Some notes on ORCHIDEE's routing scheme

Agnès Ducharne; February 19, 2019

1. How to upscale the topographic index k when using high resolution topography?

The question stems from the will to use a high resolution topographic information to drive ORCHIDEE's routing. For the PhD of Trung Nguyen-Quang, we used a data base prepared at the 30-arc-sec resolution (ca 1km at Equator) based on HydrosHEDS (itself available at 3 and 15 arc-sec, but excluding land areas north of 60°N). This resolution is finer than the one of ORCHIDEE, which requires to properly deal with the way we upscale the topographic information at the scale of ORCHIDEE grid cells (or rather sub-basins or HTUs, for hydrological transfer units, which compose a grid-cell).

This topographic information is twofold: (i) the topographic index which depends on the pixel length and its slope, (ii) the flow direction, which depends on the slope between the pixels and its neighbors, and which is important to deduce the total travel distance within a HTU.

The topographic index k is involved in the timescale of each linear reservoir involved the routing scheme:

(1) $Q=V/\tau$, with $\tau=k.g$.

Units: Q is the outflow from the reservoir, here in kg/days, V is the stored water "volume" (in kg) at the beginning of the time step (of length dt_routing in seconds), k is in km, and g depends on the type of reservoir and is given in d/km: g_{stream} = 2.4 10^{-4} d/km, g_{fast} = 3.0 10^{-3}d/km, g_{slow}=2.5 10^{-2} d/km. These values come from the parameters fast_tcst = 3.0, slow_tcst = 25.0, stream_tcst = 0.24 in the routing module, which are divided by 1000 in routing_flow. In Eq. 1, the timescale τ is therefore given in days, and in the code, Q is further converted to kg/dt_routing by multiplying by 86400/dt_routing (see Eq. 7 below, and related questions).

The topographic index describes the influence of topography on the timescale, based on a simplification of Manning's formula, thus only valid, a priori, for the stream reservoirs:

(2) $k = d/\mathrm{sqrt}(slope) = \mathrm{sqrt}(d^3/dz)$,

where d is the stream length in the pixel, assumed to be the pixel length, and dz is a vertical elevation change at the pixel scale.

Eventually, the timescale of one HTU is given by:

(3) $\tau = g \,.\, d / \mathrm{sqrt}(slope)$

The data compiled from HydroSHEDS at the 1-km resolution by Ana Schneider directly give the slope (calculated as the maximum slope among the 8 possible directions from one pixel), and k= d/sqrt(slope), in which the pixel length d varies geographically (it accounts for the exact pixel length, estimated as the square root of the pixel area, and for the flow direction, with a factor √2 along the diagonals).

At the pixel scale, the timescale τ=k.g can be seen as the lag time between a unit input to the reservoir, and the resulting outflow. Knowing the pixels that belong to a sub-basin, the goal is to combine the local timescales t into the equivalent sub-basin timescale <T>. The different test cases below are illustrated in Figure 1.

a) Time lag T_i from one pixel i to the sub-basin outlet

$$T_i = \sum_{j \in streamline} \tau_j = \sum_{j \in [i,outlet]} \tau_j$$

b) Let's imagine a unit runoff over three pixels along a 3-pixel streamline

Pixel 1 is upstream, pixel 2 is in the middle, pixel 3 is downstream, so that T1>T2>T3. We further assume that each pixel has the same area. Thus, the lag T_i of each pixel contributes to the sub-basin lag with a 1/3 weight.

$$< T > = \sum_{i \in sub-basin} \alpha_i T_i$$, where α_i is the areal fraction of pixel i in the sub-basin.

c) Case of 3 branched network of 3 pixels (1 confluence)

Pixel 3 is downstream, and pixels 1 and 2 each have T_i > T3 (T1=T2 in the illustration). Again, considering the resulting time lag in case of unit input in all pixels leads to the same expression as above.

d) Generalization

This easily generalizes to any kind of sun-basin network in ORCHIDEE, since the reservoirs are linear (so the result of 2 unit inputs is 2 unit outputs), and we assume the entire sub-basin always receives a uniform input.

The resulting calculation can be further be simplified using the "hierarchy" information, which gives the cumulative value of k from one pixel to the large basin outlet at sea:

$$H_i = \sum_{j \in [i,sea]} k_i$$

$$gH_i = \sum_{j \in [i,sea]} \tau_j = T_i + gH_{outlet}$$, where H_{outlet} is the value of hierarchy at the sub-basin's outlet.

It follows that the equivalent sub-basin timescale is:

(4) $< T > = g \left(\sum_{i \in sub-basin} \alpha_i H_i \right) - gH_{outlet}$, where α_i is the areal fraction of each pixel i in the sub-basin.

The above formulation is normally better than what's presently coded in ORCHIDEE, which relies on a simple mean of the k of all the "topography" cells (at the 0.5° resolution). Such a mean leads to underestimate <T>, and to underestimate all the more as the resolution of topography is finer compared to the one of ORCHIDEE.

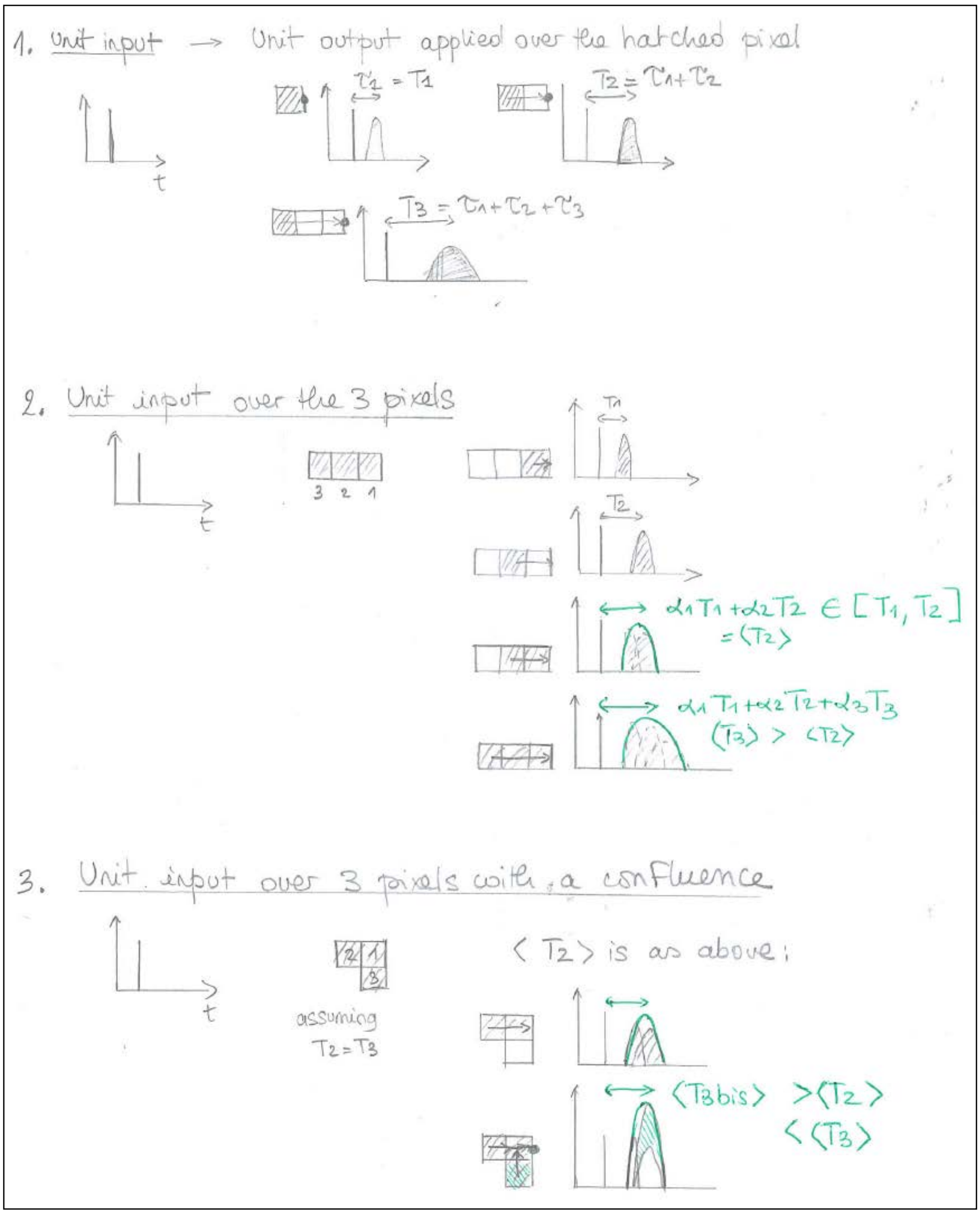

Figure 1. Combination of local timescale τ_i into sub-basin equivalent timescale <T> in simple cases.

2. Scaling problem when using the 0.5° topography

This problem has been identified when looking at the runs performed at different resolutions when preparing CMIPv1 (end of 2017, beginning of 2018). Based on physical considerations, the routing timescales should increase when the grid-cell size do, following the length of the HTUs (Eq. 3). Yet, it is not the case in the standard version of ORCHIDEE, which erroneously leads to results that are resolution dependent. The reason is that, in a given river basin, you get all the more grid-cells and HTUs as the resolution is fine, so the total routing time increases at higher resolutions if you don't correct by shorter timescales to match smaller HTUs.

Why are timescales independent from the model's spatial resolution? It comes from the routine routing_globalize, where the grid-scale topographic index (basin_topoind) is calculated as the average of the values from each contributing 0.5° pixel (topoind_bx):

```
basin_topoind(ib,ij) = basin_topoind(ib,ij) + topoind_bx(basin_pts(ij,iz,1),basin_pts(ij,iz,2))
basin_topoind(ib,ij) = basin_topoind(ib,ij)/REAL(basin_sz(ij),r_std)
```

In this part of the code, the first line is looped over basin_sz(ij), which gives the nb of 0.5° pixels in the HTU (see routing_simplify). As a result, topoind does not increase when the HTU gets larger, while it should.

Illustration of the impact. Figure 2 plot was prepared by Vladislav Bastrikov and compares four simulations of river discharge. They rely on the same code (trunk [r4438] with Zobler soil map). Simulations FG2 are forced by CRU-NECP at 2°, and FG3 by WFDEI_GPCC_v1 at 0.5°.

Simulations ending with ref (red and blue) use the default values of the g parameters (slow_tcst = 25.0, fast_tcst = 3.0, stream_tcst = 0.24), and they show the important sensitivity of the simulated discharge to the meteorological forcing. An important dependence to resolution is also visible, related to the proper location of the measurement station over the grid-mesh. This difficulty probably explains the lower discharges at 2° (red) compared to 1° (blue). Note that Matthieu Guimberteau has proposed a look-up table to place the main stations optimally at 0.5, 1, and 2°: https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/Ancillary

Besides, a close inspection of the timing of peak discharge suggests it occurs sooner at lower resolution (red before blue, clear for the Ob and the Mississippi for instance). This effect is consistent with the upscaling error reported below (same HTU-scale timescale at coarser resolution, thus with less HTU along the main river course, leading to smaller travel time to the outlet and measurement station).

This is confirmed by the other two simulations, in which we tried to correct the tcst parameters to make the two simulations closer to the 1° behavior:

- FG2.4438z.tcst (green): parameters multiplied by 2 (slowed down) to correct from 2° to 1° (SLOW_TCST = 50.0, FAST_TCST = 6.0, STREAM_TCST = 0.48)
- FG3.4438z.tcst (brown): parameters divided by 2 (accelerated) to correct from 0.5° to 1° (SLOW_TCST = 12.5, FAST_TCST = 1.5, STREAM_TCST = 0.12)

This choice was made since the tcst were tuned at the 1° resolution by Ngo-Duc et al. (2007) to reproduce the Senegal River discharge.

As expected, the 2° peak flow is delayed (FG2, from red to green), while the 0.5° peak flow comes earlier (FG3, from blue to brown). This effect is particularly visible for the Amazon, Orinico and Brahmapoutra, but it is overall very small. In the Amazon and Orinoco, peak discharge happens at the same month for the green and brown simulations, but with a different volume, coming from the different forcing and/or station location problems. Overall, the impact of the correction is small, and this simple method might not be enough.

Link to the upscaling method of section 1? To use the upscaling method proposed in section 1, we require the average of the high-resolution (here 0.5°) hierarchies inside the HTUs, to be subtracted by the hierarchy of the outflow pixel. This requires to combine the output of hierar_method='MEAN' (which overlooks the effect of 0.5° area) and hierar_method = 'OUTP'. Presently, the HTU-scale hierarchy is defined by hierar_method = 'OUTP'. Question: how comes OUTP is not equivalent to hierar_method='MINI', which should also correspond to the outlet of the HTU, at least if hierarchy does increase from headwaters to the oceans?

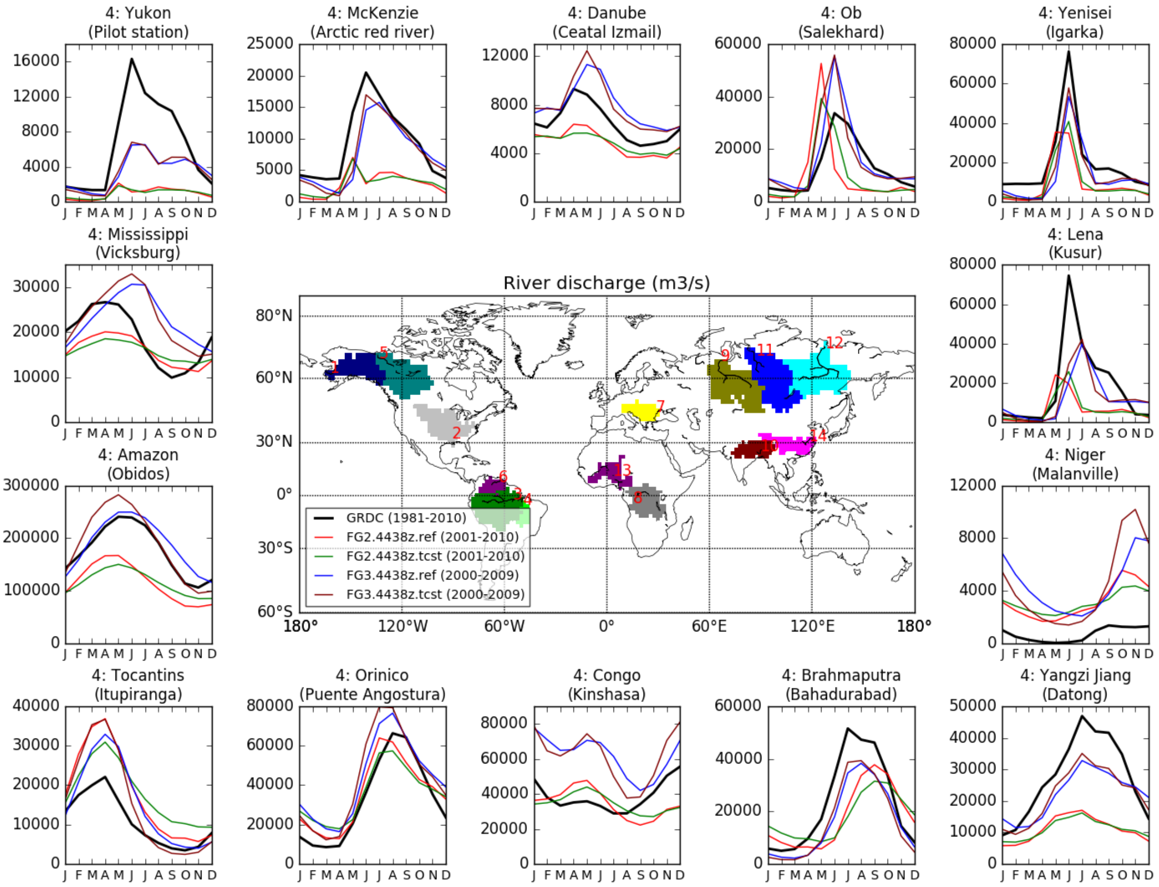

Figure 2. Comparison of river discharge simulated off-line at different resolutions with ORCHIDEE and the standard routing based on topographic info at 0.5°. Observed river discharge from GRDC appear in black. More explanations in the text.

3. Further questions regarding the validity of Eq.1

Eq. 1 is equivalent to the following differential equation:

(5) $\frac{dV}{dt} = -\frac{V}{\tau}$, which integrates as $V(t) = V_0 \exp(-t/\tau)$, in which time and timescales are have the same unit.

Over one time step of length dt_routing (given in seconds), assuming that the outflow starts from the volume V* at the beginning of time step, and assuming V* includes the inflow, we should get:

(6) $$Q = V^* - V(dt_{routing}) = V^*\left[1 - \exp(-dt_{routing}/86400\tau)\right]$$

The above equation assumes, as in the code, that τ is given in days.

This is different from what is found in the code, which corresponds to an explicit finite difference scheme:

(7) $$Q = \frac{V^*}{\tau}\frac{dt_{routing}}{86400}$$

```
flow = MIN(slow_reservoir(ig,ib)/((topo_resid(ig,ib)/1000.)*slow_tcst*one_day/dt_routing),&
                  & slow_reservoir(ig,ib))
```

We find here again that the timescale τ is given by `topo_resid(ig,ib)/1000.)*slow_tcst` (in d/km).

As shown in Table 1, the difference between the two equations, corresponding to the error over a time step, increases with dt_routing. This error also increases when τ decreases, which is consistent with the Courant-Friedrichs-Lewy stability criterion for finite difference methods:

(8) $\frac{v\,\Delta t}{\Delta x} \leq 1$, equivalent to $\Delta t \leq \tau$,

where v is the propagation velocity, inversely related to τ, and Δt and Δx are the time step and spatial step used for finite-differencing (thus dt_routing, and the "length" of the sub-basins in ORCHIDEE). The above criterion shows that the smaller τ, the higher the velocity, and the more unstable the scheme, unless Δt and Δx are adapted.

Table 1. Differences between two integrations of Eq. 5: analytical (K_Eq5, numbered Eq. 6 in text), and with an Euler scheme (K_Eq6, numbered Eq. 7 in text).

Comment	tau	tau	dt_routing		Tau/dt_routing	K_Eq5	K_Eq6	Error (%)	
	(days)	(sec)	(h)	(sec)	(-)	« True »		per time step	« per day »
Tau = 10 days	10	864000	1	3600	240	240,5003472	240	-0,208044283	-4,99306278
	10	**864000**	**24**	**86400**	**10**	**10,50833194**	**10**	**-4,837418036**	**-4,83741804**
Median of tau/slowr	67	5788800	1	3600	1608	1608,500052	1608	-0,031088083	-0,74611398
at 0.5°+Ngo-Duc	**67**	**5788800**	**24**	**86400**	**67**	**67,50124378**	**67**	**-0,74256969**	**-0,74256969**
Tau = 2 years	730	63072000	1	3600	17520	17520,5	17520	-0,002853827	-0,06849185
	730	**63072000**	**24**	**86400**	**730**	**730,5001142**	**730**	**-0,068461886**	**-0,06846189**
Tau = 20 years	7300	630720000	1	3600	175200	175200,5	175200	-0,000285387	-0,0068493
	7300	**630720000**	**24**	**86400**	**7300**	**7300,500011**	**7300**	**-0,006849002**	**-0,006849**
Tau = 200 years	73000	6307200000	1	3600	1752000	1752000,5	2E+06	-2,85355E-05	-0,00068485
	73000	**6307200000**	**24**	**86400**	**73000**	**73000,5**	**73000**	**-0,000684928**	**-0,00068493**

Note finally that other methods exist for integrating Eq. 5 with a finite-difference scheme. A simple one at the scale of a reach is the convex-routing method, cf Dingman (2002), p 427-431, see Fig 2.

Another method is linked with the Muskingum model, which differs from ORCHIDEE's routing because it uses two parameters (k = 1/ τ, and x, which describes wave diffusion) to relate inflow (I) and ouflow (Q):

(9) $dV/dt = I - Q$

(10) $V = k\,x\,I - k(1\text{-}x)Q$

The routing in ORCHIDEE is a simplification with x=0. It is noteworthy that an efficient matrix-based solution of the Muskingum method has been developed for complex river networks by David et al. (2011), and that many variations were developed to describe the effect of river discharge on the routing parameters k and x (Muskingum-Cunge method, cf Todini 2007; stage de M2 de Zhao (2007), encadré par A. Ducharne, avec tests de plusieurs méthodes d'intégration numérique).

BOX 9-5

Derivation of the Convex Routing Model

The conservation-of-mass relation can be expressed in finite-difference form as

$$QI_i - QO_i = \frac{V_{i+1} - V_i}{\Delta t}, \qquad \textbf{(9B5-1)}$$

where the subscripts denote values at successive instants, each separated by Δt. Δt is the "time step" for the routing procedure, which is selected by the analyst (see Box 9-6) and must be less than or equal to the reach travel time, T^*.

From Equation (9-28), we can write

$$V_i = T^* \cdot QO_i, \qquad \textbf{(9B5-2)}$$

and putting Equation (9B5-2) into (9B5-1) yields

$$QI_i - QO_i = \frac{T^* \cdot (QO_{i+1} - QO_i)}{\Delta t}. \qquad \textbf{(9B5-3)}$$

If we assume that T^* is known for a reach, that QI_i is known for all i during an event (the inflow hydrograph), and that an initial outflow value QO_0 is known, Equation (9B5-3) can be rearranged into a form useful for routing, namely,

$$QO_{i+1} = CX \cdot QI_i + (1 - CX) \cdot QO_i, \qquad \textbf{(9B5-4)}$$

where $CX (= \Delta t / T^*)$ is the routing coefficient [Equation (9-30)]. Equation (9B5-4) is known as the **convex routing equation.**

It can be shown that the convex routing model preserves continuity (i.e., total output always equals total input). From Equation (9B5-4), we see that the behavior of the model is controlled by CX, which must be in the range $0 < CX \leq 1$. As explained in the text, the smaller the value of CX, the more the flood wave is modified in moving through the reach.

Figure 3. The convex-routing method, from Dingman (2002).

4. Steady state volume for initialization

By definition, steady state is achieved when the outflow equals the inflow. It thus defines equilibrium between inflow and outflow, the equilibrium that is seeked by spin-up when no analytic solution is available. Since the time to reach equilibrium is commensurate to τ, we have the chance that a linear reservoir model has a simple analytic steady state solution. We define V_{ss} as the corresponding volume, which can be deduced from the long-term means of inflow (mean recharge thus drainage in ORCHIDEE, called D_m) and outflow (mean slowflow in ORCHIDEE, called Q_m).

V_{ss} can then be deduced directly from the differential equation (Eq. 5):

(11) $D_m = Q_m = \frac{V_{ss}}{\tau}$ which leads to $V_{ss} = \tau \, D_m$

Of course, for this equation to give a useful result, it has to be used with consistent units: if you want V_{ss} in mm, and have τ in days, then D_m must be in mm/d.

Equation 11 is particularly useful when τ is large, as the equilibrium volume is large as well, and initializing to zero would lead to incorrect results. This methods was used by Schneider (2017) [p83], who tested the impact of new formulations of τ for the slow reservoir, based on the Boussinesq equation.

5. Realism of the stream timescale values

We want to assess the realism of g_{stream} = 2.4 10^{-4} d/km = 20 s/km = 0.02 s/m.

Taken as is, this value corresponds to a velocity of 50 s/m, which is very fast compared to the widely accepted value of 0.5 s/m for large fluvial rivers. The reason is that the effect of slope is not yet accounted: v=sqrt(slope)/g

From Carlston (1969), we get that the slope of large US rivers falls in [0.0001,0.0005]:

slope=0.0001 => v ≈ 0.5 s/m

slope=0.0005 => v ≈ 1,1 s/m

So the default value of g_{stream} in ORCHIDEE seems to have the correct order of magnitude, although a bit too fast.

The corresponding timescales depend on the length d of "travel", which is theoretically the stream length in the calculation unit, related to the grid-cell size in ORCHIDEE. They are calculated in Table 2, for values of slope and length that are typical when running ORCHIDEE at the 0.5° resolution and coarser.

Table 2. Estimates of timescale τ [d], as a function of slope and length of the calculation unit.

τ [d]		Slope [m/m]			
		0.0001	0.0005	0.001	0.01
d [km]	25	**0.06**	0.027	0.019	0.006
	50	**0.12**	**0.054**	0.038	0.012
	100	**0.24**	**0.107**	**0.076**	0.0024

Compared to the CFL criterion expressed (Eq. 8), Table 2 shows that the finite difference scheme is never stable (dt_routing $\leq \tau$) with dt_routing = 1d, very rarely with dt_routing = 3h = 0.125 d, and not always with dt_routing = 1h = 0.042d (the corresponding "good" timescales values, so that 0.042 $\leq \tau$, appear in bold). This analysis confirms that the routing time step must be reduced when performing the routing at higher resolutions.

6. References

Carlston, Charles W (1969). Downstream variations in the hydraulic geometry of streams; special emphasis on mean velocity, American Journal of Science, 267,499-509.

David Cédric H., Maidment David R., Niu Guo-Yue, Yang Zong-Liang, Habets Florence, and Eijkhout Victor (2011). River Network Routing on the NHDPlus Dataset. *J. Hydrometeor*, **12**, 913–934. doi: http://dx.doi.org/10.1175/2011JHM1345.1

Dingman, S.L. (2002). Physical hydrology, 2nd edition. Prentice Hall.

Ngo-Duc, T., K. Laval, G. Ramillien, J. Polcher, and A. Cazenave (2007), Validation of the land water storage simulated by Organising Carbon and Hydrology in Dynamic Ecosystems (ORCHIDEE) with Gravity Recovery and Climate Experiment (GRACE) data, *Water Resour. Res.*, 43, W04427, doi:10.1029/2006WR004941.

Nguyen-Quang T, Polcher J, Ducharne A, Arsouze T, Zhou X, Schneider A, Fita L (2018). ORCHIDEE-ROUTING: revising the river routing scheme using a high resolution hydrological database. GMD, 11, 4965-4985, doi:10.5194/gmd-11-4965-2018. GMD link

Schneider, A (2017). Estimation of the base flow time constant for global scale applications, Université Pierre et Marie Curie, co-directrice : A. Jost (METIS), soutenance le 22 juin 2017. PhD in PDF

Todini, E. (2007). A mass conservative and water storage consistent variable parameter Muskingum-Cunge approach, Hydrol. Earth Syst. Sci., 11, 1645-1659, https://doi.org/10.5194/hess-11-1645-2007

Zhao, Yan (2007). Modélisation de la transformation des écoulements en débits dans le bassin de la Seine, Stage de M2 SDUEE, Parcours Hydrologie Hydrogéologie, UPMC, Année 2006-2007 PDF in English