HELLENIC REPUBLIC
MINISTRY OF DEVELOPMENT
GENERAL SECRETARIAT FOR RESEARCH AND INNOVATION
HELLENIC FOUNDATION FOR RESEARCH AND INNOVATION

Greece 2.0 NATIONAL RECOVERY AND RESILIENCE PLAN
"BASIC RESEARCH FINANCING" (Horizontal support for all Sciences)
ID 16618 – Subproject 1 (MIS: 5163923)

WP2
D2.1 Report and codes of modeling algorithms

Spatially Explicit Digital Twin of the Greek Agro-Hydro-System

ID 14815

Greece 2.0 NATIONAL RECOVERY AND RESILIENCE PLAN
Funded by the European Union NextGenerationEU

Plan Details

Report and codes of modeling algorithms
Deliverable number: D2.1
Creation Date: 17/12/2024
Last modification date: 10/11/2025
Dissemination Level: Public

1. Introduction

Work Package 2 (Development of the Digital Twin) is at the core of the DT-Agro project, aiming to build an operational Digital Twin that simulates and monitors the functioning of agro-hydrological systems in Greece. This deliverable presents the work performed during months 1-14, focusing on the integration and evaluation of simulation algorithms that constitute the computational core of DT-Agro. The purpose of this deliverable is to document, validate and deliver all algorithms developed and integrated within the core of DT-Agro. This Digital Twin is developed under the AgroHydroLogos model, a spatially distributed agro-hydrological framework. During the reporting period, the team adapted and improved algorithms for the simulation of the processes of the model. The main goals include the integration of algorithms to ensure computational efficiency, the adaptation of the modeling framework to be compatible with Earth Observation (EO) inputs, and the preparation of the model for integration into the Digital Twin environment to support dynamic agro-hydrological simulations at high spatial resolution. The model enables the ingestion of EO data (e.g. global soil data, AgERA5 meteorological parameters, Copernicus data) and other inputs. A full description of the algorithms and their implementation within the DT-Agro is included, marking the transition toward the complete version which will be reported in D2.2.

2. Overview of the DT-Agro framework

DT-Agro is being developed around the AgroHydroLogos modeling framework (Soulis and Dercas, 2007). The AgroHydroLogos model operates as an extension of the GIS software package ArcGIS (Environmental Systems Research Institute-ESRI, Redlands, CA, USA) making use of its geospatial analysis and spatial data management capabilities, which are critical for the needs of this study involving vast amounts of spatial input data and results. The model provides gridded outputs of the main hydrological balance components, vegetation water deficit and irrigation water needs, on a daily or monthly basis at the country level, yet it is simple and flexible, as its conceptual scheme draws on well-established but simplified methods for the simulation of the involved hydrological processes.

AgroHydroLogos is structured to simulate the soil–plant–atmosphere system, including hydrological, crop, soil, and energy balance components. Each grid cell represents a homogeneous unit defined by soil, land use, and topography, maintaining water and energy balance through physically based and empirical formulations. Its distributed architecture enables the estimation of infiltration, evapotranspiration, surface runoff, and percolation across complex terrain, making it suitable for regional-scale applications.

In the context of Work Package 2 (WP2) the model is designed for direct integration with EO and other spatial data sources. Datasets such as AgERA5 meteorological data, Copernicus NDVI, CLC Backbone, surface soil moisture, and EU-DEM provide model inputs for initialization, calibration, and state updating. This ensures a two-way data flow, where EO data not only drive the model but also update state variables dynamically.

Several improvements were implemented within WP2 to transform AgroHydroLogos into the DT-Agro Digital Twin, with emphasis on computational efficiency, openness and interoperability. The core simulation engine was fully recoded in C++ to substantially increase performance and to enable execution on multiple platforms (desktop, HPC and cloud environments). This redesign builds on and further enhances key features of AgroHydroLogos, including the serialization algorithm (Soulis, 2013) and the dynamic spatial interpolation of meteorological data (Soulis et al., 2020), and adopts a dual spatial resolution scheme (a coarser 1000 m grid for meteorological forcings and a finer 100 m grid for agro-hydrological components). In addition, the parallelisation of the algorithms and an improved serialization procedure allow the independent processing of hydrologically separate regions, so that smaller spatial units can be executed for calibration experiments or real-time update operations.

To support the transition from a standalone model to a fully functional Digital Twin, the DT-Agro architecture was redesigned to facilitate the integration of multiple dynamic input data streams (e.g. meteorological fields, soil moisture, vegetation dynamics, land-cover changes) from remote sensing products and global platforms. In parallel, the graphical user interface was migrated from the proprietary ArcGIS/ArcMap environment to an independent, open-source implementation based on Python. The connection with GIS functionality is maintained through *Rasterio* and related geospatial libraries, ensuring efficient handling of raster data while keeping the DT-Agro platform completely open, easier to adapt to new requirements, and straightforward to integrate into other platforms and operational workflows.

Finally, a pilot application at farm-parcel scale is being used for model calibration and validation, based on the collection of existing data from databases and local irrigation organisations on water consumption, crop production, hydrology, and soil properties. These data are used in parallel with the EO data and are also utilized for EO data evaluation. The resulting data form the basis for subsequent analysis and service design under WP5, supporting the interpretation and operational use of DT-Agro outputs.

Greece 2.0 NATIONAL RECOVERY AND RESILIENCE PLAN
Funded by the European Union NextGenerationEU

3. Description of Developed Algorithms

The development of DT-Agro requires the evaluation, adaptation, and integration of simplified but robust algorithms within the AgroHydroLogos model to simulate the key processes of the soil–plant–atmosphere system. The goal is to ensure that each process is computationally efficient, compatible with EO inputs, and capable of producing spatially distributed results.

3.1 Hydrological Algorithms

The hydrological algorithms of DT-Agro form the core of the Digital Twin, representing the main surface and subsurface processes of the soil–plant–atmosphere system in a physically consistent but computationally simplified way. They are derived from, and extend, the AgroHydroLogos modelling framework, which has already been applied at national scale in Greece for the assessment of water balance components, crop water deficits and irrigation water requirements.

The model operates on a fully distributed grid, where each cell represents a homogeneous unit defined by soil, land cover and topography. A daily water balance is solved for a reference soil volume bounded by the rooting depth, providing spatially distributed estimates of:

- infiltration and surface runoff,
- soil water storage and deep percolation,
- baseflow,
- reference and actual evapotranspiration, and
- crop water deficit and irrigation water requirements.

All fluxes are computed in a way that is compatible with EO-derived inputs (e.g. NDVI, imperviousness, land cover) and with the dual spatial resolution design of DT-Agro, which uses a coarser grid for meteorological forcings (e.g. target 1000 m) and a finer grid (e.g. target 100 m) for agro-hydrological processes. Initial applications use the same grid for simplicity and the dual-grid approach is developed to increase computational efficiency.

3.1.1 Reference soil volume and water balance

The conceptual scheme is based on a reference soil volume representing the root zone of each grid cell. The water content of this volume controls most processes of the water balance (actual evapotranspiration, surface runoff, infiltration, deep percolation) as well as irrigation demand.

For each day, the change in soil water storage is computed as:

- **Inputs:** precipitation, irrigation, and (where relevant) snowmelt;

- **Losses:** surface runoff, actual evapotranspiration, and deep percolation below the root zone.

The water balance is solved sequentially, starting from interception and surface runoff generation, followed by infiltration, soil-moisture updating, deep percolation and baseflow, and finally evapotranspiration and irrigation demand.

The water balance equation is:

$$RSWC_i = RSWC_{i-1} + P_i + IRR_i - Q_i - aET_i - DI_i \quad (1)$$

where $RSWC_i$ and $RSWC_{i-1}$ (mm) are the water contents of the reference soil volume of the current and the previous time step, respectively, P_i (mm) is the total precipitation, IRR_i is the irrigation water depth, Q_i (mm) is the direct runoff depth, αET_i (mm) is the actual evapotranspiration depth, and DI_i (mm) is the deep infiltration depth at the current time step. The reference soil volume is defined as the topsoil layer, which is limited by the rooting depth. For simplicity, the interception storage is also included in *RSWC*.

Soil water content is the main initial condition for the application of the model, but its determination is very difficult due to the very high spatial and temporal variability. To address this issue, a warm-up period at the beginning of the model application is required, during which soil water content is regulated during very dry summer periods or very wet winter periods at the various locations of the modelled area, when *RSWC* gets close to zero or equal to the maximum soil water holding capacity of the root zone, respectively.

An additional feature of the application is that the initial moisture is set to a rational value depending on the day of the year and the soil hydraulic properties to reduce the length of the required warm-up period. In future developments, the model will be able to receive, as an input, the initial soil moisture predicted by remote sensing or other methods.

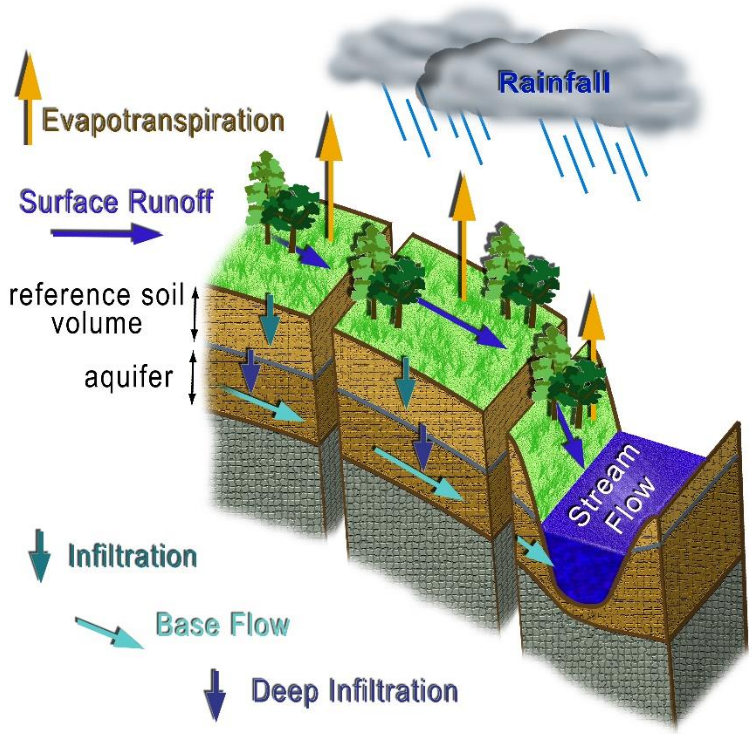

Figure 1. Schematic representation of the interactions determining the water balance of the reference soil volume.

In this scheme (Figure 1), the simulation of the main hydrological processes is based on simplified but well-established methods requiring a minimum number of parameters and relying on readily available or easy-to-grasp spatial data.

3.1.2 Surface runoff and infiltration (SCS-CN with explicit impervious areas)

Surface runoff is calculated using the Soil Conservation Service - Curve Number (SCS-CN) method, which provides a robust and parsimonious description of direct runoff volumes based on land-cover, hydrologic soil group and antecedent wetness. In DT-Agro, the SCS-CN implementation has been extended along two directions:

1. **Dynamic adjustment to soil moisture**, and
2. **Explicit consideration of impervious areas within each grid cell**, using a new simplified method developed under DT-Agro (Palli-Gravani et al. 2025).

Base CN for the pervious fraction

For each grid cell, a base CN value representing the pervious part of the cell is assigned from a knowledge base that combines:

- soil information (from ISRIC / ESDAC and national soil maps),
- land-cover and crop information from Copernicus products (e.g. CLC / CLC Backbone, HRL) and

- the IACS database, which provides detailed crop and management data at parcel level.

These datasets are harmonised and resampled to the 100 m modelling grid, and the resulting "pervious CN" is fully compatible with the standard SCS-CN documentation.

Impervious fraction and simplified Two-CN formulation

To represent the effect of impervious surfaces on runoff generation in a simple but physically meaningful way, DT-Agro adopts the new simplified method proposed by Palli-Gravani et al. (2025), which explicitly separates each spatial unit into an impervious and a pervious part.

In the Digital Twin, this concept is generalised to the grid-cell scale:

- the **fraction of impervious area** in each 100 m cell is derived from the Copernicus Land Monitoring Service (Imperviousness Density 2021 and related layers), aggregated to the model grid;
- the remaining fraction is treated as pervious and characterised by the pervious CN described above.

Following the simplified Two-CN formulation, the runoff response of each cell is represented as the combination of:

- an impervious sub-area with CN = 100, and
- a pervious sub-area with CN = CN_pervious,

with a common initial abstraction ratio λ consistent with SCS-CN documentation.

The effective cell-scale CN is thus a function of both the pervious CN and the impervious fraction, allowing the model to reproduce the strong influence of even small impervious patches on runoff generation, particularly in otherwise highly permeable areas. This approach has been shown to provide performance comparable to the more complex Two-CN method, while remaining simple, parsimonious and fully compatible with the original SCS-CN framework.

Dynamic link to soil moisture

In addition to the explicit treatment of impervious areas, DT-Agro links the CN of the pervious part to the simulated soil moisture of the reference soil volume:

- under dry conditions, CN is reduced towards a lower limit,
- under wet conditions, CN increases towards an upper limit,

thus moving from discrete AMC classes to a continuous soil-moisture-driven runoff response. This strategy, already tested in previous national-scale AgroHydroLogos applications, improves the representation of storm-to-storm variability while preserving compatibility with the SCS-CN method.

Infiltration

For each time step, infiltration is implicitly obtained as the difference between effective precipitation (after interception and initial abstraction) and the simulated surface runoff, subject to the local water holding capacity of the reference soil volume. When the soil water content approaches saturation, excess water is routed to deep percolation; under dry conditions, infiltration predominantly replenishes the soil store.

3.1.3 Soil moisture, deep infiltration and baseflow

Soil water storage in the reference volume is updated using the water balance equation, with volumetric soil-moisture bounds defined by field capacity and wilting point. Soil hydraulic properties (porosity, field capacity, wilting point, saturated hydraulic conductivity) are defined based on available soil datasets and literature values.

i. Infiltration below the rooting depth is represented through a Brooks–Corey-type relationship linking unsaturated hydraulic conductivity to soil moisture, with a free-drainage lower boundary. Depending on the amount of the initial water content, there are three options: **Initial water content lower than saturation:** When the initial water content is lower than the water content at saturation, the soil hydraulic conductivity is given by Equation (2):

$$K = K_s \left(\frac{\theta}{\theta_s}\right)^{\frac{1}{n}} \quad (2)$$

where K is the soil hydraulic conductivity at a time t, K_s is the saturated hydraulic conductivity, θ is the water content at a time t, θ_s is the soil water content at saturation and n is a shape factor. The change in soil water content over time is therefore expressed as (Equation 3).

$$\frac{d\theta}{dt} = K_s \left(\frac{\theta}{\theta_s}\right)^{\frac{1}{n}} \quad (3)$$

Solving this equation with initial value $\theta(0) = WTC$, where WTC is the initial water content, yields:

$$\theta(t) = \left(WTC^{1-b} + \frac{K_s}{\theta_s^b}(b-1)t\right)^{\frac{1}{1-b}} \quad (4)$$

For a daily time step, the soil water content at the end of the day is: $\theta(D) =$

$$\left(WTC^{1-b} + \frac{K_s}{\theta_s^b}(b-1)D\right)^{\frac{1}{1-b}} \quad (5)$$

where $b = 1/n$, D is the duration of a day.

ii. **Initial water content exceeding saturation but decreasing below saturation during the day:** When the initial water content exceeds saturation but falls below saturation within the daily time step, water movement occurs in two successive phases. During the first phase, free water drains through the soil profile at a constant velocity equal to the saturated hydraulic conductivity K_s. This phase continues until the total water content at the grid cell reaches soil water content

Greece 2.0 NATIONAL RECOVERY AND RESILIENCE PLAN
Funded by the European Union NextGenerationEU

at saturation. The time at which this occurs, t_0, is defined by the condition $\theta(t_0) = \theta_s$ and is given by: $t_0 = \frac{WTC-\theta_s}{K_s}$, where WTC is the initial water content and θ_s is the soil water content at saturation.
After this free-drainage phase, soil water movement is governed by unsaturated flow. In this phase, the temporal evolution of soil water content follows the same unsaturated hydraulic conductivity relationship used in the previous case (Equations 2 &3).

Equation (3) is a differential equation of θ. Solving this differential equation with initial value $\theta(t_0) = \theta_s$ yields the following expressions for soil water content θ:
For the fee drainage phase:

$$\theta(t) = WTC - K_s t, 0 \leq t \leq t_0 \quad (6)$$

For the unsaturated flow phase:

$$\theta(t) = \left(\theta_s^{1-b} + \frac{K_s}{\theta_s^b}(b-1)(t-t_0)\right)^{\frac{1}{1-b}}, t_0 < t \leq \mathrm{D} \quad (7)$$

where $b = 1/n$ and D is the duration of the daily time step.
The soil water content at the end of the day is therefore given by:

$$\theta(D) = \left(\theta_s^{1-b} + \frac{K_s}{\theta_s^b}(b-1)(D-t_0)\right)^{\frac{1}{1-b}} \quad (8)$$

iii. **Initial total water content higher than saturation throughout the day.** When the initial total water content exceeds soil water content at saturation and remains above saturation for the entire duration of the daily time step, water movement is dominated by free drainage. In this case, excess water moves through the soil profile at a constant velocity equal to the saturated hydraulic conductivity K_s, and no transition to unsaturated flow occurs during the day.
Under these conditions, the temporal evolution of soil water content in the reference soil volume is described by a linear decrease:

$$\theta(t) = WTC - K_s t \quad (9)$$

For a daily time step of duration D, the soil water content at the end of the day is given by: $\theta(D) = WTC - K_s D$ (10)
Deep infiltration is then computed as the difference between the initial and final soil water contents: $DI = WTC - \theta(D)$ (11)
or equivalently:

$$DI = \begin{cases} WTC - \left(WTC^{1-b} + \frac{K_s}{\theta_s^b}(b-1)D\right)^{\frac{1}{1-b}}, WTC \leq \theta_s \\ WTC - \left(\theta_s^{1-b} + \frac{K_s}{\theta_s^b}(b-1)\left(D - \frac{WTC-\theta_s}{K_s}\right)\right)^{\frac{1}{1-b}}, \theta_s < WTC < K_s D + \theta_s \\ K_s D, WTC \geq K_s D + \theta_s \end{cases} \quad (12)$$

Deep infiltration feeds a conceptual groundwater store at the cell or sub-basin scale, from which baseflow is simulated using simple linear recession equations following earlier AgroHydroLogos applications. Groundwater storage is updated daily and produces baseflow as long as storage remains above a defined threshold, allowing the model to reproduce delayed contributions to streamflow with limited computational cost.

3.1.4 Evapotranspiration and crop water use

Actual evapotranspiration (aET) is computed in two steps:

1. **Reference evapotranspiration (ETo)** is estimated using the FAO Penman–Monteith method from daily meteorological data (temperature, humidity, wind speed and radiation).
2. **Potential crop evapotranspiration (ETp)** is obtained by multiplying ETo by a crop coefficient (Kc), which reflects canopy and phenological characteristics.

Kc values are assigned from a knowledge base for each land-cover / crop type and are dynamically adjusted using EO-derived vegetation indicators (e.g. NDVI from Copernicus CLMS), capturing intra-seasonal variations in crop development and canopy cover.

The transition from ETp to **aET** is governed by a soil-moisture-dependent stress coefficient (Kst):

- when soil water in the reference volume is above a crop-specific threshold, Kst ≈ 1 and aET ≈ ETp;
- under water-deficit conditions, Kst reduces evapotranspitation proportionally to the degree of soil water depletion.

This formulation couples atmospheric demand with soil water availability and is essential for representing agricultural drought and irrigation water needs.

3.1.5 Irrigation water requirements

For irrigated crops, DT-Agro estimates **net irrigation requirements** using two approaches:

1. as the daily difference between a target evapotranspiration under optimal irrigation and the simulated aET under rain-fed conditions. The target is expressed as a fraction Sc of ETp (typically close to 0.9 for full irrigation, but adjustable to represent deficit-irrigation strategies). This approach calculates the water deficit for irrigated crops as well as for rainfed crops and natural vegetation, allowing a consistent estimation of irrigation needs across land-cover types.
2. Irrigation management is simulated considering standard soil water content thresholds to initiate and end an irrigation event. The thresholds are defined based on

the crop and the irrigation management practices. The irrigation water is added to the reference soil volume at the end of the day.

At the grid-cell level, these net requirements are then combined with information on irrigation systems and water sources (from IACS and other databases) to derive gross abstractions by applying typical losses for distribution and on-farm applications. These post-processing steps are performed in a GIS environment.

3.1.6 Runoff routing and spatial aggregation

Surface runoff and baseflow generated at each grid cell are routed along a predefined drainage network derived from the Copernicus Digitlal Elevation Model (DEM) and related products. Local slopes, upstream drainage areas and land-cover-dependent roughness coefficients are used to estimate flow velocities for overland and channel flow (Manning-type formulations). Travel times are accumulated along the flow paths, enabling the computation of daily discharges at arbitrary points of the drainage system with limited computational overhead (Soulis 2013).

This routing scheme is adequate for the daily time step and the regional-scale objectives of DT-Agro; it does not aim to reproducing sub-daily flood hydrographs but rather at ensuring mass balance consistency and realistic timing of flows.

3.1.7 Meteorological forcing

Meteorological inputs (precipitation, temperature, radiation, humidity, wind speed) are obtained through the hybrid approach described in Section 3.4, which combines historical observations from approximately 140 meteorological stations with AgERA5 reanalysis to produce bias-corrected “virtual stations” time series at each station location. These virtual stations provide continuous, homogenised daily data from the start of the AgERA5 record up to the present, even where the original station records contain large gaps.

To capture the strong spatial variability of climate in Greece while controlling storage and CPU requirements, DT-Agro adopts a dynamic spatial interpolation scheme:

- for each cell of the meteorological grid, the weights of the available stations are pre-computed using inverse-distance weighting with topographic adjustments;
- during model execution, daily fields are obtained by applying these weights to the time series, instead of re-interpolating the full spatial field at each time step.

This approach has negligible additional computational cost compared to purely lumped forcings, while preserving the spatial structure of meteorological variables that drive the water balance.

3.1.8 Role within the Digital Twin

Overall, the hydrological algorithms of DT-Agro preserve the conceptual structure of AgroHydroLogos but have been re-implemented and extended to:

- operate efficiently on a dual-resolution grid with C++ and parallel execution,
- exploit EO-derived datasets (NDVI, imperviousness, land cover, soil moisture, DEM) for parameterisation and state updating, and
- incorporate the new impervious-aware SCS-CN formulation for each grid cell, supporting a seamless transition from a conventional model to a fully functional, EO-driven Digital Twin of the Greek Agro-Hydro-System.

3.2 Crop Growth

Crop growth in DT-Agro is represented through a water-driven approach consistent with FAO guidelines on crop evapotranspiration and yield response to water. In this framework, the crop development and water stress are not simulated through a detailed physiological growth model, but are inferred from the balance between potential and actual evapotranspiration.

For each grid cell and day, potential crop evapotranspiration (ETp) is calculated from the reference evapotranspiration (ETo) using crop-specific coefficients *Kc* that vary with crop type and phenological stage, following FAO-type Kc curves. Actual evapotranspiration (ETa) is then derived by applying a water-stress coefficient *Kst* that depends on the soil water content in the reference soil volume.

The ratio ETa/ETp is used as the central indicator of crop water satisfaction and, consequently, crop growth conditions. Values close to 1 indicate that water requirements are fully met and no significant water stress is expected, while persistent reductions in ETa relative to ETp (ETa/ETp < 1) indicate increasing levels of water deficit and reduced growth. For irrigated crops, irrigation scheduling in the model is designed to keep ETa close to ETp, within an acceptable reduction threshold (e.g. ETa/ETp $\gtrsim$ 0.9) that is consistent with FAO recommendations for avoiding significant yield reductions.

Over the growing season, the model aggregates the ETa/ETp ratio over relevant growth stages to derive crop water stress indices and to approximate the relative impact of water deficits on crop performance. In this way, crop growth and yield response are directly linked to the integrated water deficit signal, expressed through ETa/ETp, rather than through a separate, empirical yield model. This formulation allows the Digital Twin to remain computationally efficient and transparent, while still being fully compatible with FAO-type analyses of yield response to water availability.

The crop-related parameters required by this module (planting dates, rooting depths, Kc curves, sensitivity to water stress by growth stage) are stored in the model's knowledge base for all major crops. Within the DT-Agro framework, these parameters are updated or refined using Earth Observation products, for example by adjusting crop coefficients or growth stages based on NDVI time series from Copernicus. This enables a consistent, FAO-based description of crop growth that is dynamically constrained by both the simulated soil–water balance and EO-derived vegetation signals.

This part of DT-Agro is currently under active development.

3.3 Soil Erosion

The soil-erosion component of DT-Agro provides spatially distributed estimates of potential soil loss and sediment delivery to the drainage network, consistent with the hydrological algorithms and grid-based structure of the Digital Twin. Soil loss is estimated using a RUSLE-type formulation applied at the 100 m modelling grid, followed by a sediment-delivery and routing scheme that quantifies the fraction of the eroded material transported downslope and reaching the stream network.

3.3.1 RUSLE-based estimation of gross soil loss

Gross soil loss in each 100 m grid cell is computed with a RUSLE-type equation:

$$A = R \cdot K \cdot LS \cdot C \cdot P \tag{13}$$

where:

- A is the long-term average soil loss
- R is the rainfall erosivity factor,
- K is the soil erodibility factor,
- LS is the topographic factor (slope length and steepness),
- C is the cover–management factor, and
- P is the support practice factor.

All factors are derived on the same grid to match the spatial resolution of the agro-hydrological core:

- **The R factor** is taken from climatological rainfall erosivity maps (or derived from long-term rainfall data) and represents the average erosive power of rainfall at each cell.
- **The K factor** is computed from harmonised soil information (global and national soil maps), using soil texture, organic matter and related properties.
- **The LS factor** is calculated from the DEM and the derived flow accumulation grid, following standard RUSLE formulations: longer and steeper hillslopes yield larger LS values.

- **The C factor** depends on land cover and crop type, using Copernicus land-cover products together with IACS information on crops and management. Within the Digital Twin, C can evolve in time based on EO-derived vegetation indices (e.g. NDVI) to represent seasonal changes in canopy cover.
- **The P factor** accounts for conservation practices (contouring, terracing, strip cropping, etc.), using available information from IACS, regional databases or scenario assumptions.

In this way, RUSLE provides a consistent estimate of **potential soil loss** for each cell, driven by local rainfall, soil, topography and land management.

3.3.2 Temporal disaggregation of erosivity using daily rainfall and irrigation

To represent the temporal variability of erosion and to make the erosion module consistent with the daily water-balance simulation, DT-Agro disaggregates the long-term erosivity factor R into daily erosivity values that depend on the actual daily rainfall and, where relevant, on irrigation events.

For each cell and year, a daily erosivity proxy $E_{\text{raw}}(t)$ is defined as a function of the daily rainfall depth $P_{\text{day}}(t)$and the applied irrigation depth $I_{\text{day}}(t)$:

$$E_{\text{raw}}(t) = \begin{cases} \left[P_{\text{day}}(t) + \gamma_I \, I_{\text{day}}(t)\right]^{\beta}, & \text{if } P_{\text{day}}(t) + \gamma_I \, I_{\text{day}}(t) > P_0 \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where:

- P_0 is a small threshold (e.g. 3–5 mm) below which rainfall–irrigation events are considered non-erosive,
- β is an exponent (typically between 1 and 2) that reflects the non-linear increase of erosivity with event depth,
- γ_I is a dimensionless coefficient that represents the relative erosive efficiency of irrigation compared to natural rainfall.

The parameter $\gamma_I$allows the model to distinguish between different irrigation systems:

- $\gamma_I \approx 1$ for sprinkler irrigation on bare or sparsely covered soils, which behaves similarly to rainfall,
- intermediate values (e.g. 0.3–0.7) for surface or furrow irrigation on sloping fields, where runoff along furrows can detach and transport soil,
- $\gamma_I \approx 0$ for drip/micro-irrigation, where the erosive effect is negligible.

The raw daily erosivity values are then normalised over the year (or growing season) to obtain daily erosivity fractions:

$$f(t) = \frac{E_{\text{raw}}(t)}{\sum_t E_{\text{raw}}(t)} \tag{15}$$

Finally, the daily R factor for each day t is defined as:

$$R_{\text{day}}(t) = R_{\text{clim}} \cdot f(t) \tag{16}$$

where R_{clim} is the climatological R factor at that cell.

By construction, the daily values satisfy:

$$\sum_{t \in \text{year}} R_{\text{day}}(t) = R_{\text{clim}} \tag{17}$$

so the annual erosivity is consistent with the long-term RUSLE factor, while its intra-annual distribution is controlled by the actual sequence of rainfall and irrigation events. Days with intense rainfall or irrigation on erodible, sloping and poorly covered fields therefore contribute more to annual soil loss than days with small events.

Daily soil loss at cell $(i, j)$is then computed as:

$$A_{\text{day}}(i,j,t) = R_{\text{day}}(i,j,t) \cdot K(i,j) \cdot LS(i,j) \cdot C(i,j,t) \cdot P(i,j,t) \tag{18}$$

where C and P may also vary in time (e.g. due to changing crop cover or conservation practices).

This formulation allows DT-Agro to:

- remain compatible with the standard RUSLE concept of a long-term R factor,
- explicitly link daily erosion risk to the simulated daily rainfall and irrigation, and
- analyse the timing of erosion events within the season.

3.3.3 Sediment delivery ratio and grid-based routing

RUSLE provides gross soil loss at the cell scale, but only a fraction of this material reaches the stream network. To represent this, DT-Agro uses a sediment delivery ratio (SDR) for each grid cell, which expresses the proportion of eroded soil that is delivered to the next downstream cell.

The SDR is defined as a function of terrain and connectivity characteristics, such as:

- distance or travel time to the nearest stream,
- local slope and flow accumulation,
- presence of depressions or flat areas favouring deposition,
- land cover and surface roughness along the flow path.

Cells that are directly connected to the channel network by short, steep flow paths have SDR values close to 1, while cells on long, gentle or poorly connected hillslopes have lower SDR values.

For each day and each cell, the delivered sediment is computed as:

$$S_{\text{delivered}}(i,j,t) = A_{\text{day}}(i,j,t) \cdot SDR(i,j) \quad (19)$$

The delivered sediment is then routed downstream along the same flow directions and travel-time structure used by the hydrological module:

- each cell passes its delivered sediment to the next downstream cell,
- optional simple deposition or decay terms are applied during routing,
- at selected nodes (e.g. river reaches, outlets, reservoirs) the model accumulates daily sediment loads.

This results in a fully distributed, grid-based sediment-routing scheme, which is implemented directly on the grid and tightly coupled to the Digital Twin hydrology and topography.

3.3.4 Integration with the Digital Twin

The erosion and sediment-delivery module is integrated within DT-Agro's spatial and EO-driven framework:

- All RUSLE factors (R, K, LS, C, P) are derived from the same DEM, land-cover, soil and management datasets used by the hydrological core.
- The temporal disaggregation of R relies on the same daily rainfall and irrigation series used by the water-balance algorithms.
- Land-cover–dependent factors (especially C) are updated in near real time using EO-derived vegetation indices, allowing the Digital Twin to reflect actual crop development and management changes.

In this way, DT-Agro quantifies both on-site soil degradation (long-term soil loss at cell level) and off-site impacts (sediment export and loads in the river network), and assesses how changes in climate, irrigation management, crop patterns and conservation practices modify erosion and sediment transport across scales.

This part of DT-Agro is currently under active development.

3.4 EO Data Acquisition and Pre-Processing Algorithms

The DT-Agro Digital Twin relies on a continuous and automated flow of Earth Observation (EO) and other geospatial data to drive simulations, initialise model states, and periodically update key variables. In WP2, a series of Python-based algorithms and processing routines

Greece 2.0 NATIONAL RECOVERY AND RESILIENCE PLAN Funded by the European Union NextGenerationEU

were developed to implement this EO data pipeline. These routines handle the **acquisition, decoding, reprojection, resampling, quality control and formatting** of EO products and ancillary datasets from multiple sources, transforming them into inputs that are directly compatible with the DT-Agro modelling core.

The EO pipeline is designed to be **modular, reproducible and re-runnable**, so that exactly the same code can be used for historical reprocessing, near-real-time updates and future extensions of the Digital Twin.

3.4.1 Overall workflow and design principles

The EO data workflow is implemented as a set of modular Python scripts organised by data source and product type (meteorological forcing, vegetation indices, land cover and imperviousness, soil properties, topography, surface soil moisture, etc.). Each module:

- connects to the relevant data service or repository (e.g. Copernicus Climate Data Store, Copernicus Land Monitoring Service, ESA hubs, ISRIC/ESDAC, national datasets),
- downloads the requested data for a specified spatial and temporal window,
- performs product-specific decoding (unzip, read NetCDF4/GRIB/GeoTIFF, apply scale factors and quality flags),
- reprojects and resamples all layers to the DT-Agro target grid (EGSA87 / EPSG:2100; 100 m cell size for most static and agro-hydrological variables; 1 km for meteorological forcing), and
- exports the final datasets in standardised formats (GeoTIFF for rasters, NetCDF/xarray for time-varying grids, CSV/Parquet for time series).

The workflow uses open-source libraries such as xarray, netCDF4, rasterio/rioxarray, pandas and geopandas, and stores intermediate products in a structured directory hierarchy with consistent naming conventions (dataset, variable, year, version). This ensures traceability and facilitates integration with the DT-Agro model configuration.

3.4.2 Meteorological data: station database, AgERA5 sampling and bias correction

For the meteorological forcing of DT-Agro, a hybrid strategy is adopted that combines:

- historical observations from a curated database of approximately 140 meteorological stations across Greece, and
- AgERA5 reanalysis data, which provide spatially and temporally complete fields.

The station database includes daily records of key variables (precipitation, minimum and maximum temperature, relative humidity, wind speed, solar radiation). These in situ data are of high quality but exhibit important limitations:

- many stations have large temporal gaps (several missing years),
- the time coverage is heterogeneous across stations, and
- real-time access is not yet available for most of them.

At the same time, previous studies and additional analyses carried out in WP2 have shown that AgERA5 (and similar global reanalyses) can display substantial errors and systematic biases when used directly for local agricultural applications in complex Mediterranean terrain. This is particularly evident in test areas in Greece, such as Nemea, where raw reanalysis data do not reproduce station measurements with sufficient accuracy for irrigation and crop-water studies.

To address these issues, DT-Agro uses a station-wise regression and bias-correction approach, producing "virtual" meteorological stations at the locations of the 140 real stations:

1. Sampling AgERA5 at station locations
    - For each of the 140 stations, complete daily time series of the required variables are extracted from AgERA5, from the start of the AgERA5 record up to the present date.
    - This extraction is performed directly at the coordinates of each station (point sampling), using the Copernicus Climate Data Store (CDS) API and automated Python routines.
2. Regression-based bias correction
    - For each station and each variable, periods with available observed station data are used to develop regression relationships between the observations and the corresponding AgERA5 time series.
    - Typically, simple linear models of the form

$$X_{\text{obs}} = a + b\, X_{\text{AgERA5}} \quad (20)$$

are fitted, with coefficients a and b estimated separately for each station and variable. More complex transformations can be introduced where needed, but the baseline implementation remains parsimonious and transparent.

    - These calibrated regressions are then applied to the full AgERA5 time series at each station location (including periods without observations), yielding bias-corrected daily series that inherit the temporal completeness and continuity of AgERA5, but are statistically adjusted to match the local station climatology.
3. Virtual stations and storage
    - The resulting corrected series are stored as "virtual-station" records, one per real station and variable.

Greece 2.0 NATIONAL RECOVERY AND RESILIENCE PLAN Funded by the European Union NextGenerationEU

  - These virtual stations act as continuous, homogenised forcing series for the model and can be used exactly like real station data in the subsequent spatial interpolation step (Section 3.1).

In the current phase of DT-Agro, these 140 virtual stations form the backbone of the dynamic interpolation scheme that generates gridded meteorological forcing on the 1 km model grid (see Section 3.1). The same framework can accommodate additional stations: as more historical station datasets are processed and new stations become available, new virtual stations can be created or existing ones recalibrated, progressively increasing the density and robustness of the forcing network without modifying the model core.

The raw AgERA5 gridded products are still archived for reference, diagnostics and potential use in ungauged regions, but all operational DT-Agro simulations use the bias-corrected virtual-station time series as input to the interpolation procedure.

3.4.3 Vegetation indices and crop-condition indicators

Dynamic vegetation information is a key driver of the DT-Agro crop and evapotranspiration modules. A dedicated workflow has been implemented for NDVI from the Copernicus Land Monitoring Service (CLMS), with provisions for extention to additional vegetation indices (e.g. LAI, FAPAR) in subsequent phases.

1. CLMS NDVI acquisition and decoding
   - NDVI products are downloaded as global or pan-European NetCDF4 files through automated Python scripts.
   - The routines loop over the required dates, spatial extent (Greek territory plus buffer) and product tiles.
   - For each file, NDVI values and the associated quality flags are read.
2. Scaling and quality control
   - NDVI values are converted from digital numbers to physical values according to the product definition, for example:

$$NDVI_{\text{real}} = DN \times 0.004 - 0.08 \quad (21)$$

   - Quality flags are stored and used to mask invalid or low-confidence pixels (e.g. clouds, snow, missing data). Only high-quality pixels contribute to subsequent compositing and statistics.
3. Reprojection, resampling and export
   - NDVI rasters are clipped to the Greek territory, reprojected to EGSA87 (EPSG:2100) and resampled to the 100 m DT-Agro grid (typically with bilinear interpolation).

  - The preprocessed NDVI stacks are exported either as:
    - multi-band GeoTIFFs (one band per date), or
    - netCDF/xarray datasets with dimensions (x, y, time).
4. Link to the modelling core
  - Time series of NDVI at grid-cell or parcel level are used to derive NDVI-based crop coefficients (Kc) and cover factors (C), which are ingested by the crop growth, evapotranspiration and erosion modules.
  - The approach is consistent with previous studies where Sentinel-2 and other satellite imagery were used to validate crop declarations and monitor crop conditions in Greece, and represents the operational implementation of those methods within the Digital Twin.

3.4.4 Land cover, imperviousness and static geospatial layers

A second group of EO and geospatial products provides the static or slowly varying spatial context for the simulations: land cover, impervious surfaces, soil properties and topography. These layers are critical for the parameterisation of hydrological and erosion processes and for the delineation of model units.

1. Land cover and imperviousness
  - Land-cover data are obtained from CLC Backbone and other Copernicus Land Monitoring Service products and are complemented by IACS information in agricultural areas, which provides field-level crop and management details.
  - High Resolution Layer (HRL) Imperviousness products are used to quantify the fraction of impervious surfaces in each 100 m grid cell. These fractions are directly used in the SCS-CN parameterisation and in the new runoff-generation method that explicitly integrates impervious areas at cell level (see Section 3.1).
  - Land-cover and imperviousness datasets are:
    - clipped to the Greek territory,
    - reprojected to EGSA87, and
    - resampled to 100 m using appropriate methods (majority or mode for categorical land cover; area-weighted aggregation for imperviousness density).
2. Soil properties

  - Soil information from ISRIC, ESDAC and national soil maps is preprocessed to derive the parameters required by the DT-Agro model, such as texture classes, organic carbon, hydraulic properties and erodibility factors.
  - Raster layers are generated for each key variable and resampled to the 100 m grid.
  - These rasters feed directly into parameter estimation routines for:
    - SCS Curve Number (CN) and the separation of pervious and impervious contributions,
    - rooting-zone water-holding capacity, and
    - the K factor in the RUSLE-based erosion module.

3. Topography (DEM and derivatives)
  - Topographic information is derived from the Copernicus DEM / EU-DEM.
  - The DEM is clipped to the Greek territory, reprojected to EGSA87, and resampled to 100 m.
  - From this DEM, a set of derivatives is computed: slope, aspect, flow direction, flow accumulation and channel-network masks.
  - These derivatives are used by:
    - the hydrological routing and water-balance algorithms (Section 3.1), and
    - the LS factor and connectivity indicators in the soil-erosion module (Section 3.3).

All static rasters are stored in a harmonised, project-wide geodatabase, ensuring that every component of DT-Agro uses consistent representations of land cover, soil and topography.

3.4.5 Surface soil moisture and auxiliary EO products

To support model initialisation, calibration and state updating, the EO pipeline also includes routines for surface soil moisture and other auxiliary products, which are progressively integrated within DT-Agro.

- Surface soil moisture products from Copernicus and other global EO datasets are downloaded, reprojected to EGSA87 and resampled to either 1 km or 100 m, depending on their native resolution and uncertainty.
- These products are used to:
  - evaluate simulated soil moisture patterns at selected dates, and

  - explore simple state-updating strategies (for example, bias correction of upper-layer soil moisture prior to the start of the irrigation season).
- Similar routines are implemented for additional EO layers (e.g. snow cover, LAI/FAPAR), allowing their incorporation as the scope of the Digital Twin expands.

3.4.6 Harmonisation, quality control and integration with DT-Agro

Across all EO and ancillary datasets, particular attention is given to harmonisation, quality control and traceability, so that the modelling core can treat the data in a consistent and robust way:

- Common projection and resolution
  - All rasters are stored in EGSA87 (EPSG:2100).
  - The core modelling resolution is 100 m for agro-hydrological and erosion processes, and 1 km for meteorological forcing. Aggregations or disaggregations between these resolutions are performed using methods appropriate to each variable type (e.g. conservative resampling for precipitation, bilinear for temperature, majority for categorical land cover).
- Quality masks and metadata
  - Quality flags from EO products (e.g. NDVI QC layers) are kept alongside the main variables and used to mask unreliable pixels.
  - Each processed dataset includes basic metadata (source product, version, processing date, scripts used), allowing full traceability and reproducibility of the EO pipeline.
- Automation and reproducibility
  - The workflows are designed to run unattended, enabling routine updates of the EO and meteorological data archive.
  - All steps from download to final raster/CSV/NetCDF output are scripted, avoiding manual GIS operations and ensuring that datasets can be regenerated if new versions or corrections become available.

Together, these EO data acquisition and pre-processing algorithms form the data backbone of the DT-Agro Digital Twin. They provide a robust, scalable and fully documented pipeline from external EO and climate services to the internal data structures used by the hydrological, crop and erosion modules described in Sections 3.1–3.3, and they prepare the ground for future near-real-time and data-assimilation capabilities developed in D2.2 and subsequent work packages.

4. Integration Framework

The DT-Agro Digital Twin is implemented as a modular system that couples the C++ simulation core with Python-based EO and data-handling modules within a coherent integration framework. This framework defines the module structure, interfaces and data flow between components, ensuring that EO data, meteorological forcing, model parameters and outputs are consistently exchanged and that the system can be extended and maintained in a transparent way.

The architecture is organised in three main layers:

1. **Data and EO layer** – responsible for acquiring, pre-processing and storing all external datasets (meteorological stations and AgERA5, Copernicus products, soil and DEM layers, IACS data).
2. **Model core layer** – implemented in C++, hosting the AgroHydroLogos-based simulation engine (hydrological balance, crop water use, soil erosion, routing) and the related parameterisation routines.
3. **Application and interface layer** – implemented in Python, providing configuration management, run control, input/output handling, and user-facing tools (scripts, GUI components, workflows).

The following subsections describe the main elements of this integration fremework.

4.1 System architecture and module structure

At the centre of DT-Agro lies the core hydrological–agricultural model, recoded in C++ and operating on the reference soil volume at 100 m resolution (Section 3.1). Around this core, a set of Python modules implements:

- EO data retrieval and pre-processing (Section 3.4),
- meteorological data retrieval, bias correction and interpolation,
- parameter derivation from static geospatial layers (soil, land cover, imperviousness, DEM),
- configuration parsing and simulation control, and
- output post-processing and export (e.g. aggregation by basin, administrative unit, crop type).

Each module exposes a clearly defined interface:

- input: file paths, variable names, temporal and spatial domain, configuration options;
- output: standardised rasters (GeoTIFF/NetCDF), time series (CSV/Parquet) or binary arrays ready to be passed to the C++ core.

The model executable is designed to be called from the Python layer with a configuration file describing:

- simulation period and time step,
- domain and grid definition,
- paths to meteorological and EO-driven inputs (virtual station fields, NDVI-based Kc, CN maps, RUSLE factors),
- active processes (hydrology, crop, erosion), and
- output options (variables to be saved, temporal aggregation, etc.).

This modular organisation enables DT-Agro to run in different contexts (development, calibration, operational mode) simply by changing configuration files and input directories, without modifying the core algorithms.

4.2 Integration of EO and model components

The EO data pipeline (Section 3.4) and the model core are connected through a set of well-defined intermediate products:

- Meteorological grids at 1 km derived from the virtual station network, created via regression-based bias correction of AgERA5 against historical station data. These grids provide the daily forcing fields (precipitation, temperature, radiation, etc.) required by the hydrological and crop modules.
- 100 m parameter rasters for soils (hydraulic properties, erodibility), land cover and imperviousness (for CN and C factors), and DEM derivatives (slope, flow accumulation, channel masks).
- Time-varying vegetation layers (NDVI and derived Kc and C factors) at 100 m resolution.

The Python integration layer:

1. checks that all required EO and parameter layers are available for the simulation domain and period,
2. arranges them into the file structure expected by the C++ model (e.g. one directory per variable, with standardised filenames), and
3. writes the corresponding configuration entries that map each process (runoff, evapotranspiration, erosion) to its required input datasets.

During execution, the C++ core reads these datasets in a streaming fashion (using the serialisation algorithm and regional parallelisation where applicable), updates the state variables of the Digital Twin and writes the requested outputs. The Python layer subsequently triggers additional post-processing steps (e.g. computing seasonal indicators, aggregating to basins, or exporting maps and time series).

4.3 Rainfall and temperature gradients, bias correction and interpolation

A novel component of the integration framework is the combined treatment of spatial gradients and bias correction for meteorological variables over Greece. Starting from the 140-station database and the corresponding AgERA5 series, DT-Agro:

- constructs bias-corrected "virtual stations" at each station location (Section 3.4.2), and
- uses these virtual stations to infer spatial patterns of rainfall and temperature across the country.

The interpolation and gradient algorithms are implemented in Python and can be summarised as follows:

- For each variable (e.g. precipitation, minimum/maximum temperature), long-term statistics and daily anomalies are computed for all virtual stations.
- Relationships between these statistics and physiographic factors (elevation, distance from the sea, large-scale climatic regions) are analysed to derive rainfall and temperature gradients in space and, where appropriate, in time (e.g. seasonal lapse rates).
- These gradients are then used within the dynamic interpolation scheme to refine the weighting of stations for each grid cell, particularly in areas with strong elevation or coastal contrasts.

The resulting procedure constitutes a novel approach and code base for:

- meteorological data retrieval (AgERA5 + stations),
- station-wise bias correction, and
- spatial interpolation that explicitly accounts for rain and temperature gradients across Greece.

All algorithms are encapsulated in reusable Python modules, so that the gradient fields and interpolation weights can be recomputed if new stations or updated EO datasets become available.

4.4 Interface, input–output handling and code organisation

The Python interface also acts as a **run manager** and I/O handler:

- **Input handling**

  - Reads user-defined configurations (YAML/JSON or similar).
  - Validates paths and variable names for EO rasters, meteorological time series and parameter files.
  - Prepares binary or NetCDF input sets in the format expected by the C++ engine.
- **Execution control**
  - Calls the C++ executable with appropriate arguments.
  - Manages parallel runs over multiple hydrologically independent regions, exploiting the serialisation and regionalisation capabilities of AgroHydroLogos/DT-Agro.
  - Logs run metadata (start/end time, configuration hash, code version).
- **Output handling**
  - Collects model outputs (e.g. daily soil moisture, ET, runoff, erosion indicators) from temporary directories.
  - Converts them into user-friendly formats (GeoTIFF maps, CSV time series), and, where needed, aggregates them to administrative units, river basins or irrigation districts.

The codebase is organised into clearly labelled folders (e.g. eo_preprocessing, meteo_bias_correction, parameter_maps, dt_agro_core, postprocessing). A detailed list of modules, dependencies and environment requirements is provided in the Annex (Code documentation), enabling other partners or future users to reproduce the full DT-Agro pipeline.

4.5 Digital Twin functionality beyond the hydrological model

Beyond running a hydrological model with EO-derived inputs, the integration framework is designed to support the full Digital Twin functionality:

- A complete historical run of DT-Agro is used to build a baseline representation of the Agro-Hydro-System over several decades, using the best available EO and meteorological data.
- As new EO products (e.g. updated NDVI, soil moisture, land-cover changes) and meteorological data become available, the system can be re-run or updated for the most recent period, updating both input variables (e.g. revised cover factors, imperviousness, irrigation patterns) and state variables (e.g. soil moisture, crop water status).
- The architecture supports the incorporation of near-real-time meteorological forecasts in future work, enabling DT-Agro to explore short-term trajectories of the

system (e.g. anticipated soil moisture evolution or irrigation needs under forecast weather).

In this way, the integration framework transforms AgroHydroLogos from a static simulation model into a dynamic Digital Twin that can:

- ingest and assimilate new EO and meteorological information,
- maintain an updated picture of the current state of the Greek Agro-Hydro-System, and
- support scenario analysis and decision-making in subsequent work packages.

5. Code documentation

The requirements and dependencies of the model program, as well as the file structures are as follows:

i. C++
  - Requirements
    - C++20 or newer
    - C++ Standard Library
    - C++ compiler for desired operating system (tested on Windows using the GNU GCC compiler)
  - File Structure
    - Model
      - Config.cpp
      - data_results_model.cpp
      - function_pick.cpp
      - main_model.cpp
      - Model.cpp
      - Config.h
      - day2int.h
      - function_pick.h
      - Interpolated.h
      - Model.h
      - points.h
      - Spatial.h
      - Station.h
      - TimeInterpolated.h
      - TimeSpatial.h
      - TimeStation.h
      - type_check.h
    - Serialize Raster
      - main_serialized_raster.cpp

        - Transform.cpp
        - Transform.h
      - Regrid Raster
        - main_serialized_raster.cpp
        - Transform.cpp
        - Transform.h
      - Shared files
        - container_ostream.h (for debugging only)
        - matrix.h
        - pow.h
        - read_data.h
        - replace.h
        - split.h
        - write_data.h
        - yeartime.h

ii. Python
  - Requirements
    - Python Standard Library
    - Python 3.12 or newer
  - Dependencies
    - rasterio
    - numpy
  - Modules/Files
    - ModelControl
      - __init__.py
      - filepick.py
      - data_conversions.py
      - date_pick.py
      - data_types.py
      - main.py
    - write_data.py
    - read_data.py

The file "data_types.py" is used as an input file. At the top of this file, the user can adapt the addresses of different model executables to suit their needs. The model is written in a modular form to facilitate future additions and modifications, as well as runs that only use some of the functionalities provided. The interface is written in Python, as well as the conversion of rasters to an internal binary format readable by the model. Computation-intensive parts are written in C++ to make the process faster.

Raster inputs (e.g. flow direction, flow accumulation) can be provided in TIFF or ADF format. Inputs that are taken directly from stations (e.g. rainfall, reference evapotranspiration) are provided in CSV format with a comma as a separator and a dot as a decimal point. Inputs that depend on time (e.g. NDVI) must be accompanied by a CSV file that mentions the dates, along with the corresponding file names. All specified rasters and the associated CSV files must be in the same folder.

Raster outputs are produced in TIFF format. All outputs are saved in folders specified by the user. The model also accepts data that are already in the internal binary format, allowing efficient reruns with different parameters settings without repeating unnecessary preprocessing steps..

To ensure an organized workspace, the model creates an internal folder called "_model_data" at a user specified location . This foldercontains the subfolders "_matrix", "_vector", "_station" and "_util", which are used to store pre-serialization binary raster data, post-serialization binary raster data, station data and miscellaneous utility files respectively. Placing additional data manually in this folder does not affect model results but is not recommended.

The total list of functionalities is as follows:

- Runoff calculation
- Runoff Routing (requires runoff calculation)
- K_c calculation
- Evapotranspiration
- Irrigation needs (requires evapotranspiration)
- Deep Infiltration (requires water balance)
- Water Balance (requires runoff, evapotranspiration and deep infiltration)

There are also two debug functionalities inside the code: one interpolates the rain data from station-based to raster format, and one returns the internal states produced by the serialisation process to facilitate inspection and visualisation.

In addition to the main model, two auxiliary C++ programs are provided: one converts rasters to a serialized (Soulis, 2013) format in correct hydrological order, and one converts serialised data back to a raster format. The serialisation utility requires a Digital Elevation Model, flow direction and flow accumulation data for the specified area. The deserialisation utility requires the internal state file produced during serialisation (stored in the "_util" folder as "data.ubin", in binary format).

If serialisation is performed within the workflow, the program automatically locates the data.ubin file. If serialised data are provided externally, the user specifies the corresponding data.ubin file manually.

6. Final Notes

The work presented in this report is best viewed as a snapshot of an evolving and dynamic system, rather than a final, static product. The DT-Agro platform and the underlying AgroHydroLogos-based modelling framework are being progressively transformed into a full Digital Twin of the Greek Agro-Hydro-System, capable of continuously ingesting EO and meteorological data, updating internal states and supporting operational decision-making. This constitutes an ambitious and long-term endeavour, and many of the components described here are subject to further refinement, extension and tighter integration in the coming phases of the project.

The current version already includes major advances:

- recoding and optimisation of the model core in C++ and Python,
- implementation of dual-resolution hydrological, crop and erosion algorithms,
- development of a hybrid meteorological forcing scheme based on bias-corrected virtual stations, and
- establishment of an EO data pipeline that links Copernicus and other geospatial datasets with the model parameterisation and state variables.

These developments represent a significant and essential step towards a robust operational Digital Twin, but they do not fully cover the scope of DT-Agro. Additional modules and processes will be gradually incorporated, including:

- nutrient and pollutant load budgets (e.g. nitrogen, phosphorus, agrochemicals) from source to stream,
- water-quality and pollution indicators at field and basin scales,
- soil health and degradation indices (organic carbon dynamics, salinity, compaction, erosion risk), and
- improved representations of management practices, irrigation technologies and adaptation measures.

In parallel, the EO and data-integration components will continue to expand, with more stations entering the virtual network, additional satellite products being assimilated, and new routines for near-real-time updating and scenario analysis. As these elements mature and are connected to other work packages, the Digital Twin progressively evolves from a research-grade prototype to a fully operational tool for monitoring, planning and supporting policies in Greek agriculture.

In this sense, the present report documents the foundation on which the DT-Agro Digital Twin will continue to be built. It summarises the current status of the algorithms, data flows and software components, while acknowledging that ongoing refinement, testing and extension are inherent to the development of such a complex and ambitious system.

References

Soulis, K., & Dercas, N. (2007). Development of a GIS-based Spatially Distributed Continuous Hydrological Model and its First Application. Water International, 32(1), 177–192. https://doi.org/10.1080/02508060708691974

Soulis K.X., 2013, "Development of a simplified grid cells ordering method facilitating GIS-based spatially distributed hydrological modeling." Computers & Geosciences, 54, 160-163, DOI:http://dx.doi.org/10.1016/j.cageo.2012.12.003.

Palli-Gravani, S., Soulis, K., Wałęga, A. (2025). Simplified Approach to consider Watersheds Heterogeneity in Direct Runoff Estimation using SCS–CN Model [Manuscript Under Review in Hydrological Sciences Journal]