Large-scale flood mapping - Tutorial

September 2020

Guénolé Choné ([g.chone@yahoo.ca](mailto:g.chone@yahoo.ca))

Julia Mazgareanu ([julie.mazga@gmail.com](mailto:julie.mazga@gmail.com))

Concordia River Management Lab

# Purpose

The purpose of this document is to deliver step-by-step instructions about how to apply the large-scale flood modeling methodology developed by the River Management Lab at Concordia University. The procedure is exemplified on the Etchemin watershed.

# Requirements

## Software requirements and installation

* ArcGIS Desktop version 10.6 or above, with Advanced license and Spatial analyst extension. The 3D Analyst extension can also be useful but is not required.
* LISFLOOD-FP 7.0.6 (included). This version of LISFLOOD-FP requires GDAL. A standalone version of GDAL for windows can be downloaded here: <http://download.gisinternals.com/>. More specifically, the “GDAL core components” version 3.0.4, release 1911, for x64 processors, have been successfully used with the provided Lisflood version: <http://download.gisinternals.com/sdk/downloads/release-1911-x64-gdal-3-0-4-mapserver-7-4-3/gdal-300-1911-x64-core.msi>. Note that the GDAL installation path must be manually added to the PATH environment variable of Windows.
* ArcGIS tools developed by the lab, available at <https://github.com/gchone>. Three folders are available in this github. The two folders “FloodTools” and “RiversTools” are ArcGIS toolboxes, both required. The folder “RasterIO” contains a single file (RasterIO.py), which is required by the two ArcGIS toolboxes and must be placed within their respective folders.
  + **NB:** this version of the tutorial corresponds to version 2.2 of the tools found on the Github under <https://github.com/gchone/ConcordiaRiverLab-FloodTools/releases/tag/v2.2>

## Data requirements

For a given watershed to be modelled, the following data are required:

* LiDAR DEMs, along with metadata about the LiDAR acquisition dates and flight paths as lines or polygons. Only LiDAR data of the study area is required.
* If the LiDAR data do not cover the entire watershed, the 10m provincial DEM of the remaining area: <https://www.donneesquebec.ca/recherche/fr/dataset/modeles-numeriques-d-altitude-a-l-echelle-de-1-20-000>
* GRHQ database containing vectorized stream network data (<https://vgo-telechargement.portailcartographique.gouv.qc.ca/mobile.aspx?gpm=710908e1-9eff-409a-9423-8607e585511c>)
* CIRCA 2000 Canadian Land Cover (<https://open.canada.ca/data/en/dataset/97126362-5a85-4fe0-9dc2-915464cfdbb7>)
* CEHQ historical discharge data (<http://cehq.gouv.qc.ca/hydrometrie/historique_donnees/default.asp>)
* Updated flood discharge. The discharge data from the “Atlas hydroclimatique du Québec méridional”, version 2020, is included.

## Data included in tutorial

### Necessary basic data needed for modeling any given watershed (found in *Data)*:

* fortecontenance.shp: points of dams and reservoirs used for the lake model component of flood simulations
* Noeuds\_Hydrotel\_2020\_long-lat\_revise\_20\_juin.shp: updated Hydrotel data points containing flood discharges of 20, 100 and 350 year recurrence intervals
* circa200manning.csv: table linking Manning’s n values with landcover types, used as join table to landuse data
* lakemodel.xlsx: pre-made spreadsheet to fill in for calculating projected water surface of lakes and reservoirs for each flood discharge
* Stations\_jaugeage\_Quebec.shp: Quebec’s gaging stations
* River networks of the INFO-Crue project

### Example dataset

The *Example\_Etchemin* folder contains an example of the application of the large-scale floodmapping methodology, with all intermediary results. For easier accessibility, rasters have been clipped to a small subarea of the watershed. Lisflood results also reflect this.

# Some important notes before beginning

All files that go into the tools described herein have to have the same cell size, number of columns and rows, extent and spatial reference of the flow direction raster (or DEM). If this condition is not respected, the tools will not run or will display errors/warnings. As such, make sure to specify this in the Environments pane of each tool; be it ArcMap-specific ones or those described in this document. Furthermore, make sure the resulting rasters are snapped to the flow direction/DEM and that all overlap correctly.

For the sake of increased clarity, ArcGIS geoprocessing tools will be signified by an orange colour (ex: *Polygon To Line*), while homemade tools, as well as associated scripts, will be demarcated by a blue colour (ex: *D4 flow direction*).

Rasters and shapefiles should not be stored in a geodatabase. In addition, all rasters must be of the GRID type (and hence can have 13 characters maximum). When using homemade tools, select input data from the Catalog, not from the Table of content.

# Process overview

The large-scale modelling approach takes place in two main steps.

First, the raw data must be processed in order to create the inputs with the right format for LISFLOOD-FP. These inputs are mainly 6 rasters containing the following information:

* DEM
* Channel width
* Channel mask
* Bed elevation
* Flood discharge
* Manning’s n values for the floodplain

Then, the LISFLOOD-FP simulations can take place.

# Data preparation

## Identification of the river networks

The very first step of the methodology is to identify the river networks for which the flood assessment is needed. The INFO-Crue project already defines a river network (file *niveaux\_services\_fichier\_global.shp*). However, the large scale flood modelling process only need to be applied to a subpart of this network. The rules are:

* Excluding all reaches classified as “Aucune couverture” in the INFO-Crue network (field “NS\_VS1”)
* Including all reaches classified as “Modèle hydraul \_complet” (regardless of the drainage area)
* Including all reaches classifies as “Large-échelle” or “Modèle hydraul\_simplifié” with a drainage area of more than 50 km2.
* Including reaches classifies as “Large-échelle” or “Modèle hydraul\_simplifié” with a drainage area between 40 km2 and 50 km2 if the same stream is already included in the network (i.e. reaches identified by the previous rules are extended upstream until the drainage area drops below 40km2. No new streams are added by this process).

In order to compute the drainage area, a first 10m resolution DEM at the watershed scale must be computed. Note that this DEM is only a temporary one, only used for preliminary drainage area computation. It should not be used further in the process.

### Drainage area computation

* Make sure you have DEMs:
  + At 1m resolution (from LiDAR data) for the study reaches and their floodplain
  + At any resolution for the rest of the watershed if it’s not entirely covered by LiDAR data (NB: polygons of watersheds are available in the GRHQ, layer *UDH*)
* If holes are present in the DEM, they need to be filled:
  + Replace NoData in the DEM by an unrealistic low value (ex. -999). This can be done in the *Raster Calculator* by:

Con(IsNull(“dem”), -999, “dem”)

* + *Fill* the DEM
  + *SetNull* values that remained at -999
* *Aggregate* individual LiDAR tiles from 1m to 10m using the **minimum**

This can be done with the Iterate Rasters function in ModelBuilder (https://desktop.arcgis.com/en/arcmap/10.3/tools/modelbuilder-toolbox/iterate-rasters.htm)

* *Mosaic to New Raster* to merge 10m DEM tiles into one single DEM

Input: all 10m LiDAR tiles

Parameters: Pixel Type: 32 float; Number of Bands: 1

* If the LiDAR data do not cover the whole watershed:
  + *Mosaic to New Raster* to merge the coarse resolution DEMs into one single DEM
  + *Project Raster* to change spatial reference of the resulting DEM to the same one as the 10m LiDAR DEM.
  + *Resample* the raster to 10m (*Snap* to the 10m LiDAR DEM)
  + *Mosaic to New Raster* to add the rest of the watershed to the 10m LiDAR DEM. With the default *Mosaic Operator* (*LAST*), the complementary DEM must be added first, then the 10m LiDAR DEM, in order for the 10m LiDAR DEM to be kept where the two DEMs overlapped.
* *Fill*
* *Flow direction*
* *Flow accumulation*

Flow accumulation result is a drainage area in a number of pixels. With a 10m raster, it can be transformed into square-kilometers by dividing it by 10 000.

### Creation of the linear river network

* using the Flow Accumulation result *Reclassify* can be used to set a specific value (i.e. 1) to rivers with more than 50 km2 of drainage area, another value (i.e. 2) to rivers between 40 and 50km2 of drainage area, and NoData elsewhere.
* *Raster to Polyline* can turn the result into a shapefile

To create the final shapefile of the river network, a copy of the INFO-Crue can be made and manually edited to match the set of rules for river inclusion and exclusion. The shapefile with rivers classified according to their drainage area will help you identify which reaches need to be cut or deleted.

The resulting river network must be validated with the project manager.

## LiDAR survey selection and DEM generation

### LiDAR survey selection

In the case of certain watersheds, several LiDAR surveys can be available. It is then required to select which survey to use where they overlap. This choice must be mainly based on the discharge in the rivers when the LiDAR data were acquired. There are no straight up instructions that can be followed, as the situation will greatly vary from one watershed to another. As a general guideline, and by order of importance:

* LIDAR surveys for which the exact days of acquisition can be identified are preferable
* LiDAR surveys with low discharges are preferable
* The most recent LIDAR surveys are preferable

### Creation of the polygonal river network

A temporary version of the river network, with rivers represented as polygons, is required for communication. This polygon river network can be created using the polygons from the GRHQ database (layer *RH\_S*), with manual editing to cut out upstream parts and unwanted tributaries. For streams for which a GRHQ polygon doesn’t exist, a simple buffer (*Buffer* tool) with a crude approximation of the half-width value can be done and merge to the previous results.

### DEM generation

The selection of LiDAR surveys, with their order of preference, as well as the polygonal river network must be send to the GIS contact at the DEH. They will produce accordingly the DEM and provide it.

## Identification of lakes and reservoirs

Because the hydraulic modeling uses a steady flow (constant discharge), lakes and reservoirs must be excluded from the hydraulic modeling, and thus properly identified.

### Identifying lakes and reservoirs from the GRHQ

* Lakes are identified in the GRHQ layer *RH\_S* by the field *TYPECE = 21 – Lac*. Select these by the *Select By Attributes* tool, and export them as a new shapefile.

Output: lakes/lakesGRHQ.shp

* The lakes on the studied area can be identified by a *Selection by Location*, using the intersection with the polygonal river network.

Output: lakes/lakesGRHQ\_onnetwork.shp

* Small lakes can be included in the hydraulic simulation, as their effect on hydraulic can be neglected: a threshold of 5 ha for small rivers, or 10 ha for bigger ones, can be used. The field *SUP\_AREA* can be used to select and delete them from the lakes shapefile.

*NB: The lake included in this tutorial is smaller than 5ha, but only serves an illustrative purpose. The same principle applies to lakes larger than 5ha.*

The largest reservoirs are also included in the GRHQ (*TYPECE = 23 – Réservoir*) and can be extracted the same way. This was not done on the Etchemin River as there is no such reservoir over the studied area.

### Identifying reservoirs from the *Répertoire des barrages*

The government database of dams (<https://www.cehq.gouv.qc.ca/barrages/default.asp>) can be used to identify dams and reservoirs over the studied area. To help with this task, this data has been turned into a shapefile (*fortecontenance.shp*). Only the major dams (category “Forte contenance”) have been included.

Reservoirs created by these dams must be digitized manually. To identify the reservoir, a longitudinal profile of the water surface elevation can be drawn with the 3D Analyst toolbar.

*Output: lakesfinal.shp*

NB: This step was not done on the Etchemin River, as all reservoirs on the study network are already included as lakes in the GRHQ database.

## River polygons

A polygon of the wetted channel, when the LiDAR data were acquired, is required.

### Automated channel detection

This method aims to automatically produce the channel polygons. It combines an edge detection algorithm with a filling algorithm to produce a polygon representing the wetted channel based on the LiDAR DEM. It is suggested that this step should be done on a LiDAR of 3m resolution, aggregated by the minimum. The default set of parameters defined by the tools described below are based on this resolution. Work is under progress to refine this method.

The basic steps for this are:

* *Canny Edge Detection* to automatically detect edges based on a slope raster created from a high resolution DEM

*NB: "Canny Edge Detection" is not part of version 2.2 of the toolset. It is available in the "master" fork on the GitHub and works on ArcGIS PRO only.*

*Inputs: a slope raster of the LiDAR covering the area of interest, created with Slope*

*The parameters sigma and upper and lower thresholds must be left to default if using a 3m resolution LiDAR. They need to be adjusted otherwise.*

*A desired output should detect the river banks as much as possible, especially in the case of smaller channels. Also, the detected bank edge should be as much as possible continuous.*

*Output: edge raster, where a value of 255 represents the edges.*

*The output needs to be transformed to a binary raster, where 1 represents the edges and NoData everything else.*

* *Détection du chenal* to fill the channel, from the channel centerline to the edges detected in the previous step.

*Inputs: channel elevations, i.e. a raster containing the LiDAR elevations along the flow direction path of that LiDAR*

*LiDAR DEM (e.g. lidar3m aggregated by the minimum)*

*Maximal iteration number and maximum allowed height difference should be left to default (these indicate how sensitive the filling procedure should be; generally, higher numbers result in more filling and vice versa, as well as in increased computational times)*

*Raster used to exclude areas from the analysis represents the binary edges raster*

*(Optional) Channel lines (NB: not used in the current version of the methodology)*

*Diffusion by D4 (leave box unchecked; D8 results in the filling done in the diagonal direction as well, which is not desired when the edges are diagonal, as the filling “escapes” and fills overbank)*

*Post-processing and post-validation unchecked (NB: not used in the current version of the methodology)*

*Output: raster of the filled channel*

* *Création du polygone de surface de l'eau* to transform the raster into a polygon, while also applying a post-processing

*Inputs : raster of the filled channel*

*Maximal width to fill should be double the LiDAR resolution, i.e. 6m*

*Minimal width of channels should be raster resolution, i.e. 3m*

*NB : this tool will soon be available on the GitHub*

* *Smooth Polygon* with a tolerance of 12m
* *Simplify Polygon* with a tolerance of 1m

### Manual editing of the polygons

The automated channel detection produces an approximation of the channel, but these polygons must be manually checked and corrected to better fit the wetted channel on the LiDAR data. Digitization of the streams can be done using a hillshade raster of the network of interest at 1m resolution (produced with *Hillshade*). GRHQ polygons of the channel (RH\_S) can be used as a reference in the cases where they are a good representation of the LiDAR channel (manual polygon can be traced along the GRHQ polygon). Note that:

* At the downstream end of the river, the river polygon must extend further than the DEM.
* Lake polygons (*lakesfinal.shp*) can be added directly to the results. No manual editing is required as the polygons around lakes don’t need to be as precise as for the rivers.
* The river network, including lakes, must be one and only one polygon. This can be checked by using *Dissolve* (with *create multipart features* unchecked): the resulting attribute table must contain one and only one row.

The produced polygons should fit as best as possible the LiDAR data. However, tests are currently done to better assess the degree of manual editing needed.

Output: Lisfloodinputs\channelpoly.shp

## Linear river network

This step aims to produce a linear network of the rivers.

* Over wide rivers, the linear network of the GRHQ can be used (layer *RH\_L*). It is however necessary to check that the river lines always fall inside the river polygons, and correct them if necessary.
* The GRHQ data is often too approximate for small streams. Manual digitization of the stream centerlines can be done.
* Alternatively, a centerline can be derived from the river polygons:
  + *Polygon To Line* to create lines of the banks

Output: temp\polybanks.shp

* + Edit the result in order to delete the transversal lines and cut the downstream and upstream ends into a new shapefile: *temp\polycuts.shp*.

NB: temp\polybanks.shp was copied to temp\polybanks2.shp before editing to show the edits

* + *Collapse Dual Lines To Centerline* to create a centerline

Parameter: Maximum Width = the maximum width of the channels, not accounting for the lakes. 180m was used for the Etchemin River.

Output: temp\collapsed\_banks.shp

* + Edit the result to improve it

NB: temp\collapsed\_banks.shp was copied to river\_network.shp before editing to show the edits

* When multi-channels exist for a river, only keep the main one
* Oxbows or side channel with no flow must be excluded
* In a similar way as with the channel polygon, the network can be checked by using *Dissolve* (one shape should be created by river reach)

Output: temp\rnetwork\_d.shp

* Create a watershed-scale river network
  + *Merge* together the linear river network from the GRHQ (layer *RH\_L*), for the studied watershed and the ones surrounding it.
  + Simplify the network by deleting streams with a Strahler order of 0.
  + Identify the streams that intersect the channel polygons (*channelpoly.shp*) using the *Select by Location…* tool (in the *Selection* menu), and delete them
  + *Merge* together the result with *river\_network.shp*

Output: watershed\_net.shp

## DEM preparation

This step aims to create 10m DEMs of the study reaches (from LiDAR data) and for the whole watershed, as well as generate basic files used in several subsequent steps.

* Make sure you have DEMs:
  + At 1m resolution (from LiDAR data) for the study reaches and their floodplain
  + At any resolution for the rest of the watershed if it’s not entirely covered by LiDAR data (NB: polygons of watersheds are available in the GRHQ, layer *UDH*)
* If holes are present in the DEM, they need to be filled:
  + Replace NoData in the DEM by an unrealistic low value (ex. -999). This can be done in the *Raster Calculator* by:

Con(IsNull(“dem”), -999, “dem”)

* + *Fill* the DEM
  + *SetNull* values that remained at -999
* *Aggregate* individual LiDAR tiles from 1m to 10m using the **mean**

This can be done with the Iterate Rasters function in ModelBuilder (https://desktop.arcgis.com/en/arcmap/10.3/tools/modelbuilder-toolbox/iterate-rasters.htm)

* *Mosaic to New Raster* to merge 10m DEM tiles into one single DEM

Input: all 10m LiDAR tiles

Parameters: Pixel Type: 32 float; Number of Bands: 1

Output: DEMs\10m\lidar10m\_avg

NB: For illustrative purposes, all subsequent steps are illustrated on a small subsection of the Etchemin watershed. The 1m DEM and hillshade are in DEMs\1m.

* If the LiDAR data do not cover the whole watershed:
  + *Mosaic to New Raster* to merge the coarse resolution DEMs into one single DEM
  + *Project Raster* to change spatial reference of the resulting DEM to the same one as the 10m LiDAR DEM.
  + *Resample* the raster to 10m (*Snap* to the 10m LiDAR DEM)
  + *Mosaic to New Raster* to add the rest of the watershed to the 10m LiDAR DEM. With the default *Mosaic Operator* (*LAST*), the complementary DEM must be added first, then the 10m LiDAR DEM, in order for the 10m LiDAR DEM to be kept where the two DEMs overlapped.

NB: This step was not done on Etchemin River, as the LiDAR data covered the entire watershed.

* burn streams into DEM (at the watershed scale):
  + *Polyline* *to Raster* (with *Cellsize*, *Coordinate System*, *Extent*, and *Snap* defined by the 10m watershed-scale DEM)

Input: watershed\_net.shp

Output: temp/stream\_l

* + *Raster calculator*:

Input: Con(IsNull("stream\_l"), "lidar10m\_avg", "lidar10m\_avg"-100)

NB: lidar10m\_avg must be replaced by the watershed-scale DEM if they differ.

Output: DEMs\10m\lidar10m\_burn

* *Fill*

Input: DEMs\10m\lidar10m\_burn

Output: DEMs\10m\lidar10m\_fill

* *Flow direction*

Input: DEMs\10m\lidar10m\_fill

Output: DEMs\10m\lidar10m\_fd

* *Flow accumulation*

Input: DEMs\10m\lidar10m\_fd

Output: DEMs\10m\lidar10m\_facc

* *Clip* the flow direction and the flow accumulation results to the studied area (use *lidar10m\_avg* for the *Output Extent*)

NB: This step was not done on the Etchemin River, as the LiDAR data covered the entire watershed.

* Create a point shapefile having the same spatial reference as the DEM and place it/them at the upstream end of the river line, on the flow path displayed by the flow accumulation

Output: dep\_pts.shp

## Channel width

* *Placement sections transversales* to place cross sections at a specific interval. The distance used can vary with the size of the river. A distance of 100m can be used for small head streams, while a distance of 500m can be used for major rivers. If different distances are used within the same watershed, several files must be created and then merged manually.

Input:

DEMs\10m\lidar10m\_fd

dep\_pts.shp

Distance entre sections = 100

Output: width\xsect100.shp

* Edit the points in order to add more points where the width of river changes rapidly. New points must be added on the flow path. This can be easily done by using Placement sections transversales without a distance parameter (it creates a point on every raster cell along the flow path) and use the result to snap the new points.

Output: width\xsect\_flowpath.shp / width\xsect.shp

* *Create Thiessen Polygons*

Input: width\xsect.shp

Output: width\thiessen.shp

* Verify and edit the Thiessen polygons for them to split the river polygons in reaches. Check in particular confluences, sharp meanders, area around lakes, and places of sharp changes of width. (The Thiessen polygon should extend completely from one bank to the opposite bank.)

Output: width\thiessen\_edit.shp

* *Intersect* Thiessen polygons with rivers (polygons and lines)

Input: width\thiessen\_edit. shp, temp\river\_network.shp/ channelpoly.shp

Output: width\reaches\_line.shp/ width\reaches\_poly.shp

* *Dissolve* according to *INPUT\_FID*

Input: width\reaches\_line.shp/ width\reaches\_poly.shp

Output: width\reaches\_line\_d.shp/ width\reaches\_poly\_d.shp

* *Add Geometry Attributes* to calculate geodesic length and area

Input: width\reaches\_line\_d.shp/ width\reaches\_poly\_d.shp

* *Add Field* to xsect.shp to store width (Type: Float)
* *Join Field* to add areas and lengths to xsect.shp

Input: width\xsect.shp (FID), width\reaches\_line\_d.shp (Input\_FID)/ width\reaches\_poly\_d.shp (Input\_FID)

* *Calculate Field* to calculate "width" field = area/length
* Delete points that fall into lakes (by selecting them by location)

Output: width\xsect\_minuslakes.shp

* *Point to Raster* based on field "width" (match Output Coordinates, Processing Extent and Snap to computed flow direction raster)

Input: width\xsect\_minuslakes.shp

Output: width\ch\_width\_pts

* *D4 flow direction* to extract the D4 path of the flow direction

Input: DEMs\10m\lidar10m\_fd

DEMs\10m\lidar10m\_fill

dep\_pts.shp

Output: d4fd

* *Interpolation lineaire* to linearly interpolate obtained width points

Input: d4fd

dep\_pts.shp

width\ch\_width\_pts

Output: width\width

## Channel mask

* *Polygon to Raster* to transform the channel polygon into a raster (match Output Coordinates, Processing Extent and Snap to computed flow direction raster)

Input: channelpoly.shp

Output: mask\frompoly

* *Raster Calculator* to have 1 where the raster made from the channel polygon has a value, or where the D4 has a value. This ensures that the streams are always comprised of at least one cell.

Con(IsNull("mask\frompoly"),Con(IsNull("d4fd") == 0,1),1)

Output: mask\mask\_temp

* *Raster to Polygon* to transform the channel mask back into a polygon in order to be able to subsequently modify it according to channel size

Input: mask\mask\_temp

Output: mask\mask\_poly.shp

* Display width, and change the symbology to “Classified”. Use 2 classes, with a break value at 15m
* **Edit the *mask\_poly.shp* file to cut out reaches that will be modelled with the SubGrid technique (width < 15 = no mask). *NB: The mask should at best represent areas >15m. For reaches smaller than ~1km where the width varies, follow the general characteristic of the larger section, e.g. in the image below, the general characteristic is defined by green (=SubGrid), so this entire section should be cut out from the mask.*
* *Polygon to Raster* to transform the result back into a raster (match Output Coordinates, Processing Extent and Snap to computed flow direction raster)

Input: mask\mask\_poly.shp

Output: mask\mask

## Water surface detection– method #1

The aim of this step is to extract water surface elevation along the studied rivers. This process gives better result with a raster at fine resolution. A 3m resolution is often a good compromise (finer resolution can be too long to process and produce too big files). For huge watersheds, a 5m resolution could be needed.

* *Aggregate* individual LiDAR tiles from 1m to 3m using the **minimum** (Iterate Rasters function)
* *Project Raster* to change spatial reference of tiles, if needed (Iterate Rasters function)
* *Mosaic to New Raster* to merge 3m DEM tiles into one single DEM

Input: all 3m LiDAR tiles

Parameters: Pixel Type: 32 float; Number of Bands: 1

Output: lidar3m\_min

NB: this LiDAR is found under DEMs\lidar3m\_min for the subsection treated in the documentation

* Identify the bridges that need to be corrected. This can be done by looking at the DEM hillshade. Create polygon of the bridges (having the same coordinate system as the DEM). The bridges should be small rectangles that contain at least one full cell of a lower elevation than the high elevation of the bridge.

Ouput: watersurface\bridges.shp

* *Correction des ponts et ponceaux* to correct bridges and culverts on the 3m DEM

Input: DEMs\lidar3m\_min, watersurface\bridges.shp

Output: watersurface\m1\lidar3m\_corr

* *Correction de la surface de l'eau* to correct the elevation of the water surface

Input:

watersurface\m1\lidar3m\_corr

temp\polycuts.shp : lignes des extremités des polygones de cours d’eau

channelpoly.shp

temp\river\_network.shp

Output: watersurface\m1\lidar3m\_forws

* *Fill* and *Flow Direction* to generate the flow direction raster

Input: watersurface\m1\lidar3m\_forws

Intermediary result: watersurface\m1\lidar3m\_fill

Output: watersurface\m1\lidar3m\_fd

* *Lissage de la surface de l’eau* to extract the water surface along the flow path. The DEM to use for the water surface is the “fill” one. The DEM to use for the estimation of the error is the one with the bridges corrected.

Input:

watersurface\m1\lidar3m\_fd

dep\_pts.shp

watersurface\m1\lidar3m\_fill

watersurface\m1\lidar3m\_corr

Output: watersurface\m1\ws3m

* *Allocation euclidienne pour matrice à virgule flottante* to project laterally the water surface values over a specified distance

Input: watersurface\m1\ws3m

Output: watersurface\m1\ws3m\_b

* *Resample* to turn the raster into a 10m one. Extent and Snap define by *lidar10m\_avg*.

Input: watersurface\m1\ws3m\_b

Bilinear resampling

Output: watersurface\m1\ws10m

* *Clip* to match the characteristics of the DEM

Input: watersurface\m1\ws10m

Extent: DEM\10m\lidar10m\_avg

Parameter Maintain clipping extent = True

Output: watersurface\m1\ws10m\_c

* *Bréchage*

Input:

watersurface\m1\ws10m\_c

d4fd

dep\_pts.shp

Output: watersurfac\m1\ws10mbr\_m1

## Water surface detection – method #2

This step extracts the water surface using a different approach which further distinguishes between large and small channels. The two water surfaces obtained in this section (one for large streams, one for small streams) will be applied at the whole watershed, but subsequently merged according to the channel mask to form a global water surface.

### Large channels

In the case of large channels (width > 15m) a process of filtering, linear regression and smoothing is employed. This uses a compounded DEM (aggregation: 1m to 5m by mean; 5m to 10m by minimum)

* *Aggregate* individual LiDAR tiles from 1m to 5m using the **mean** (Iterate Rasters function)
* *Project Raster* to change spatial reference of tiles, if needed (Iterate Rasters function)
* *Mosaic to New Raster* to merge LiDAR tiles into one single DEM comprising the watershed (Pixel Type: 32 float, Number of Bands: 1)

*Input: all reprojected 5m LiDAR tiles*

*Output: lidar5m\_avg*

NB: this LiDAR is found under DEMs\lidar5m\_avg for the subsection treated in the documentation

* *Clip* to comprise area of study, if necessary
* Identify the bridges that need to be corrected. This can be done by looking at the DEM hillshade. Create polygon of the bridges (having the same coordinate system as the DEM). The bridges should be small rectangles that contain at least one full cell of a lower elevation than the high elevation of the bridge. *(NB: Skip if already done in step 5.7.)*

Ouput: watersurface\bridges.shp

* *Correction des ponts et ponceaux* to correct bridges and culverts on this 5m DEM

*Input: watersurface\bridges.shp*

*Output: watersurface\m2\lid5mavg\_corr*

* *Aggregate* from 5m to 10m using the minimum to obtain the compounded DEM

*Input: watersurface\m2\lid5mavg\_corr*

*Output: watersurface\m2\comp10m*

* *Fill*

Input: watersurface\m2\comp10m

Output: watersurface\m2\comp10m\_fill

* *Flow direction*

Input: watersurface\m2\comp10m\_fill

Output: watersurface\m2\comp10m\_fd

* *Extraction de la surface de l’eau, méthode Info-CRUE 1*

Input:

watersurface\m2\comp10m\_fd

dep\_pts.shp

watersurface\m2\comp10m

Output: watersurface\m2\comp10m\_rawws

* *Lissage de la surface de l’eau*

Input:

watersurface\m2\comp10m\_fd

dep\_pts.shp

watersurface\m2\comp10m\_rawws

watersurface\m2\comp10m

Output: watersurface\m2\comp10m\_wssm

* *Allocation euclidienne pour matrice à virgule flottante* to project laterally the water surface values over a specified distance

Input: watersurface\m2\comp10m\_wssm

Output: watersurface\m2\comp10m\_wssmb

* *Bréchage*

Input:

watersurface\m2\comp10m\_wssmb

d4fd

dep\_pts.shp

Output: watersurface\m2\ws10mbr\_m2

### Small channels

In the case of small channels (width <15m), a breach-only procedure is applied.

* *Aggregate* individual LiDAR tiles from 1m to 5m using the **minimum** (Iterate Rasters function)
* *Project Raster* to change spatial reference of tiles, if needed (Iterate Rasters function)
* *Mosaic to New Raster* to merge LiDAR tiles into one single DEM comprising the watershed (Pixel Type: 32 float, Number of Bands: 1)

*Input: all reprojected 5m LiDAR tiles*

*Output: lidar5m\_min*

NB: this LiDAR is found under DEMs\lidar5m\_min for the subsection treated in the documentation

* *Clip* to comprise area of study, if necessary
* *Correction des ponts et ponceaux* to correct bridges and culverts on this 5m DEM

*Input: watersurface\bridges.shp*

*Output: watersurface\m2\lid5mmin\_corr*

* *Aggregate* from 5m to 10m using the minimum

*Input: watersurface\m2\lid5mmin\_corr*

*Output: watersurface\m2\lidar10m\_min*

* *Fill*

Input: watersurface\m2\lidar10m\_min

Output: watersurface\m2\lid10mmin\_f

* *Flow direction*

Input: watersurface\m2\lid10mmin\_f

Output: watersurface\m2\lid10mmin\_fd

* *Bréchage*

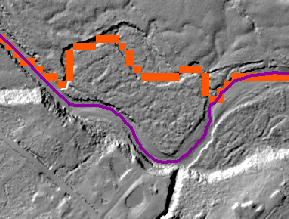
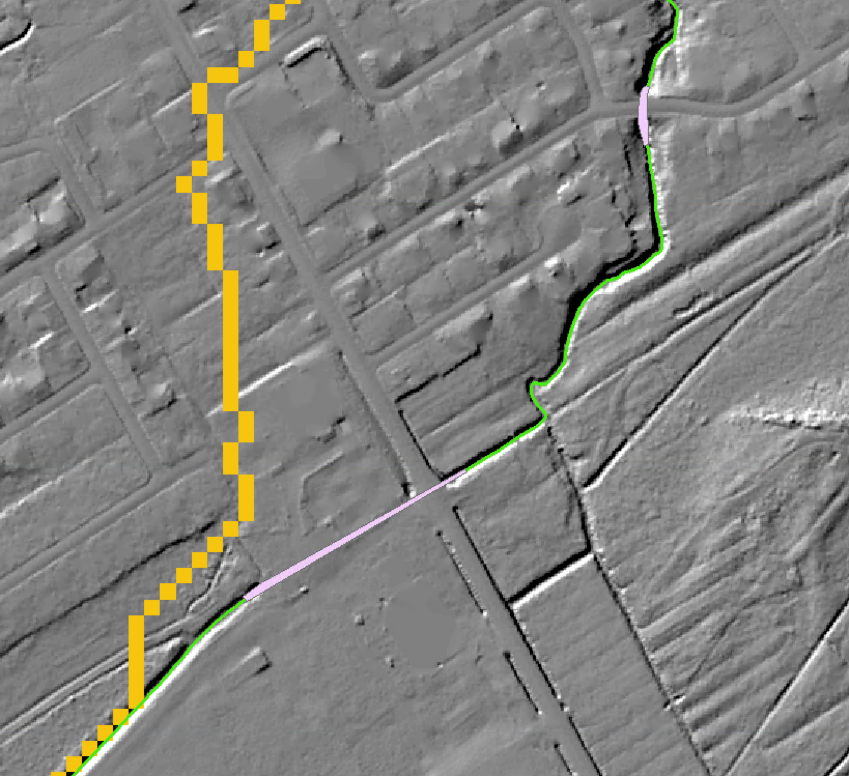
Input:

watersurface\m2\lidar10m\_min

watersurface\m2\lid10mmin\_fd

dep\_pts.shp

Output: watersurface\m2\breach

* Where there is no mask (i.e. only for small streams, where the breach applies), compare the location of the breached flow with the river network to make sure the rivers flow were they are supposed to (e.g. image below, left: flow through side channels is accepted; right: bridge/culvert polygon was too narrow, resulting in a wrong flow path of the breach). In the latter case, the bridges need to be corrected and the procedure rerun.
* *Allocation euclidienne pour matrice à virgule flottante* to laterally project the water surface values over a specified distance

Input: watersurface\m2\breach

Output: watersurface\m2\breach\_b

* *Extract by Mask*

Input: watersurface\m2\breach\_b, d4fd

Output: watersurface\m2\breachd4fd

### Global water surface

* *Raster Calculator* to merge both water surfaces according to the mask

*Con(IsNull("mask") == 1, "breachd4fd", "ws10mbr\_CONC ")*

*Output: watersurface\m2\merged\_ws*

* Raster Calculator to replace NoData outside the DEM by a high elevation in order to safeguard against filling everything in the subsequent step

Con(IsNull("lidar10m\_avg")==0, Con(IsNull("merged\_ws")==1, 9999,"merged\_ws"))

Output: watersurface\m2\fake\_z\_ws

* Fill to ensure that there are no steps in the water surface elevation at confluences as a result of the merging of the two different methods

Input: watersurface\m2\fake\_z\_ws

Output: watersurface\m2\fake\_z\_fill

* SetNull to replace 9999m with NoData

Input: watersurface\m2\fake\_z\_fill, Value 9999

Output: watersurface\m2\global\_ws

## LiDAR discharge

Discharges when the LiDAR were acquired are extracted by the DEH from the *Atlas hydroclimatique du Québec meridional*, version 2020. A table must be prepared, with reaches codes (according to the Atlas 2020) as rows and dates as columns. This table is sent to the hydrological resource at the DEH, which fill it and send it back.

* Join the discharge table to the shapefile containing the Hydrotel points based on the *Code\_tron (make sure that points of no interest are excluded and if necessary, check “Keep only matching records”)*
* Export result to new shapefile

*Output: Q\qlidar\qpts.shp*

* manually snap these Hydrotel points to the flow accumulation raster, just upstream the confluence, on each stream

Output: Q\qlidar\qpts\_snapped.shp

* *Extract Multi Values to Points*, to add the flow accumulation to these points
* Add a new fields (*Add Field*, type = Double, name them according to date with the suffix “\_corr”, max. 10 characters) to store the discharges adjusted by proportionality to the drainage area, using *Field Calculator*:

*e.g. 190605cor = [19\_06\_05]/ [Superficie]\* [lidar10m\_f]/10000.*

* *Point to Raster* each corrected date *(match Output Coordinates, Processing Extent and Snap to computed flow direction raster)*

Input: Q\qlidar\qpts\_snapped.shp

Output: e.g. Q\qlidar\19\_06\_05

* *Spatialisation des débits* with each of these points to interpolate discharge as a function of basin area

Input:

DEMs\10m\Lidar10m\_fd

dep\_pts.shp

DEMs\10m\lidar10m\_facc

e.g. Q\qlidar\19\_06\_05

Interpolation entre les points de débit **unchecked**

Output: e.g. Q\qlidar\19\_06\_05\_s

Repeat for all discharge dates

* *Clip* according to flypath that corresponds to the date (e.g. flypath taken on 19\_06\_05) by opening the attribute table of the flightpath.shp and selecting the corresponding polygon while doing the clip *(snap to flow direction/DEM)*

*Input:*

*Q\qlidar\19\_06\_05*

*Q\qlidar\flightpath.shp (with the right polygon selected)*

*Check Use Input Features for Clipping Geometry*

*Uncheck Maintain Clipping Extent*

*Output: e.g. Q\qlidar\19\_06\_05\_c*

Repeat for all discharge dates

* *Mosaic to New Raster* to merge all qlidar segments into one (*Pixel Type: 32 float, Number of Bands: 1, match Output Coordinates, Processing Extent and Snap to flow direction/DEM*)

*Input: all clipped qlidar segments*

*Mosaic Operator: MEAN (NB: sometimes an order of preference applies based on how trustworthy the LiDAR survey is, e.g. a weektime survey would be least trustworthy. In those cases, the Mosaic Operator should reflect order of preference)*

*Output: Q\qlidar\qlidar*

* *Allocation euclidienne pour matrice à virgule flottante* to project laterally the water surface values (to make sure there are values all along the d4 flow path)

Input: Q\qlidar\qlidar

Distance: 100

Output: Q\qlidar\qlidard4

## Bed elevation

* *Évaluation du lit* to calculate the water surface elevation along the flow direction path

Input:

d4fd

dep\_pts.shp

width\width

watersurface\ws10mbr (m1 or m2)

Q\qlidar\qlidard4

Output: bedelevation\bed (\_m1/m2)

## Manning’s n for floodplain

* *Merge* and *Clip* Circa Landuse shapefiles over basin/study area

Output: ManningFP\landuse\_clip.shp

* *Add Join* to created landuse shapefile with data from circa200manning.csv, based on *"covtype"*
* *Polygon to Raster* with Manning’s n assigned as Value Field *(match Output Coordinates, Processing Extent and Snap to flow direction/DEM)*

Input: ManningFP\landuse\_clip.shp

Output: ManningFP\n\_floodplain

# Simulations (Méthode 2D)

Before being able to run the Méthode2D tools, we still have some intermediary steps to deal with: process lakes and reservoirs, and identify simulation type (SubGrid Channel or SuperGridChannel).

## Lakes and reservoirs

### Estimation of flood water level for lakes and reservoirs

A statistical model was built to link water level reaches in lakes and reservoirs in Quebec, for 20 years, 100 years and 350 years of return period, by Olcese et al. (2020, in redaction). This model used gauged lakes to explained water level using multiple linear regression, with discharge downstream the lake and lake area as explanatory variables.

The *lakemodel.xls* spreadsheet contains the equations to apply this model. Required data for each lake are:

* The discharges for 20 years, 100 years and 350 years of return period (q20, q100, q350)
* Lake area. Lakes from the GRHQ, as well as the shapefile *fortecontenance.shp* from the Répertoire des barrages, already provide this information
* Water surface elevation of the lake from the LiDAR (lidar10m\_avg). This can be estimated with the *3D Analyst surface profile* tool, by interpolating a line along the reservoir, displaying a graph of this and visually estimating the elevation.

### Processing lakes for Lisflood simulations

* Copy the polygons of the lakes

Input: lakes\_final.shp or lakes\_GRHQonnetwork.shp (see section 5.4)

Output: lakes\_forsim.shp

* Create an additional feature in *lakes\_forsim.shp* at the downstream end of the study area. This is to assign the model a downstream boundary condition. Most often, this represents the water level in the Saint Lawrence that is associated with a particular recurrence interval and that is obtained from reports found on the InfoCrue platform.

*NB: This does not apply to the Etchemin River, since the subsections ends at a lake.*

* *Optional:* If lakes have extremely uneven shapes that, it is useful to create small polygons, roughly rectangle, where the river path flows into lakes. This will improve the creation of tiles.
* *Optional:* Additional features can be added to the *lakes\_forsim.shp* to exclude some area from the hydraulic simulations.
* Add 3 fields in *lakes\_forsim.shp* to store the z20, z100, and z350 calculated in *lakemodel.xls* for each lake, as well as for the most downstream polygon.

## Simulation type

Hydraulic simulation for streams larger than one cell must be done with the SuperGC technique, while hydraulic simulation for streams less than one cell wide must be done with the SubGC technique. Identifying which stream must be run with the SuperGC or the SubGC technique is done by modifying the channel mask (see step 5.6).

## Tiling

The aim of this step is to divide the stream into multiple, smaller reaches, on which Lisflood is run independently, in order to ease processing. The tool to be used is *Découpage en zones.*

*Inputs:*

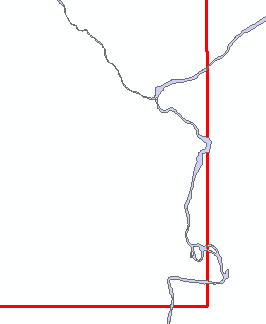
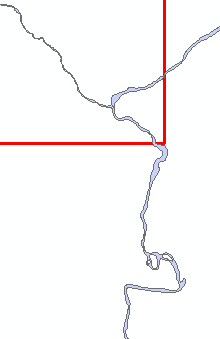
*DEMs\10m\lidar10m\_fd*

*lakes\lakes\_forsim.shp*

*dep\_pts.shp*

*simulations\zones (NB: create a new empty folder called “zones” to store all outputs of this tool)*

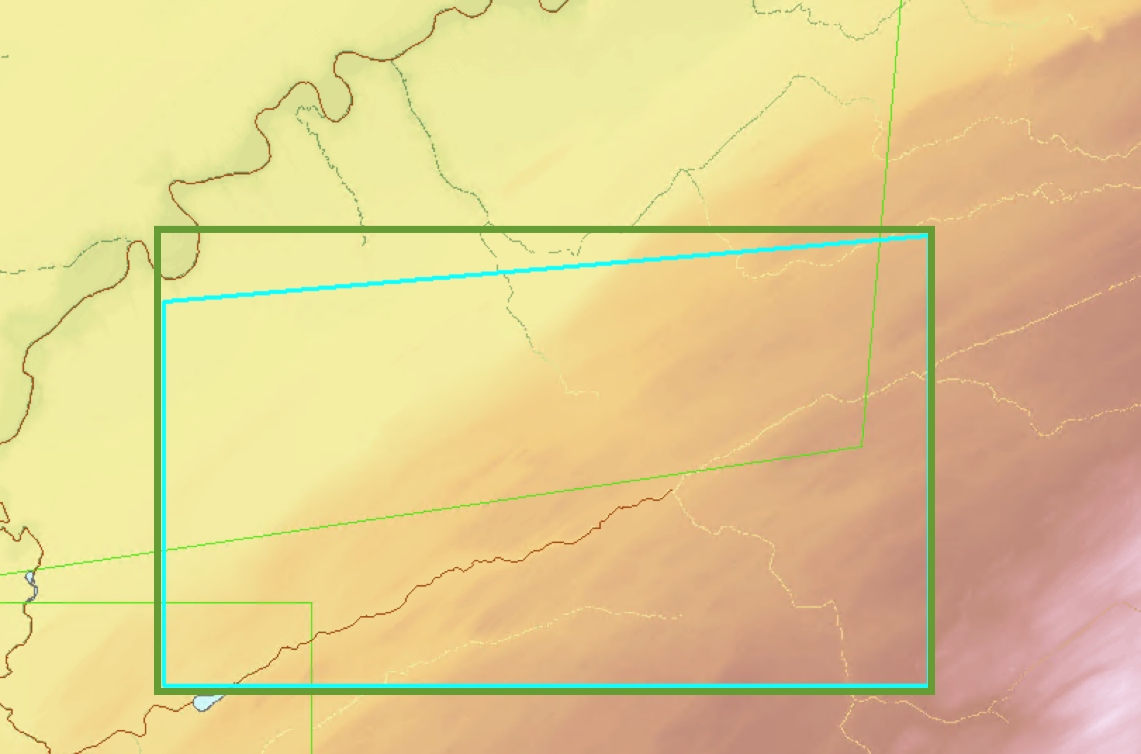
Two shapefiles are created by this tool, in the “zones” folder: *polyzones.shp* and *sourcepoints.shp*. The tiles are represented by the polygons in *polyzones.shp*. They must be checked and corrected if necessary. In particular, at confluences, a zone should not cut into meanders (see image below). Also, check in the attribute table of the polyzones that the “Lake\_ID” corresponds to the lake that the zone (defined by the “grid\_code”) is stopping at. Manually edit this if this is not the case.



*Corner of a zone (used for the tributary in the upper-right corner) that need to be corrected, and the corrected zone.*

The sourcepoints determine where the processing starts for that zone and should be located inside each polyzone. When editing the polyzones, ensure that the sourcepoint is inside it and located on the flow path defined by the flow accumulation raster.

If zones are to be deleted, the corresponding source point should be deleted as well.

The simulation tiles will be based on the envelope of the shapes in *polyzones.shp,* which will be created in the next step. The modified polyzones do not have to be perfectly rectangular, but attention must be given to their shape in relation to how the envelope is going to be created:

*Turquoise= modified polyzone; Green=envelope that will be created.*

## Hydraulic simulations with LISFLOOD-FP

The hydraulic simulations are run with two steps.

### Preparation of the simulations

* *Préparation des simulations hydrauliques*

*Inputs:*

*DEMs\10m\lidar10m\_fd*

*DEMs\10m\lidar10m\_facc*

*Simulations\zones*

*DEMs\10m\lidar10m\_avg*

*Width\width*

*Bedelevation\bed*

*ManningFP\n\_floodplain*

*Mask\mask*

*Simulations\simulations (NB: create a new empty folder to store all outputs of this tool. This will also store the final results. Name should be indicative of simulation done, e.g. sim\_bedconcm2)*

This tool creates additional files in the “zones” folder as well as the input files to be used by Lisflood in the simulation folder (all variables clipped according to the zones in .txt format, .bci and .par files). Furthermore, it creates the inbci points (points of discharge) and outbci points (points of elevation). *NB: See Lisflood manual for more details about .bci, .par files.*

### Running of the simulations

The simulations can be run in two modes: classical (using rasters of spatialized flood discharges of a given recurrence) or forecasting (using a csv file of a range of specific discharges).

The classical mode can be run with a high flood discharge in order to test the model and anticipate any issues, before running multiple discharges in the forecasting mode.

*NB: The data provided along this documentation contains only files related to the forecasting mode.*

#### Classical mode:

* *Lancement des simulations avec LISFLOOD-FP*

*Inputs:*

*Simulations\zones*

*Simulations\simulations*

*filepath containing Lisflood executable on your computer (e.g.* *C:\Users\user\_x\LISFLOOD\_FP\_v7c\LISFLOOD\_FP\_v7c)*

*q\qflood\q20*

*lakes\lakesforsim.shp (Field for boundary condition z20)*

*bedelevation\bed*

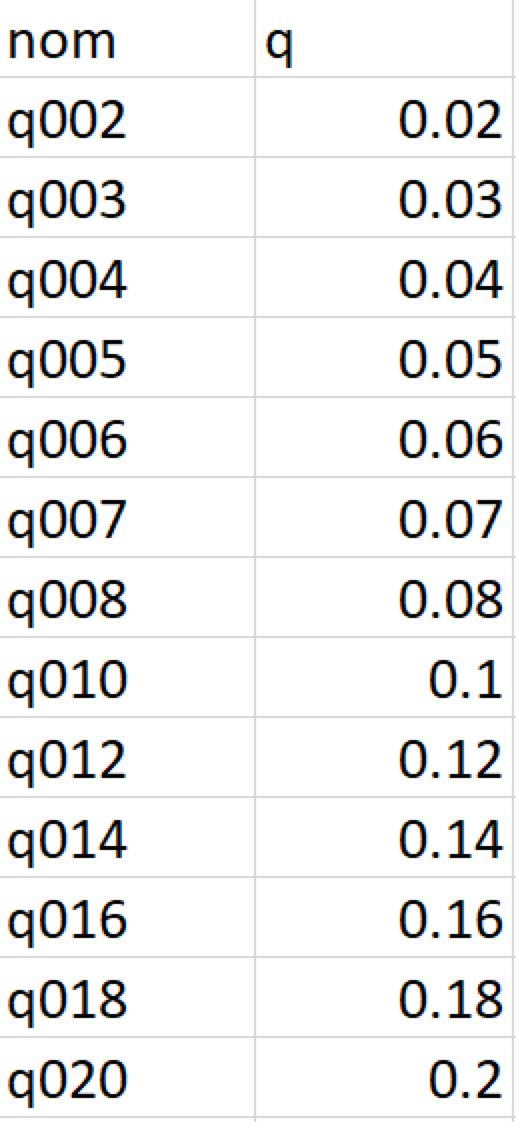
This tool writes the .bdy files for each zone. It creates a “res” folder where the simulations are being run. Here, the .mass files can be copy-pasted to check the status of the simulation (i.e. compare Qin and Qout to see if and how long until the steady state is reached). This tool also converts the Lisflood results (i.e. each elev\_zone#, the merged lisflood\_res and velocities) back into ArcGIS format.

“lisflood\_res” represents the merged end result of the simulation and contains flood elevation values in meters. *(NB: It is created by Mosaic to New Raster of all elev\_zone# rasters, by using the maximum as a merging rule.)*

#### Forecasting mode

In the forecasting mode, a csv file of specific discharges (discharge per km2 of drainage area) is provided to Lisflood and simulations are run automatically for this range of discharges. Two aspects need to be adjusted: the discharge and the downstream boundary conditions for lakes and the outlet. The simulations can be run using the same simulation folder created with *Préparation des simulations hydrauliques* in step 6.5.1.

The csv containing the specific discharges needs to have the format illustrated below. An example “qspecific.csv” is also included in the data.



The range of specific discharges is, in this case, defined by a lower limit of 0.02 m3.s-1.km-2 to an upper limit determined by the maximum q350 Hydrotel value of the study watershed. To calculate the value:

* open *qpts\_snapped.shp* and create a new field “q350spec” (Field type = Double). Calculate field by dividing q350 by the drainage area:

*q350spec = [Q350\_50e] / [Superficie]*

* sort in decreasing order and the maximum q350spec will be the upper limit of the specific discharges (i.e. in this case, the range is 0.02-1.12).
* add as many values in between as needed. For example, the current methodology follows this rule, which roughly allows 30cm of elevation difference between individual flood maps:
  + - increase of 0.01, from 0.02 to 0.08 m3.s-1.km-2
    - increase of 0.02, from 0.08 to 0.5 m3.s-1.km-2
    - increase of 0.05, from 0.5 to the maximum

*NB: if the difference between individual flood maps is larger than 30cm for more than 1% of the linear network, an additional specific discharge has to be included. For the Etchemin Watershed, a smaller subset of this range was selected as an example.*

Similarly, the downstream boundary conditions must be updated to reflect the forecasting mode.

* open *lakesforsim.shp* and add a field called “qlidm03” (Field type = Double). Following the current methodology, the field is calculated by subtracting 0.3cm from the zlidar, as follows:

*qlidm03 = [zlidar] – 0.3*

*NB: If there is no zlidar in the lakesforsim.shp, copy information from lakemodel.xlsx into a new field “zlidar”. Zlidar needs to be estimated for the most downstream polygon as well, and then 0.3m subtracted.*

* *Lancement des simulations de prevision avec LISFLOOD-FP*

*Inputs:*

*Simulations\zones*

*Simulations\simulations*

*filepath containing Lisflood executable on your computer (e.g.* *C:\Users\user\_x\LISFLOOD\_FP\_v7c\LISFLOOD\_FP\_v7c)*

*simulations\qspecific.csv*

*lakes\lakesforsim.shp (Field for boundary condition zlidm03)*

*bedelevation\bed*

This version does not create a “res” folder, but instead stores the elev\_zone# results in separate folders for each specific discharge based on the name given in the qspecific.csv (e.g. q002). The watershed-wide, merged result is res\_# (e.g. res\_q002).