DOI: 10.14529/jsfi250406

Variational Data Assimilation in the Constructor of Dynamic Soil Carbon Models

Siumbel K. Shangareeva[1], ***Victor M. Stepanenko***[1,2], ***George M. Faykin***[1], ***Alexander I. Medvedev***[1], ***Irina M. Ryzhova***[3], ***Vladimir A. Romanenkov***[3]

This work presents an automatic adjoint-model construction within the Carbon Cycle Model Constructor (CCMC) that enables variational data assimilation (VDA) for estimating the initial state of soil dynamic carbon models. The adjoint is generated once from the generic pool-flux representation used in CCMC, which allows efficient gradient evaluation and iterative optimization of the initial pool vector without constructing a model-specific adjoint. The proposed approach is tested with two soil carbon models: SOCS (Soil Organic Carbon Saturation) and RothC (Rothamsted model). Data assimilation experiments are performed using long-term field observations of soil carbon content. The entire VDA workflow, including the adjoint solver and optimization algorithm, is implemented in the same Fortran code base as CCMC. CCMC+VDA implementation is fully compatible with the MPI+OpenMP TerM land surface model and provides a reusable, scalable foundation for variational soil-carbon data assimilation on modern supercomputers.

Keywords: data assimilation, carbon dynamic models, adjoint model, automatic differentiation.

Introduction

Variational data assimilation (VDA) is a class of mathematical methods and numerical techniques used to reduce uncertainty in the external parameters of mathematical models by optimizing performance metrics (objective or cost function) with respect to observed data and prior estimates. VDA relies on the adjoint-equation framework to compute the gradient of the objective (cost) function. Among geophysical models, hydrodynamical models of the atmosphere and the ocean have most benefited from deep integration of VDA into research and operational systems over the last decades [18]. Land surface models have benefited less from VDA, even though they include many parameters and initial states that are not directly measurable. Specifically, this relates to terrestrial carbon-cycle models, where soil carbon pools are rarely measured in situ, and equation parameters are usually phenomenological, suggesting no method of field assessment. For the carbon cycle, VDA can potentially align model parameters and states with a variety of observations (e.g., soil carbon content, CO_2 fluxes, sensible and latent heat fluxes, etc.).

Over the past decade, the practical effectiveness of VDA-based modeling systems has been demonstrated at both regional and global levels [7, 8, 11, 17, 22, 23]. In [11], the authors propose a step-by-step data assimilation system that sequentially optimizes the parameters of the ORCHIDEE model to improve the model's estimation of terrestrial carbon uptake using three data streams: Moderate Resolution Imaging Spectroradiometer (MODIS)-Normalized Difference Vegetation Index (NDVI) satellite observations, net ecosystem exchange (NEE) and latent heat (LE) flux measurements from FLUXNET stations, and atmospheric CO_2 concentrations modeled using the general circulation model (GCM) of the Laboratoire de Météorologie Dy-

[1]Lomonosov Moscow State University, Research Computing Center, Moscow, Russian Federation
[2]Moscow Center of Fundamental and Applied Mathematics, Moscow, Russian Federation
[3]Lomonosov Moscow State University, Soil Science Faculty, Moscow, Russian Federation

namique (LMDz). In [17], the posterior parameter covariance obtained by applying a variational method to the BETHY land surface model is used to quantify forecast errors of CO_2 fluxes and atmospheric concentrations.

In these studies, adjoint model construction relies on source-to-source automatic differentiation (AD) to obtain adjoint code; the most widely used tool is TAF [7, 8, 11, 12, 16]. An alternative is to build the model within the YAO variational-assimilation platform, where the model is described as a modular graph for which an adjoint is generated automatically, as in [2]. These approaches are tightly coupled to specific model implementations. Using TAF requires restructuring and annotating the Fortran code to satisfy the constraints of the AD toolchain, YAO requires rewriting the model as a component graph within its own modeling language. As a result, adjoints must be rebuilt and revalidated for each individual model and model version.

In this work, we develop a module for automatic adjoint construction for models specified in the Carbon Cycle Model Constructor (CCMC) [5] to seek the initial conditions by VDA. The constructor is a general-purpose tool for building carbon-cycle models and is intended for integration into the INM RAS-MSU land active-layer model TerM (Terrestrial Model) [20]. By integrating adjoint generation into CCMC itself, we obtain a reusable, model-agnostic adjoint solver that automatically adapts to any CCMC-defined model without rewriting model code.

The paper is organized as follows. Section 1 briefly reviews the basic VDA framework to be then applied for carbon cycle models. Next, Section 2 presents the concept of carbon cycle model constructor (CCMC), which has been recently proposed and implemented by the authors; introduction of the VDA algorithm to CCMC code is described. Numerical experiments for testing the created CCMC+VDA code are performed with RothC and SOCS model, as detailed in Section 3; the code scalability towards larger scale problems and multicore computer systems and distributed-memory supercomputers is discussed. The Conclusions section summarizes the demonstrated properties of the developed CCMC+VDA code and draws prospects for future research.

1. Variational Data Assimilation

The variational data assimilation problem can be formulated as follows [9, 10, 14, 15]. Let us consider a model described by a system of differential equations:

$$\begin{cases} \frac{\mathrm{d}C(t)}{\mathrm{d}t} = F(C,t), \quad t \in (0,T), \\ C(0) = C_0, \end{cases} \tag{1}$$

where C is the state vector of the model (for carbon-cycle models, C typically represents a vector of carbon pools); F is the models dynamical operator; and C_0 is the unknown initial state to be determined. The optimal initial state C_0^{a} is found as the solution to the following minimization problem:

$$C_0^{\mathrm{a}} = \arg\min_{C_0} J(C_0), \tag{2}$$

$$J(C_0) = \frac{1}{2}\left(C_0 - C_0^b\right)^\top B^{-1}\left(C_0 - C_0^b\right) + \frac{1}{2}\int_0^T \left[HC(t) - y^{\mathrm{obs}}(t)\right]^\top R^{-1}\left[HC(t) - y^{\mathrm{obs}}(t)\right] \mathrm{d}t, \tag{3}$$

where C_0^b is the background (prior) initial state, B and R are the covariance matrices of background and observation errors, respectively, y^{obs} denotes the observation vector, and H is the

corresponding observation operator, projecting the model state vector to the vector of observations. In the variational assimilation approach, the gradient of the cost function $J(C_0)$ is computed by solving the adjoint problem, yielding the optimality system (1), (4), (5), which can be derived, for example, via the method of Lagrange multipliers:

$$\begin{cases} \frac{\mathrm{d}C^*(t)}{\mathrm{d}t} = -\left[\nabla_C F(C,t)\right]^\top C^*(t) + H^\top R^{-1}\left(HC(t) - y^{\mathrm{obs}}(t)\right), \quad t \in (0,T), \\ C^*(T) = 0. \end{cases} \tag{4}$$

$$\nabla_{C_0} J = B^{-1}(C_0 - C_0^b) - C^*(0) = 0. \tag{5}$$

We then solve, in sequence, the forward model (1) and the adjoint model (4); their outputs are inserted into (5) to compute the cost-function gradient, which drives an update of C_0 using an appropriate gradient-based optimization method. Iterations proceed until the change between successive estimates of C_0 becomes sufficiently small or until a maximal number of iterations is exceeded.

2. Carbon Cycle Model Constructor and Data Assimilation

2.1. CCMC Concept and Implementation

The Carbon Cycle Model Constructor [5] enables the implementation of models representable as a system of differential equations that describe the dynamics of N_p carbon pools:

$$\frac{dC_i}{dt} = F_i = \sum_{j=1}^{N_p} \sum_{k=1}^{N_{i,j}^f} F_{i,j}^k, \quad F_{i,j}^k = \prod_{m=1}^{N_{i,j,k}^m} f_{i,j,k}^m\left(\psi_{i,j,k}^m\right), \quad i = 1, \ldots, N_p, \tag{6}$$

where C_i is the carbon content in the i-th pool (a scalar variable); $N_{i,j}^f$ is the number of fluxes between pools i and j; $F_{i,j}^k$ is the k-th flux between pools i and j; $N_{i,j,k}^m$ is the number of multiplicative factors in the expression for the k-th flux between pools i and j; $f_{i,j,k}^m(\cdot)$ is the m-th single-argument function in the expression for the k-th flux between pools i and j; and $\psi_{i,j,k}^m$ is the argument of the m-th function, representing a biotic or abiotic driver of the process, which can be either one of the pools or an external given variable.

The set of possible forms of $f_{i,j,k}^m(\psi_{i,j,k}^m)$ in most carbon cycle models reduces to a number of standard functional dependencies (linear, exponential, Michaelis–Menten, etc.). This allows most models to be implemented within a single code in which the model structure is specified by a collection of standardized multiplicative factors $f = f(\psi)$. Accordingly, the constructor provides an interface for setting the number of pools, the graph of fluxes between pools, and the factors $f_{i,j,k}^m$ so that a solver for the general system (6) implements the model in a such specified configuration.

In order to implement a data assimilation system in CCMC for initial-state estimation for any model specified in CCMC, it is necessary to set the initial guess C_0^b, the error-covariance matrices B and R, and the observation operator H, as well as construct the adjoint model. As can be seen from (4), this requires computing partial derivatives of the models dynamics operator with respect to the carbon pools. The next section describes how this differentiation can be automated for models formulated within CCMC framework, i.e., those that can be specified by (6).

2.2. Automatic Construction of Adjoint Models in CCMC

CCMC solves the forward problem (1) numerically using a first-order explicit time-stepping scheme with a fixed time step Δt. The adjoint problem (4) for the general CCMC equation (6) takes the form:

$$\frac{\mathrm{d}C^*(t)}{\mathrm{d}t} = -\left[\frac{\partial F}{\partial C}(C(t),t)\right]^\top C^*(t) + H^\top R^{-1}\left[HC(t) - y^{\mathrm{obs}}(t)\right], \quad F = (F_1,\ldots,F_{N_p}). \tag{7}$$

To construct the adjoint problem (7), it is necessary to evaluate $\frac{\partial F_i}{\partial C_l}$, where $i,l = 1,\ldots,N_p$. It can be shown that

$$\frac{\partial F_i}{\partial C_l} = \sum_{j=1}^{N_p}\sum_{k=1}^{N_{i,j}^f} F_{i,j}^k \sum_{m=1}^{N_{i,j,k}^m} \frac{1}{f_{i,j,k}^m\left(\psi_{i,j,k}^m\right)} \frac{\partial f_{i,j,k}^m\left(\psi_{i,j,k}^m\right)}{\partial C_l}, \quad l = 1,\ldots,N_p. \tag{8}$$

In the numerical implementation, to avoid division by zero, we replace $\frac{1}{f_{i,j,k}^m\left(\psi_{i,j,k}^m\right)}$ with $\frac{1}{f_{i,j,k}^m\left(\psi_{i,j,k}^m\right)+\varepsilon}$, where ε is a small regularization constant. In the current CCMC programming code, eight basic multiplicative factor functions are available: constant, linear, hyperbolic, Michaelis–Menthen, a step function, an exponential, and two piecewise-linear functions, which form the extendable library of functions. By specifying, for each function, the form of its derivative with respect to C_l (using a library of derivatives $df/d\psi$), i.e., $\frac{\partial f_{i,j,k}^m}{\partial C_l}$, one can compute and store $\frac{\partial F_i}{\partial C_l}$ alongside the evaluation of F_i. With $\frac{\partial F_i}{\partial C_l}$ available, the adjoint problem can be integrated in CCMC using the explicit solver used for the forward model.

2.3. Code Modifications to CCMC

To enable automatic adjoint solver construction within CCMC and to solve the data-assimilation system for restoring the initial-state vector, the constructor code was modified as follows:

1. For each base factor function $f_{i,j,k}^m$ implemented in the constructor, their derivatives with respect to C_i were added.
2. A function returning $\frac{\partial F_i}{\partial C_l}$ using formula (8) was added to the method that calculates the right-hand side F_i. The computed derivatives are stored over all time levels, since the adjoint solver requires access to their full temporal history when integrating backward in time.
3. A module was added in which, after solving the forward problem, the adjoint one is solved numerically using an explicit time scheme. The error-covariance matrices B, R, the observation operator H, and the first approximation for the optimized initial state C_0^b are also specified in this module.
4. A function was added which computes the gradient of the cost function $J(C_0)$ via (5).
5. The iterative adaptive gradient method Adagrad [4] was implemented to update the initial condition C_0 values according to the following formulas:

$$\begin{aligned} g_k &= \nabla_{C_0} J(C_{0,k-1}), \\ G_k &= G_{k-1} + g_k \odot g_k, \\ C_{0,k} &= C_{0,k-1} - \alpha \frac{g_k}{\sqrt{G_k} + \varepsilon}. \end{aligned} \tag{9}$$

Here, $\odot$ denotes the elementwise product of vectors, and the operations of squaring, division, and taking square roots of vectors are all taken elementwise; k is the iteration number, α

is the learning rate, and ε is a small constant for numerical stability. Adagrad was chosen because the scale of the pools can vary greatly, and dynamically recalculating the learning step at each iteration for each pool allows this to be taken into account, thus achieving better convergence [4]. Figure 1 illustrates the schematic flow of CCMC with the embedded adjoint and optimization components.

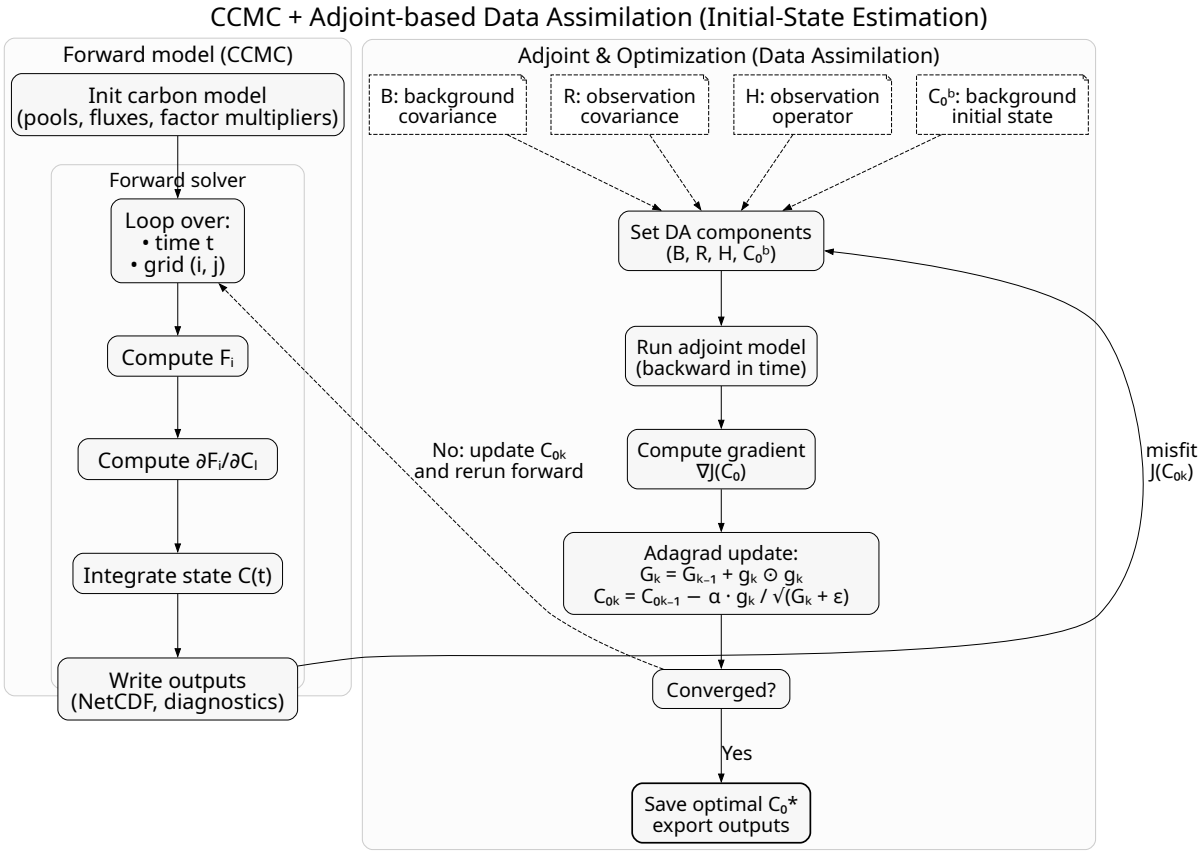


Figure 1. Flowchart of the variational data assimilation module in the carbon cycle model constructor

3. Numerical Experiments

The data-assimilation system for initial-state estimation within CCMC was tested for the two carbon-soil models implemented in CCMC: SOCS (Soil Organic Carbon Saturation) [13] and RothC (Rothamsted model) [3]. In both cases the optimization is performed in a low-dimensional state space: in a single site (zero-dimensional column) the control vector C_0 has dimension N_p, with $N_p = 2$ for SOCS and $N_p = 4$ for RothC.

3.1. SOCS Model

This model describes the dynamics of two soil-organic-matter (SOM) pools: a free (unprotected) pool and a protected pool formed through organo-mineral interactions and physical

occlusion within microaggregates [13]. The model equations read:

$$\begin{aligned}\frac{\mathrm{d}C_1}{\mathrm{d}t} &= I - (1-r)\cdot k\cdot C_1 - rkC_1\cdot\left(1-\frac{C_2}{C_m}\right) + k_d C_2,\\ \frac{\mathrm{d}C_2}{\mathrm{d}t} &= r\cdot k\cdot C_1\cdot(1-\frac{C_2}{C_m}) - k_d\cdot C_2,\end{aligned} \tag{10}$$

where C_1 is carbon in the free (unprotected) SOM pool; C_2 is carbon in the protected SOM pool; C_m is the maximum amount of organic carbon that can be protected in soil (the soils protective capacity); I is the input of organic carbon to the soil; r is the fraction of carbon transferred to the protected pool during decomposition of C_1; $(1-r)$ are respiration losses; k is the decomposition-rate coefficient for C_1; and k_d is the rate coefficient for the transition of carbon from C_2 to C_1 due to desorption and aggregate breakdown.

3.2. RothC Model

In CCMC, the soil component of the RothC model is implemented, with the prescribed rate of plant litter input to the soil. The evolution of carbon stocks is described by the following differential equations [3]:

$$\begin{aligned}\frac{\mathrm{d}C_{\mathrm{DPM}}}{dt} &= f_{\mathrm{dpm}}\cdot F_{\mathrm{lit}} - R_{\mathrm{DPM}},\\ \frac{\mathrm{d}C_{\mathrm{RPM}}}{dt} &= (1-f_{\mathrm{dpm}})\cdot F_{\mathrm{lit}} - R_{\mathrm{RPM}},\\ \frac{\mathrm{d}C_{\mathrm{BIO}}}{dt} &= f_{\mathrm{bio}}\cdot\beta_R R_S - R_{\mathrm{BIO}},\\ \frac{\mathrm{d}C_{\mathrm{HUM}}}{dt} &= f_{\mathrm{hum}}\cdot\beta_R R_S - R_{\mathrm{HUM}},\end{aligned} \tag{11}$$

where β_R is the clay fraction (particles < 0.002 mm) in the soil; F_{lit} is the input of organic matter to the soil from vegetation, crop residues, and organic fertilizers, R_S is the total respiration rate over the four pools ($R_S = \sum_i R_i$, where $i \in \{\mathrm{DPM}, \mathrm{RPM}, \mathrm{BIO}, \mathrm{HUM}\}$); R_{BIO}, R_{HUM}, R_{RPM}, and R_{DPM} are the respiration rates of the microbial biomass (BIO), long lived humified (HUM), resistant plant material (RPM), and decomposable plant material (DPM) pools, respectively; f_{dpm} is the litter-quality function; and f_{bio} and f_{hum} are partitioning coefficients that allocate incoming organic matter to the BIO and HUM pools during mineralization.

The inert organic matter (IOM) pool is calculated from the total soil carbon at the initial time and kept constant during the simulation:

$$C_{\mathrm{IOM}} = a_{q1} C_{\mathrm{tot}}^{a_{q2}}, \tag{12}$$

where C_{tot} is the total soil carbon content (the sum of all pools), $a_{\mathrm{q1}} = 0.049$, $a_{\mathrm{q2}} = 1.139$ – empirical dimensionless constants.

The terms R_i are computed as

$$R_i = k_{si} F_T(T_{\mathrm{soil}}) F_s(s) F_v(v) C_i, \tag{13}$$

where $i \in \{\mathrm{DPM}, \mathrm{RPM}, \mathrm{BIO}, \mathrm{HUM}\}$, k_{si} is the respiration rate per unit mass of pool i under standard conditions [1/s]; $F_T(T_{\mathrm{soil}})$ is the soil-temperature factor; $F_s(s)$ is the soil-moisture factor; and $F_v(v)$ accounts for vegetation cover. The standard respiration rates k_{si} used in this

study are

$$k_{s,\mathrm{DPM}} = 3.22, \qquad k_{s,\mathrm{RPM}} = 9.65, \qquad k_{s,\mathrm{BIO}} = 2.12, \qquad k_{s,\mathrm{HUM}} = 6.43.$$

The IOM pool has no respiration term ($k_{s,\mathrm{IOM}} = 0$).

The soil temperature function reflects temperature variations in the upper active soil layer (0–30 cm) and is defined as

$$F_T(T_{\mathrm{soil}}) = \frac{b_1}{1 + e^{b_2/(T_{\mathrm{soil}} - b_3)}}, \tag{14}$$

where T_{soil} is the mean monthly soil temperature [K]; $b_1 = 47.9$ (dimensionless), $b_2 = 106$ K, and $b_3 = 254.85$ K are empirical parameters.

The soil moisture factor is parameterized as

$$F_s(s) = \begin{cases} 1 - d_1 \cdot (s - s_o), & \text{for } s > s_o, \\ d_2 + d_1 \cdot \left(\frac{s - s_{\min}}{s_o - s_{\min}}\right), & \text{for } s_{\min} < s \le s_o, \\ d_2, & \text{for } s \le s_{\min}, \end{cases} \tag{15}$$

where s is the moisture of the upper (unfrozen) soil layer; s_o is the optimal soil moisture at which $F_s(s)$ attains its maximum (i.e., equals one); and $d_1 = 0.8$, $d_2 = 0.2$ are dimensionless empirical coefficients, chosen so that $d_1 + d_2 = 1$, a condition ensuring continuity of F_s. The wilting moisture is determined experimentally:

$$\begin{aligned} s_o &= \tfrac{1}{2} \cdot (1 + s_w), \\ s_{\min} &= c \cdot s_w, \end{aligned} \tag{16}$$

where s_w is the wilting moisture, and $c = 1.7$ is a dimensionless empirical coefficient.

The vegetation-cover factor is given by

$$F_v(v) = e_1 + e_2 \cdot (1 - v), \tag{17}$$

where $v \in [0, 1]$ indicates the presence of vegetation cover. The dimensionless empirical coefficients e_1 and e_2 are 0.6 and 0.4, respectively.

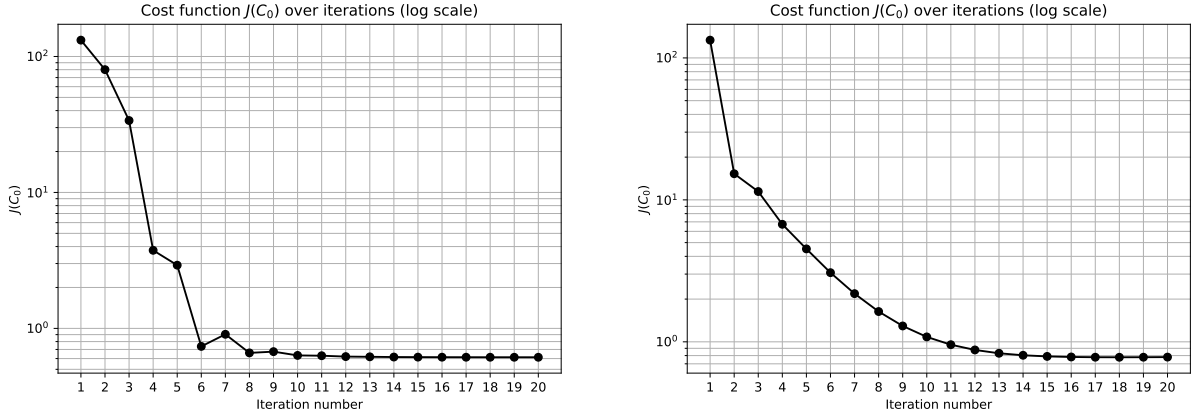
3.3. Results and Discussion

For validation of the variational data assimilation algorithm embedded into CCMC, we used experimental time series of soil carbon content from long-term fertilization field experiments [1] conducted at the Donskoy Federal Agrarian Research Center (Rostov Oblast) and at DAOS-3, the Dolgoprudny Agrochemical Experimental Station (Moscow Oblast). The external forcing time series used in the SOCS and RothC models were compiled from the above-mentioned field stations (soil temperature, soil moisture, organic carbon inputs, mean vegetation cover) together with ERA5 reanalysis data (clay content and wilting-point soil moisture). Both models were integrated with the time step of one month. Their parameter values were set as follows: $r = 0.45, C_m = 12, k_d = 0.007, k = 7.5$ for SOCS and $\beta_R = 0.2, s_w = 0.75, v = 0.8$ for RothC.

The observation operator is $H = (1, 1, \ldots, 1) \in \mathbb{R}^n$, since the measured soil carbon content is the sum of all model pools. The observation-error covariance matrix R is diagonal. Since the observations are of order 10 kg m^{-2}, we assume an absolute measurement uncertainty of 0.1 kg m^{-2}, and therefore set $R^{-1} = 1/0.1$. The background covariance matrix B is also diagonal.

Its diagonal entries are taken as 10% of the characteristic magnitude of each pool, assuming that pool errors are uncorrelated and that no detailed prior covariance information is available. The initial guess for the pool vector C_0 in each model was constructed using expert-based estimates of the typical distribution of soil carbon among pools for the corresponding soil type.

The optimization employs the Adagrad algorithm with learning rate $\alpha = 1$ which was chosen empirically as a compromise between convergence speed and stability. A total of 20 iterations were performed for both models. The convergence of the cost function $J(C_0)$ for the SOCS and RothC models is shown in Fig. 2. Figure 2 demonstrates that, for both SOCS and RothC, the VDA scheme systematically reduces the cost function over the course of the Adagrad iterations and approaches a nearly stationary value by the end of the optimization.



(a) SOCS (b) RothC

Figure 2. Cost function $J(C_0)$ over Adagrad iterations for the SOCS (a) and RothC (b) models

Figures 3 and 4 show the simulated soil carbon stock dynamics obtained with the prior initial conditions and with those estimated via data assimilation for the Rostov Oblast site. Blue dots indicate observations. Tables 1 and 2 compare the prior and optimal initial conditions and report the root-mean-square error (RMSE) of the simulated total soil carbon stocks with respect to the observed time series at the site. In both cases the reference values are the measurements, while the "prior" RMSE is computed from the forward model run started from the background initial state C_0^{b} and the "posterior" RMSE from the run started from the optimized state C_0^{a}. Quantitatively, the VDA-based initialization leads to a strong reduction of the model–data misfit. For SOCS, the RMSE decreases from 1.864 to 0.096 kg m^{-2} and for RothC, the RMSE decreases from 1.88 to 0.152 kg m^{-2}. In Figs. 3 and 4 the prior simulations systematically underestimate the measured soil carbon stocks, whereas the trajectories obtained with VDA-based initial conditions closely track the observed levels and reproduce the temporal evolution of the carbon stocks.

The established way of initializing the pools in carbon cycle models is to run the forward model under statistically stationary external forcing (litter input) for a sufficiently long time period until a quasi-steady-state of the pools is reached [21, 24]. This approach relies on long-term reconstructions of climate, carbon inputs, and land-use history, although such records are typically uncertain. It also presumes that soil carbon is close to equilibrium at the beginning of the simulation, which is often not the case [6, 24]. In our method, the initial pool vector C_0 is treated as an unknown control variable and is estimated directly from the observed soil carbon time series using VDA. This removes the need for uncertain multi-decadal forcing and does not

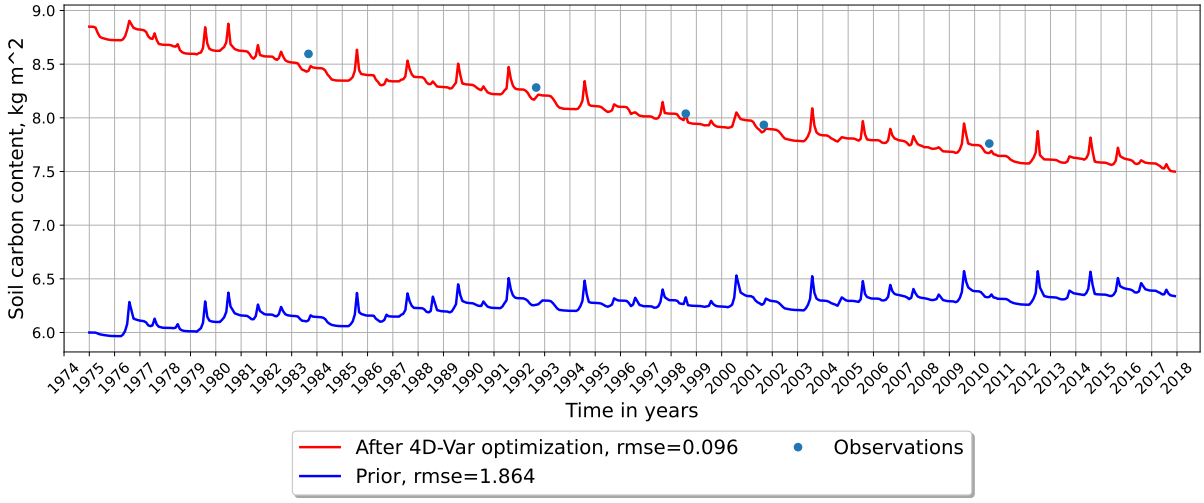


Figure 3. Soil carbon stocks at the Rostov Oblast site simulated by the SOCS model using the prior initial conditions (blue) and initial conditions estimated via data assimilation. Blue dots indicate observations

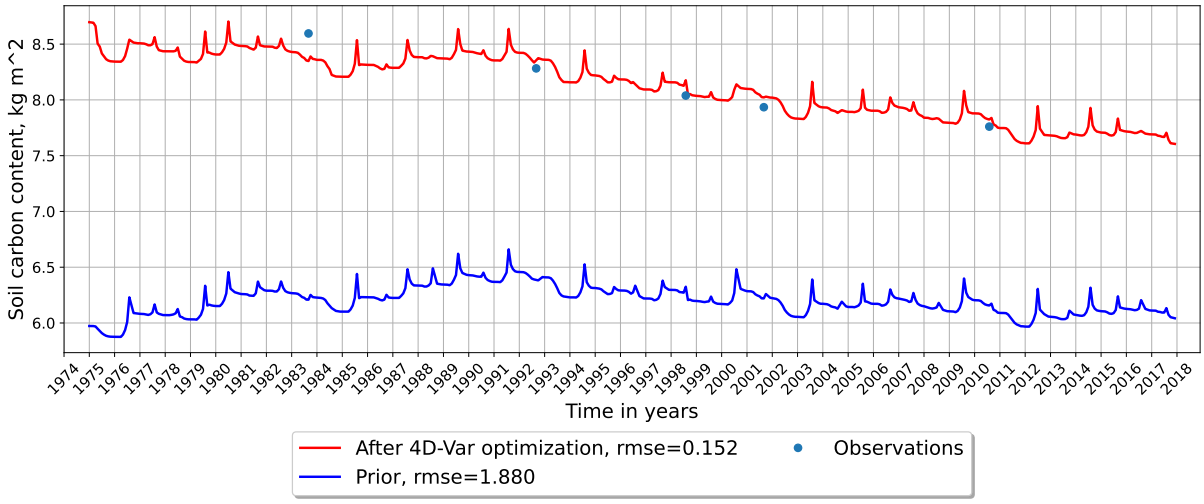


Figure 4. Same as Fig. 3, but simulations are performed with RothC model

Table 1. The prior and optimal initial conditions for SOCS model, and corresponding root-mean-square errors (RMSE)

Pool	Background (prior)	Analysis (posterior)
$C_1(0)$	0.	0.09
$C_2(0)$	6	8.76
RMSE	1.864	0.096

rely on the equilibrium assumption. A limitation of the presented VDA-based initialization is that it requires a time series of soil carbon observations, whereas spin-up can be applied when only a single measurement is available.

The developed CCMC+VDA programming code is intended for implementation into TerM land surface model. The VDA driver, adjoint solver, and Adagrad iterations are implemented in

Table 2. Same as Tab. 1, but simulations are performed with RothC model

Pool	Background (prior)	Analysis (posterior)
$C_{DPM}(0)$	0.01	0.2
$C_{RPM}(0)$	0.1	0.32
$C_{BIO}(0)$	0.05	0.24
$C_{HUM}(0)$	5.	7.13
RMSE	1.88	0.152

the same Fortran code base as CCMC and TerM. The parallel implementation of TerM uses MPI and OpenMP. The TerM computational domain is a rectangular latitude-longitude grid that is decomposed across MPI processes with nested OpenMP threads; each process/node computes all grid points in its subdomain independently of the others [19]. This decomposition is valid because TerM solves a system of one-dimensional vertical equations in the longitude-latitude cells with no horizontal coupling. CCMC+VDA module, including the Adagrad optimizer, follows the same domain decomposition and performs all optimization steps locally within each grid column, requiring no additional inter-process communication. As a result, the Adagrad-based VDA algorithm is fully compatible with the existing MPI+OpenMP parallelization of TerM and scales naturally from a single multicore node to multi-node supercomputers.

All numerical experiments reported in this paper were performed in a single-core mode on an M3 Pro processor (ARM64 architecture). The code was compiled using the GNU Fortran compiler version 15.2.0. For this configuration, a single forward model run required approximately 0.004 s for the SOCS model and 0.006 s for the RothC model. A complete variational data assimilation cycle, including 20 Adagrad iterations, required approximately 0.038 s for SOCS and 0.097 s for RothC.

Conclusion

In this work, we have implemented automatic construction of an adjoint model to the model specified in the carbon cycle model constructor (CCMC), and made the necessary modifications to the constructor code. Based on that, we have also built a solution to the system of variational data assimilation equations for restoring the initial conditions of the carbon model dynamics system. Numerical experiments with data assimilation using the SOCS and RothC models showed a substantial reduction in the model-observations misfit, confirming both the correctness of the data assimilation implementation and the practical value of the variational approach for state variables initialization.

The prospects of extending the variational assimilation method and its implementation presented in this paper include the following:

- moving from initial state estimation to *joint* optimization of initial state and model parameters; this requires adding derivatives of the base functions not only with respect to pools but also with respect to model parameters;
- adding quantification of uncertainty of the optimal solution via the posterior error-covariance matrix;
- further integrating CCMC into the land surface model TerM and optimizing the memory usage of variational data assimilation on modern supercomputers.

Thus, we have established a reproducible and scalable foundation for variational assimilation in CCMC – from automatic adjoint construction and gradient evaluation to practical validation on real soil data.

Acknowledgments

The mathematical method for data assimilation and its implementation in CCMC code was supported by the Ministry of Science and Higher Education of the Russian Federation as part of the program of the Moscow Center of Fundamental and Applied Mathematics under agreement No. 075-15-2025-345. Comparison of simulation data to observations is supported by Multidisciplinary research and educational school "The future of the planet" of the Lomonosov Moscow State University under agreement No. 23-Sh07-55.

References

1. Pryanishnikov institute of agrochemistry. `https://www.vniia-pr.ru/laboratorii/otdl-geoseti/lab-geogr-seti/` (2025), accessed: 2025-09-30

2. Benavides Pinjosovsky, H.S., Thiria, S., Ottlé, C., *et al.*: Variational assimilation of land surface temperature within the ORCHIDEE Land Surface Model Version 1.2.6. Geoscientific Model Development 10(1), 85–104 (2017). `https://doi.org/10.5194/gmd-2016-64`

3. Coleman, K., Jenkinson, D.S.: RothC-26.3-A Model for the turnover of carbon in soil. In: Evaluation of soil organic matter models: Using existing long-term datasets, pp. 237–246. Springer (1996)

4. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12, 2121–2159 (2011). `https://doi.org/10.5555/1953048.2021068`

5. Faykin, G.M., Stepanenko, V.M., Medvedev, A.I., *et al.*: Constructor of soil carbon dynamic models. Numerical Methods and Programming 26(3), 281–303 (2025). `https://doi.org/10.26089/NumMet.v26r320`

6. Herbst, M., Welp, G., Macdonald, A., *et al.*: Correspondence of measured soil carbon fractions and RothC pools for equilibrium and non-equilibrium states. Geoderma 314, 37–46 (2018). `https://doi.org/10.1016/j.geoderma.2017.10.047`

7. Kaminski, T., Knorr, W., Schürmann, G., *et al.*: The BETHY/JSBACH carbon cycle data assimilation system: Experiences and challenges. Journal of Geophysical Research: Biogeosciences 118(4), 1414–1426 (2013). `https://doi.org/10.1002/jgrg.20118`

8. Kuppel, S., Peylin, P., Chevallier, F., *et al.*: Constraining a global ecosystem model with multi-site eddy-covariance data. Biogeosciences 9(10), 3757–3776 (2012). `https://doi.org/10.5194/bg-9-3757-2012`

9. Lions, J.L.: Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles. Dunod Gauthier-Villars (1968)

10. Marchuk, G.I., Orlov, V.V.: On the theory of conjugate functions. In: Krupchinsky, P.A. (ed.) Neutron physics. Sat. articles. Atomizdat, Moscow. p. 30–45. (1961)

11. Peylin, P., Bacour, C., MacBean, N., *et al.*: A new stepwise carbon cycle data assimilation system using multiple data streams to constrain the simulated land surface carbon cycle. Geoscientific Model Development 9(9), 3321–3346 (2016). https://doi.org/10.5194/gmd-9-3321-2016

12. Raoult, N.M., Jupp, T.E., Cox, P.M., Luke, C.M.: Land-surface parameter optimisation using data assimilation techniques: the adJULES system V1.0. Geoscientific Model Development 9(8), 2833–2852 (2016). https://doi.org/10.5194/gmd-9-2833-2016

13. Ryzhova, I.M.: Analysis of soil organic matter dynamics based on minimal carbon-cycle models. Russia's Soils-Strategic Resource: Abstracts of the 8th Congress of the V. V. Dokuchaev Soil Science Society and the School of Young Scientists on Soil Morphology and Classification (Syktyvkar, 2020–2022) 2, 130–131 (2021)

14. Sasaki, Y.: An objective analysis based on the variational method. Meteor. Soc. Japan (1958)

15. Sasaki, Y.: Some basic formalisms in numerical variational analysis. Monthly Weather Review 98(12), 875–883 (1970). https://doi.org/10.1175/1520-0493(1970)098<0875:SBFINV>2.3.CO;2

16. Scholze, M., Kaminski, T., Knorr, W., *et al.*: Simultaneous assimilation of SMOS soil moisture and atmospheric CO2 in-situ observations to constrain the global terrestrial carbon cycle. Remote sensing of environment 180, 334–345 (2016). https://doi.org/10.1016/j.rse.2016.02.058

17. Scholze, M., Kaminski, T., Rayner, P., *et al.*: Propagating uncertainty through prognostic carbon cycle data assimilation system simulations. Journal of Geophysical Research: Atmospheres 112(D17) (2007). https://doi.org/10.1029/2007JD008642

18. Shutyaev, V.P.: Methods for observation data assimilation in problems of physics of atmosphere and ocean. Izvestiya, Atmospheric and Oceanic Physics 55(1), 17–34 (2019). https://doi.org/10.1134/S0001433819010080

19. Stepanenko, V.M.: River routing in the INM RAS-MSU land surface model: Numerical scheme and parallel implementation on hybrid supercomputers. Supercomputing Frontiers and Innovations 9(1), 32–48 (2022). https://doi.org/10.14529/jsfi220103

20. Stepanenko, V.M., Medvedev, A.I., Bogomolov, V.Y., *et al.*: Land surface scheme TerM: the model formulation, code architecture and applications. Russian Journal of Numerical Analysis and Mathematical Modelling 39(6), 363–377 (2024). https://doi.org/10.1515/rnam-2024-0031

21. Taghizadeh-Toosi, A., Cong, W.F., Eriksen, J., *et al.*: Visiting dark sides of model simulation of carbon stocks in European temperate agricultural soils: allometric function and

model initialization. Plant and Soil 450(1), 255–272 (2020). https://doi.org/10.1007/s11104-020-04500-9

22. Thum, T., Zaehle, S., Köhler, P., *et al.*: Modelling sun-induced fluorescence and photosynthesis with a land surface model at local and regional scales in northern Europe. Biogeosciences 14(7), 1969–1987 (2017). https://doi.org/10.5194/bg-14-1969-2017

23. Verbeeck, H., Peylin, P., Bacour, C., *et al.*: Seasonal patterns of CO2 fluxes in Amazon forests: Fusion of eddy covariance data and the ORCHIDEE model. Journal of Geophysical Research: Biogeosciences 116(G2) (2011). https://doi.org/10.1029/2010JG001544

24. Wutzler, T., Reichstein, M.: Soils apart from equilibrium–consequences for soil carbon balance modelling. Biogeosciences 4(1), 125–136 (2007). https://doi.org/10.5194/bg-4-125-2007