GRASS GIS 7:
A theoretical and practical course

Session 1 – Hydrological applications

Markus Neteler

NINA 2016, Oslo, Norway

mundialis GmbH & Co. KG
http://www.mundialis.de

Session Objectives

Overview: Watershed and hydrological modelling

- Flow calculation
- Groundwater flow
- Hydrological models
- Sediment modules
- Stream modules
- Watershed modules

Overview: Flow calculation

Flow calculation

- Stream preparation (carving) – r.carve
- Flow tracing – r.drain
- Flow computation for massive grids – r.watershed, r.terraflow
- Filling of no-data areas and sink-filling (note: not needed for watershed modelling) – r.fill.dir
- Calculation of flowlines, flowpath lengths, and flowline densities (upslope areas) – r.flow
- Calculation of topographic index and topographic wetness index) – r.topidx

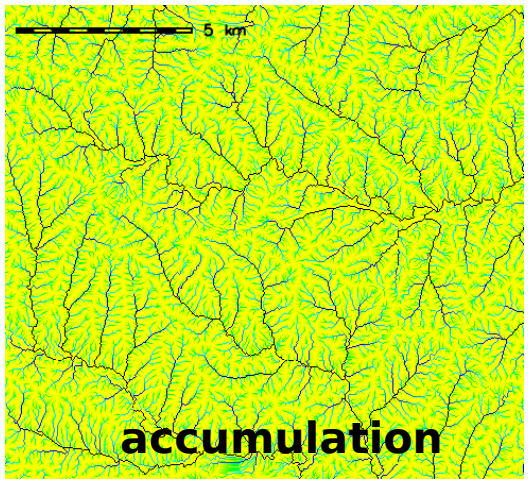
Addons:

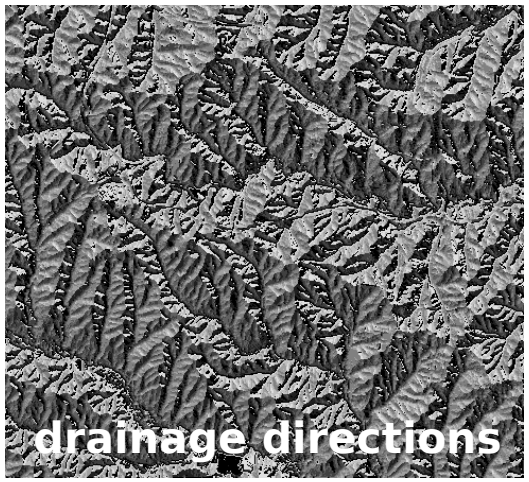
- Hydrological conditioning, sink removal – r.hydrodem
- Computation of the travel time of surface runoff to an outlet – r.traveltime
- Calculation of the longest flow path – r.lfp

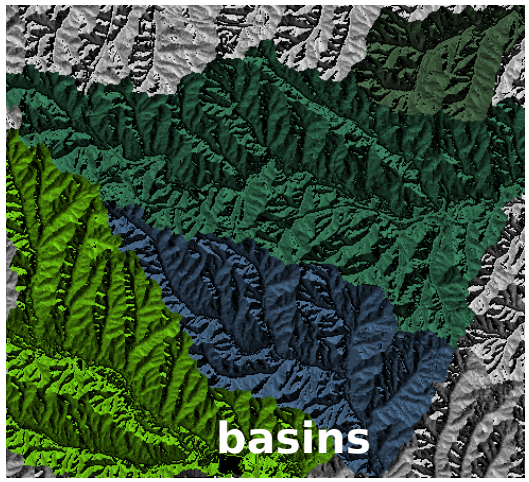
Exercise: Flow calculation 1/3

Flow calculation

```
g.region raster=elev_ned_30m -p
# we use single flow directions for this task (SFD) with -s:
r.watershed -s elev_ned_30m threshold=10000 \
  accumulation=accum_10k drainage=draindir_10k basin=basin_10k
```








```
d.mon wx0
d.rast accum_10k
d.rast draindir_10k
# shaded basin map:
d.shade color=basin_10k shade=draindir_10k
```

accumulation: Number of cells that drain through each cell (overland flow)
- see r.watershed

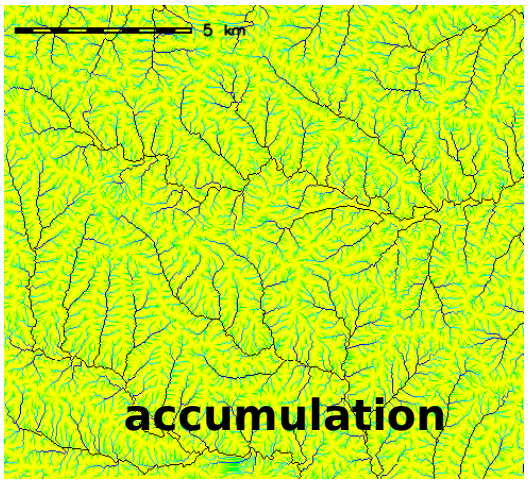
Exercise inspired by NCSU

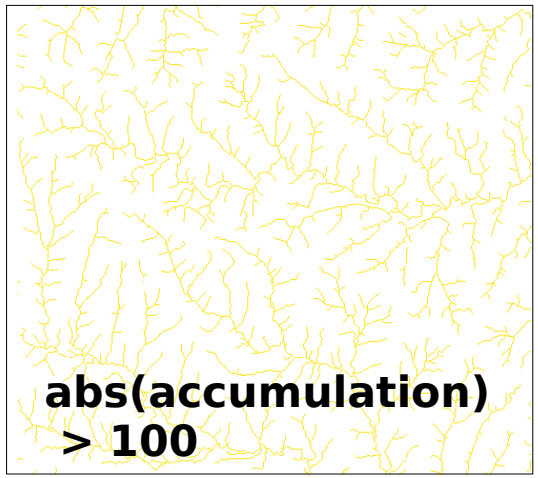
Exercise: Flow calculation 2/3

Flow calculation

Extract more detailed streams from flow accumulation raster:

```
r.mapcalc "streams_acc10k = if(abs(accum_10k) > 100, 1, null())"
```





Convert extracted raster streams to vector model:

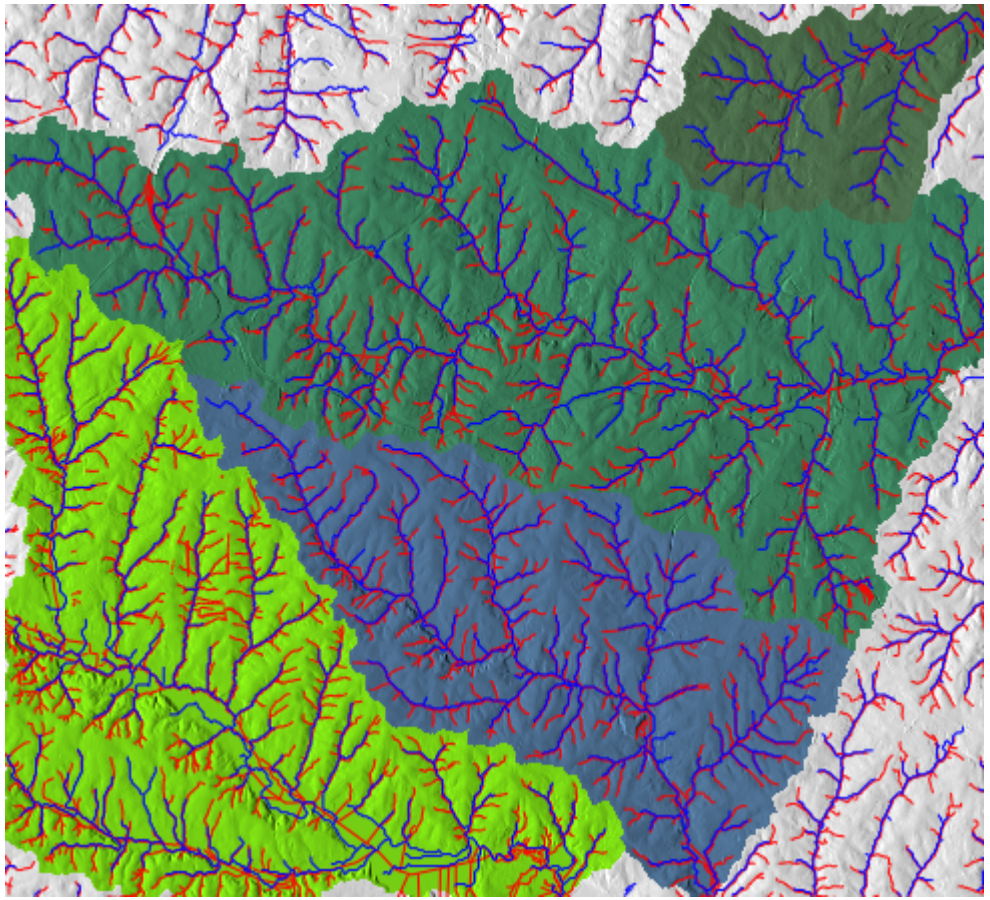
```
r.to.vect -s basin_10k output=basin_10k type=area
r.thin streams_acc10k output=streams_acc10k_thin
r.to.vect -s streams_acc10k_thin output=streams_acc10k type=line
```

Exercise: Flow calculation 3/3

Flow calculation

Compare to official Wake county streams (red):

```
d.shade color=basin_10k shade=elevation_shade brighten=40
d.vect basin_10k type=boundary
d.rast lakes
d.vect streams_acc10k color=blue
d.vect streams color=red
d.out.file mystreams
```



Questions:

- How do the DEM derived streams compare with the official stream map? (check also the “accum_10k” map)
- How to modify the mapcalc expression to make stream origins fit better with the official stream map?

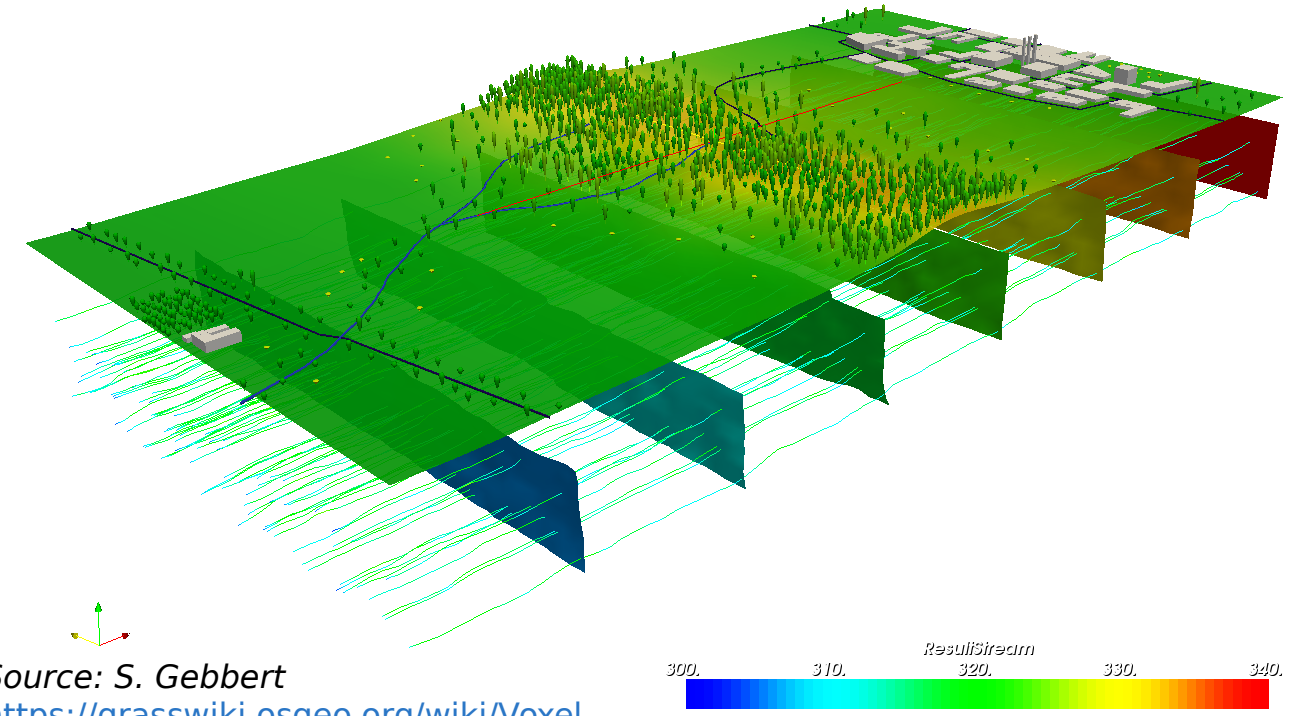
Overview: Groundwater flow

2D groundwater flow

- Numerical calculation program for transient, confined and unconfined groundwater flow in two dimensions – r.gwflow

3D groundwater flow

- Numerical calculation program for transient, confined groundwater flow in three dimensions – r3.gwflow



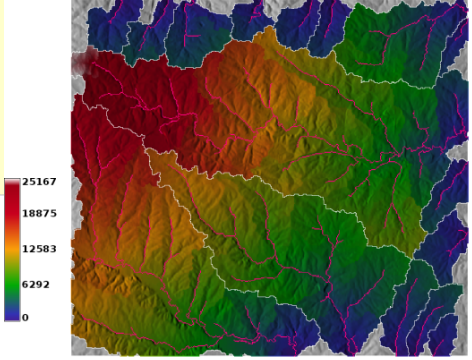
Source: S. Gebbert
https://grasswiki.osgeo.org/wiki/Voxel

Overview: Hydrological models and Sediment transport

Available models

- TOPMODEL which is a physically based hydrological model – r.topmodel
- HydroFOSS: A distributed, physically based hydrological model (GRASS GIS 6 only)
- Overland flow hydrologic simulation using path sampling method (SIMWE) – r.sim.water
- Sediment transport and erosion/deposition simulation using path sampling method (SIMWE) – r.sim.sediment

Overview: Stream modules



Stream calculation

- Performs stream network extraction – r.stream.extract

Addons – see also https://grasswiki.osgeo.org/wiki/R.stream.*_modules

- Delineates **basins** according stream network – r.stream.basins
- Calculates **local parameters** for individual streams – r.stream.channel
- Calculates **distance** to and elevation above streams and outlet – r.stream.distance
- Calculates **Strahler**'s and more streams hierarchy – r.stream.order
- Divides **network** into near straight-line segments and calculate its order – r.stream.segment
- Calculates local parameters for **slope** subsystem – r.stream.slope
- **Snap** point to modelled stream network – r.stream.snap
- Calculates **Horton**'s statistics for Strahler and Horton ordered networks created with r.stream.order – r.stream.stats

Exercise: Stream modules

Stream network extraction with r.stream.extract 1/3

```
# set region
g.region -p raster=elev_ned_30m

# calculate flow accumulation
r.watershed elevation=elev_ned_30m accumulation=elev_ned_30m_acc

# curvature to get narrow valleys
r.param.scale input=elev_ned_30m output=tangential_curv_5 size=5 \
  method=crosc

# curvature to get a bit broader valleys
r.param.scale input=elev_ned_30m output=tangential_curv_7 size=7 \
  method=crosc

# curvature to get broad valleys
r.param.scale input=elev_ned_30m output=tangential_curv_11 size=11 \
  method=crosc
```

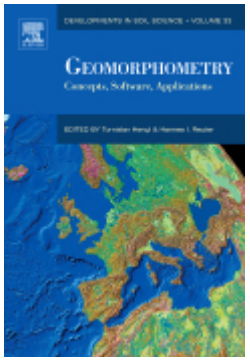
Curvatures

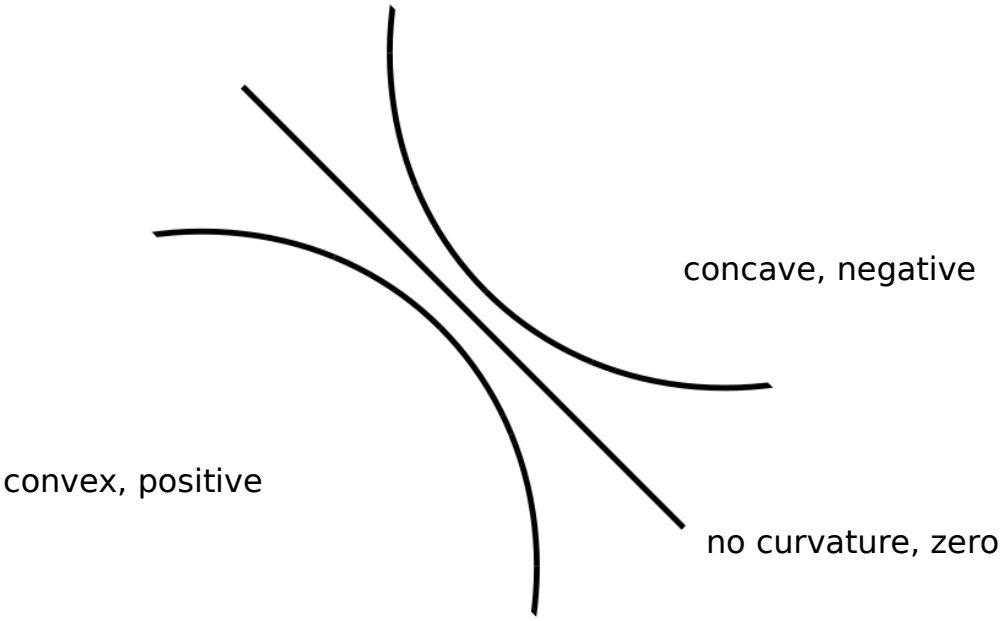
Profile curvature

curvature in the direction of steepest slope

Tangential curvature

curvature in the direction of the contour tangent





*"The **profile** (or vertical) curvature and **tangential** (or horizontal) curvature can be used to distinguish (locally) convex and concave shapes.*

***Concave tangential** curvature indicates convergence and convex divergence of flow lines.*

***Convex profile** curvature indicates acceleration of flow."*

Geomorphometry: Concepts, Software, Applications Edited by Tomislav Hengl and Hannes I. Reuter, 2009, p.150

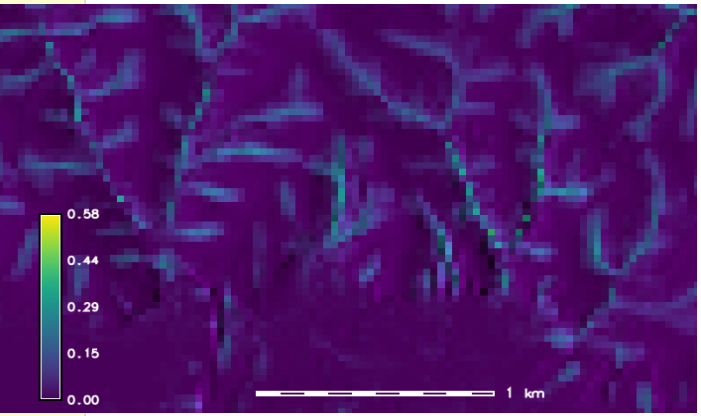
Exercise: Stream modules

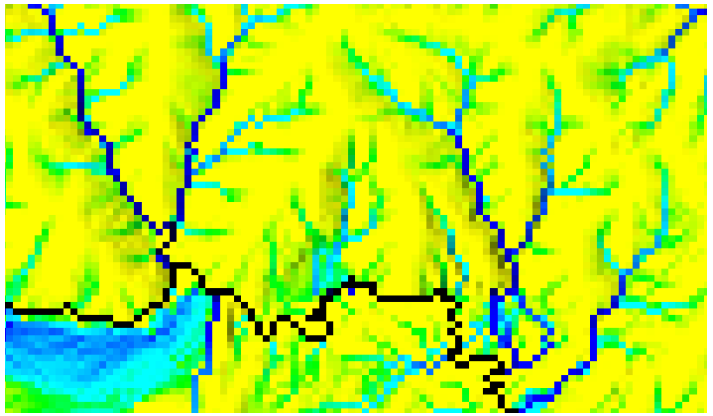
Stream network extraction with r.stream.extract 2/3

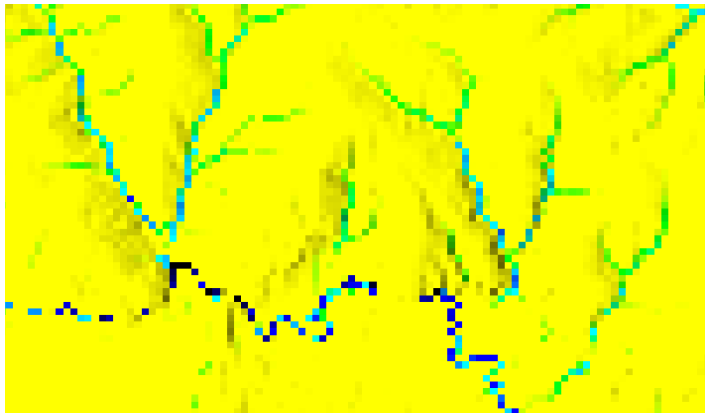
```
# create weight map (from multi-resolution input, nested if() condition):
r.mapcalc "weight = if(tangential_curv_5  < 0, -100.0 * tangential_curv_5, \
    if(tangential_curv_7  < 0, -100.0 * tangential_curv_7, \
    if(tangential_curv_11 < 0, -100.0 * tangential_curv_11, 0.000001)))"

# weigh accumulation map
r.mapcalc "elev_ned_30m_acc_weighed = elev_ned_30m_acc * weight"

# copy color table from original accumulation map
r.colors map=elev_ned_30m_acc_weighed raster=elev_ned_30m_acc
```







weight map | original accumulation | weighed accumulation

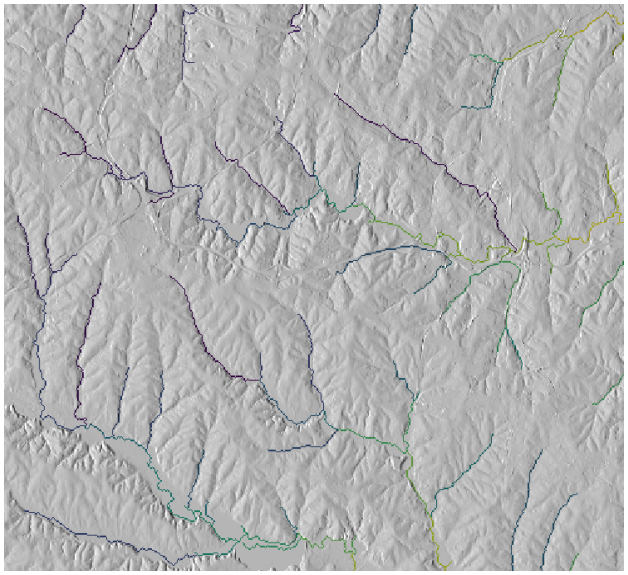
Visualided with:

```
d.shade color=elev_ned_30m_acc shade=elevation_shade brighten=80
```

Exercise: Stream modules

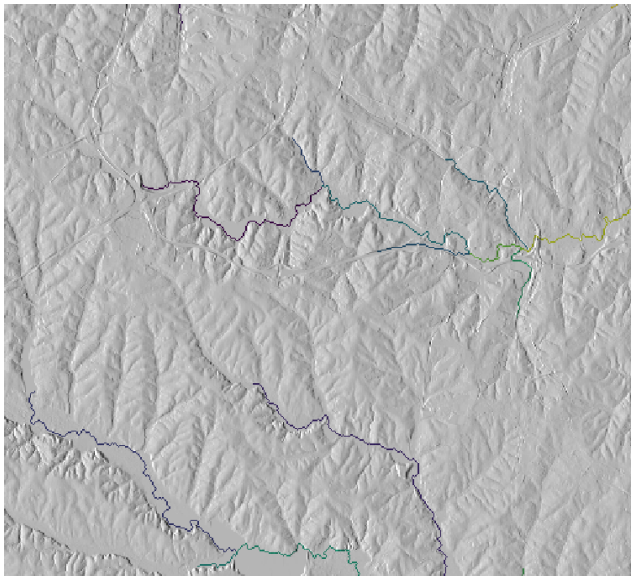
Stream network extraction with r.stream.extract 3/3

```
# extract streams using the original accumulation map
r.stream.extract elevation=elev_ned_30m \
    accumulation=elev_ned_30m_acc \
    threshold=1000 \
    stream_rast=elev_ned_30m_streams_noweight
```



Important: slightly zoom into map to avoid NULL cells

```
# extract streams from weighed map
r.stream.extract elevation=elev_ned_30m \
    accumulation=elev_ned_30m_acc_weighed \
    threshold=1000 \
    stream_rast=elev_ned_30m_streams
```



Overview: Watershed modules

Watershed (basin) calculation

- Generates a raster map layer showing watershed subbasins – r.basins.fill
- Generates a watershed basin from a drainage direction map (from r.watershed) and a set of coordinates representing the outlet point of watershed – r.water.outlet
- Watershed basin analysis program – r.watershed
- Fills a lake to a target water level from a given start point – r.lake

Addon:

- Generates the main morphometric parameters of the basin – r.basin

Exercise: Watershed modules

Watershed (basin) calculation from outlet

```
# set computational region to raster map “elevation”
g.region raster=elevation -p

# calculate drainage directions
r.watershed elevation=elevation drainage=drain_directions
# calculate outlet point related watershed
r.water.outlet input=drain_directions output=basin \
  coordinates=642455,222614

# generate outlet vector point
echo "642455,222614" | v.in.ascii -n input=- output=outlet separator=","

# visualize
d.shade shade=drain_directions \
  color=basin
d.vect outlet color=red \
  icon=basic/diamond size=10
```