Deliverable 1.3

Modelling and processing services and tools

Creator	*Department of Mathematics Tullio Levi-Civita, University of Padova. ** Forschungszentrum Jülich, Institute of Bio- and Geosciences: Agrosphere (IBG-3).

Creation date	March 22, 2018
Due date	September 1, 2018
Last revision date	September 3, 2018
Status	Final
Type	Report
Description	This deliverable proposes tools to model data and give predictions on the evolution of several quantities of interest.
Right	Public
Language	English
Citation	E. Perracchione*, M. Polato*, D. Tran*, F. Piazzon*, F. Aiolli*, S. De Marchi*, S. Kollet**, C. Montzka**, A. Sperduti*, M. Vianello*, M. Putti*, 2018. Modelling and processing services and tools.
Grant agreement	GEOEssential Deliverable 1.3., ERA-PLANET No 689443

Table of Contents

Introduction

The objective of this deliverable is to provide a survey of the mathematical foundation of the efficient and robust reduced order modelling based on machine-learning techniques that will form the framework of Task 1.3 entitled “Modelling and Processing Services” (MPS). This task aims at delivering value-added services to Essential Variable (EV) within the premises of WP1 actions on “Knowledge Management Services”. The realm of the EVs of interest encompasses a large number of Earth Observation (EO) data types that must be considered within this task, from time series of localized observations at given spatial points to distributed satellite images and model simulations. The large amount of available data necessarily calls for a drastic synthesis to be able to deliver the sought services such as, e.g., short term real-time predictions, feature extraction, data comparison, etc. Effective synthesis can be achieved only via suitable reduced order models that must be at the same time i) computationally efficient to guarantee reasonable computing times, ii) robust to be employed seamlessly on EVs of different nature, and iii) accurate to ensure meaningful analysis of the EVs of interest. Advances in machine learning techniques in combination with novel theories arising in the field of numerical analysis, and more specifically approximation theory, provide an extraordinary opportunity to accomplish the relevant tasks set forth in WP1. This document intends to demonstrate the effectiveness of this combination by developing the necessary theoretical foundation of approximation algorithms that ensure stability and accuracy, in other words robustness, and demonstrate, via simple examples, the applicability of these modelling tools for the purposes of the GEOEssential project.

The problem of combining EO datasets from different sources to gain further knowledge about EVs by the previously mentioned methods is that they are not consistent. Uncertainties and/or biases of EO data may lead to inaccuracies in the calculation of non-observed variables and not solvable knowledge gaps. One example is the closure of water and energy cycle, which is not possible when water mass and energy conservation is not considered. Therefore, to assemble a full picture of the water and energy cycle at European continental scale, the utilization of EO data is supported by process-based simulations. The physically based simulator of choice within GEOEssential is the TerrSysMP model, which simulates the full water and energy system from groundwater up to the upper atmosphere. Different terrestrial compartment models are combined in order to realize a more sophisticated physical description of water, energy and carbon fluxes across compartment boundaries and to provide a more integrated view on terrestrial processes.

This document is organized in two sections. First we detail the key ideas of the data-based modelling efforts that will be considered. Subsequently, we will briefly describe TerrSysMP that will be used, at least initially, to provide datasets to our reduced order modelling. Further analyses of the relationship between specific EO data and comparable simulations will provide simplified process chains to address EVs within the Food-Water-Energy Nexus for GEOEssential.

Data-based models

The modelling and analysis of data (EVs), for instance, coming from distributed measurements of physical quantities and satellite images will be the recurrent theme of the project. Because of the huge size that some of these datasets achieve, reduced models need to be devised. Thus we first focus on novel reduced spatial models for satellite data that will form the basis for the final time-simulators. The key idea behind this study is that, once we are able to consider a reduced model for the image, we can then model the dynamics of the considered quantities (e.g. soil moisture). The reduced order spatial models proposed in this project will be based on the so-called CATCH (Caratheodory-Tchakaloff) algorithm (refer e.g. to Piazzon at al. 2017 and Sommariva and Vianello 2018). Examples of this technique will be given, while for the evolution in time, we will consider at this preliminary stage only artificial time series. In particular, we provide accessible softwares for modelling these series and forecast the future dynamics of the considered quantities.

In the last decades many methods have been proposed for modelling experimental data, especially in the context of machine learning. Time series analysis has been faced by using many different methods, such as neural networks, deep networks (Goodfellow et al. 2016) and regression models. We focus on a theoretical founded framework concerning kernels (Shawe-Taylor and Nello Cristianini 2004). Within this realm, in the last twenty years Support Vector Regression (SVR) (Smola et al. 2004) and Support Vector Machine (SVM) (Vapnik et al. 1995) have been successful methods for many tasks.

More recently, approaches based on kernel-based interpolation or approximation (Fasshauer 2007 and Fasshauer and McCourt 2015), coupled with Reduced Order Methods (ROMs), have also been studied. In general ROMs, see e.g. (De Marchi and Santin 2015), are all routed on information compression techniques and thus find their natural applications in a wide variety of fields, such as in digital image compression, bioinformatics, signal processing and resolution of Partial Differential Equations (PDEs). In particular, when dealing with huge datasets, one usually needs to consider a surrogate model of drastically smaller size, with the aim of dealing with computationally efficient yet sufficiently accurate modelling tools. To achieve this goal, we implement methods base on, e.g., the work of Haasdonk and Santin 2017, Wirtz et al. 2015.

In what follows, after briefly discussing the technique we propose for creating reduced models for images, we analyze different approaches (SVR and ROMs) with the scope of providing effective schemes for data modelling. Moreover, we point out benefits and drawbacks of each method.

We will contribute to the GEOEssential knowledge by proposing examples of different numerical schemes for image modelling, data assimilation and data forecasting, briefly reviewed and tested on different datasets in the next sections.

Data Compression

The schemes we present in this section represent the background for efficiently dealing with huge images. To introduce them, we need some mathematical background recalled in what follows.

Caratheodory-Tchakaloff least squares

In quadrature theory, the basics are due to Tchakaloff's theorem. It states that for every compactly supported measure there exists a positive algebraic quadrature formula with a cardinality that do not exceed the dimension of the exactness polynomial space. Here we focus on polynomial Least Squares (LS) that are orthogonal projections with respect to a discrete measure.

Theorem 1. *Let μ be a multivariate discrete measure supported at a finite set $X_N = \{x_i, i = 1, \dots, N\} \in \mathbb{R}^d$, with correspondent positive weights (masses) $\Lambda_N = \{\lambda_i, i = 1, \dots, N\}$ and let $S = \mathrm{span}(\varphi_1, \dots, \varphi_L)$ a finite dimensional space of d-variate functions defined on $K \supseteq X_N$, with $P = \dim(S_{|X_N}) \leq L$. Then, there exists a quadrature formula with nodes $T_M = \{\boldsymbol{t}_i, i = 1, \dots, M\} \subseteq X_N$ and positive weights $W_M = \{w_i, i = 1, \dots, M\}$, with $M \leq P$, such that*

$$I_\mu(f) = \sum_{i=1}^{N} \lambda_i f(x_i) = \sum_{i=1}^{M} w_i f(t_i), \qquad \forall f \in S_{|X_N}.$$

In other words, such theorem shows that the original sampling set can be replaced by a smaller one. Moreover in (Piazzon at al. 2018), the authors prove that this can be done keeping practically invariant the Least Squares (LS) approximation estimates. To solve the problem outlined in Theorem 1 , i.e. the one of computing weights and nodes, one can use *Quadratic Programming*, namely the classical *Lawson-Hanson active set method* for NonNegative Least Squares (NNLS) or also Linear Programming (LP) via the classical *simplex method;* refer to (Piazzon at al. 2018) for further details.

Indeed, restricting our search to polynomials of degree n, we need to solve the following quadratic minimum problem

$$\begin{cases} \min \left|\left| V^T \boldsymbol{u} - \boldsymbol{b} \right|\right|_2, \\ \boldsymbol{u} \geq \boldsymbol{0}, \end{cases}$$

where V is the classical Vandermonde matrix and $b = V^T \lambda$. Then, the nonvanishing components of such a solution give the weights $W_M = \{w_i, i = 1, \dots, M\}$ and the indexes of the nodes $T_M = \{\boldsymbol{t}_i, i = 1, \dots, M\} \subseteq X_N$.

For the linear programming approach, we instead have to find

$$\begin{cases} \min \boldsymbol{s}^T \boldsymbol{u}, \\ V^T \boldsymbol{u} = \boldsymbol{b}, \\ \boldsymbol{u} \geq \boldsymbol{0}, \end{cases}$$

where the constraints identify a polytope (the feasible region) and the vector $\boldsymbol{s}$ is chosen to be linearly independent from the rows of V^T.

This method enables us to consider only few points to reconstruct a given function f. Precisely, let us consider two subsets: $X_N = \{x_i, i = 1, \dots, N\} \subseteq \Omega$, with $\Omega \subseteq \mathbb{R}^d$ the set of distinct data points (or data sites or nodes), arbitrarily distributed on Ω and $F_N = \{f_i, i = 1, \dots, N\}$, $f_i \in \mathbb{R}$, the associated set of data values (or measurements or function values). We model the function f by considering M points extracted via Theorem 1 and then we apply some approximation schemes, such as in this case, polynomial least squares.
Note that, for the images we can think of X_N as the set of pixels with the associated values F_N. Then, we construct the reduced model and by evaluating it, we can reconstruct the image at each pixel via the approximated values, namely y_k, $k = 1, \dots, N$.

Image Subsampling

We now give an example devoted to show how the method previously explained can be useful for constructing reduced models for satellite images. First note that the scheme is independent of the problem geometry, i.e. it can be applied to any polygonal region. In fact, once the reduced number of points is found, then we can adopt a polynomial reconstruction scheme for the function. Therefore, if the polynomial expansion consists of m coefficients $(c_1, \dots, c_m)$ we can define the Compression Ratio (CR) as

$$\mathrm{CR} = \frac{N}{m}.$$

As accuracy indicators, once the model is evaluated at *S* evaluation points, we define the Relative Root Mean Square Error (RRMSE) and Maximum Absolute Error (MAE), whose formulae are:

$$\mathrm{RRMSE} = \left(\frac{\sum_{k=1}^{S} (f_k - y_k)^2}{\sum_{k=1}^{S} f_k^2} \right)^{\frac{1}{2}}, \qquad \mathrm{MAE} = \max_{\mathrm{k}=1,\dots,\mathrm{S}} |f_k - y_k|.$$

where y_k are the approximated values at the pixels.

As example, we take the image plotted in Figure 1 (top left) and we construct our reduced model for Spain (on the polygon plotted in red). Figure 1 summarizes the steps of the algorithm, i.e. at first points are extracted and then the image is modelled. Finally, we also report an error plot (see bottom right). Furthermore, from Figure 2, we can note that the statistics of the approximated image, i.e. variance (Var) and mean (E), are comparable with the original one. As a remark, we point out that to reduce the Gibbs phenomenon one could use the method based on discontinuous basis functions proposed in (De Marchi et al. 2018). In that paper a test for the soil moisture over Portugal is presented.

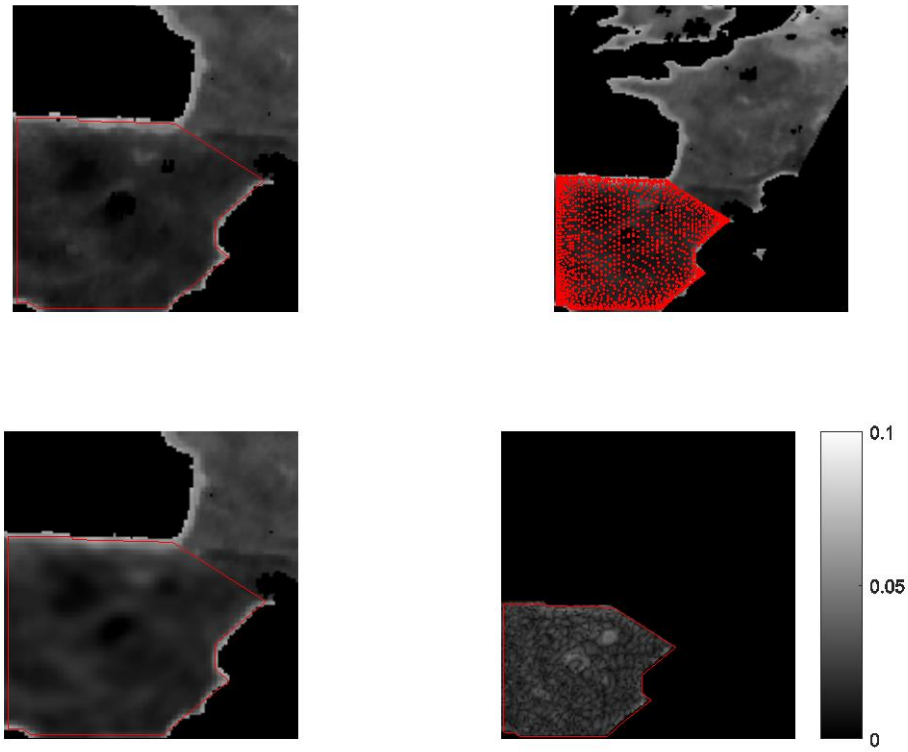

Figure 1: Reduced model for an image taken by SMAP satellite on April 2015. From left to right, top to bottom: original image over Spain with the extracted boundary of the area of interest; distribution of the extracted sampling points; reconstructed image with polynomial of degree n=28; relative (Full Scale) FS MAE.

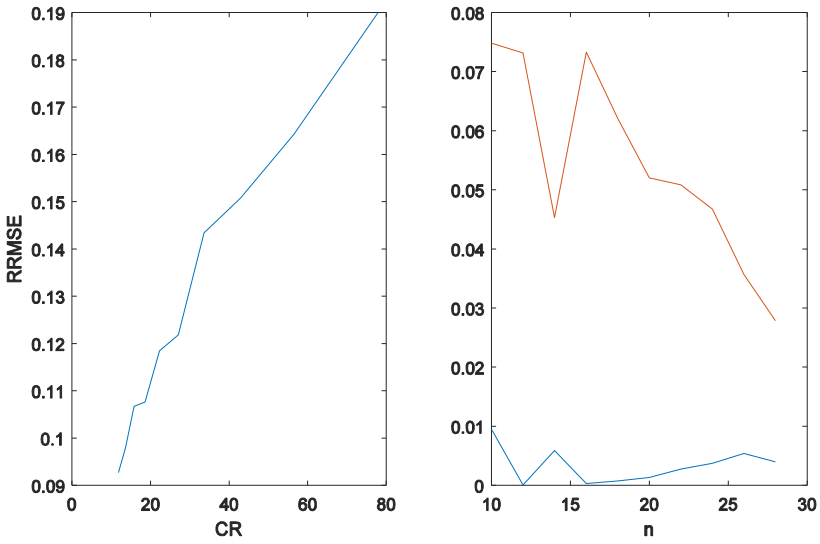

Figure 2: Left: Compression ratio versus the relative RRMSE. Right: in red is plotted $|Var(f_k) - Var(y_k)|/|Var(f_k)|$, and in blue $|E(f_k) - E(y_k)|/|E(f_k)|$.

Modelling tools

As future work, we need to study the dynamics of the images in time. To reach this aim, we need to focus on the single reduced pixels and see them as time series. Eventually, we can also work with the coefficients of polynomials. Here, in the numerical experiments, we only focus on artificial data and noisy time series. However, the schemes presented in what follows are the materials and methods for reaching the final scope with satellite images.

For time series data, we compare:

1. Kernel-based data assimilation,
2. Learning with kernels.

Kernel-based data assimilation

To introduce the first method, we need to review the basic theory of RBF interpolation approaches.

Reduced order interpolation

If we suppose to have several function values $f_i, i = 1, \dots, N$, obtained by sampling some (unknown) function $f: \Omega \to \mathbb{R}$ at the nodes $\boldsymbol{x}_i, i = 1, \dots, N$, a way to model such data consists in finding a function $R: \Omega \to \mathbb{R}$ so that:

$$R(\boldsymbol{x}_i) = f_i, \qquad i = 1, \dots, N. \tag{1}$$

The so-constructed function R is known as interpolant function. Focusing on Radial Basis Functions (RBFs), it can be expressed as:

$$R(\boldsymbol{x}) = \sum_{i=1}^{N} c_i \phi\left(\left|\left|\boldsymbol{x} - \boldsymbol{x}_i\right|\right|_2\right),$$

where ϕ is the so-called RBF. For many further details refer to e.g. (Fasshauer 2007 and Wendland 2005). For several examples of RBFs, refer to Table 1. Note that to each univariate function ϕ, we can associate a positive definite kernel $K: \Omega \times \Omega \to \mathbb{R}$ such that

$$K(x, y) = \phi\left(\left|\left|x - y\right|\right|_2\right) = \phi(r).$$

Table 1: Examples of RBFs together with their smoothness degree.

RBFs	$\boldsymbol{\phi(r)}$	Regularity
Gaussian (GA)	$e^{-\varepsilon^2 r^2}$	C^∞
Matérn (M0)	$e^{-\varepsilon r}$	C^0
Matérn (M2)	$e^{-\varepsilon r}(1 + \varepsilon r)$	C^2

Matérn (M4)	$e^{-\varepsilon r}(\varepsilon^2 r^2 + 3\varepsilon r + 3)$	C^4
Matérn (M6)	$e^{-\varepsilon r}(\varepsilon^3 r^3 + 6\varepsilon^2 r^2 + 15\varepsilon r + 15)$	C^6

Remark. Note that the RBFs depend on the so-called shape parameter $\varepsilon > 0 \in \mathbb{R}$. Such parameter, being a scale parameter can influence the accuracy of the modelling process. Nevertheless, we are not going into details and we refer the reader to (Fasshauer 2007 and Fasshauer and McCourt 2015), where strategies for selecting good values for ε are presented. In particular, to construct the model, we select a safe value for the shape parameter via the so-called Leave One Out Cross-Validation (LOOCV).

Then, if we impose the conditions (1), we reduce to solving a linear system of the form

$$Ac = f, \tag{2}$$

where $A_{ik} = \phi\left(\left|\left|\boldsymbol{x}_i - \boldsymbol{x}_k\right|\right|_2\right)$, $i, k = 1, \dots, N$, $c = (c_1, \dots, c_N)^T$ and $f = (y_1, .., y_N)^T$. Note that, since we consider *strictly positive definite* RBFs, the system (2) admits a unique solution, i.e. we deal with well-posed problems; refer e.g. to (Wendland 2005). In the following, by virtue of the equivalence between kernels and RBFs, we might write the entries of A as $A_{ik} = K(\boldsymbol{x}_i, \boldsymbol{x}_k)$.

To explain how to select the shape parameter via LOOCV, let

$$R^{[i]}(\boldsymbol{x}) = \sum_{k=1, k\neq i}^{N} c_k \phi\left(\left|\left|\boldsymbol{x} - \boldsymbol{x}_i\right|\right|_2\right),$$

be the interpolant obtained leaving out the i-th data on Ω. Moreover, let

$$e_i = \left|f_i - R^{[i]}(\boldsymbol{x}_i)\right|,$$

be the error at the i-th point. Then, the quality of the fit is determined by some norm of the vector of errors

$$\boldsymbol{e} = (e_1, \dots, e_N)^T,$$

obtained by removing in turn one of the data points and comparing the resulting fit with the known value at the removed point. This implementation would be truly expensive; indeed for each data we need to calculate the inverse of the interpolation matrix. A possible solution comes from (Rippa 1999). Precisely, we can simplify the computation to a single formula by calculating

$$e_i = \frac{c_i}{A_{ii}^{-1}},$$

where c_i is the i −th coefficient of the RBF interpolant based on the full dataset and A_{ii}^{-1} is the i −th diagonal element of the inverse of the corresponding local interpolation matrix. Then, to select the *optimal* shape parameter, we add a loop for different values of ε and we simply select the one that leads to the minimal estimated error via LOOCV. Note that the term optimal is used with abuse of notation. Each scheme used to predict the optimal values only

gives an approximation that is *close* to the one that can be found via trials and errors when the solution is known.

To comment on the convergence of the RBF interpolation method, we need to introduce the so-called *fill-distance h,* which is a measure of data distribution. It is defined as

$$h = \sup_{x\in\Omega} \min_{x_i \in X_N} ||\boldsymbol{x} - \boldsymbol{x}_i||_2.$$

Then, it can be proved (Wendland 2005) that the error decreases according to the fill distance and the interpolation with a C^{2k} smooth kernel has approximation order k. Consequently, the approximation order k is arbitrarily high for infinitely smooth strictly positive definite functions, while for strictly positive definite functions with limited smoothness the approximation order is limited by the smoothness of the function. Thus, the choice of the RBF affects the fitting process.

The interpolation approach is suitable as long as data are not affected by noise, which is not our case. Therefore, if the data is contaminated by error, i.e. $f_i = f(x_i) + \epsilon_i$, one no longer wants to fit the data exactly. Here one usually assumes that one has *Gaussian white noise*, i.e. $\epsilon = (\epsilon_1, .., \epsilon_N)^T \sim N(0, \sigma_\epsilon^2 I)$, where I is the $N \times N$ identity matrix. Then, it is customary to treat such a problem with a *Tikhonov regularization* method so that one computes the coefficients as

$$\boldsymbol{c} = (A + \lambda I)^{-1}\boldsymbol{f}, \tag{3}$$

where λ is the so-called *Tikhonov parameter*; see e.g. (Sarra 2017) for details. Note that, trivially, (3) is the solution of the following unconstrained optimization problem:

$$\min_{\mathbf{c}\in\mathbb{R}^N} (\boldsymbol{f} - A\boldsymbol{c})^T(\boldsymbol{f} - A\boldsymbol{c}) + \lambda \boldsymbol{c}^t A\boldsymbol{c}, \tag{4}$$

Indeed, since (4) is a quadratic minimization problem, setting the gradient with respect to $\boldsymbol{c}$ equal to zero is not only a necessary condition, but also sufficient to solve the problem:

$$\nabla_{\boldsymbol{c}} [(\boldsymbol{f} - A\boldsymbol{c})^T(\boldsymbol{f} - A\boldsymbol{c}) + \lambda \boldsymbol{c}^t A\boldsymbol{c}] = 0 \;\leftrightarrow\; -\boldsymbol{f} + A\boldsymbol{c} + \lambda \boldsymbol{c} = 0,$$

which shows the equivalence between (3) and (4).

Dealing with RBFs, we need to point out that there exists a conflict between theoretical accuracy and numerical stability. In fact, if a large number of interpolation nodes are involved, because of the ill-conditioning we might have inaccurate approximations. Thus, recent research focuses on stable computation of the interpolant/approximant or in reduced order methods; refer e.g. to (De Marchi et al. 2016, Wirtz et al. 2015). We now drive our attention to the problem of constructing a reduced model which involves only a smaller subset of bases, i.e. a smaller number of nodes.

Thus, following (Haasdonk and Santin 2017, Wirtz et al. 2015), given the initial set consisting of N data the aim if the one of finding a suitable subspace (reduced), spanned by M centres.

Here, the selection will be carried out by means of a greedy approach. In this sense, we will have training and validation sets. Of course the points that will form the subspace will be selected so that, the error estimated via the validation set is below a certain tolerance. The idea is depicted in Figure 3. To be more precise, the algorithm can be summarized as follows.

RBF-ROM Algorithm

1. Let the function $f: \Omega \to \mathbb{R}$. Denote by R its interpolant
2. Take $X_0 \neq \emptyset$, and for $k > 0$, if there exists an index k such that $|f(\boldsymbol{x}_k) - R(\boldsymbol{x}_k)| > \tau$, where τ is a fixed tolerance, define the new sets of nodes as:
   a. $\boldsymbol{x}_k = \text{argmax}_{\boldsymbol{x} \in X_N \setminus X_{k-1}} |f(\boldsymbol{x}) - R(\boldsymbol{x})|$.
   b. $X_k = X_{k-1} \cup \{\boldsymbol{x}_k\}$ if and only if $|f(\boldsymbol{x}_k) - R(\boldsymbol{x}_k)| > \tau$.
   c. A suitable subspace of M bases, with $M \ll N$ is found.

The convergence of the algorithm is studied in (Haasdonk and Santin 2017) and essentially follows from the fact that filling out the domain with more and more bases leads to decreasing the fill distance.

Remark. The method here proposed allows to construct a reduced model, i.e. the RBF interpolant, that can be used in data assimilation context and that enables us to provide reliable approximations on the future dynamics of a given process. More details are given in the next section.

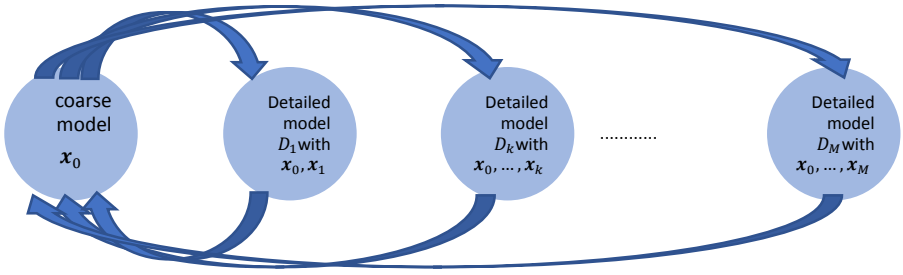

Figure 3: Communication diagram for macro and microscale model.

Kalman filter

Once we construct the model, the aim is the one of incorporating new observations and predictions into the model state of a numerical model. To this aim, we propose a well-known scheme based on the so-called Ensemble Kalman Filter (EnKF). To introduce it, we need to recall the basic features of the Extended Kalman Filter. Let us take a discrete-time nonlinear system with dynamics

$$\boldsymbol{t}_{k+1} = f(\boldsymbol{t}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k,$$

and measurements

$$\boldsymbol{l}_k = g(\mathbf{t}_k) + \boldsymbol{v}_k,$$

where in general, $\boldsymbol{t}_k, \boldsymbol{w}_k \in \mathbb{R}^d$, $u_k \in \mathbb{R}^p$, $\boldsymbol{l}_k, v_k \in \mathbb{R}^q$. We assume that $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are stationary zero-mean white noise processes with covariance matrices Q_k and Z_k, respectively. Furthermore, let $\boldsymbol{t}_0, \boldsymbol{w}_k$ and $\boldsymbol{v}_k$ be uncorrelated. The scope is to construct estimates $\boldsymbol{t}_k^a$ of the state $\boldsymbol{t}_k$ using the measurements so that

$$\mathrm{tr}(\mathrm{E}[\boldsymbol{\delta}_k^a(\boldsymbol{\delta}_k^a)^t]),$$

is minimized, where $\boldsymbol{\delta}_k^a = \boldsymbol{t}_k - \boldsymbol{t}_k^a$.
When the dynamics is linear, i.e.

$$f(\boldsymbol{t}_k, \boldsymbol{u}_k) = B_k \boldsymbol{t}_k + C_k \boldsymbol{u}_k.$$

$$g(\boldsymbol{t}_k) = D_k \boldsymbol{t}_k.$$

we define the analysis state error covariance $P_k^a \in \mathbb{R}^{d\times d}$ as $P_k^a = \mathrm{E}[\boldsymbol{\delta}_k^a(\boldsymbol{\delta}_k^a)^T]$. Furthermore, we introduce the forecast state error covariance $P_k^f \in \mathbb{R}^{d\times d}$, defined by

$$P_k^f = = \mathrm{E}\left[\boldsymbol{\delta}_k^f(\boldsymbol{\delta}_k^f)^T\right],$$

and

$$P_{\boldsymbol{t},\boldsymbol{l}_k}^f = \mathrm{E}\left[\boldsymbol{\delta}_k^f(\boldsymbol{l}_k - \boldsymbol{l}_k^f)^T\right] = P_k^f D_k^T, \qquad P_{\boldsymbol{l},\boldsymbol{l}_k}^f = \mathrm{E}\left[(\boldsymbol{l}_k - \boldsymbol{l}_k^f)(\boldsymbol{l}_k - \boldsymbol{l}_k^f)^T\right] = D_k P_k^f D_k^T + Z_k,$$

where $\boldsymbol{l}_k^f = D\boldsymbol{t}_k^f$, $\boldsymbol{\delta}_k^f = \boldsymbol{l}_k - \boldsymbol{l}_k^f$. Then, the Kalman filter iterations can be summarized in the following two steps:

1. Analysis step:

$$K_k = P_{\boldsymbol{t},\boldsymbol{l}_k}^f \left(P_{\boldsymbol{l},\boldsymbol{l}_k}^f\right)^{-1}, \qquad P_k^a = (I - K_k D_k)P_f^k, \qquad \boldsymbol{t}_k^a = \boldsymbol{t}_k^f + K_k(\boldsymbol{l}_k - D_k \boldsymbol{t}_k^f).$$

2. Forecast step:

$$\boldsymbol{t}_{k+1}^f = B_k \boldsymbol{t}_k^a + C_k \boldsymbol{u}_k, \qquad P_{k+1}^f = B_k P_k^a B_k^T + Q_k.$$

When the dynamics is nonlinear, one usually makes use of the Extended Kalman Filter (EKF), where in the forecast step:

$$\boldsymbol{t}_{k+1}^f = f(\boldsymbol{t}_k^a, \boldsymbol{u}_k), \qquad P_{k+1}^f = B_k P_k^a B_k^T + Q_k,$$

and for the data assimilation we have:

$$\boldsymbol{t}_k^a = \boldsymbol{t}_k^f + K_k\left(\boldsymbol{l}_k - g\left(\boldsymbol{t}_k^f\right)\right), \qquad K_k = P_k^f D_k^T\left(D_k P_k^f D_k^T + Z_k\right)^{-1},$$
$$P_k^a = P_k^f - P_k^f D_k^T\left(D_k P_k^f D_k^T + Z_k\right)^{-1} D_k P_k^f,$$

where $B_k \in \mathbb{R}^{d\times d}$ and $D_k \in \mathbb{R}^{q\times d}$ are given by

$$B_k = \frac{\partial f(\boldsymbol{t},\boldsymbol{u})}{\partial \boldsymbol{t}}|_{\boldsymbol{t}=\boldsymbol{t}_k^a}, \qquad D_k = \frac{\partial g(\boldsymbol{t})}{\partial \boldsymbol{t}}|_{\boldsymbol{t}=\boldsymbol{t}_k^a}.$$

While when the dynamics is linear the Kalman filter produces optimal estimates of the state, the EnKF for non-linear model is a suboptimal estimator, where the statistical errors are predicted by producing an ensemble $T_k^f = \left(\boldsymbol{t}_k^{f_1}, \dots, \boldsymbol{t}_k^{f_s}\right)^T$ and f_i refers to the i −th forecast ensemble member. Then, we define the ensemble mean $\bar{\boldsymbol{t}}_k^f \in \mathbb{R}^d$ as

$$\bar{\boldsymbol{t}}_k^f = \frac{1}{s}\sum_{i=1}^{s} \boldsymbol{t}_k^{f_i}.$$

To approximate the state $\boldsymbol{t}_k$, we first introduce the ensemble error matrix $S_k^f \in \mathbb{R}^{d\times s}$

$$S_k^f = \left[\boldsymbol{t}_k^{f_1} - \bar{\boldsymbol{t}}_k^f, \dots, \boldsymbol{t}_k^{f_s} - \bar{\boldsymbol{t}}_k^f\right],$$

and the ensemble of output error $S_{\boldsymbol{l}_k}^a \in \mathbb{R}^{d\times s}$

$$S_{\boldsymbol{l}_k}^a = \left[\boldsymbol{l}_k^{f_1} - \bar{\boldsymbol{l}}_k^f, \dots, \boldsymbol{l}_k^{f_s} - \bar{\boldsymbol{l}}_k^f\right].$$

Taking into account the notation previously introduced we approximate P_k^f by $\hat{P}_k^f$, $P_{\boldsymbol{t},\boldsymbol{l}_k}^f$ by $\hat{P}_{\boldsymbol{t},\boldsymbol{l}_k}^f$ and $P_{\boldsymbol{l},\boldsymbol{l}_k}^f$ by $\hat{P}_{\boldsymbol{l},\boldsymbol{l}_k}^f$, with

$$\hat{P}_k^f = \frac{1}{s-1}S_k^f\left(S_k^f\right)^T, \qquad \hat{P}_{\boldsymbol{t},\boldsymbol{l}_k}^f = \frac{1}{s-1}S_k^f\left(S_{\boldsymbol{l}_k}^f\right)^T, \qquad \hat{P}_{\boldsymbol{l},\boldsymbol{l}_k}^f = \frac{1}{s-1}S_{\boldsymbol{l}_k}^f\left(S_{\boldsymbol{l}_k}^f\right)^T.$$

Therefore, the spread of the ensemble members around the mean is the error between best estimate and actual state, while we can see the ensemble mean as the best forecast estimate of the state. Then, for each $i = 1, \dots, s$, we define

$$\boldsymbol{t}_k^{a_i} = \boldsymbol{t}_k^{f_i} + \hat{K}\left(\boldsymbol{l}_k^i - g\left(\boldsymbol{t}_k^{f_i}\right)\right),$$

and the perturbed observations $\boldsymbol{l}_k^i = \boldsymbol{l}_k + \boldsymbol{v}_k^i$, where $\boldsymbol{v}_k^i$ is a zero mean random variable with normal distribution and covariance Z_k. Then, letting

$$\bar{\boldsymbol{t}}_k^a = \frac{1}{s}\sum_{i=1}^{s} \boldsymbol{t}_k^{a_i}.$$

the analysis error covariance P_k^a is approximated by

$$\hat{P}_k^a = \frac{1}{s-1} S_k^a (S_k^a)^T, \quad \text{with} \quad S_k^a = \left[\boldsymbol{t}_k^{a_1} - \bar{\boldsymbol{t}}_k^a, \dots, \boldsymbol{t}_k^{a_s} - \bar{\boldsymbol{t}}_k^a \right].$$

Finally, in agreement with the linear Kalman filter, we get $\boldsymbol{t}_{k+1}^{f_i} = f\left(\boldsymbol{t}_k^{a_i}, \boldsymbol{u}_k\right) + \boldsymbol{w}_k^i$, where $\boldsymbol{w}_k^i$ are from a normal distribution with zero average and covariance Q_k.

After introducing $\widehat{K}_k = \hat{P}_{\boldsymbol{t},\boldsymbol{l}_k}^f \left(\hat{P}_{\boldsymbol{l},\boldsymbol{l}_k}^f\right)^{-1}$, we summarize the two main steps as follows:

1. Analysis step:

$$\boldsymbol{t}_k^{a_i} = \boldsymbol{t}_k^{f_i} + K_k \left(\boldsymbol{l}_k + \boldsymbol{v}_k^i - g\left(\boldsymbol{t}_k^{f_i}\right) \right).$$

2. Forecast step:

$$\boldsymbol{t}_{k+1}^{f_i} = f\left(\boldsymbol{t}_k^{a_i} + \boldsymbol{v}_k^i\right).$$

In our case, the model f is constructed via the RBF interpolant R on the reduced number of basis, as explained before. Then, to build the forecast step, we perturb with random Gaussian noise the optimal shape parameter found via LOOCV. In this way, we are able to obtain an ensemble and to give reliable previsions on the future dynamics.

Learning with kernels

Machine learning (Schölkopf and Smola 2001) is a field of computer science which aims to discover patterns from data. In particular, the goal of supervised learning is to discover which relation links inputs and outputs. Given $X_N = \{x_i, i = 1, \dots, N\} \subseteq \Omega$, and $F_N = \{f_i, i = 1, \dots, N\} \subseteq \mathbb{R}$, we formally define the training set $D = X_N \times F_N$, in which we assume there exists a function (i.e., relation) such that $\forall (x_i, y_i) \in D, g(x_i) \approx f_i$. Again, a learning algorithm tries to find a function (a.k.a. model or hypothesis) that approximates the unknown function g as tightly as possible.

Kernel-based methods are one of the most used machine learning approaches. The basic idea behind these kinds of methods is related to the so-called kernel trick which allows to implicitly compute vector similarities (defined in terms of dot-product) in potentially infinite dimensional spaces. In the following we give a brief overview about kernel functions and kernel methods in machine learning.

It is well-known that $K(x, y)$ admits the following expansion

$$K(x, y) = \Phi(x)^T \Phi(y),$$

where $\Phi: \Omega \to H$ is a mapping function from Ω to the embedding (a.k.a. feature) space H (Shawe-Taylor and Cristianini 2004). An important observation is that, by defining a kernel function, we are implicitly defining a new representation of the inputs via the embedding function Φ.

Kernel Machines

Kernel machines are a particular family of machine learning methods which are based on kernels. Most of them try to minimize the following optimization problem

$$\arg\min_{g \in H} L(g(x_1), \ldots, g(x_N)) + \Lambda ||g||_H,$$

where L is a loss (or cost) function associated to the empirical error, Λ is a trade-off parameter which gives more or less importance to the regularization term (smoothness function) represented by the norm of the functional g. It can be demonstrated that (Representer Theorem, Schölkopf et al. 2001) the solution of such kind of problems can be computed by

$$g(x) = w^T \Phi(x) = \sum_{i=1}^{N} c_i K(x_i, x),$$

which means that g can be expressed as a hyperplane in a feature space defined by the function Φ, and hence it can be seen as a weighted sum of kernels between the input vector and the vectors in the training set. The intuition behind this result is that, by using the embedding function Φ the data are projected into a (usually higher dimensional) space in which the hypothesis we are looking for becomes a linear function that can be expressed in terms of training examples. This concept is depicted in Figure 4 on a classification problem.

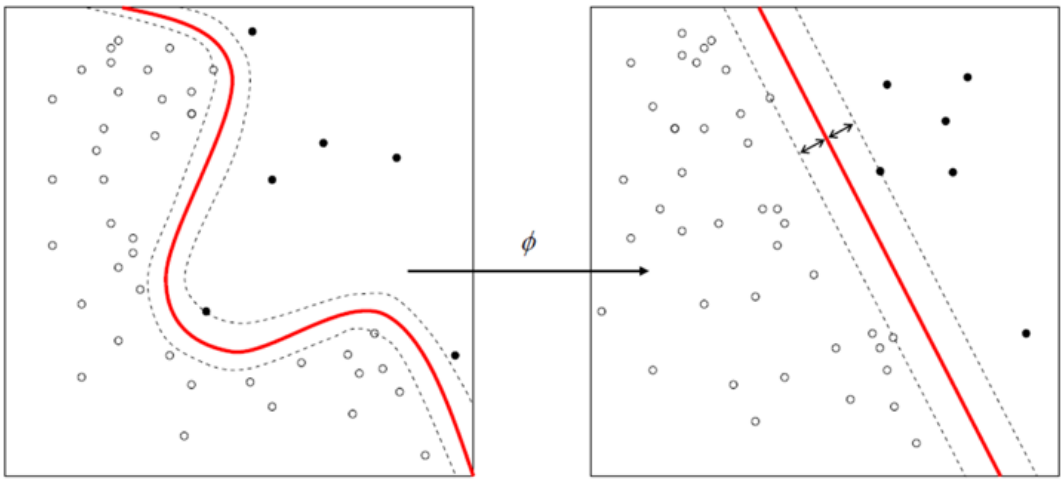

Figure 4: Example of feature mapping and corresponding classification boundary.

The kernel trick essentially is to define the value of the kernel in terms of original space itself without even defining the representation given by Φ.

This modularity allows to define kernel matrices which are independent w.r.t. the algorithm which is going to use it. So, by using the same algorithm many different kernels can be used and only the kernel computation needs to be done.

Support vector regression

Support vector machine is the most famous and successful kernel method. It has demonstrated state of the art performances in many learning tasks. It is usually used for classification tasks, but it can be easily adapted to regression (SVR) (Smola and Schölkopf 2004, Cortes and Vapnik 1995). In the following we provide the SVR optimization problem (in its primal from)

$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{N}(\zeta_i + \zeta_i^*),$$

subject to:

$$\begin{aligned} &y_i - w^T\phi(x_i) - b \leq \epsilon + \zeta_i, \\ &w^T\phi(x_i) + b - y_i \leq \epsilon + \zeta_i^*, \\ &\zeta_i, \zeta_i^* \geq 0, \end{aligned}$$

for $i = 1, \dots N$, where the objective function aims to minimize the squared norm of the hypothesis in order to get a smooth function, while maintaining the number of errors as low as possible. C represents the trade-off hyper-parameter. The hyper-parameter ϵ indicates the width of the "tube" in which the examples can fall into without being counted as errors. Figure 5 shows this concept.

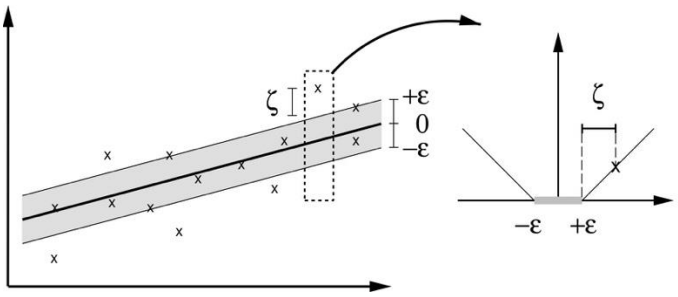

Figure 5: SVR depiction: on the left the epsilon-tube and on the right the linear penalization given to the mistaken points.

This problem is usually solved in its dual form defined as

$$\min_{\boldsymbol{c},\boldsymbol{c}^*} \frac{1}{2}\sum_{i,j=1}^{N}(c_i - c_i^*)(c_j - c_j^*)K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{N}(c_i + c_i^*)\epsilon - \sum_{i,j=1}^{N}(c_i - c_j)y_i$$

subject to $\forall i$:

$$\begin{aligned} &\sum_{i=1}^{N}(c_i - c_i^*) = 0 \\ &c_i, c_i^* \in [0, C] \end{aligned}$$

and consequently the final regression model takes the following form:

$$g(\boldsymbol{x}) = \sum_{i \in SV} (c_i - c_i^*) K(\boldsymbol{x}, \boldsymbol{x}_i) + b,$$

where SV is the set of training examples $\boldsymbol{x}_i$ such that the corresponding $c_i$or c_i^* are not both zero, and they are called support vectors.

Preliminar results

In this section, we summarize the results of the data-based models obtained via reduced order schemes and machine learning methods.

The data

As dataset for the current study on time series, we consider the data collected in the South-Eastern part of the Veneto Region and available at

http://voss.dmsa.unipd.it/

The above mentioned dataset has been created for an experimental study of the organic soil compaction and prediction of the land subsidence related to climate changes in the South-Eastern area of the Venice Lagoon catchment (VOSS - Venice Organic Soil Subsidence). Such data were collected with the contribution of the University of Padova (UNIPD) from 2001 to 2006. Different physical quantities, measured each hour, are available. For instance, here we consider the temperature sampled one meter below the soil and the measurements collected via the potentiometer. Examples of such data are shown in what follows.

Experiments for reduced order schemes

In this section, we test the kernel-based method coupled with data assimilation on two different data sets (refer to Figure 6). The first data are rescaled measurements of the temperature at one meter under soil, while the second ones correspond to samples obtained via the potentiometer. Both measurements are sampled each hour. They correspond to about ten and three months measurements in 2002. With the method based on reduced order schemes, we can select, as test cases, an arbitrary number of points, indeed, they are not based on learning features, i.e. there is no need to see seasonality and periodicity in the data. Note that there are missing data, due to some damages of instruments. Anyway, the method is robust for this situation indeed it works for scattered data.

The method here proposed starts with the original datasets, then it extracts a certain number of bases, i.e. points (see Figure 6). For the first dataset we take the Matérn M6 RBF, while, since the second dataset is less smooth, we consider a function with lower regularity, i.e. the Matérn M2 kernel. Being time series, once the model is constructed, the data assimilation previously described is used to approximate the data from the last reduced basis (let us say

x_B) till the end of the test set (see Figure 6). To have a feedback on the accuracy, we compute the RRMSE and MAE on the forecasted values. The results are summarized in Table 2.

As a final remark, we point out that the proposed scheme ca be used iteratively. Let us suppose to have at a certain fixed time T_1 a set of M reduced based, then we apply the Kalman filter as long as the model fits the data whiting the prefixed tolerance. And if at a certain time T_2 , where usually $T_2 \gg T_1$, the $(M+1)$ −th basis is added, then we adjust the model according to that and again repeat the forecast step. In this sense, it assumes the form of a data assimilation procedure. For an example, refer to Figure 7.

Table 2. Accuracy indicators obtained via RBF-reduced order methods and Kalman filter.

N	M	B	$N-B$	RRMSE	MAE
7819	34	6944	875	$1.30\text{E}-02$	$2.28\text{E}-02$
2075	158	1991	94	$3.81\text{E}-02$	$4.25\text{E}-02$

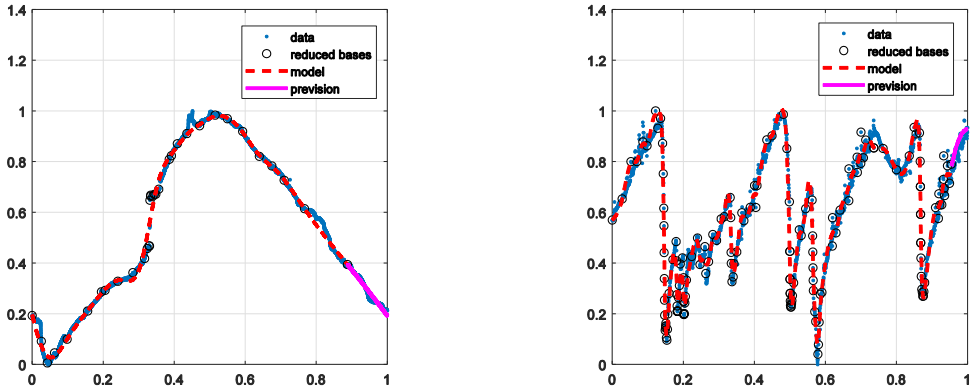

Figure 6: Graphical results via RBF-reduced order methods and Kalman filter. Left: temperature data, right: potentiometer samples.

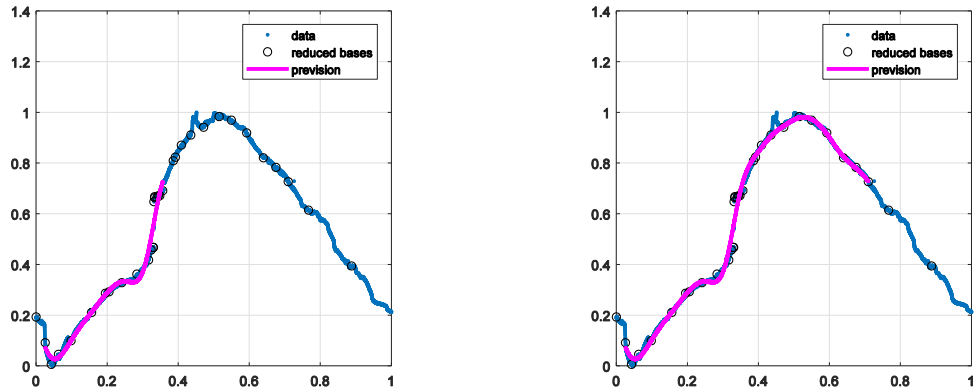

Figure 7: The RBF-reduced order methods and Kalman filter applied iteratively on the temperature dataset. The figure shows the progresses of the algorithm for two different time steps.

Experiments for learning schemes

Here we show the results achieved using SVR on the dataset concerning the temperatures at one meter under soil registered over 2 years. First, we test the SVR using the gaussian kernel. During the test we use, in temporal order, the first 98% of the dataset as training set and last 2% as the test set. So, the model has to predict the soil temperature for the last (roughly) 11 days (280 temperature values) of the dataset. We used a 5-fold cross validation in order to validate the SVR parameter C and the kernel hyper-parameter ε. In order to build the training set we use a sliding window approach where an instance is defined as the vector containing 48 consecutive temperature values (without missing values), and the corresponding target value is the very next temperature value. The achieved results are reported in Figure 8.

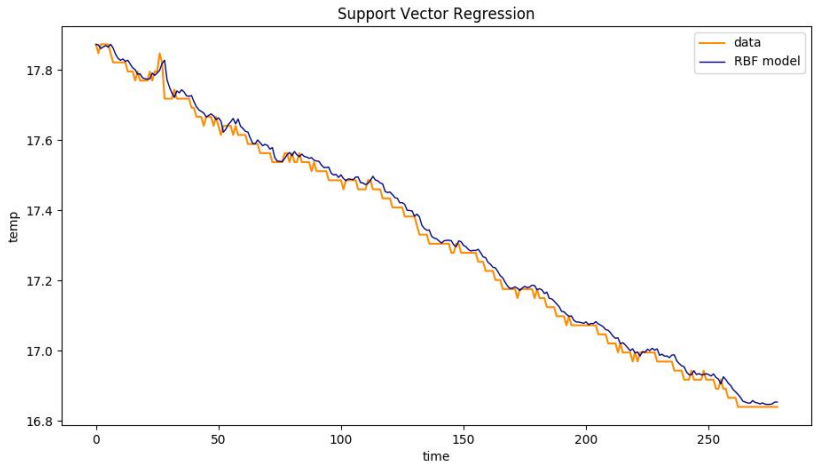

Figure 8: SVR prediction using RBF kernel.

In order to predict the temperature at the next time step in the test setting the actual previous 48 temperature values (and not the predicted ones) have been used. In order to build a system able to predict for a complete day in the future (24 hours), we carry out a second experiment in which we built one model for each hour in the future. So, the prediction at 1 hour in the future is done via the first model, 2 hours in the future by the second model and so on. In this way the method is completely agnostic about the future and it do not use any information about it in order to give a prediction. The achieved results are reported in Figure 9.

Table 3 summarizes the achieved RMSE by both methods and it also provides the range of values validated for all the hyper-parameters.

As a remark, we point out that the surrogate reduced order models, can be used to train the kernel machines. Usually, in these cases results are more robust, refer to (Aminian Shahrokhabadi et al. 2018) and the related Python software freely available at https://github.com/makgyver/vlabtestrepo/.

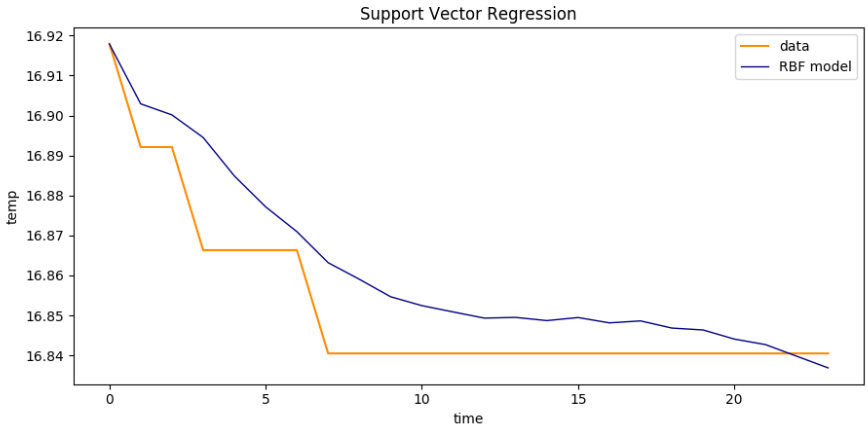

Figure 9: results achieved by the 24 SVR models, one for each hour in the future.

Table 3: Results achieved by the SVR models.

#Models	C	ε	RRMSE
Single	$10^{-2}, \dots, 10^{5}$	$10^{-7}, \dots, 10^{2}$	$2.22\mathrm{E}-02$
24	$10^{-2}, \dots, 10^{5}$	$10^{-7}, \dots, 10^{2}$	$1.17\mathrm{E}-02$

Physically-based models

The terrestrial system modelling platform (TerrSysMP) was developed to simulate the interaction between lateral flow processes in river basins with the lower atmospheric boundary layer (Shresta at al. 2014). It features three model components: COSMO – the weather prediction model of the German Weather Service, CLM – the community land surface model hosted at NCAR, the subsurface model ParFlow, and an external coupler OASIS3 that drives the system. This platform allows explicit modelling of land-atmosphere interactions across scales ranging from meters to kilometers via mosaic and downscaling/upscaling approaches. New parameterizations for root water uptake, additional plant function types, downscaling algorithms, and CO2 exchange processes have recently been implemented into TerrSysMP. The Centre for High-Performance Scientific Computing in Terrestrial Systems (HPSC TerrSys) is operating TerrSysMP in a forecasting setup over North Rhine-Westphalia and Europe. The model results are made publicly available daily as videos via the YouTube Channel of HPSC TerrSys https://www.youtube.com/channel/UCGio3ckQwasR5a_kJo1GdOw. The fully coupled TerrSysMP is initialized every day in the early morning hours in a two- and six-node configuration on the Jülich Supercomputing Centre JURECA HPC system for a lead time of a day for the NRW domain at 1km/0.5km spatial resolution and about 3 days at 12km resolution for the EURO-CORDEX pan-European model domain. The simulations encompass different fluxes and states of the terrestrial hydrologic and energy cycles from aquifers across the land surface into the atmosphere. The atmospheric boundary conditions are kindly provided by the ECMWF for the European and the DWD for the NRW simulations. Although the creation and publication of the results are fully automatic, the system is not considered and meant in

any way as an "operational forecast" but rather a "monitoring run". It can be used to gain further insight in TerrSysMP physics features, it helps with technical model development, but primarily, it provides an insight in the fully coupled terrestrial water budgets.

Atmospheric processes are simulated with COSMO-DE (Baldauf at 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 at al. 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; Kollet and Maxwell 2006). 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.

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 et al. 2013). 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. The data that are exchanged in TerrSysMP via OASISMCT are schematically shown in Fig. 11. ParFlow provides CLM with its calculated subsurface pressure and saturation values for the first 10 subsurface layers and in return CLM provides the upper boundary condition for ParFlow, consisting of the recharge values 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 and latent heat flux), outgoing long-wave radiation, momentum flux and albedo as a lower boundary condition to COSMO-DE. In turn, COSMO-DE provides forcing data to CLM including air pressure, air temperature, wind velocity, incoming short-wave and long-wave radiation, specific humidity and precipitation. 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 important 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. Refer also to Figure 10.

Modelling and processing services for SDGs workflows

The presented models and methods will be utilized to fulfil the aim of GEOEssential, i.e. providing workflows to address SDGs by EO of EVs. While TerrSysMP is able to provide consistent data of the full system at climate relevant time periods (3 decades), EO data is typically mission specific (~5 years) and focuses on single variables. In order to investigate the links between integrative simulation (TerrSysMP) and EVs as well as EO and EVs by the proposed methods (e.g. learning schemes), example EV workflows will be implemented. The general aim is to support the Global indicator framework for the Sustainable Development Goals and targets of the 2030 Agenda for Sustainable Development.

One example workflow is targeting towards Goal 15: Protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss. More specifically, goal 15.3 (By 2030, combat desertification, restore degraded land and soil, including land affected by desertification, drought and floods, and strive to achieve a land degradation-neutral world)

will be addressed by an investigation of drought occurrence, drought severity, and drought temporal change. Drought indices will be calculated from EO and simulation data, and analyzed regarding their suitability for SDG evaluation.

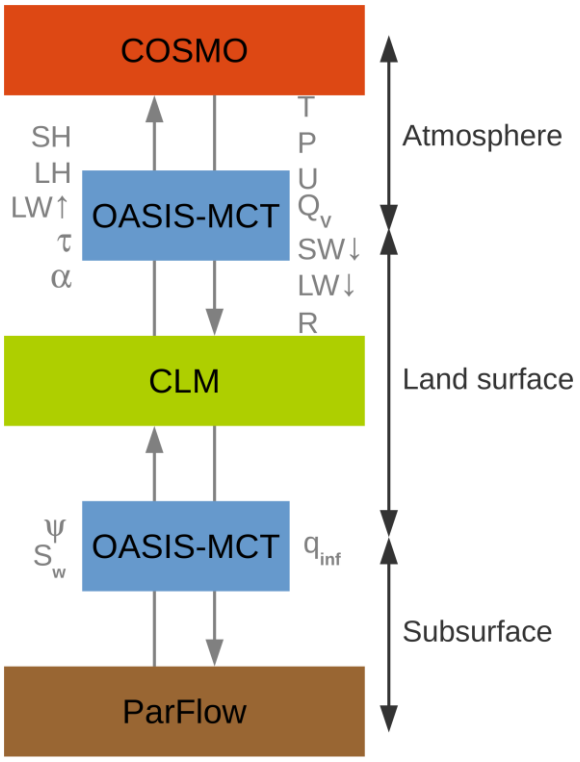

Figure 10. Coupling of the TerrSysMP component models ParFlow (subsurface), CLM (land surface) and COSMO-DE (atmosphere) by OASIS-MCT. The exchanged fluxes and state variables are: subsurface pressure, subsurface saturation, net infiltration flux, sensible heat flux, latent heat flux, outgoing long-wave radiation, momentum flux, albedo, air pressure, air temperature, wind velocity, incoming short-wave radiation, incoming long-wave radiation, specific humidity and precipitation.

The Soil Water Deficit Index (SWDI, calculated by soil moisture, field capacity and wilting point information) as an agricultural index and the Palmer Drought Severity Index (PDSI, calculated from precipitation and evapotranspiration) as a meteorological index will be investigated. Machine learning will ease and fasten the linkage between EO data and SDG indicators.

Summary

The main scope of this deliverable is the one of analyzing robust reduced methods in the machine learning framework. This can also be seen as the background for Task 1.6 entitled "Data Fusion" (DF). The current document is divided into two main sections that respectively deal with data- and physically-based models. As pointed out in the manuscript the former takes advantage of being independent of any physical assumptions.

Furthermore, aside comparing these two approaches and making reliable tests, as work in progress, we also planned to merge such schemes together. This would allow us to obtain reliable predictions on the evolution of the considered EVs. Indeed, after modelling the spatial counterpart of the images via TerrSysMP, we are able to retain only the meaningful information and then predicting the time evolution via machine learning tools. Few examples in this direction are already provided in the current deliverable. In this sense, this document demonstrates the effectiveness of reduced order tools by developing the necessary theoretical foundation and by demonstrating, via simple examples, the applicability of these modelling approaches for the purposes of the GEOEssential project.

References

Aminian Shahrokhabadi, M., E. Perracchione and M. Polato. 2018. "Learning with reduced kernel-based methods: Environmental and financial applications". Preprint.

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

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

Cortes, C., and V. N. Vladimir. 1995. "Support-vector networks". *Machine Learning*. 20: 273–297.

De Marchi, S., A. Idda, and G. Santin. 2016. "A Rescaled Method for RBF Approximation". Approximation Theory XV: San Antonio 2016, vol. 201, 39–59.

De Marchi, S., F. Marchetti, and E. Perracchione. 2018. "Jumping with Variably Scaled Discontinuous Kernels (VSDKs)". Preprint.

De Marchi, S., and G. Santin. 2015. "Fast computation of orthonormal basis for RBF spaces through Krylov space methods". *BIT* 55: 949–966.

Fasshauer, G.E. 2007. "Meshfree Approximations Methods with Matlab". *World Scientific*, Singapore.

Fasshauer, G.E., and M.J. McCourt. 2015. "Kernel-based Approximation Methods Using Matlab". *World Scientific*, Singapore.

Gasper, F., K. Goergen, P. Shrestha, M. Sulis, J. Rihani, M. Geimer, and S. Kollet. 2014. "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.

Gillijns, S., O.B. Mendoza, J. Chandrasekar , B.L.R. De Moor, D. S. Bernstein, and A. J. Ridley. 2006. "What is the ensemble Kalman filter and how well does it work?". *American Control Conference* 1–6.

Goodfellow, I., Y. Bengio and A. Courville. 2016. "Deep Learning". MIT Press, Cambridge, MA, USA.

Haasdonk, B., and G. Santin. 2017. "Greedy Kernel Approximation for Sparse Surrogate Modelling". *Proceedings of the KoMSO Challenge Workshop on Reduced-Order Modeling for Simulation and Optimization*.

Jones, J. E. and C.S. Woodward. 2001. "Newton–Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems", *Adv. Water Resour.*

24:763–774.
Kollet, S. and R.M. Maxwell. 2008. “Capturing the influence of groundwater dynamics on land surface processes using an integrated, distributed watershed model”. *Water Resour. Res.* 44:W02402.
Maxwell, R.M. 2013. “A terrain-following grid transform and preconditioner for parallel, large-scale, integrated hydrologic modeling”, *Adv. Water Resour.* 53:109–117.
Oleson, K.W., G.Y. Niu, Z.L. Yang, D.M. Lawrence, P.E. Thornton, P.J. Lawrence, R. Stöckli, R.E. Dickinson, G.B. Bonan, S. Levis, A. Dai, and T. Qian. 2008. “Improvements to the Community Land Model and their impact on the hydrological cycle”. *J. Geophys. Res.-Biogeo*. 113:G01025.
Piazzon, F., A. Sommariva and M. Vianello. 2017. “Caratheodory-Tchakaloff Least Squares”. *Sampling Theory and Applications 2017, IEEE Xplore Digital Library*.
Rippa, S. 1999. “An algorithm for selecting a good value for the parameter c in radial basis function interpolation”. *Adv. Comput. Math.* 11: 193–210.
Sarra, S.A. 2017. “The Matlab radial basis function toolbox”. *J. Open Research Software,* 5:1– 10.
Schölkopf, B., R. Herbrich, and A.J. Smola. 2001. “A Generalized Representer Theorem”. *Computational Learning Theory. Lecture Notes in Computer Science*. 111: 416–426.
Schölkopf, B., and A.J. Smola. 2001. “Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond”. MIT Press, Cambridge, MA, USA.
Shawe-Taylor, J., and N. Cristianini. 2004. “Kernel Methods for Pattern Analysis”. Cambridge University Press, New York, NY, USA.
Shrestha, P., M. Sulis, M. Masbou, S. Kollet, and C. Simmer. 2014. “A scale-consistent terrestrial systems modeling platform based on COSMO, CLM, and ParFlow”, *Mon. Weather Rev.* 142:3466–3483.
Smola, A. J., and B. Schölkopf. 2004. “A tutorial on support vector regression”. *Statistics and Computing*. 14:199–222.
Sommariva, A. and M. Vianello. 2018. “Nearly optimal nested sensors location for polynomial regression on complex geometries”. *Sampl. Theory Signal Image Process.* 17:95–101.
Valcke, S. 2013. “The OASIS3 coupler: a European climate modelling community software”. Geosci. Model Dev. 6:373–388.
Wendland, H. 2005. “Scattered Data Approximation”. *Cambridge Monogr. Appl. Comput. Math.*, vol. 17, Cambridge Univ. Press, Cambridge.
Wirtz, D., N. Karajan and B. Haasdonk. 2015. “Surrogate modelling of multiscale models using kernel methods”. *Int. J. Numer. Met. Eng.* 101:1–28.