

Introducing earthkit-hydro

Earthkit 1.0 training

Development Section

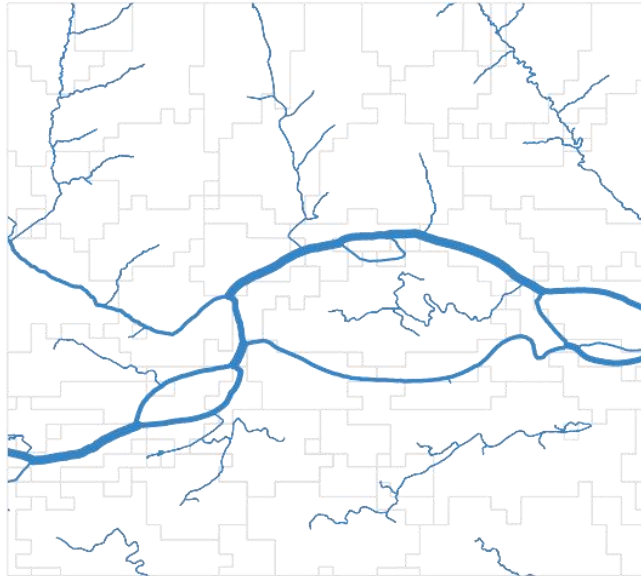


Hydrology – a special case?

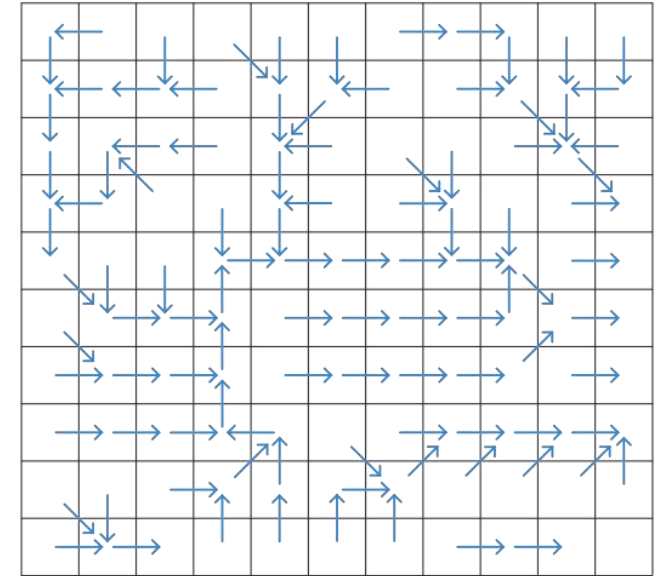
- Complex spatial dependencies governed by a river network



Rio Negro and Solimões River



Vector network



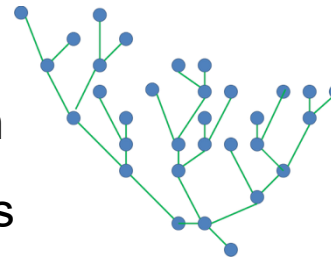
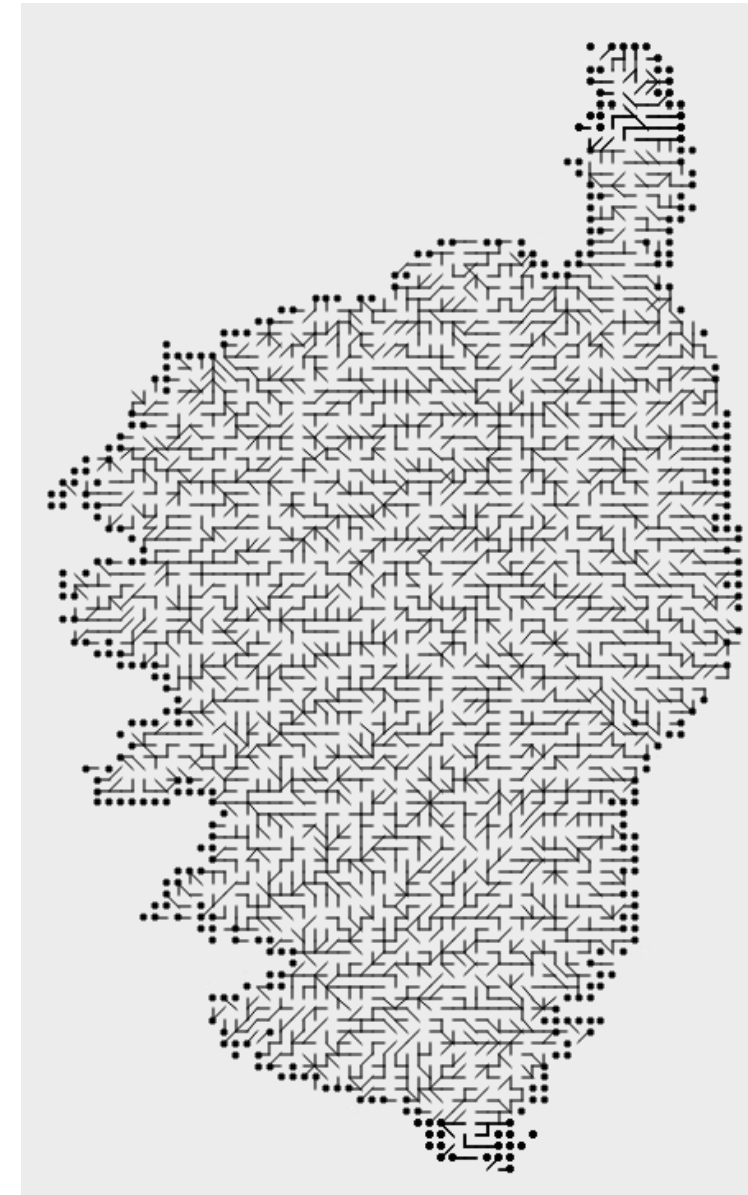
Grid-based network

Hydrology – a special case?

- ⚠ Many tools for specific river network formats
- ⚠ Performance – some statistics can be complex

earthkit-hydro

- Abstraction of river network through a graph
- Quick access to precomputed river networks
- Efficient statistics computations
- High-level (xarray) and array-level (numpy/torch/...) interfaces
- Designed for machine learning (GPU support, differentiable)
- Bifurcation support
- Both grid and vector-based networks



Loading river networks

Many precomputed river networks

```
import earthkit.hydro as ekh

ekh.river_network.load("efas", "5")
```

Or bring your own network

```
ekh.river_network.create(
    | path, river_network_format
)
```

- many formats: pcr_d8, esri_d8, merit_d8, cama, etc.

domain	version	Details	Note	Attribution
"efas"	"5"	1arcmin European		[1]
"efas"	"4"	5km European	Smaller domain than v5	[1]
"glofas"	"4"	3arcmin global	60° South to 90° North	[2]
"glofas"	"3"	6arcmin global	60° South to 90° North	[2]
"cama_01min"	"4"	1arcmin global		[3]
"cama_03min"	"4"	3arcmin global		[3]
"cama_05min"	"4"	5arcmin global		[3]
"cama_06min"	"4"	6arcmin global		[3]
"cama_15min"	"4"	15arcmin global		[3]
"hydrosheds_30sec"	"1"	30arcsec global	56° South to 84° North	[4]
"hydrosheds_05min"	"1"	5arcmin global	56° South to 84° North	[4]
"hydrosheds_06min"	"1"	6arcmin global	56° South to 84° North	[4]
"grit"	"1"	30m global (vector)	segments network	[5]

[1] Choulga, Margarita, et al. (2023): LISFLOOD static and parameter maps for Europe. European Commission, Joint Research Centre (JRC) [Dataset] PID: <http://data.europa.eu/89h/f572c443-7466-4adf-87aa-c0847a169f23>

[2] Choulga, Margarita, et al. (2023): LISFLOOD static and parameter maps for GloFAS. European Commission, Joint Research Centre (JRC) [Dataset] PID: <http://data.europa.eu/89h/68050d73-9c06-499c-a441-dc5053cb0c86>

[3] Yamazaki, Dai, et al. (2019): MERIT Hydro: A high-resolution global hydrography map based on latest topography datasets. Water Resources Research, vol.55, pp.5053-5073, 2019, DOI: 10.1029/2019WR024873

[4] Lehner, Bernhard, et al. (2008): New global hydrography derived from spaceborne elevation data. Eos, Transactions, 89(10): 93-94. Data available at <https://www.hydrosheds.org>.

[5] Wortmann, Michel, et al. (2025). Global River Topology (GRIT): A bifurcating river hydrography. Water Resources Research, 61(5), DOI: 10.1029/2024WR038308

Xarray and Standard Array APIs

- earthkit-hydro allows multiple array backends (numpy, cupy, torch, etc.)

```
import earthkit.hydro as ekh
```

```
network = ekh.river_network.load("efas", "5").to_device("cuda:0")
```

... and allows the network to be sent to a specific device

```
network = ekh.river_network.load("efas", "5").to_device("cuda:0")
```

- 2-level APIs: high-level (Xarray) and low-level (standard array)

```
# xarray-oriented operation (returns xarray)
ekh.submodule.operation(...)
```

```
# array-oriented operations (returns arrays)
ekh.submodule.array.operation(...)
```

```
ekh.catchments.sum(network, example_da, locations = {
    "gauge_1": (70.475, 28.32),
    "gauge_2": (42.225, 50.24)
})
```

xarray.DataArray 'precip' (step: 2, node_index: 2)

```
array([[ 6710.97380367, 272520.24025306],
       [ 6716.52716166, 272441.37894161]])
```

Coordinates:

step	(step)	int64	0 1	
node_index	(node_index)	int64	26066 4153831	
lat	(node_index)	float64	28.32 50.24	
lon	(node_index)	float64	70.47 42.23	
name	(node_index)	object	417 'gauge_1' 'gauge_2'	

```
ekh.catchments.array.sum(network, example_arr, locations = {
    "gauge_1": (70.475, 28.32),
    "gauge_2": (42.225, 50.24)
})
```

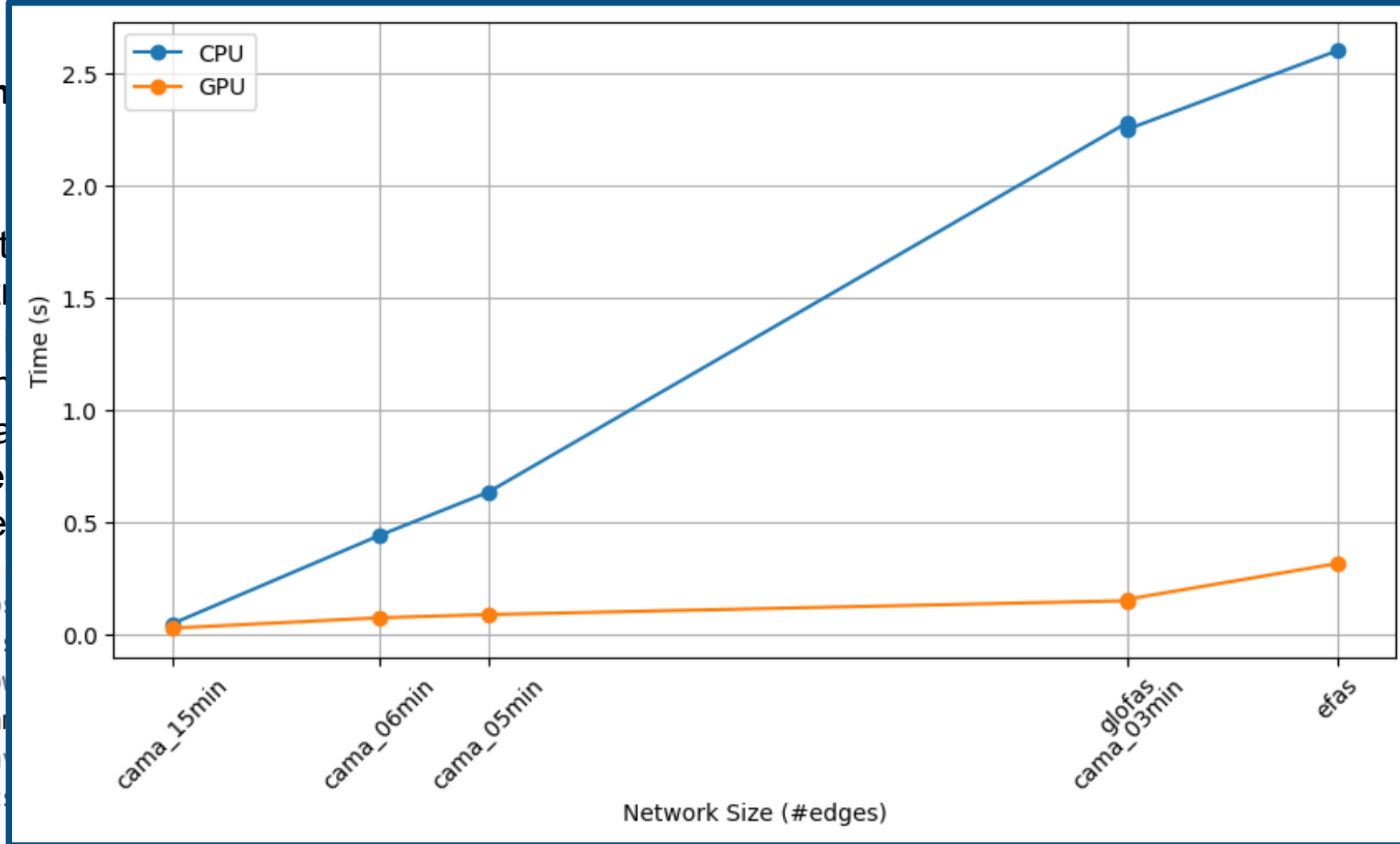
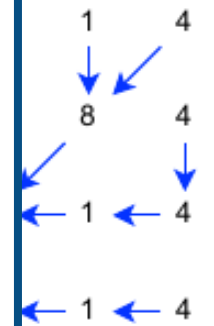
```
array([[ 6710.97380367, 272520.24025306],
       [ 6716.52716166, 272441.37894161]])
```

Efficient statistics

Efficient accum

- upstream,
- catchment
- Many met
 - sum,
 - min, r
 - media
 - Mode
- GPU acce

```
# sum over up
ekh.upstream.
# max over do
ekh.downstream
# catchment a
ekh.catchment
```



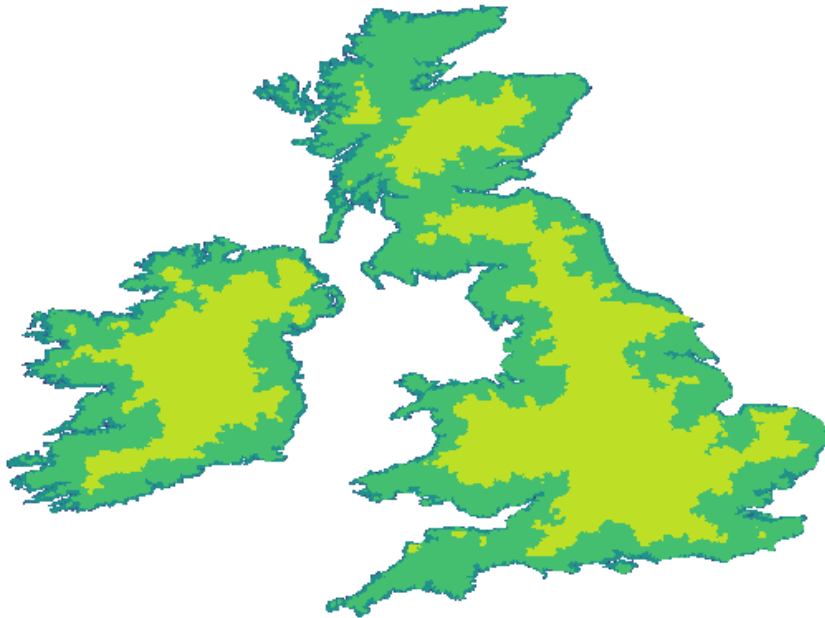
And more!



Distance calculations

```
earthkit.hydro.distance.max(river_network, locations, field=None,  
    upstream=False, downstream=True, return_type=None,  
    input_core_dims=None)
```

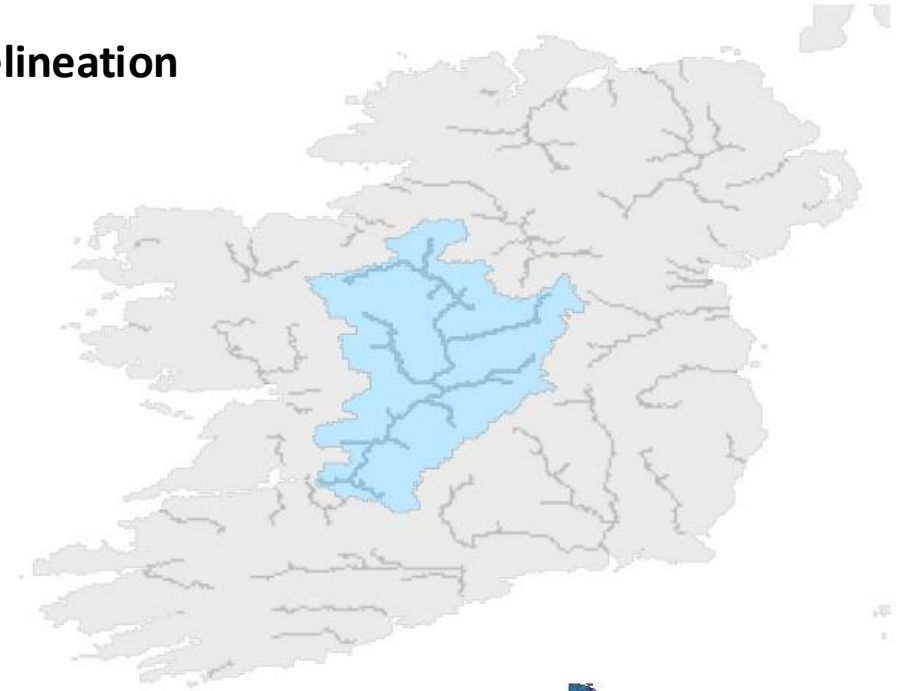
[\[source\]](#)



Catchment delineation

```
earthkit.hydro.catchments.find(river_network, locations, overwrite=True,  
    return_type=None, input_core_dims=None)
```

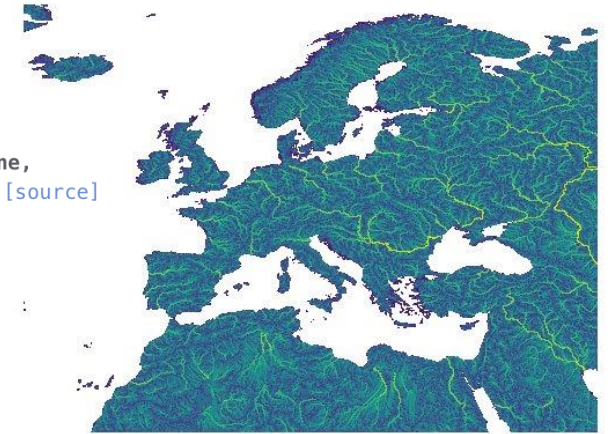
[\[source\]](#)



Streamorder (topological properties)

```
earthkit.hydro.streamorder.strahler(river_network, return_type=None,  
    input_core_dims=None)
```

[\[source\]](#)



Explore earthkit-hydro and earthkit

Project website



earthkit.ecmwf.int



Read *the* Docs



[earthkit-hydro
.readthedocs.io](https://earthkit-hydro.readthedocs.io)



GitHub



[github.com/ecmwf/
earthkit-hydro](https://github.com/ecmwf/earthkit-hydro)

