Universal differential equations for estimating terrestrial evaporation

Olivier Bonte[1], Diego G. Miralles[1], Akash Koppa[2] & Niko E.C. Verhoest[1]

[1]Hydro-Climate Extremes Lab (H-CEL), Ghent University, Ghent, Belgium

[2]Laboratory of Catchment Hydrology and Geomorphology (CHANGE), EPFL Valais Wallis, Sion, Switzerland

GHENT UNIVERSITY · FACULTY OF BIOSCIENCE ENGINEERING · H-CEL Hydro-Climate Extremes Lab, Ghent University

Sharing is encouraged

This presentation participates in OSPP · EGU · Outstanding Student & PhD candidate Presentation contest

1. Introduction

Terrestrial evaporation (E)

- Crucial role in modulating climate dynamics and water resources.
- Not directly observable from space, limited in-situ observations.
- Challenging to model due to variety of atmospheric drivers and environmental stressors.
- Objective: Improve E modelling by leveraging the universal differential equation framework.

Approaches to modelling evaporation

Process-based models	Hybrid models	Machine learning
LSM: CLM, HTESSEL…	GLEAM4	FLUXCOM-X
RS: JPL-PT, PM-MOD…	3SEB-FR	

2. Universal differential equations

Universal Differential Equations (UDEs) combine process-based ordinary differential equations (ODEs) with data-driven universal approximators (U), such as neural networks, enabling the integration of physical models and machine learning.

State-space model

For the vector or state variables $\boldsymbol{x}$ with inputs $\boldsymbol{u}$, outputs $\boldsymbol{y}$ and parameters $\boldsymbol{\theta}$, the general state-space model (SSM) is given by:

$$\frac{d\boldsymbol{x}}{dt} = g(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})$$
$$\boldsymbol{y} = h(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}^*)$$

Depending on the model type, g is given by:

ODE	Neural ODE	UDE
$(g_1 \circ g_2 \circ g_3)(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})$	$U(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})$	$(g_1 \circ g_2 \circ U)(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})$

with g_i a process-based submodel.

Training data-driven universal approximators

Example: $U(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_U)$ replaces $\boldsymbol{y}_{\text{int}} = g_3(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_3)$

- Offline training: $\min_{\boldsymbol{\theta}_U} J(U(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_U), \boldsymbol{y}_{\text{int,obs}})$
  - (+) Easier implementation: U trained independent of SSM
  - (-) $\boldsymbol{y}_{\text{int,obs}}$ does not always exist, possibly unstable hybrid model
- Online training: $\min_{\boldsymbol{\theta}_U} J(\boldsymbol{y}, \boldsymbol{y}_{\text{obs}})$
  - (+) Directly optimise on output of interest $\boldsymbol{y}$
  - (-) Need for SSM implemented in a differentiable programming framework to efficiently calculate $\frac{\partial J}{\partial \boldsymbol{\theta}_U}$

Differentiable programming frameworks

Programming frameworks compatible with automatic differentiation (AD) and/or adjoint models:

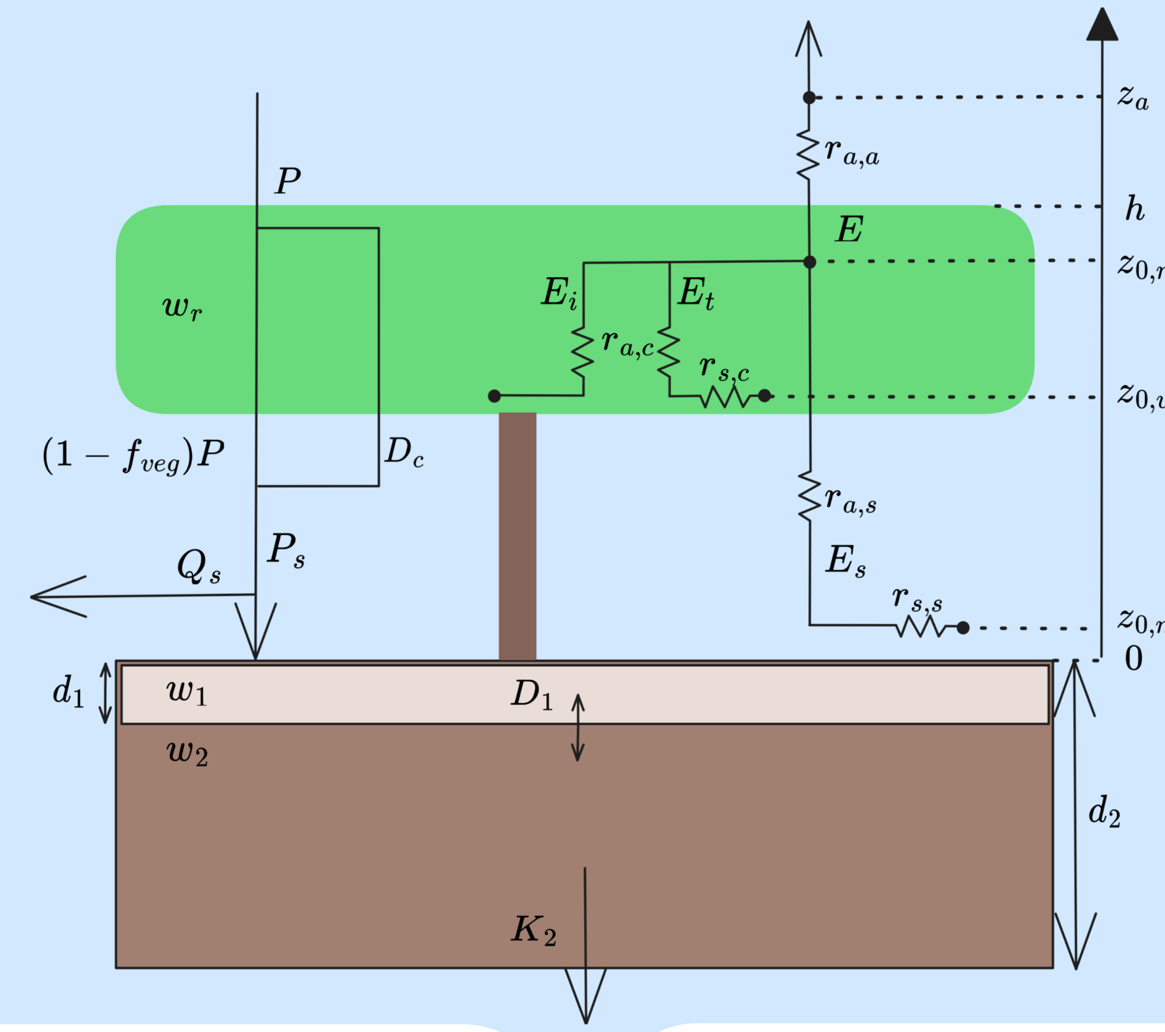
JAX · PyTorch · SciML (Framework of choice)

3. Model and method

Model description: 3SEB-FR

The three source energy balance force-restore (3SEB-FR) model aims to combine a relatively complex description of surface fluxes (3SEB) with a simpler soil moisture scheme (FR) to account for water stress limitation on E.

Subscript	Component
s	Soil
c	Canopy
a	Air



3SEB

The Shuttleworth-Wallace 2SEB model [1], which accounts for soil evaporation (λE_s) and transpiration (λE_t), is extended to explicitly include interception (λE_i). Each of the λE components is calculated using the Penman-Monteith equation (PM) with varying inputs of available energy (A) and aerodynamic/surface resistances (r_a/r_s):

$$\lambda E = \lambda E_i + \lambda E_t + \lambda E_s =$$
$$f_{wet}\lambda E_{PM}(0, r_{ac}, A_c) + (1 - f_{wet})\lambda E_{PM}(r_{sc}, r_{ac}, A_c)$$
$$+ \lambda E_{PM}(r_{ss}, r_{as}, A_s)$$

FR

The SSM is based on the ISBA-2L model [2]:

$$\frac{dw_1}{dt} = \frac{C_1}{\rho_w d_1}(P_s - Q_s - E_s) - D_1$$
$$\frac{dw_2}{dt} = \frac{1}{\rho_w d_2}(P_s - Q_s - E_s - E_t) - K_2$$
$$\frac{dw_r}{dt} = f_{veg}P - E_i - D_c$$

Following [3], a wet vegetation fraction (f_{wet}) is defined based on w_r.

Numerical ODE solution

Despite evidence that the numerical method chosen to solve the nonlinear differential equations of the hydrological SSM has a large effect on the numerical error, many hydrological models use ad hoc "sequential flux update" approaches [4]. By implementing the model in Julia, a wide variety of established numerical solvers is available via the DifferentialEquations.jl software and hence their effect on the simulation can be assessed.

Example: SSM with $\boldsymbol{x} = [x_1, x_2]^T$ and $g(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) = g(\boldsymbol{x})$

Sequential flux updates

$$x_1(t + \Delta t) = x_1(t) + g([x_1(t), x_2(t)]^T)\Delta t$$
$$x_2(t + \Delta t) = x_2(t) + g([x_1(t + \Delta t), x_2(t)]^T)\Delta t$$
$$\Delta t = \text{fixed}$$

(-) Order of state updating matters

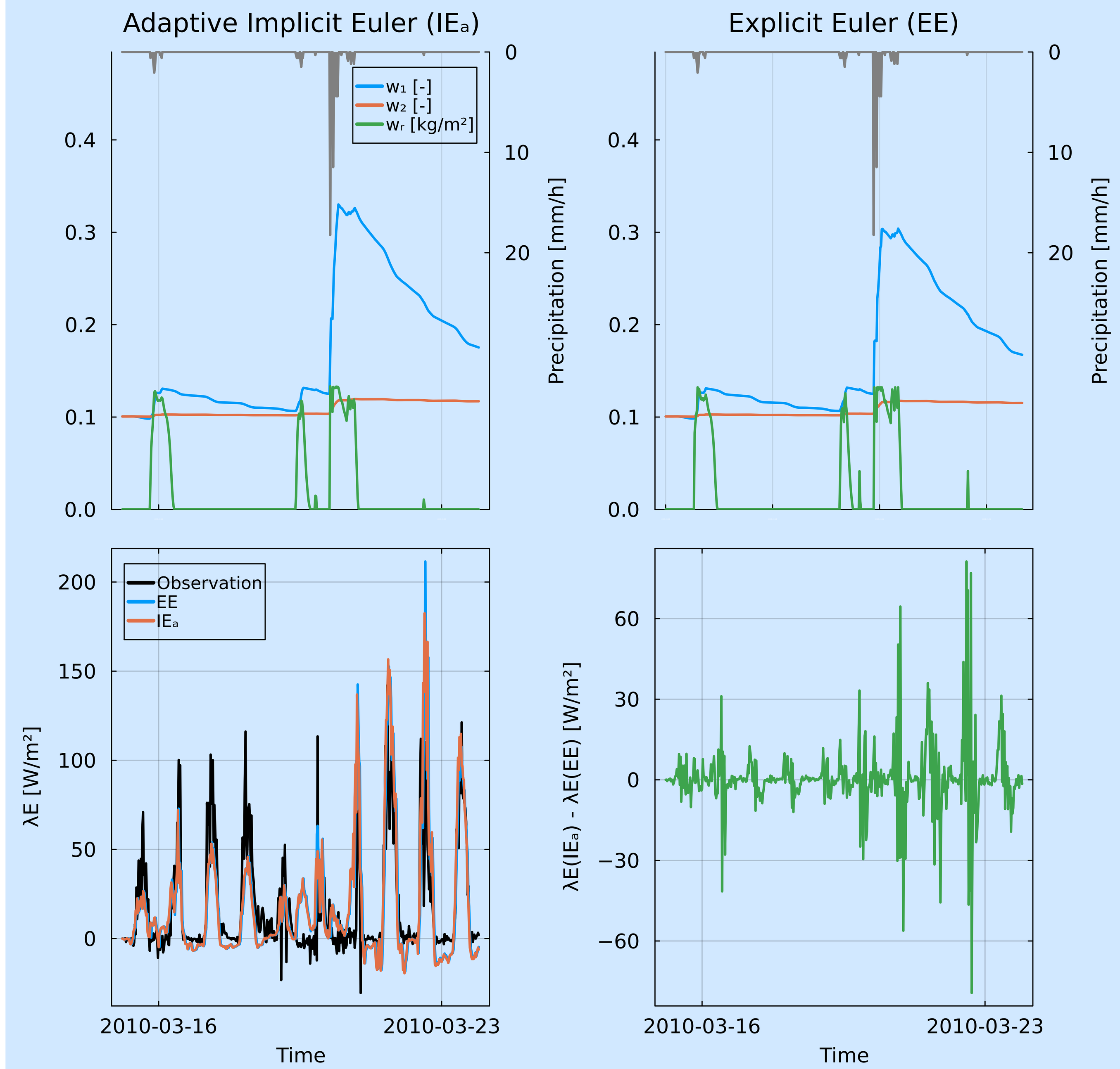
(-) No separation of conceptual model and numerical implementation
$\Rightarrow$ no distinction between conceptual and numerical error

Established numerical solvers

1) **Explicit Euler** (EE): $\boldsymbol{x}(t + \Delta t) = \boldsymbol{x}(t) + g(\boldsymbol{x}(t))\Delta t$, Δt = fixed
   - (+) Computationally cheap for large enough Δt
   - (-) Stability can require very small $\Delta t \Rightarrow$ hydrologists often apply manual corrections for states to be within physical bounds
2) **Adaptive Implicit Euler** (IE): $\boldsymbol{x}(t + \Delta t) = \boldsymbol{x}(t) + g(\boldsymbol{x}(t + \Delta t))\Delta t$, $\Delta t \neq$ fixed
   - (+) Stable method, avoids overshoots
   - (+) Computationally efficient adapting of Δt based on model stiffness to match user-defined error tolerance

4. Results

Model forcings and λE observations from the ICOS eddy covariance site BE-Bra from FluxDataKit [5] at 30' temporal resolution. Therefore, fixed-step schemes are applied with Δt =30'. The EE scheme is applied with manual state correction (cf. pane 3.).



- Difference in numerical scheme alters the simulated states and fluxes.
- Choice of forcing interpolation method affects stability of the ODE solver (not shown).

5. Future perspectives

Hybrid model development

- Leverage UDE framework to replace the empirical r_{sc} formulation by an online-trained U at the BE-Bra site.
- Extend the analysis to multiple eddy covariance sites, allowing the prediction of r_{sc} by U to be modulated based on static ecosystem characteristics.
- Benchmark the model predictions by comparing to an empirical machine learning method from [6].

Numerical considerations

- Further experiment with different numerical solvers considering numerical error, stability and computational expense.
- Explore the different options on how to calculate gradients of numerical solutions of ODEs outlined in [7] and assess how these interact with the solvers.

References:

[1] Shuttleworth, W. J., & Wallace, J. S. (1985). Evaporation from sparse crops-an energy combination theory. Q.J.R. Meteorol. Soc., 111, 839-855. https://doi.org/10.1002/qj.49711146910

[2] Deardorff, J. W. (1978). Efficient prediction of ground surface temperature and moisture, with inclusion of a layer of vegetation. J. Geophys. Res., 83(C4), 1889–1903. https://doi.org/10.1029/JC083iC04p01889

[3] Mahfouf, J., & Noilhan, J. (1996). Inclusion of gravitational drainage in a land surface scheme based on the force-restore method. Journal of Applied Meteorology and Climatology, 35(6), 987-992. https://doi.org/10.1175/1520-0450(1996)035<0987:IOGDIA>2.0.CO;2

[4] Clark, M. P., & Kavetski, D. (2010). Ancient numerical daemons of conceptual hydrological modeling: 1. Fidelity and efficiency of time stepping schemes. Water Resour. Res., 46, W10510, https://doi.org/10.1029/2009WR008894

[5] Hufkens, K., & Stocker, B. (2024). FluxDataKit v3.4: A comprehensive data set of ecosystem fluxes for land surface modelling (v3.4) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.13748398

[6] Abramowitz G., Ukkola, S., Hobeichi, S., Cranko, J. P., Lipson, M., De Kauwe, M. G., Green, S., Brenner, C., Frame, J., Nearing, G., Clark, M., Best, M., Anthoni, P., Arduini, G., Boussetta, S., Caldararu, S., Cho, K., Cuntz, M., Fairbairn, D., … Zeng, Y. (2024). On the predictability of turbulent fluxes from land: PLUMBER2 MIP experimental description and preliminary results. Biogeosciences, 21, 5517–5538. https://doi.org/10.5194/bg-21-5517-2024

[7] Sapienza, F., Bolibar, J., Schäfer, F., Groenke, B., Pal, A., Boussange, V., Heimbach, P., Hooker, G., Pérez, F., Persson, P., & Rackauckas, C. (2024). Differentiable programming for differential equations: a review. arXiv. https://doi.org/10.48550/arXiv.2406.09699

Olivier.Bonte@UGent.be

https://github.com/olivierbonte/DifferentiableEvaporation