JULES in UniFHy

David Livings[1] Bryan Lawrence[1]
Rich Ellis[2] Grenville Lister[3]

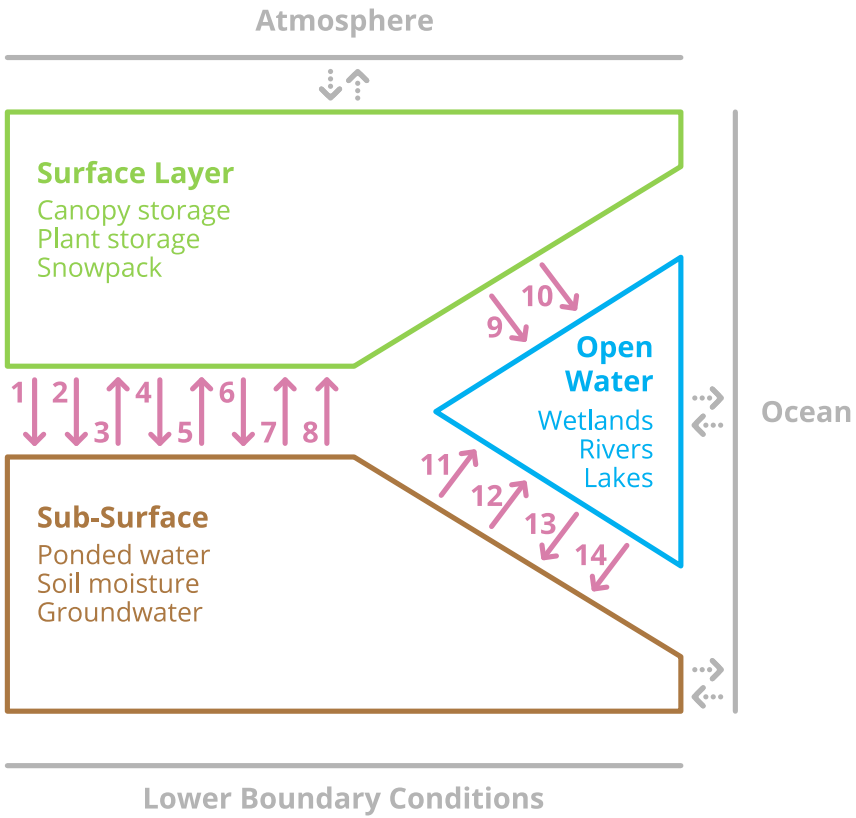
[1]NCAS, University of Reading

[2]CEH Wallingford (retired)

[3]NCAS, University of Reading (retired)

JULES Annual Science Meeting
9 September 2026

UniFHy (Unified Framework for Hydrology)



Source: Hallouin et al. (2022)

- UniFHy is a Python framework for hydrological modelling created as part of the HydroJULES project.
- It splits the model into three components with standardised interfaces.
- A library of components is available.
- JULES is being split into components for UniFHy.

Standard Interface Variables

Surface Layer ↔ Sub-Surface

Surface Layer → Sub-Surface

Canopy liquid throughfall and snow melt flux
Transpiration flux from root uptake
Direct water evaporation flux from soil

Sub-Surface → Surface Layer

Soil water stress for transpiration
Soil water stress for direct soil evaporation

Standard Interface Variables

Open Water

Surface Layer → Open Water

Direct throughfall flux

Water evaporation flux from open water

Sub-Surface → Open Water

Surface runoff flux delivered to rivers

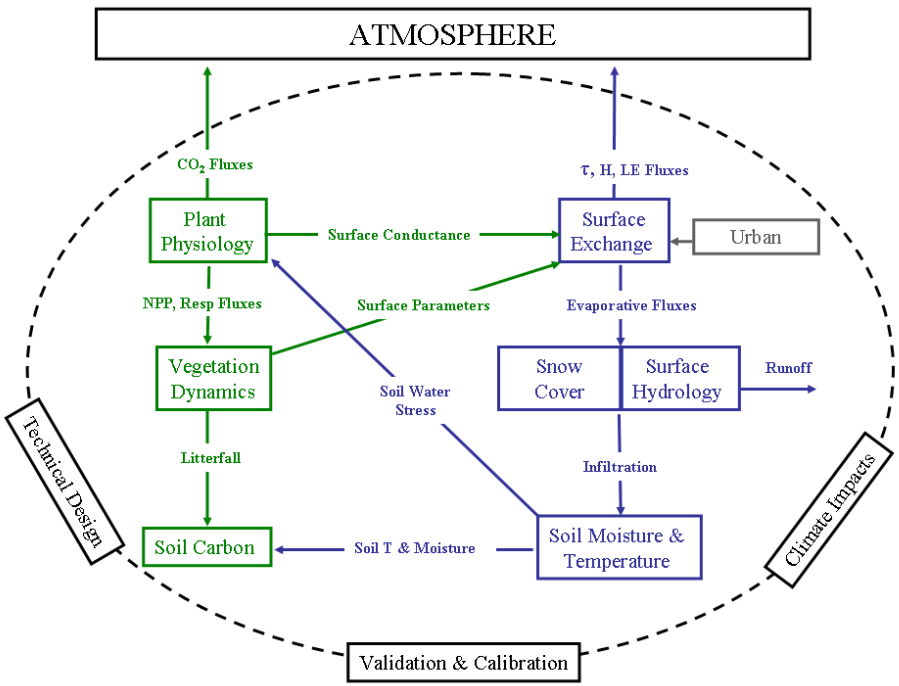
Net groundwater flux to rivers

Open Water → Sub-Surface

Open water area fraction

Open water surface height

JULES in UniFHy



Source: Best et al. (2011)

- Preliminary study based on documentation identified additional interface variables required to cover things such as thermodynamics.
- Work now is to split Sub-Surface off from Surface Layer.
- Strategy is to start at bottom of code and work up. By the time the split reaches the top, we should have a set of components usable in UniFHy.

Two Lessons Learnt

- Even more extra interface variables were discovered on examining the code. UniFHy has 14; at least as many again are required to support JULES.
- The JULES source code is in a terrible state (recent releases have subroutines with over 200 arguments).

Progress

- Science code and initialisation code have been split.
- Control code is now being split.
- A first version of split JULES (omitting TRIFFID and some features added since the documentation papers of 2011) will be ready soon.
- The JULES Fortran components will then be tested and wrapped in Python for use in UniFHy.
- Time permitting, omitted features will then be reintroduced.

Extra Hydrological Interface Variables

Sub-Surface → Surface Layer

Available moisture for transpiration

Available moisture for evaporation from bare soil

Soil moisture in layer at surface

Surface Layer → Sub-Surface

Initial surface runoff

Extra Thermodynamic Interface Variables

Generic

Surface Layer → Sub-Surface

Heat flux into the ground

Sub-Surface → Surface Layer

Temperature of top soil layer

Thermal conductivity of top soil layer

Heat capacity of top soil layer

Extra Thermodynamic Interface Variables

Required for JULES zero-layer snow model

Surface Layer → Sub-Surface

- Snow depth
- Number of snow layers
- Temperature of top soil layer [again]

Extra Interface Variables for Carbon Fluxes and Vegetation Dynamics

Surface Layer $\rightarrow$ Sub-Surface

- Litterfall
- Net primary productivity

Sub-Surface $\rightarrow$ Surface Layer

- Soil respiration
- Methane flux from wetlands

Extra Interface Variables for Fire

Sub-Surface → Surface Layer

Amount of DPM available to burn

Amount of RPM available to burn

Extra Interface Variable for Tiling

- JULES divides a gridbox into tiles representing different surface types.
- UniFHy doesn't currently have a tile concept, although one is planned for addition in the future.
- Until then, a loss of information at the interface must be accepted.
- When tiles are added to UniFHy, surface tile area fraction can be added as an interface variable (Surface Layer $\rightarrow$ Sub-Surface) to support the case where vegetation competition is turned on and area fractions can vary.

How Far Should the UniFHy Interface Be Extended?

It depends on why we want JULES embedded in UniFHy:

JULES user or developer wants all the extra interface variables.

Hydrological modeller using JULES to model components they aren't interested in wants as few extra interface variables as possible.

Seeker after modular land system modelling is pulled both ways.

Final Thoughts

- When faced with a mismatch between a model and an interface you can adapt the model to the interface, adapt the interface to the model, or do a bit of both.
- It would be good to have an objective criterion for identifying generic interface variables based on a fundamental analysis of the nature of flows of matter, energy, and information between components.
- A mismatch could then be taken as a sign that the model was doing something wrong and should be fixed.