Notes from Dr. Tarboton:

The following logic is used to implement this approach

Inputs are a stream vector file (e.g. NHD or NHD High Res) and a DEM (denoted z).

1. Convert the stream vectors to a raster that shares the same dimensions (columns, rows, cell size and edge coordinates as the DEM), denoted srfv (stream raster from vector) with values 1 on stream and 0 off stream
2. Burn srfv into z using raster calculator zb = z-100 \* srfv (or any big number in place of 100)
3. Fill pits in burned dem. Result is zbfel
4. D8FlowDir with input zbfel and output flow directions p
5. Mask D8 flow directions to only have flow directions on streams. The raster calculation pm = p/srvf achieves this.
6. Apply new TauDEM flow direction conditioning tool "Flowdircond" that operates on pm and z and produces zfdc (flow direction conditioned elevations) by tracking down the D8 flow directions and ensuring there is no uphill elevation.

Code Implementation:

Step 0: Select the high resolution flowlines for any given huc unit

echo "==: create high resolution flowline shp from HRNHDPlus"

echo "=CMD= ogr2ogr ${n}-hrflows.shp $dsnhdhr NHDFlowline -where \"REACHCODE like '${hucid}%'\""

Tstart

[ ! -f "${n}-hrflows.shp" ] && \

ogr2ogr ${n}-hrflows.shp $dsnhdhr NHDFlowline -where "REACHCODE like '${hucid}%'" \

&& [ $? -ne 0 ] && echo "ERROR creating high resolution flowline shp." && exit 1

Tcount hrflowline

Step 1 in Dr. Tarboton’s note: Convert the stream vectors to a raster

python Flowline\_Rasterize.py -flowline ${n}-hrflows.shp -DEM ${n}.tif -srfv ${n}srfv.tif

Step 2 in Dr. Tarboton’s note: Burn srfv into z using raster calculator zb = z-100 \* srfv

python gdal\_calc.py -A ${n}.tif -B ${n}srfv.tif --outfile=${n}zb.tif --calc="A-100\*B"

Step 3 in Dr. Tarboton’s note: Fill pits in burned dem

echo "==: taudem burnin pitremove"

echo "=CMD= mpirun -np $np $taudem/pitremove -z ${n}zb.tif -fel ${n}zbfel.tif"

Tstart

[ ! -f "${n}zbfel.tif" ] && \

mpirun -np $np $taudem/pitremove -z ${n}zb.tif -fel ${n}zbfel.tif \

&& [ $? -ne 0 ] && echo "ERROR creating pitremove burnin DEM." && exit 1

Tcount pitremove

Step 4 in Dr. Tarboton’s note: D8FlowDir with input zbfel and output flow directions p

echo "==: taudem burnin d8"

echo "=CMD= mpirun -np $np $taudem/d8flowdir -fel ${n}zbfel.tif -p ${n}bp.tif -sd8 ${n}bsd8.tif "

Tstart

[ ! -f "${n}bp.tif" ] && \

mpirun -np $np $taudem/d8flowdir -fel ${n}zbfel.tif -p ${n}bp.tif -sd8 ${n}bsd8.tif \

&& [ $? -ne 0 ] && echo "ERROR creating burnin d8 raster." && exit 1

Tcount d8

Step 5 in Dr. Tarboton’s note: Mask D8 flow directions to only have flow directions on streams

python gdal\_calc.py -A ${n}bp.tif -B ${n}srfv.tif --outfile=${n}bpm.tif --calc="A/B"

Step 6 in Dr. Tarboton’s note: Apply new TauDEM flow direction conditioning tool "Flowdircond"

mpiexec -n 4 flowdircond -z ${n}.tif -p ${n}bpm.tif -zfdc ${n}zbfdc.tif

After the process is completed, we should replace the original DEM with the output file named “${n}zbfdc.tif” in our previous HAND workflow