference simulation was run for 6 months (January–June 2013, 181 days) with an hourly time step for both ParFlow and CLM. Observation data (soil water content) from this reference run are collected at 16 observation points (Fig. 5), which are evenly distributed over the whole virtual catchment. Observations are taken at different depths ranging from the uppermost model layer (-1 cm) in the south to the sixth model layer (-65 cm) in the north. Additionally, four verification points are defined to assess the effect of soil moisture assimilation in between the observation points.

For the data assimilation experiment, an ensemble of 128 realisations of subsurface parameters ($\log_{10}(K_s)$) and meteorological forcings was created. The $\log_{10}(K_s)$ fields were also generated with unconditioned sequential Gaussian simulation with the same geostatistical parameters as
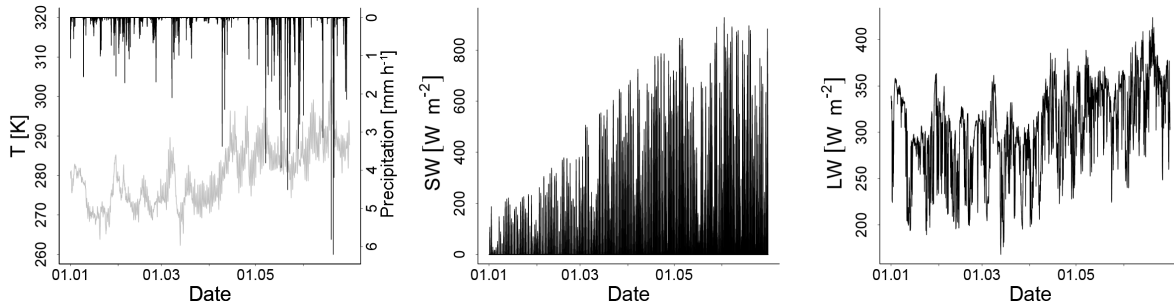
Figure 6. Hourly meteorological forcings for twin experiment from 1 January to 30 June 2013. Left panel shows 2 m air temperature and precipitation, middle panel shows incoming short-wave radiation and right panel shows incoming long-wave radiation.

for the reference field. Only the sill value was increased to $0.2 \log_{10}(\text{m}^2 \text{h}^{-2})$. The ensemble of meteorological forcings was generated by perturbing precipitation rates from the DWD reanalysis data with multiplicative noise sampled from a uniform distribution $U(0.5, 1.5)$. For each realisation, daily perturbation factors were sampled from the uniform distribution and the hourly precipitation values were multiplied with the corresponding perturbation factor. The daily perturbation factors were randomly sampled; i.e. no temporal correlation was considered.

First, the ensemble was used to perform an open-loop simulation (i.e. no observation data are assimilated) for the whole simulation period (January–June 2013). This simulation serves as a spin-up and benchmark for the following data assimilation run. Data assimilation was performed for the second half of the simulation period (April–June 2013) after the ensemble was spun-up for the first 3 months (January–March). Observation data from the reference run (soil moisture content) were assimilated on a daily basis for all 16 observation points. The measurement error for all observations was set to $0.02 \, \text{m}^3 \, \text{m}^{-3}$ and measurement errors were assumed to be spatially uncorrelated. The measurement data were used to jointly update the pressure and $\log_{10}(K_s)$ fields in ParFlow with an augmented state vector approach, resulting in 1.6 million unknowns for the data assimilation problem.

5.2 Scaling behaviour of TerrSysMP–PDAF

In order to check the computational efficiency of TerrSysMP–PDAF in a high-performance computational environment, we performed a weak scaling study on the supercomputer JUQUEEN located at Forschungszentrum Jülich (Germany). JUQUEEN consists of 28 672 IBM Blue-Gene/Q compute nodes with a total of 458 752 processors and 448 TB main memory. Each compute node consists of 17 cores (16 for computation, 1 for operating system services) running at 1.6 GHz and 16 GB main memory. The compute nodes allow for simultaneous multi-threading (SMT) up to a factor of 4, which means that up to 64 processes can run on one node. JUQUEEN uses a static memory mapping

to processors, so one processor can utilise a maximum of 1 GB main memory (256 MB in case of four-way SMT). The whole system reaches a Linpack performance of 5.0 Petaflops. More details on the system architecture of JUQUEEN can be found in Gasper et al. (2014).

In a weak scaling study, which is typically performed for such kinds of systems, the workload per processor is held constant and the problem size linearly increases with the number of processors. As we are not interested in the scaling properties of TerrSysMP itself, which have been described in detail by Gasper et al. (2014), we keep the number of processors for each model realisation constant and increase the number of model realisations along with the number of processors. For the scaling study, the synthetic model set-up described in Sect. 5.1 is used but only the first 20 assimilation time steps (1–20 April 2013) are calculated. For each model realisation 128 processors were used, which keeps the workload per processor constant. The partitioning of processors for one realisation was 96 for ParFlow and 32 for CLM, which was found to be the most optimal ratio for both models in terms of simulation time and computational efficiency. Furthermore, preliminary tests suggested that using 32 processors per node (two-way SMT) on JUQUEEN was the best compromise between execution time and memory requirements of ParFlow and CLM. For the weak scaling study, the number of realisations was increased from 8 to 256 and the corresponding number of processors ranged from 1024 to 32 768. Between each step of the scaling the number of realisations and processors was doubled (see Table 1 for information on all investigated scaling steps). The lowest number of realisations (processors) was set to 8 (1024) because this is the lowest possible job size on JUQUEEN given our chosen set-up (128 processors per model realisation, 32 ranks per compute node). The scaling behaviour can be assessed by calculating parallel efficiency E for weak scaling:

$$E(n_p) = \frac{T_{1024}}{T(n_p)}, \tag{1}$$

where $E(n_p)$ is the parallel efficiency of n_p processors, T_{1024} is the execution time with 1024 processors and $T(n_p)$ is the execution time with n_p processors. The timing informa-

No. processors	No. compute nodes*	No. realisations
1024	32	8
2048	64	16
4096	128	32
8192	256	64
16 384	512	128
32 768	1024	256

* A compute node on JUQUEEN consists of 16 + 1 physical cores but allows for simultaneous multi-threading up to a factor of 4. For the weak scaling study 32 ranks per compute node were used for simulations.

tion for the weak scaling study was acquired by instrumenting TerrSysMP–PDAF with the parallel performance tool Scalasca (Geimer et al., 2010) (version 2.2.1). Note that no special optimisation (such as critical path analysis) was performed to acquire the timing information with Scalasca.

A specific problem that occurs for assessing the parallel performance of ensemble methods like the EnKF in TerrSysMP–PDAF is that the simulation times for different ensemble members varies according to the assigned forcings and parameter sets. This, of course, can introduce some load balance issues because the filtering step introduces an effective barrier for the parallel computation. This implicit barrier causes the processors, for which the computation of the specific realisation is already finished for the current time step, to wait until the computation of the remaining model realisations is finished before they can proceed to the filtering step. This is typically not the case when parallel performance is measured for a deterministic model (as for example in Kollet et al., 2010; Gasper et al., 2014). In this case, the same model set-up is extended spatially for keeping a fixed workload per processor, meaning that the internal model processes during the calculation stay the same when the weak scaling behaviour is assessed by simultaneously increasing the domain size and the number of processors.

Therefore, the scalability of TerrSysMP–PDAF was first tested with a homogeneous ensemble where all ensemble members are identical to the reference run that was used to generate the observation data. This means that for all ensemble members the reference $\log_{10}(K_s)$ field and the deterministic (unperturbed) forcings were used. As a result, there is no variability in the ensemble for this set-up. Although this idealised set-up is not meaningful from a methodological perspective (as all ensemble members are identical) this will provide information about the scaling of TerrSysMP in a pure technical sense and helps to gain insight into the computational limits for performing data assimilation with TerrSysMP in a massively parallel environment. In a second step, the scaling was investigated for the heterogeneous en-

semble that is described in Sect. 5.1. For this set-up, also the load imbalance caused by variable forcing and parameter sets is taken into account in the scaling results. Note that results from this scaling set-up heavily depend on the chosen uncertainty description and the model dynamics of the chosen assimilation time period. Furthermore, the settings of the solver and the time stepping that is used to solve the transient variably saturated groundwater flow equations in ParFlow influence the scaling behaviour in this case. Therefore, results from this study are only meant to provide an example on how the scaling could look like for a typical application of TerrSysMP–PDAF for a coupled land surface–subsurface environment. In this study, no attempt was made to optimise the time stepping and solver settings of ParFlow in order to decrease load imbalance issues.

Aside from the above-mentioned effect of ensemble heterogeneities, another important issue that influences the parallel performance of data assimilation algorithms are the input/output (I/O) settings of the model code. Compared to a deterministic model run, the I/O operations multiply with the number of realisations in a data assimilation run. This can create a certain bottle neck for the code performance when large amounts of output data are written to disk simultaneously. Usually, in data assimilation applications the model output is restricted to the most important variables and mostly include the assimilated state variable and possibly other state variables or parameters that are jointly updated with measurement data or provide diagnostic information on the model performance. In some cases, detailed information on the distribution of certain variables is required, which means that output files from all ensemble members are needed for this variable. In other cases, knowledge on the statistics (e.g. mean and standard deviation) of a certain variable is sufficient. As ensemble output might be of importance for the parallel efficiency, we also compared three scenarios with a varying degree of model output:

– no model output;

– mean and standard deviation of simulated pressure and updated $\log_{10}(K_s)$ fields are calculated during model execution and are written to file by the filter communicator;

– output files for simulated pressure and updated $\log_{10}(K_s)$ fields are written for all ensemble members.

These three I/O scenarios were compared for the idealised test case (homogeneous ensemble) as well as the test case with a heterogeneous ensemble. This gives a total of six scaling scenarios. For each of these scenarios, the parallel efficiency was calculated with Eq. (1) separately meaning that the parallel efficiency is always normalised to the respective simulation with 1024 processors (T_{1024}).

Figure 7 shows the scaling behaviour and timing information for the ideal (homogeneous ensemble) and non-ideal
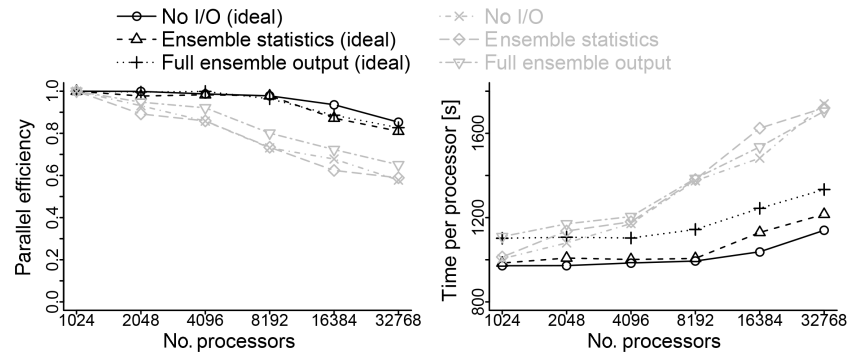
Figure 7. Scaling behaviour (left) and timing information (right) for TerrSysMP–PDAF for a weak scaling test on JUQUEEN. Black lines show results for an idealised test case (identical ensemble members) and grey lines show results for a heterogeneous ensemble. The number of ensemble members is increased from 8 to 256. Each ensemble member used 32 processors for CLM and 96 processors for ParFlow.

(heterogeneous ensemble) scaling runs. The parallel efficiency for the ideal test case stays very high (> 0.8) for all output scenarios within the investigated range of resources. The scenarios with model output show a slight reduction of parallel efficiency for a higher number of processors (> 8192). From the absolute timing information one can see that the scenario with full ensemble output requires systematically more time than the scenarios with no model output and statistical output only. The scenario with statistical output requires approximately the same simulation time as the no I/O scenario for a lower number of processors but then levels off for higher processor numbers (> 8192).

The parallel efficiency with the more realistic setting (heterogeneous ensemble) in Fig. 7 generally shows a stronger and faster decrease with increasing resource allocation. The differences to the ideal test case are of the order of 10–20 %, which is mainly caused by the load imbalance within the heterogeneous ensemble. Nevertheless, the parallel efficiency for the heterogeneous set-up is still around 0.6 for the largest tested processor allocation. From the timing information in Fig. 7 one can see that the scenario with full ensemble output requires more CPU time than the other scenarios for a lower number of processors. For a higher resource allocation, the differences between the I/O scenarios tend to vanish, which is a significant difference to the idealised test case. This behaviour is probably related to the load imbalance within the ensemble, which leads to a certain time delay in the writing of output files. On the contrary, in the idealised test case all ensemble members finish the model forward integration at approximately the same time meaning that all ensemble members tend to write output files synchronously. Note that for the heterogeneous case, an offline coupling between TerrSysMP and PDAF would also lead to a certain time overhead due to I/O operations because the writing and reading of restart files after the assimilation step would also occur simultaneously before the next model integration.

More detailed information on the timing of individual components of TerrSysMP–PDAF for the idealised scenario

can be found in Fig. 8. Here, the total execution time is categorised in four components: model initialisation, model integration, data assimilation and model shut down. These categories sum up to the total execution time and are normalised to the total number of processors that are used for the respective simulation. From the absolute values of the different model components it can be seen that by far most of the time is dedicated to the model integration. The computation time for model initialisation and the assimilation step have a similar order of magnitude but also exhibit some differences with respect to the I/O scenarios. The finalisation step only consumes a negligible part of the simulation time. In the initialisation phase, there is a significant increase of computation time for all three I/O scenarios when the number of processors exceeds 8192. This increase is related to the fact that many common input files are shared among the different realisations. The access to these shared input files on the storage system can be a bottleneck especially for a higher number of processors. Furthermore, the initialisation and set-up of the parallel communication through OASIS-MCT leads to a certain communication overhead when moving to a higher number of processors. During the model integration and the assimilation step, the scenario with the complete ensemble output again shows a significant deviation compared to the no I/O scenario. The reason is that in both phases the ensemble output is written to disk. At the end of the model integration the output files for the state variable (pressure) are written and at the end of the assimilation phase the same is done for the updated parameter fields (K_s). For the scenario with statistics output a slightly different pattern can be observed. The run times for this scenario are similar to the no I/O scenario for lower processor number (up to 8192) for both the model integration and the assimilation step. When the number of processors is further increased, the calculation of ensemble statistics leads to a certain overhead, which degrades the performance compared to the no I/O scenario. Also in this case the ensemble statistics are calculated and printed out at the end of the model integration (pressure fields) and at the
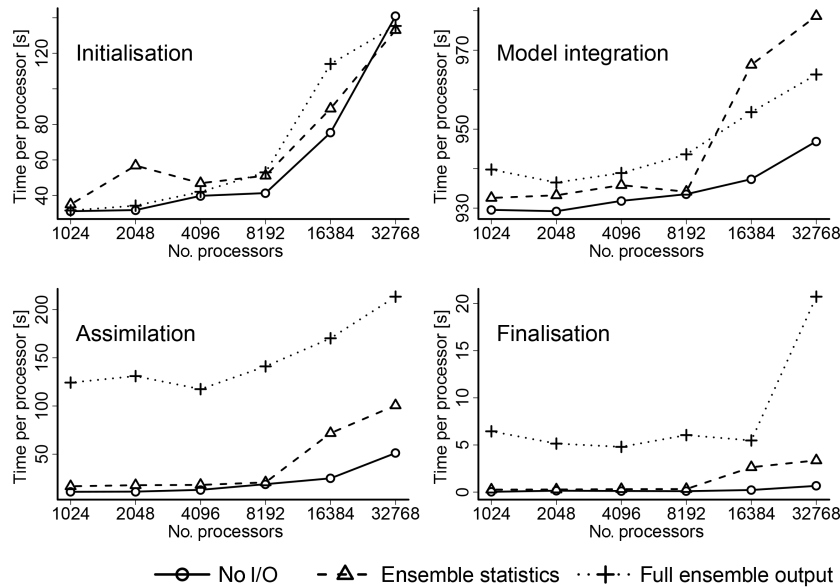
Figure 8. Timing information for individual components of TerrSysMP–PDAF for three I/O scenarios for the ideal test case.

end of the assimilation phase (updated K_s). In the finalisation phase there is also a certain offset for the scenario with full ensemble output but the required computation time is in general very low compared to the other parts of the program.

The scaling results for the idealised set-up are in general very good as the parallel efficiency stays above 0.8 even for a large number of processors. This is an indication that the coupling between TerrSysMP and PDAF is working very well in a technical sense. Furthermore, the results show that the filter algorithms implemented in PDAF scale well to an even higher number of processors than reported before in Nerger and Hiller (2013). The bottleneck of the parallel performance is mainly the initialisation phase (reading operations and set-up of OASIS-MCT communication) and the output operations. Here, parallel I/O concepts could help to further improve the parallel performance of TerrSysMP–PDAF. The scaling results for the more realistic heterogeneous ensemble are also promising for the application of TerrSysMP–PDAF for more complex land surface–subsurface data assimilation problems. Generally, for any given model set-up, the scaling behaviour of the data assimilation problem will particularly depend on the numerical robustness of the deterministic forward model towards ensemble perturbations. Critical situations with respect to convergence could occur, e.g. for strong heterogeneities in the subsurface parameterisation (e.g. hydraulic conductivities) or for the coupling of overland and subsurface flow. For the latter case, especially the computationally demanding onset and offset of overland flow at particular grid cells (e.g. due to heavy rainfall or recession events) can have a negative influence on the scaling behaviour of the deterministic forward model (Kollet and Maxwell, 2006; Osei-Kuffuor et al.,

2014). If only a subset of realisations is affected by such convergence problems, also the scalability of the ensemble propagation might be influenced negatively. Therefore, it is important to configure the deterministic forward model well with respect to numerical stability and execution time. This can be achieved, for example, through the correct choice of solver parameters, an adequate spatio-temporal discretisation of the problem and a proper choice of model parameters and ensemble perturbations.

5.3 Data assimilation results

Figure 9 shows time series of simulated soil water content at four verification points (see Fig. 5) along the south–north direction for both the open loop (upper row) and the assimilation run (lower row). Results for the open-loop simulations already show that the temporal dynamics of the reference run are well represented by the ensemble and that the changes in soil moisture very much depend on the dynamics of the meteorological forcing data. Assimilation of soil moisture data leads to a reduction of the ensemble spread and a reduction of the mismatch between the ensemble mean of forecasted soil moisture and the reference values. Additionally the absolute average error (AAE) of soil moisture content θ is used to assess the model performance of the open loop and the assimilation run:

$$\text{AAE}_\theta = \frac{1}{N_t} \sum_i^{N_t} |\bar{\theta}_i^{\text{sim}} - \theta_i^{\text{ref}}|, \tag{2}$$

where $\bar{\theta}^{\text{sim}}$ is the ensemble mean of simulated soil moisture, θ^{ref} is the reference soil moisture content and N_t is the number of assimilation time steps. AAE_θ averaged over
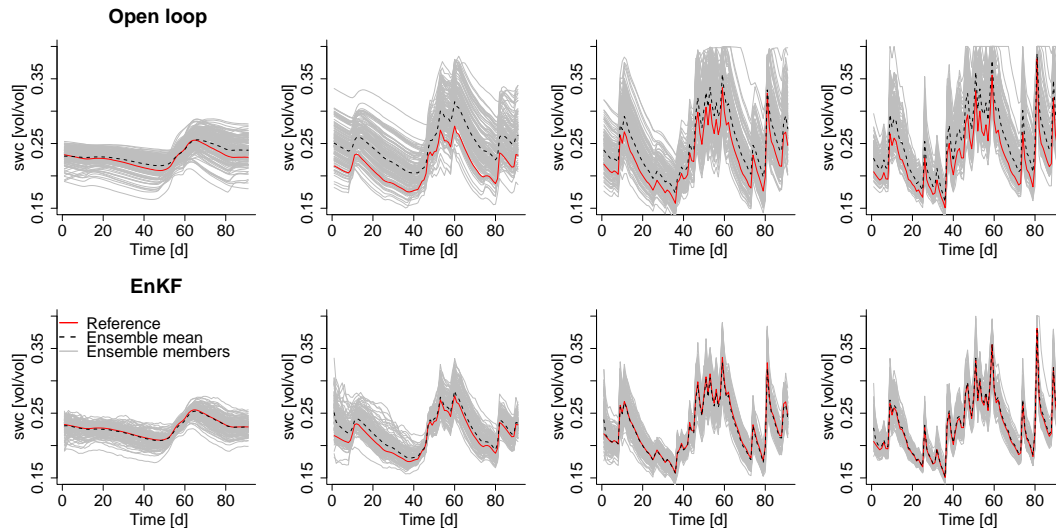
Figure 9. Simulated soil water content at the four verification nodes in Fig. 5 (from north to south) for April–June 2013 (91 days). Upper row shows results for open-loop simulations and lower row for assimilation.

the uppermost 10 model layers reduced from $0.0135 \, \mathrm{m^3 \, m^{-3}}$ (open-loop simulation) to $0.0096 \, \mathrm{m^3 \, m^{-3}}$ (assimilation experiment) through the assimilation of soil moisture data (the 10 lower model layers were excluded from this calculation because they are constantly saturated during the whole simulation period). In Fig. 10 AAE_θ is shown for one specific model layer at $-65 \, \mathrm{cm}$ depth. It can be seen from Fig. 10 that AAE_θ is reduced in large parts of the model domain, which means that data assimilation was not only effective at the observation locations but also significantly improved the model performance in the surrounding grid cells. Several spots in the model domain, e.g. at the southern boundary and in the north-east, show less improvement, which may be related to the assigned boundary conditions and the fact that the distance to observation points is larger at the model boundaries reducing the correlation between observation points and those grid cells. The effect of soil moisture assimilation on land surface fluxes (latent and sensible heat flux) was also analysed for this set-up. The total AAE values (averaged over all grid cells and time steps) in the open-loop run were $1.003 \, \mathrm{W \, m^{-2}}$ for sensible heat flux and $1.212 \, \mathrm{W \, m^{-2}}$ for latent heat flux. The spatial pattern of errors in land surface fluxes is closely related to those of AAE_θ in Fig. 10. In principal, the calculation of land surface fluxes within TerrSysMP can be affected by (1) plant physiological parameters, (2) meteorological forcings that affect stomatal conductance and (3) the availability of water in the subsurface. In the chosen set-up neither plant physiological parameters nor meteorological forcings (with the exception of precipitation) were perturbed, so the variability of land surface fluxes is mainly influenced by the availability of water in the rooting zone. The relatively low errors in the open-loop simulation indicate that the variability of soil moisture content

only had a limited effect on land surface fluxes in the chosen set-up and that most of the model domain is not affected by water limitation. With the assimilation of soil moisture contents, the total AAE values of sensible (latent) heat fluxes were reduced to 0.730 (0.876) $\mathrm{W \, m^{-2}}$, which is a relative improvement of about 27 %. Nevertheless, the absolute magnitude of land surface flux errors and the improvements by data assimilation are relatively low due to the fact that the system was not affected by water limitation throughout the simulation period.

In the presented data assimilation experiment, soil moisture data from the reference run are also used to simultaneously update the $\log_{10}(K_\mathrm{s})$ fields of the ensemble. In Fig. 11 the reference field of $\log_{10}(K_\mathrm{s})$ is compared with the average $\log_{10}(K_\mathrm{s})$ field of the initial ensemble and the average $\log_{10}(K_\mathrm{s})$ field after the assimilation period. It becomes obvious that the correction of $\log_{10}(K_\mathrm{s})$ values through the assimilation of soil moisture observations leads to a significant improvement of the estimated average $\log_{10}(K_\mathrm{s})$ field. Compared to the initial estimate of $\log_{10}(K_\mathrm{s})$, the updated average $\log_{10}(K_\mathrm{s})$ field includes the main structural features of the reference field, e.g. the higher $\log_{10}(K_\mathrm{s})$ values in the eastern part and the lower values in the western part. Again, as for AAE_θ, the improvement is less pronounced at the model boundaries especially in the southern part. This can again be related to the lower observation density at the model borders.

5.4 Applicability at hyper-resolution

The problem size of the TerrSysMP model used for the scaling study and the verification example in the previous two subsections is very close to typical state-of-the-art applications of integrated land surface–subsurface models at the catchment scale. However, integrated modelling is also con-
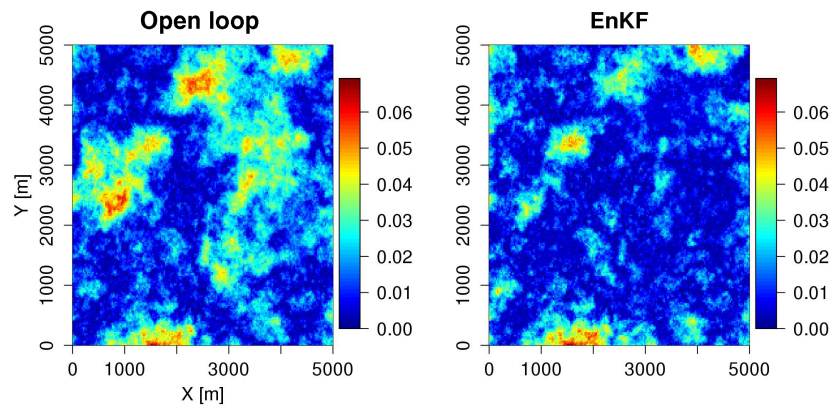
Figure 10. Absolute average error of soil water content AAE$_\theta$ for open loop (left) and assimilation (right) at a depth of $-65\,\mathrm{cm}$ from April to June 2013.
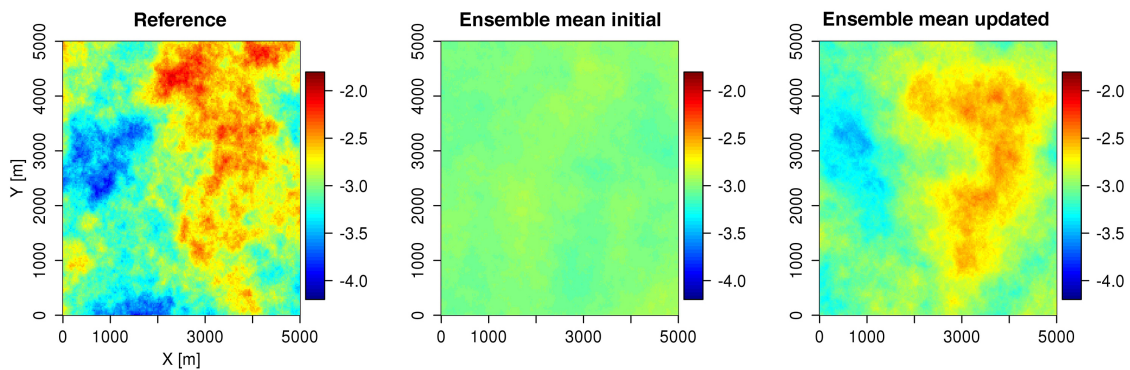
Figure 11. Log-transformed saturated hydraulic conductivity fields of the reference (left), the initial ensemble mean (middle) and the updated ensemble mean at the end of the assimilation period (right).

tinuously moving forward towards higher model resolutions (e.g. Maxwell et al., 2015), which was identified as one of the forthcoming challenges in Earth system modelling (e.g. Wood et al., 2011; Bierkens et al., 2014). Therefore, it was also tested whether the TerrSysMP–PDAF data assimilation framework is applicable for models with a much bigger problem size.

For this purpose, the problem size of the forward model was increased by a factor of 25 by increasing the horizontal model resolution to 5 m (1000×1000 grid cells) leading to 20 million grid cells for the subsurface part of TerrSysMP.

The model input for the synthetic reference and the ensemble was re-gridded to this higher model resolution. The $\log_{10}(K_\mathrm{s})$ fields for the synthetic reference and the individual ensemble members were additionally perturbed with small-scale noise, which was introduced to resemble a certain sub-scale variability within the original 25 m grid cells. The small-scale perturbation fields were generated with the parallel Gaussian simulation algorithm implemented in ParFlow with a horizontal correlation length of 20 m and a standard deviation of 0.2 log units. The reference $\log_{10}(K_\mathrm{s})$ fields for

the 25 and 5 m resolution models are shown in Figs. 11 and 12, respectively. The set-up for the data assimilation experiment for the high-resolution model was identical to the 25 m resolution case, i.e. 90 days of model spin-up and daily assimilation of 16 soil moisture observations for 91 days with a joint state–parameter estimation.

The simulations for the 5 m resolution model were run using four complete racks (65 536 physical cores with two-way SMT) on JUQUEEN to solve the data assimilation problem for 40 million unknowns. The simulation time for the assimilation period (91 days) with this configuration was about 4.5 h. Figure 12 shows the initial and updated ensemble mean of $\log_{10}(K_\mathrm{s})$ for the 5 m resolution model. As for the 25 m resolution model, the main features of the reference field, e.g. the high conductivity parts in the eastern part of the model domain, were retrieved through the update of $\log_{10}(K_\mathrm{s})$ values with soil moisture data. However, the updated $\log_{10}(K_\mathrm{s})$ patterns do not match exactly, which can be explained by the different support range of observations for the two model resolutions and the additional sub-scale variability added in the 5 m resolution model.
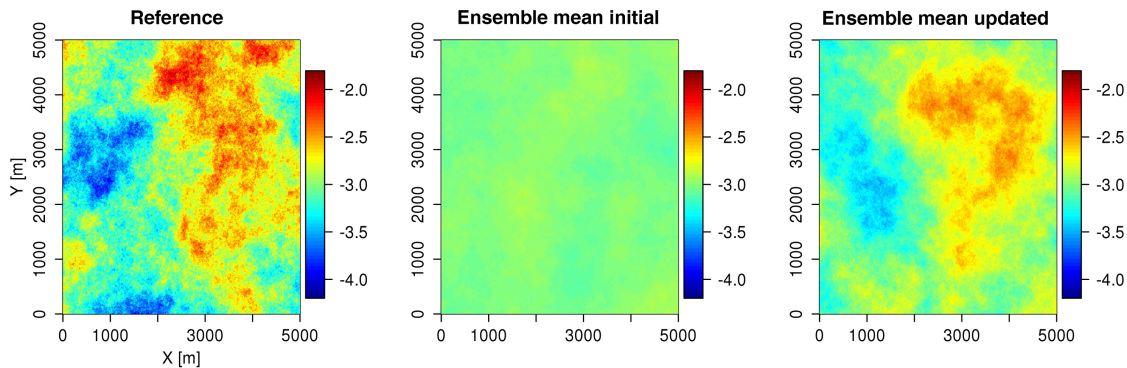
Figure 12. Log-transformed saturated hydraulic conductivity fields of the reference (left), the initial ensemble mean (middle) and the updated ensemble mean at the end of the assimilation period (right) for the 5 m resolution model.

Of course, the model set-up that was used here is relatively simple in terms of model dynamics compared to typical real-world applications of integrated Earth system models. Topography, heterogeneous land surface parameters and spatially distributed meteorological forcings usually lead to a much more complex model behaviour, which also leads to far longer simulation times compared to the model set-up used in this study. This will make data assimilation with high-resolution integrated models for real-world applications very challenging with respect to the amount of necessary computational resources. Nevertheless, these simulations with a relatively simple high-resolution model set-up show that the TerrSysMP–PDAF framework is technically able to cope with data assimilation problems where the problem size of the forward model is in the range of tens of millions grid cells. Such problem sizes will become more common especially in the context of integrated hydrological modelling on the catchment scale (e.g. to better resolve small-scale variabilities in hydraulic parameters) as well as for large-scale applications in order to improve hydrological and meteorological forecasts on the basin and the continental scale.

6 Conclusions and outlook

In this paper, we presented a modular high-performance data assimilation framework for the land surface–subsurface part of the integrated terrestrial system modelling platform TerrSysMP. In TerrSysMP, land surface processes are modelled with CLM 3.5 and subsurface processes with ParFlow where both models are coupled via the exchange of states and fluxes with the coupling software OASIS-MCT. The data assimilation system for this model was established with the Parallel Data Assimilation Framework (PDAF), which provides a suite of efficient and scalable data assimilation algorithms. The coupling between TerrSysMP and PDAF is done in a fully integrated fashion meaning that the model ensemble as well as the infrastructure for data assimilation is only

initialised once and the data assimilation system is continuously integrated forward in time without the need of system calls to the model or re-initialisation of any of the system components. The data exchange between TerrSysMP and PDAF is done completely via main memory, which avoids the need for a frequent reading and writing of model restart files. TerrSysMP as well as PDAF are fully parallelised and the data exchange between the two components makes effective use of the domain decomposition in the models. This significantly reduces the memory requirements of the system because the global state(-parameter) vector does not need to be stored completely in any part of the filter algorithm. In addition to the parallelism in the model integration (provided by the component models of TerrSysMP) and in the filtering step (provided by PDAF) also the ensemble propagation is running fully parallel. The data assimilation system for TerrSysMP is designed in a modular fashion; i.e. assimilation can either run with the coupled land surface–subsurface model (ParFlow + CLM coupled via OASIS-MCT) or with one of the stand alone models (ParFlow or CLM). This provides the user with some flexibility regarding the model choice because for certain modelling purposes the use of a single compartment model (subsurface or land surface) may be sufficient in the context of data assimilation, whereas in other situations a fully coupled approach may be more adequate. Currently, pressure and soil moisture data can be assimilated in ParFlow. These data are used in ParFlow for a state update of the 3-D pressure field but they can also be used for a joint update of saturated hydraulic conductivities or Mannings coefficients. If the assimilation system is only running with CLM, soil moisture data can be assimilated directly into CLM.

In this study we also provide a scaling study on the massively parallel supercomputing environment JUQUEEN, which shows that the assimilation system runs efficiently and scales well even for a large number of processors (32 768). These results are promising for the application of the data assimilation system for large-scale applications or high-

resolution models, which require a huge amount of computational resources and therefore also benefit from an efficient implementation of the ensemble propagation and the filtering step. Additional tests with a high-resolution model set-up where 20 million states and 20 million parameters were updated simultaneously (as compared to 0.8 million states and 0.8 million parameters in the scaling study) revealed that the infrastructure of the proposed TerrSysMP–PDAF framework is well suited for such large problem sizes. Results from the scaling study also showed that the output strategy (ensemble output vs. statistical output) as well as load balancing issues between the different ensemble members can have a certain influence on the parallel efficiency, which should be carefully taken into consideration when data assimilation is performed with a large amount of computational resources.

In further work we plan to include also the atmospheric compartment model of TerrSysMP (COSMO-DE) in the assimilation system. This will allow us to investigate the effect of data assimilation in a fully coupled system from the subsurface to the atmosphere. It is also planned to extend the data assimilation system to make full use of the functionality of PDAF with respect to filter variants and assimilation options (e.g. localisation and smoothing). Furthermore, the data assimilation system will be extended with additional measurement operators for soil moisture assimilation including measurement operators for active and passive radar remote sensing data and cosmic ray sensors.

Code availability

The source code of TerrSysMP–PDAF is added as a supplement to this article.

The Supplement related to this article is available online at doi:10.5194/gmd-9-1341-2016-supplement.

References

Anderson, J. L.: An ensemble adjustment Kalman filter for data assimilation, Mon. Weather Rev., 129, 2884–2903, doi:10.1175/1520-0493(2001)129<2884:AEAKFF>2.0.CO;2, 2001.

Anderson, J., Hoar, T., Raeder, K., Liu, H., Collins, N., Torn, R., and Avellano, A.: The data assimilation research testbed: a community facility, B. Am. Meteorol. Soc., 90, 1283–1296, doi:10.1175/2009bams2618.1, 2009.

Andreadis, K. M. and Lettenmaier, D. P.: Assimilating remotely sensed snow observations into a macroscale hydrology model, Adv. Water Resour., 29, 872–886, doi:10.1016/j.advwatres.2005.08.004, 2006.

Ashby, S. and Falgout, R.: A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations, Nucl. Sci. Eng., 124, 145–159, 1996.

Bailey, R. T. and Baù, D.: Estimating geostatistical parameters and spatially-variable hydraulic conductivity within a catchment system using an ensemble smoother, Hydrol. Earth Syst. Sci., 16, 287–304, doi:10.5194/hess-16-287-2012, 2012.

Baldauf, M., Seifert, A., Förstner, J., Majewski, D., Raschendorfer, M., and Reinhardt, T.: Operational convective-scale numerical weather prediction with the COSMO model: description and sensitivities, Mon. Weather Rev., 139, 3887–3905, doi:10.1175/mwr-d-10-05013.1, 2011.

Barbu, A. L., Calvet, J.-C., Mahfouf, J.-F., and Lafont, S.: Integrating ASCAT surface soil moisture and GEOV1 leaf area index into the SURFEX modelling platform: a land data assimilation application over France, Hydrol. Earth Syst. Sci., 18, 173–192, doi:10.5194/hess-18-173-2014, 2014.

Barker, D., Huang, X.-Y., Liu, Z., Auligné, T., Zhang, X., Rugg, S., Ajjaji, R., Bourgeois, A., Bray, J., Chen, Y., Demirtas, M., Guo, Y.-R., Henderson, T., Huang, W., Lin, H.-C., Michalakes, J., Rizvi, S., and Zhang, X.: The Weather Research and Forecasting Model's Community Variational/Ensemble Data Assimilation System: WRFDA, B. Am. Meteorol. Soc., 93, 831–843, doi:10.1175/bams-d-11-00167.1, 2012.

Bateni, S. and Entekhabi, D.: Surface heat flux estimation with the ensemble Kalman smoother: joint estimation of state and parameters, Water Resour. Res., 48, W08521, doi:10.1029/2011WR011542, 2012.

Bierkens, M. F. P., Bell, V. A., Burek, P., Chaney, N., Condon, L. E., David, C. H., de Roo, A., Döll, P., Drost, N., Famiglietti, J. S., Flörke, M., Gochis, D. J., Houser, P., Hut, R., Ke-

une, J., Kollet, S., Maxwell, R. M., Reager, J. T., Samaniego, L., Sudicky, E., Sutanudjaja, E. H., van de Giesen, N., Winsemius, H., and Wood, E. F.: Hyper-resolution global hydrological modelling: what is next?, Hydrol. Process., 29, 310–320, doi:10.1002/hyp.10391, 2014.

Bishop, C., Etherton, B., and Majumdar, S.: Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects, Mon. Weather Rev., 129, 420–436, doi:10.1175/1520-0493(2001)129<0420:ASWTET>2.0.CO;2, 2001.

Burgers, G., van Leeuwen, P. J., and Evensen, G.: Analysis scheme in the ensemble Kalman filter, Mon. Weather Rev., 126, 1719–1724, doi:10.1175/1520-0493(1998)126<1719:ASITEK>2.0.CO;2, 1998.

Butts, M., Drews, M., Larsen, M. A., Lerer, S., Rasmussen, S. H., Grooss, J., Overgaard, J., Refsgaard, J. C., Christensen, O. B., and Christensen, J. H.: Embedding complex hydrology in the regional climate system – Dynamic coupling across different modelling domains, Adv. Water Resour., 74, 166–184, doi:10.1016/j.advwatres.2014.09.004, 2014.

Camporese, M., Paniconi, C., Putti, M., and Salandin, P.: Ensemble Kalman filter data assimilation for a process-based catchment scale model of surface and subsurface flow, Water Resour. Res., 45, W10421, doi:10.1029/2008WR007031, 2009.

Carrera, M. L., Bélair, S., and Bilodeau, B.: The Canadian Land Data Assimilation System (CaLDAS): Description and Synthetic Evaluation Study, J. Hydrometeor., 16, 1293–1314, doi:10.1175/jhm-d-14-0089.1, 2015.

Chen, Y. and Zhang, D.: Data assimilation for transient flow in geologic formations via ensemble Kalman filter, Adv. Water Resour., 29, 1107–1122, doi:10.1016/j.advwatres.2005.09.007, 2006.

Crow, W. T. and Wood, E. F.: The assimilation of remotely sensed soil brightness temperature imagery into a land surface model using Ensemble Kalman filtering: a case study based on ESTAR measurements during SGP97, Adv. Water Resour., 26, 137–149, doi:10.1016/s0309-1708(02)00088-x, 2003.

Davin, E. L., Stöckli, R., Jaeger, E. B., Levis, S., and Seneviratne, S. I.: COSMO-CLM2: a new version of the COSMO-CLM model coupled to the Community Land Model, Clim. Dynam., 37, 1889–1907, doi:10.1007/s00382-011-1019-z, 2011.

De Lannoy, G. J. M., Reichle, R. H., Houser, P. R., Pauwels, V. R. N., and Verhoest, N. E. C.: Correcting for forecast bias in soil moisture assimilation with the ensemble Kalman filter, Water Resour. Res., 43, W09410, doi:10.1029/2006wr005449, 2007.

Erdal, D., Neuweiler, I., and Wollschläger, U.: Using a bias aware EnKF to account for unresolved structure in an unsaturated zone model, Water Resour. Res., 50, 132–147, doi:10.1002/2012WR013443, 2014.

Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte-Carlo methods to forecast error statistics, J. Geophys. Res.-Oceans, 99, 10143–10162, doi:10.1029/94JC00572, 1994.

Ferguson, I. M. and Maxwell, R. M.: Role of groundwater in watershed response and land surface feedbacks under climate change, Water Resour. Res., 46, W00F02, doi:10.1029/2009wr008616, 2010.

Ford, T. W. and Quiring, S. M.: Influence of MODIS-derived dynamic vegetation on VIC-simulated soil moisture in Okla-

homa, J. Hydrometeorol., 14, 1910–1921, doi:10.1175/jhm-d-13-037.1, 2013.

Gasper, F., Goergen, K., Shrestha, P., Sulis, M., Rihani, J., Geimer, M., and Kollet, S.: Implementation and scaling of the fully coupled Terrestrial Systems Modeling Platform (TerrSysMP v1.0) in a massively parallel supercomputing environment – a case study on JUQUEEN (IBM Blue Gene/Q), Geosci. Model Dev., 7, 2531–2543, doi:10.5194/gmd-7-2531-2014, 2014.

Geimer, M., Wolf, F., Wylie, B. J. N., Ábrahám, E., Becker, D., and Mohr, B.: The Scalasca performance toolset architecture, Conc. Comput.: Practice and Experience, 22, 702–719, doi:10.1002/cpe.1556, 2010.

Gharamti, M., Kadoura, A., Valstar, J., Sun, S., and Hoteit, I.: Constraining a compositional flow model with flow-chemical data using an ensemble-based Kalman filter, Water Resour. Res., 50, 2444–2467, doi:10.1002/2013WR014830, 2014.

Ghent, D., Kaduk, J., Remedios, J., Ardö, J., and Balzter, H.: Assimilation of land surface temperature into the land surface model JULES with an ensemble Kalman filter, J. Geophys. Res., 115, D19112, doi:10.1029/2010jd014392, 2010.

Han, X., Hendricks Franssen, H.-J., Li, X., Zhang, Y., Montzka, C., and Vereecken, H.: Joint assimilation of surface temperature and L-band microwave brightness temperature in land data assimilation, Vadose Zone J., 12, 3, doi:10.2136/vzj2012.0072, 2013.

Han, X., Hendricks Franssen, H.-J., Montzka, C., and Vereecken, H.: Soil moisture and soil properties estimation in the Community Land Model with synthetic brightness temperature observations, Water Resour. Res., 50, 6081–6105, doi:10.1002/2013wr014586, 2014.

Hendricks Franssen, H. J. and Kinzelbach, W.: Real-time groundwater flow modeling with the ensemble Kalman filter: joint estimation of states and parameters and the filter inbreeding problem, Water Resour. Res., 44, W09408, doi:10.1029/2007WR006505, 2008.

Hunt, B. R., Kostelich, E. J., and Szunyogh, I.: Efficient data assimilation for spatiotemporal chaos: a local ensemble transform Kalman filter, Physica D, 230, 112–126, doi:10.1016/j.physd.2006.11.008, 2007.

Jones, J. E. and Woodward, C. S.: Newton–Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems, Adv. Water Resour., 24, 763–774, doi:10.1016/s0309-1708(00)00075-0, 2001.

Kollet, S. J. and Maxwell, R. M.: Integrated surface–groundwater flow modeling: a free-surface overland flow boundary condition in a parallel groundwater flow model, Adv. Water Resour., 29, 945–958, doi:10.1016/j.advwatres.2005.08.006, 2006.

Kollet, S. J. and Maxwell, R. M.: Capturing the influence of groundwater dynamics on land surface processes using an integrated, distributed watershed model, Water Resour. Res., 44, W02402, doi:10.1029/2007wr006004, 2008.

Kollet, S. J., Maxwell, R. M., Woodward, C. S., Smith, S., Vanderborght, J., Vereecken, H., and Simmer, C.: Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources, Water Resour. Res., 46, W04201, doi:10.1029/2009wr008730, 2010.

Kumar, P. and Kaleita, A. L.: Assimilation of near-surface temperature using extended Kalman filter, Adv. Water Resour., 26, 79–93, doi:10.1016/s0309-1708(02)00098-2, 2003.

Kurtz, W., Hendricks Franssen, H.-J., Brunner, P., and Vereecken, H.: Is high-resolution inverse characterization of heterogeneous river bed hydraulic conductivities needed and possible?, Hydrol. Earth Syst. Sci., 17, 3795–3813, doi:10.5194/hess-17-3795-2013, 2013.

Kurtz, W., Hendricks Franssen, H.-J., Kaiser, H.-P., and Vereecken, H.: Joint assimilation of piezometric heads and groundwater temperatures for improved modeling of river-aquifer interactions, Water Resour. Res., 50, 1665–1688, doi:10.1002/2013wr014823, 2014.

Li, C. and Ren, L.: Estimation of unsaturated soil hydraulic parameters using the ensemble Kalman filter, Vadose Zone J., 10, 1205–1227, doi:10.2136/vzj2010.0159, 2011.

Li, L., Zhou, H., Gómez-Hernández, J. J., and Hendricks Franssen, H. J.: Jointly mapping hydraulic conductivity and porosity by assimilating concentration data via ensemble Kalman filter, J. Hydrol., 428–429, 152–169, doi:10.1016/j.jhydrol.2012.01.037, 2012.

Li, N., McLaughlin, D., Kinzelbach, W., Li, W., and Dong, X.: Using an ensemble smoother to evaluate parameter uncertainty of an integrated hydrological model of Yanqi basin, J. Hydrol., 529, 146–158, doi:10.1016/j.jhydrol.2015.07.024, 2015.

Liu, G., Chen, Y., and Zhang, D.: Investigation of flow and transport processes at the MADE site using ensemble Kalman filter, Adv. Water Resour., 31, 975–986, doi:10.1016/j.advwatres.2008.03.006, 2008.

Maxwell, R. M.: A terrain-following grid transform and preconditioner for parallel, large-scale, integrated hydrologic modeling, Adv. Water Resour., 53, 109–117, doi:10.1016/j.advwatres.2012.10.001, 2013.

Maxwell, R. M. and Miller, N. L.: Development of a coupled land surface and groundwater model, J. Hydrometeorol., 6, 233–247, doi:10.1175/jhm422.1, 2005.

Maxwell, R. M., Lundquist, J. K., Mirocha, J. D., Smith, S. G., Woodward, C. S., and Tompson, A. F. B.: Development of a coupled groundwater–atmosphere model, Mon. Weather Rev., 139, 96–116, doi:10.1175/2010mwr3392.1, 2011.

Maxwell, R. M., Condon, L. E., and Kollet, S. J.: A high-resolution simulation of groundwater and surface water over most of the continental US with the integrated hydrologic model ParFlow v3, Geosci. Model Dev., 8, 923–937, doi:10.5194/gmd-8-923-2015, 2015.

Montzka, C., Moradkhani, H., Weihermüller, L., Franssen, H.-J. H., Canty, M., and Vereecken, H.: Hydraulic parameter estimation by remotely-sensed top soil moisture observations with the particle filter, J. Hydrol., 399, 410–421, doi:10.1016/j.jhydrol.2011.01.020, 2011.

Nerger, L. and Hiller, W.: Software for ensemble-based data assimilation systems – implementation strategies and scalability, Comput. Geosci., 55, 110–118, doi:10.1016/j.cageo.2012.03.026, 2013.

Nowak, W.: Best unbiased ensemble linearization and the quasi-linear Kalman ensemble generator, Water Resour. Res., 45, W04431, doi:10.1029/2008WR007328, 2009.

Oleson, K. W., Dai, Y., Bonan, G. B., Bosilovichm, R., Dickinson, R. E., Dirmeyer, P., Hoffman, F., Houser, P., Levis, S., Niu, G.-Y., Thornton, P. E., Vertenstein, M., Yang, Z.-L., and Zeng, X.: Technical Description of the Community Land Model (CLM), NCAR Technical Note NCAR/TN-461+STR, National Center for Atmospheric Research, Boulder, Colorado, doi:10.5065/D6N877R0, 2004.

Oleson, K. W., Niu, G.-Y., Yang, Z.-L., Lawrence, D. M., Thornton, P. E., Lawrence, P. J., Stöckli, R., Dickinson, R. E., Bonan, G. B., Levis, S., Dai, A., and Qian, T.: Improvements to the Community Land Model and their impact on the hydrological cycle, J. Geophys. Res.-Biogeo., 113, G01025, doi:10.1029/2007jg000563, 2008.

OpenDA: The OpenDA data-assimilation toolbox, available at: http://www.openda.org (last access: 10 September 2015), 2013.

Osei-Kuffuor, D., Maxwell, R. M., and Woodward, C. S.: Improved numerical solvers for implicit coupling of subsurface and overland flow, Adv. Water Resour., 74, 185–195, doi:10.1016/j.advwatres.2014.09.006, 2014.

Pasetto, D., Niu, G.-Y., Pangle, L., Paniconi, C., Putti, M., and Troch, P.: Impact of sensor failure on the observability of flow dynamics at the Biosphere 2 LEO hillslopes, Adv. Water Resour., 86, 327–339, doi:10.1016/j.advwatres.2015.04.014, 2015.

Pebesma, E. J.: Multivariable geostatistics in S: the gstat package, Comput. Geosci., 30, 683–691, 2004.

Rahman, M., Sulis, M., and Kollet, S.: The subsurface-land surface-atmosphere connection under convective conditions, Adv. Water Resour., 83, 240–249, doi:10.1016/j.advwatres.2015.06.003, 2015.

Rasmussen, J., Madsen, H., Jensen, K. H., and Refsgaard, J. C.: Data assimilation in integrated hydrological modeling using ensemble Kalman filtering: evaluating the effect of ensemble size and localization on filter performance, Hydrol. Earth Syst. Sci., 19, 2999–3013, doi:10.5194/hess-19-2999-2015, 2015.

R Core Team: R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, available at: http://www.R-project.org/, last access: 10 September 2015.

Reichle, R. H., Kumar, S. V., Mahanama, S. P. P., Koster, R. D., and Liu, Q.: Assimilation of satellite-derived skin temperature observations into land surface models, J. Hydrometeorol., 11, 1103–1122, doi:10.1175/2010jhm1262.1, 2010.

Ridler, M. E., van Velzen, N., Hummel, S., Sandholt, I., Falk, A. K., Heemink, A., and Madsen, H.: Data assimilation framework: linking an open data assimilation library (OpenDA) to a widely adopted model interface (OpenMI), Environ. Modell. Softw., 57, 76–89, doi:10.1016/j.envsoft.2014.02.008, 2014.

Rihani, J. F., Maxwell, R. M., and Chow, F. K.: Coupling groundwater and land surface processes: idealized simulations to identify effects of terrain and subsurface heterogeneity on land surface energy fluxes, Water Resour. Res., 46, W12523, doi:10.1029/2010wr009111, 2010.

Rodell, M., Houser, P. R., Jambor, U., Gottschalck, J., Mitchell, K., Meng, C.-J., Arsenault, K., Cosgrove, B., Radakovich, J., Bosilovich, M., Entin, J. K., Walker, J. P., Lohmann, D., and Toll, D.: The Global Land Data Assimilation System, B. Am. Meteorol. Soc., 85, 381–394, doi:10.1175/bams-85-3-381, 2004.

Sabater, J. M. n., Rüdiger, C., Calvet, J.-C., Fritz, N., Jarlan, L., and Kerr, Y.: Joint assimilation of surface soil moisture and LAI observations into a land surface model, Agr. Forest Meteorol., 148, 1362–1373, doi:10.1016/j.agrformet.2008.04.003, 2008.

Shi, L., Song, X., Tong, J., Zhu, Y., and Zhang, Q.: Impacts of different types of measurements on estimating

unsaturated flow parameters, J. Hydrol., 524, 549–561, doi:10.1016/j.jhydrol.2015.01.078, 2015.

Shi, Y., Davis, K. J., Zhang, F., Duffy, C. J., and Yu, X.: Parameter estimation of a physically based land surface hydrologic model using the ensemble Kalman filter: a synthetic experiment, Water Resour. Res., 50, 706–724, doi:10.1002/2013wr014070, 2014.

Shi, Y., Davis, K. J., Zhang, F., Duffy, C. J., and Yu, X.: Parameter estimation of a physically-based land surface hydrologic model using an ensemble Kalman filter: a multivariate real-data experiment, Adv. Water Resour., 83, 421–427, doi:10.1016/j.advwatres.2015.06.009, 2015.

Shrestha, P., Sulis, M., Masbou, M., Kollet, S., and Simmer, C.: A scale-consistent terrestrial systems modeling platform based on COSMO, CLM, and ParFlow, Mon. Weather Rev., 142, 3466–3483, doi:10.1175/mwr-d-14-00029.1, 2014.

Song, X., Shi, L., Ye, M., Yang, J., and Navon, I. M.: Numerical comparison of iterative ensemble Kalman filters for unsaturated flow inverse modeling, Vadose Zone J., 13, 2, doi:10.2136/vzj2013.05.0083, 2014.

Su, H., Yang, Z.-L., Dickinson, R. E., Wilson, C. R., and Niu, G.-Y.: Multisensor snow data assimilation at the continental scale: the value of gravity recovery and climate experiment terrestrial water storage information, J. Geophys. Res., 115, D10104, doi:10.1029/2009jd013035, 2010.

Tian, W., Li, X., Cheng, G.-D., Wang, X.-S., and Hu, B. X.: Coupling a groundwater model with a land surface model to improve water and energy cycle simulation, Hydrol. Earth Syst. Sci., 16, 4707–4723, doi:10.5194/hess-16-4707-2012, 2012.

Tippett, M., Anderson, J., Bishop, C., Hamill, T., and Whitaker, J.: Ensemble square root filters, Mon. Weather Rev., 131, 1485–1490, doi:10.1175/1520-0493(2003)131<1485:ESRF>2.0.CO;2, 2003.

Valcke, S.: The OASIS3 coupler: a European climate modelling community software, Geosci. Model Dev., 6, 373–388, doi:10.5194/gmd-6-373-2013, 2013.

Valcke, S., Craig, T., and Coquart, L.: OASIS3-MCT User Guide: OASIS3-MCT 2.0, Tech. Rep. TR/CMGC/13/17, CERFACS/CNRS SUC URA No 1875, CERFACS (Centre Européen de Recherche et Formation Avancée en Calcul Scientifique), Toulouse, France, available at: http://www.cerfacs.fr/oa4web/oasis3-mct_3.0/oasis3mct_UserGuide.pdf (last access: 29 October 2015), 2013.

van Genuchten, M. T.: A closed-form equation for predicting the hydraulic conductivity of unsaturated soils, Soil Sci. Soc. Am. J., 44, 892–898, 1980.

Williams, J. L. and Maxwell, R. M.: Propagating subsurface uncertainty to the atmosphere using fully coupled stochastic simulations, J. Hydrometeorol., 12, 690–701, doi:10.1175/2011jhm1363.1, 2011.

Wood, E. F., Roundy, J. K., Troy, T. J., van Beek, L. P. H., Bierkens, M. F. P., Blyth, E., de Roo, A., Döll, P., Ek, M., Famiglietti, J., Gochis, D. J., van de Giesen, N., Houser, P., Jaffé, P. R., Kollet, S., Lehner, B., Lettenmaier, D. P., Peters-Lidard, C., Sivapalan, M., Sheffield, J., Wade, A., and Whitehead, P.: Hyperresolution global land surface modeling: meeting a grand challenge for monitoring Earth's terrestrial water, Water Resour. Res., 47, W05301, doi:10.1029/2010wr010090, 2011.

Wu, C.-C. and Margulis, S. A.: Feasibility of real-time soil state and flux characterization for wastewater reuse using an embedded sensor network data assimilation approach, J. Hydrol., 399, 313–325, 2011.

Wu, C.-C. and Margulis, S. A.: Real-time soil moisture and salinity profile estimation using assimilation of embedded sensor datastreams, Vadose Zone J., 12, 1, doi:10.2136/vzj2011.0176, 2013.

Xu, J. and Shu, H.: Assimilating MODIS-based albedo and snow cover fraction into the common land model to improve snow depth simulation with direct insertion and deterministic ensemble Kalman filter methods, J. Geophys. Res.-Atmos., 119, 10684–10701, doi:10.1002/2014jd022012, 2014.