Large-scale flood mapping - Tutorial

July 2020

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

Julia Mazgareanu

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 Rivière Noire.

# 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.

## Data requirements

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

* LiDAR along with metadata about the LiDAR acquisition dates and flight paths as lines or polygons (e.g. <https://www.foretouverte.gouv.qc.ca>). 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>)
* Flood discharge. The discharge data from the “Atlas hydroclimatique du Québec méridional”, version 2018, 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
* QmedHydrotel.shp: Hydrotel data points containing median discharge
* QfloodHydrotel.shp: Hydrotel data points containing simulated discharges of a 20, 100 and 350 year return period
* 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

### Rivière Noire data

The *Example\_RiviereNoire* folder contains an example of the application of the large-scale floodmapping methodology, with all intermediary results.

# 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*), whil*e* 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 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
* Manning’s n values for the floodplain
* Channel width
* Bed elevation
* Channel mask
* Flood discharge

Then, the LISFLOOD-FP simulations can take place.

# Data preparation

## River polygons

A polygon of the wetted channel, when the LiDAR data were acquired, is required. Work is under progress to find a semi-automatic method to detect the channels. However, no automatic method can produce a result with the same details and level of confidence than manual digitalization of the channel.

* Digitalization of the streams can be done using the hillshade raster at 1m resolution (produced with *Hillshade*). For wide rivers (approximately 15m wide or more), polygons are available in the GRHQ (layer *RH\_S*, Field *TYPECE = 10 – Cours d’eau*). It is recommended to check and modify if necessary the GRHQ polygons for them to match better what is observed with the LiDAR data.
* At the downstream end of the river, the river polygon must extent further than the DEM.
* 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.

Output: channelpoly.shp

This document and the associate tools on github will be updated when a semi-automated method will be determined.

## Linear river network

This step aim 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 digitalization of the streams 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 the downstream and upstream ends)

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 With = the maximum width of the channels, not accounting for the lakes. 100m was used for the rivière Noire.

Output: temp\collapsed\_banks.shp

* + Edit the result to improve it

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 than 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 creating basic files used to 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: lidar10m\_avg

* If the LiDAR data do not cover the whole watershed:
  + *Mosaic to New Raster* to merge the coarse resolution DEMs one single DEM
  + *Project Raster* to change spatial reference of the resulting DEM to the same one than 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, than the 10m LiDAR DEM, in order for the 10m LiDAR DEM to be kept where the two DEMs overlapped.

This step was not done on the Rivière Noire 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\lidar10m\_burn

* *Fill*

Input: DEMs\lidar10m\_burn

Output: DEMs\lidar10m\_fill

* *Flow direction*

Input: DEMs\lidar10m\_fill

Output: DEMs\lidar10m\_fd

* *Flow accumulation*

Input: DEMs\ lidar10m\_fd

Output: DEMs\lidar10m\_facc

* *Clip* the flow direction and the flow accumulation results to the studied area (use *lidar10m\_avg* for the *Output Extent*) (not done on the rivière Noire as the LiDAR data covered the entire watershed)
* Create point shapefile and place it/them at the upstream end of your river line , on the flow path displayed by the flow accumulation

Output: dep\_pts.shp

## Exclusion of lakes and reservoir

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

### 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 river polygons (*channelpoly.shp*).

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 on small rivers, or 10 ha for bigger ones, can be used. The field *SUP\_AREA* can be used to select them and delete them from the lakes shapefile.

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 rivière Noire as there are 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 digitalized manually. Do identify the reservoir, a longitudinal profile of the water surface elevation can be drawn with the 3D Analyst toolbar.

On the rivière Noire, the dam controlling the Lac noir water level is the only dam in the *Répertoire des barrages*. The area upstream this dam was included in the polygon for the Lac noir from the GRHQ.

Output: lakes/lakes\_final.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/lidar10m\_fd, dep\_pts.shp

Parameter: Distance entre sections = 100

Output: 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: xsect.shp

* *Create Thiessen Polygons*

Input: xsect.shp

Output: 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.

Output: thiessen2.shp

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

Input: thiessen2. shp, river\_network.shp/ channelpoly.shp

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

* *Dissolve* according to *INPUT\_FID*

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

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

* *Add Geometry Attributes* to calculate length and area

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

* *Point to Raster* based on field "width"

Input: xsect\_minuslakes.shp

Output: ch\_width\_pts

! **Important:** In Environments Pane within tool, match Output Coordinates, Processing Extent and Snap to computed flow direction raster!

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

Input: flowdir\_b, lidar10m\_fill, dep\_pts.shp

Output: d4fd

* *Interpolation lineaire* to linearly interpolate obtained width points

Input: d4fd, dep\_pts.shp, ch\_width\_pts

Output: width

## 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* to generate a 3m DEM by the minimum

Input: DEMs/lid1m

Output: DEMs/lid3m\_min

* Identify the bridges that need to be corrected. This can be done by looking at the DEM hillshade. Create a polygon of the bridges.

Ouput: watersurface /bridges.shp

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

Input: DEMs/lid3m\_min, watersurface /bridges.shp

Output: watersurface /lidar3m\_br

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

Input:

watersurface /lidar3m\_br

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

channelpoly.shp

stream\_nerwork.shp

Output: watersurface /lidar3m\_forws

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

Input: watersurface /lidar3m\_forws

Intermediary result: watersurface /lidar3m\_fill

Output: watersurface/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 /lidar3m\_fd

dep\_pts.shp

watersurface /lidar3m\_fill

watersurface /lidar3m\_br

Output: watersurface/ws3m

* *Allocation euclidienne pour matrice à virgule flottante* to project laterally the water surface values

Input: watersurface/ws3m

Output: watersurface/ws3m\_alloc

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

Input: watersurface/ ws3m\_alloc

Bilinear resampling

Output: watersurface/ws10m

* *Clip*

Input: watersurface/ws10m

Extent: DEM/lidar10m\_avg

Parameter Maintain clipping extent = True

Output: watersurface/ws10m\_c

* *Bréchage*

Input:

watersurface/ws10m\_c

d4fd

dep\_pts.shp

Output: watersurface/ws10mbr\_FLT

## Water surface detection – method #2

* *Aggregate* individual LiDAR tiles from 1m to 5m 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)

* *Project Raster* to change spatial reference of tiles, if needed

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 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

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

Input: bridges.shp

Output: lidar5m\_corr

\*Note: there are no bridges along the river used as an example here.

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

Input: lidar5m\_corr

Output: comp10m

* *Fill*

Input: comp10m

Output: comp10m\_fill

* *Flow direction*

Input: comp10m\_fill

Output: comp10m\_fd

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

Input: comp10m\_fd, dep\_pts.shp, comp10m

Output: comp10m\_rawws

* *Lissage de la surface de l’eau*

Input: comp10m\_fd, dep\_pts.shp, comp10m\_rawws, comp10m

Output: comp10m\_wssm

* *Allocation euclidienne pour matrice à virgule flottante* to project laterally the water surface values

Input: comp10m\_wssm

Output: comp10m\_wssmb

* *Bréchage*

Input:

comp10m\_wssmb

d4fd

dep\_pts.shp

Output: watews10mbr\_CONC

## LiDAR discharge

* find dates when LiDAR was acquired
  + if flightpaths are lines, *Buffer* to transform to polygon strips using a 500m buffer (the span of a LiDAR survey is more or less 1km. Some metadata like the flight altitude and the scan angle can help obtaining a better assessment).
  + *Dissolve* polygons by date (uncheck *Create multipart features*)

Input: flightpath.shp

Output: qlidar.shp

* find discharge data of open watershed gauging stations corresponding to those dates from the CEHQ website
* create a new column in attribute table of dissolved flight lines polygons and insert discharge values (Field name = gaging station number)
* *spaghetti and meatballs* technique used on overlapping polygons (because it is not clear which day, and so which discharge, is associated with those areas where polygons are overlapping => an average discharge is calculated for those areas) - for more details see https://www.esri.com/arcgis-blog/products/arcgis-desktop/analytics/more-adventures-in-overlay-counting-overlapping-polygons-with-spaghetti-and-meatballs/
* *Feature to Line*

Input: flightpath.shp

uncheck Preserve attributes

Output: spagline.shp

* *Feature to Polygon* – these are polygons that do not overlap anymore

Input: spagline.shp

Output: spag.shp

* *Feature to Point*

Input: spag.shp

check Inside

Output: meatballs.shp

* *Spatial Join* (the geoprocessing tool, not by right-click -> *Joins and Relates*)

Target: meatballs.shp

Join: qlidar.shp

uncheck Keep all target features

right-click Q in Field Map of Join Features > Merge rule > Mean

Output: qlid\_meatball.shp

* *Join Field*

Input: spag.shp by FID

Join: qlid\_meatball.shp by ORIG\_FID

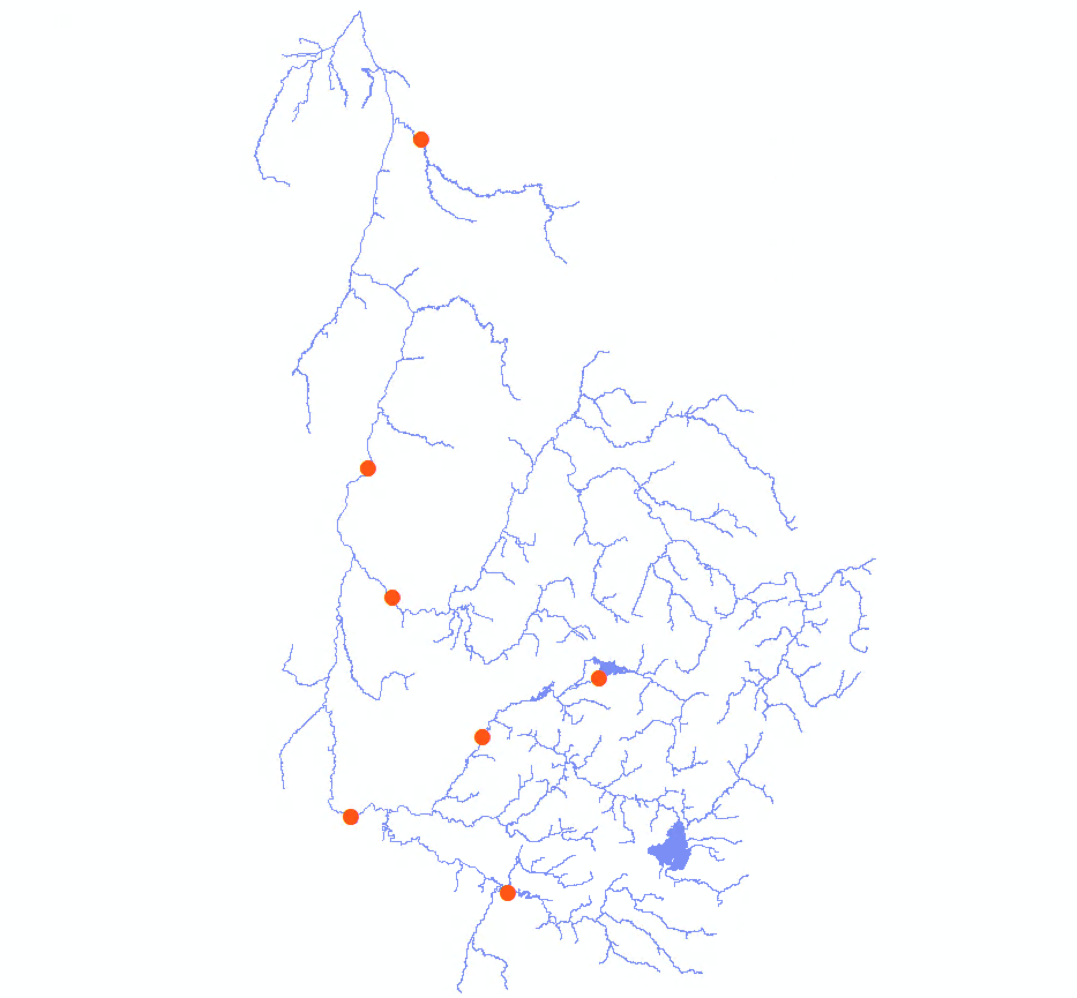
Join Field: discharge values by gaging station number

* *Polygon to Raster*

Input: spag.shp (fields storing discharges)

Output: stationQ

! **Important:** In Environments Pane within tool, match Output Coordinates, Processing Extent and Snap to computed flow direction raster!



**Note:** In case of more complex watershed network configurations, where the LiDAR discharge needs to be derived from multiple gauging stations (see example on the left), a special process involving spatial weighting based on the flow accumulation (basin area and distance from station) was used.

* From QmedHydrotel.shp, select only points inside of your study area and export to new shapefile
* manually snap these Hydrotel points to the flow accumulation raster, just upstream the confluence, on each stream

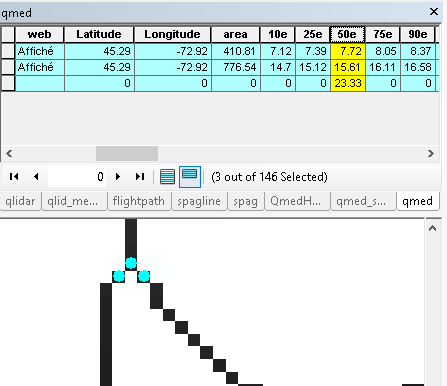
Output: qmed\_snapped.shp

NB: There are no confluences in the discharge data for the rivière Noire

* *Extract Multi Values to Points*, to add the flow accumulation to the points
* Add a new field named *qmed* (*Add Field*, type = Double), and adjust the discharges by proportionality to the drainage area, using *Field Calculator*:

*Qmed = [50e]/ [area]\* [lidar10m\_f]/10000.*

* Add a point downstream of confluences, with the sum of the two discharges upstream for *qmed*

Example of points at a confluence, with flows going from south to north. The last line in the attribute table, with the sum of the discharge values of the two other points, is the point after the confluence (NB: the correction of the discharges by the flow accumulation values wasn’t done in that case)

* *Point to Raster* snapped Hydrotel points based on Field *qmed*

Input: qmed\_snapped.shp

Output: qmed\_pts

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

Input:

Lidar10m\_fd

dep\_pts.shp

lidar10m\_facc

qmed\_pts

Check Interpolation entre les points de débit

Output: qmed

* *Raster* *Calculator* to divide result by interpolated discharge value at the gauging station to obtain a spatial variation rule of the discharge over the study area

"qmed"/3.416566

Output: q\_spvar

* *Raster Calculator* to multiply this result with the Q obtained from the LiDAR to calculate a normalized LiDAR discharge.

"q\_spvar"\*"stationq"

Output: 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: qlidar

Distance: 100

Output: qlidard4

## Bed elevation

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

Input:

d4fd

dep\_pts.shp

width

ws10mbr

qlidard4

Output: bed

## Manning’s n for floodplain

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

Output: 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

Input: landuse\_clip.shp

Output: n\_floodplain

! **Important:** In Environments Pane within tool, match Output Coordinates, Processing Extent and Snap to computed flow direction raster!

## Simulation type

Hydraulic simulation for streams larger than one cell must be done with the SuperGC technic, while hydraulic simulation for stream less than one cell wide must be done with the SubGC technic. Identify which stream must be run with the SuperGC technic or the SubGC technic can be done by cutting down appropriately the linear river network and adding this information in a new field.

* Add Field (Type = Short Integer) to river\_network.shp, name: SuperGC
* Display width, and change the symbology for “Classified”. Use 2 classes, with a break value at 15
* Edit the river\_network.shp lines. Set the field SuperGC to 2 for every line. Set which reaches are wider than 15m (SuperGC = 1), and which reaches are narrower than 15m (SuperGC = 0). Cut the lines if necessary, but avoid doing too small reaches (less than 1kmlong roughly). Setting initially the field to 2 can help keeping track of which reach has been assessed during the process.

## Channel mask

* *Polygon to Raster* to transform the channel polygon into a raster

Input: channelpoly.shp

Output: frompoly

! **Important:** In Environments Pane within tool, match Output Coordinates, Processing Extent and Snap to computed flow direction raster!

* *Raster Calculator* to have 1 where the raster made from the channel polygon have a value, or where the D4 have a value (for streams to be of at least one cell)

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

Output: mask\_temp

* *Raster to Polygon* to transform back the channel mask into a polygon

Input: mask\_temp

Output: mask\_poly.shp

* Display *river\_network.shp* so that it appears with two different colors, depending on the value of the field SuperGC (in the menu *Properties* -> *Symbology* -> *Categories*)
* Edit the *mask\_poly.shp* file to cut out reaches that will be modelled with the SubGrid technic (SuperGC = 0)
* *Polygon to Raster* to transformback the result in a raster

Input: mask\_poly.shp

Output: mask

! **Important:** In Environments Pane within tool, match Output Coordinates, Processing Extent and Snap to computed flow direction raster!

## Flood discharge

* *Join* QfloodHydrotel\_max, using the field *station,* to qmed\_snapped, using the field *id2*. Choose the option *Keep only matching records*.
* Export the result

Output: qflood\_snapped

* Add three new field named *q20cor, q100cor, q350cor* (*Add Field*, type = Double), and adjust the discharges by proportionality to the drainage area, using *Field Calculator*:

*Ex: q20cor = [MAX\_q20]/ [area]\* [lidar10m\_f]/10000.*

* *Point to Raster* snapped Hydrotel points based on discharge field for each scenario (i.e. q20cor, q100cor, q350cor)

Input: QfloodHydrotel.shp, Field q20

Output: q20\_pts

Repeat for q100 and q350

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

Input:

Lidar10m\_fd

dep\_pts.shp

lidar10m\_facc

q20\_pts

« Interpolation entre les points de débit » must be unchecked

Output: q20

Repeat for q100 and q350

# Simulations (Méthode 2D)

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

## Slope of the water surface

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

Input: ws10mbr

Distance: 100

Output: ws10mbr\_b

* *Pente* using *ws10mbr\_b* as a substitute to the DEM as the surface from which to calculate the slope

Input:

ws10mbr\_b

lidar10m\_fd

dep\_pts.shp

200 for the “Distance de pente”

Output: slope and slope\_fp.shp

## 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 yeasr 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. 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

Output: lakes\_forsim.shp

* Create an additional feature the *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.
* Create small polygons, roughly rectangle, where rivers flow into lakes. This will improve the creation of tiles.
* Additional features can be added to the *lakes\_forsim.shp* to exclude some area from the hydraulic simulations.
* Add 3 fields to store the z20, z100, and z350 in the lake polygons at the end of river reaches.

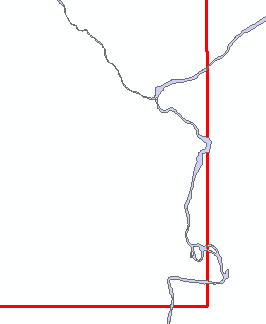
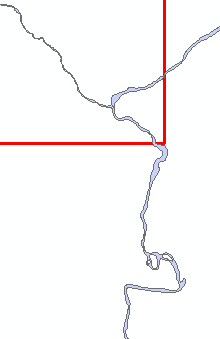
In the case of the rivière Noire example, arbitrary values were used for the downstream end.

## 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*



Two shapefiles are output by this tool, in the « zone » 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. The simulation tiles will be based on the envelope of the shapes in *polyzones.shp*.



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

## Hydraulic simulations with LISFLOOD-FP

The hydraulic simulations are run with two steps.

* *Préparation des simulations hydrauliques*



This tool creates additional files in the “zones” folder as well as the input files to be used by Lisflood in the output folder, with the exception of the .par and the .bdy files.

* *Lancement des simulations avec LISFLOOD-FP*

