

```

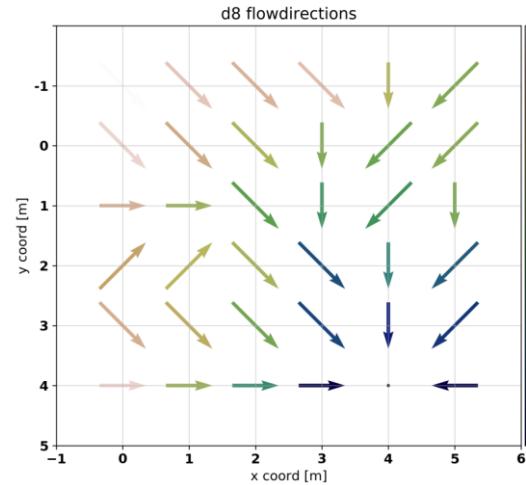
assert len(idx) == len(x.shape)

if np.isscalar(radius): radius = tuple([radius
                                         for i in range(len(x.shape))])

slices = []
paddings = []
for axis in range(len(x.shape)):
    if idx[axis] is None or radius[axis] is None:
        slices.append(slice(0, x.shape[axis]))
        paddings.append((0, 0))
        continue

    r = radius[axis]
    l = idx[axis] - r
    r = idx[axis] + r

```



# FlowDir\_v010

Sebastian Multsch, Florian Krebs

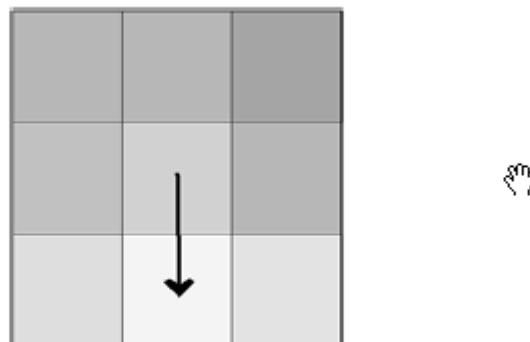
14.08.2019

# Single-neighbour flow

## 3. FLOW ALGORITHMS

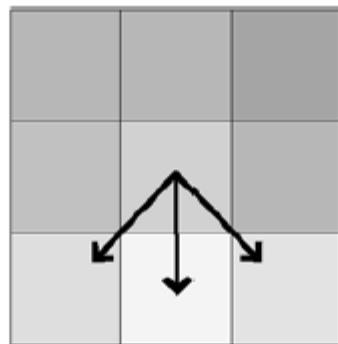
### 3.1 Single-neighbour flow algorithms

The most basic flow algorithm is the so-called “D8”, sometimes referred to as *method of the steepest descent* (O’Callaghan and Mark, 1984). From each cell, all flow is passed to the neighbour with the steepest downslope gradient (Figure 1) resulting in 8 possible drainage directions — hence the name D8. It can model convergence (several cells draining into one), but not divergence (one cell draining into several cells). Ambiguous flow directions (the same minimum downslope

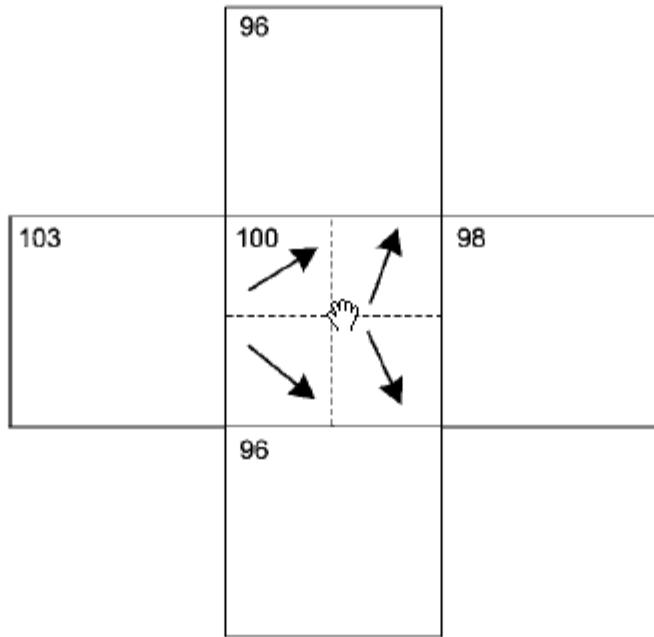


**FIGURE 1** Single flow direction assigned to the central pixel in a  $3 \times 3$  neighbourhood. Grey values represent elevation increasing with darkness of the cell.

## Multiple-neighbour flow



**FIGURE 2** Multiple flow directions assigned to the central pixel in a  $3 \times 3$  neighbourhood using MFD. Grey values represent elevation increasing with darkness of the cell. Multiple flow directions are assigned and a fraction of the mass of the central cell is distributed to each of the three lower cells that the arrows point to. All mass fractions together must sum to one in order to conserve mass.



**FIGURE 4** Flow directions assigned to quarter pixels using the Mass-Flux Method. Numbers refer to pixel elevations in this example. © 2005 Rivix LLC, used with permission.

# Triangulated irregular network (TIN) (voronoi cells)

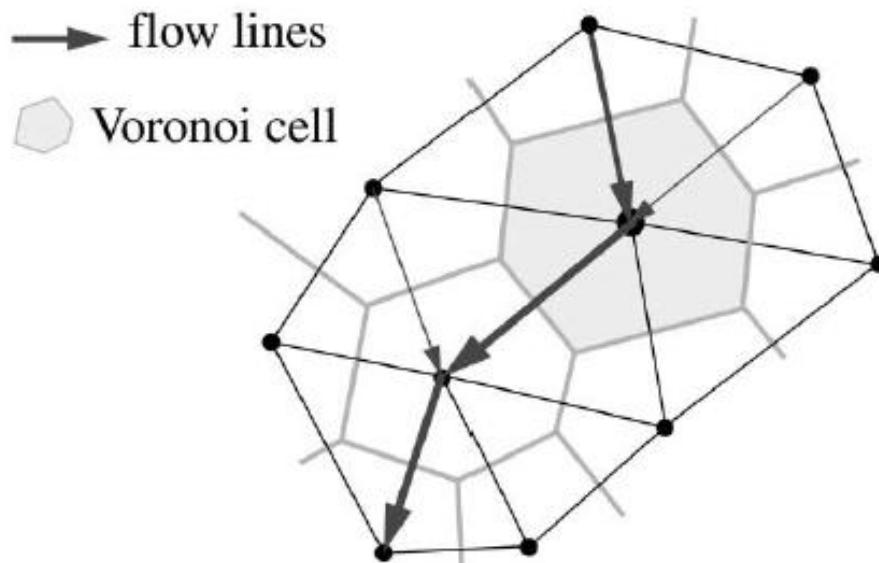
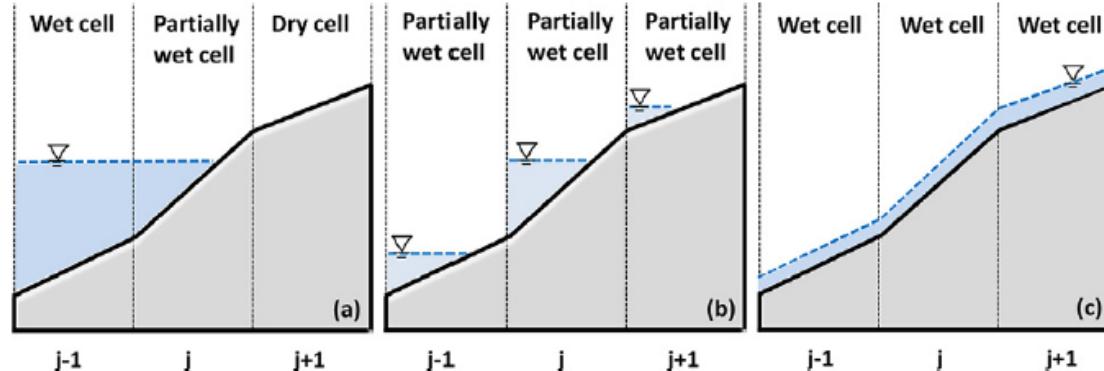


Fig. 5. Illustration of steepest-descent flow routing in TIN framework.

# Water depth



**Fig. 1.** Flow regimes near the wet and dry boundary on a sloped bed: (a) the quiescent flow regime characteristic of hydraulic applications, such as flooding and wave runup; (b) an incorrect representation of flow regime using the concept (a) for hydrological applications; and (c) the sheet flow representation of a flow regime for hydrological applications with low runoff on a steeply sloped bed.

# ArcGIS flow directons



There are eight valid output directions relating to the eight adjacent cells into which flow can approach presented in Jenson and Domingue (1988).

78	72	69	71	58	49
74	67	56	49	46	50
69	53	44	37	38	48
64	58	55	22	31	24
68	61	47	21	16	19
74	53	34	12	11	12

Elevation surface



2	2	2	4	4	8
2	2	2	4	4	8
1	1	2	4	8	4
128	128	1	2	4	8
2	2	1	4	4	4
1	1	1	1	4	16

Flow direction

32	64	128
16		1
8	4	2

Direction coding

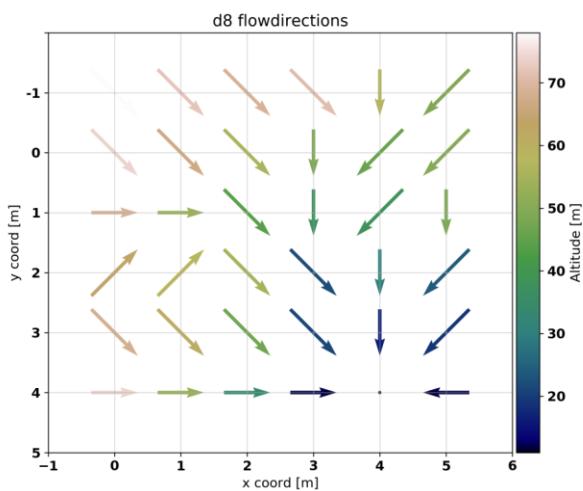
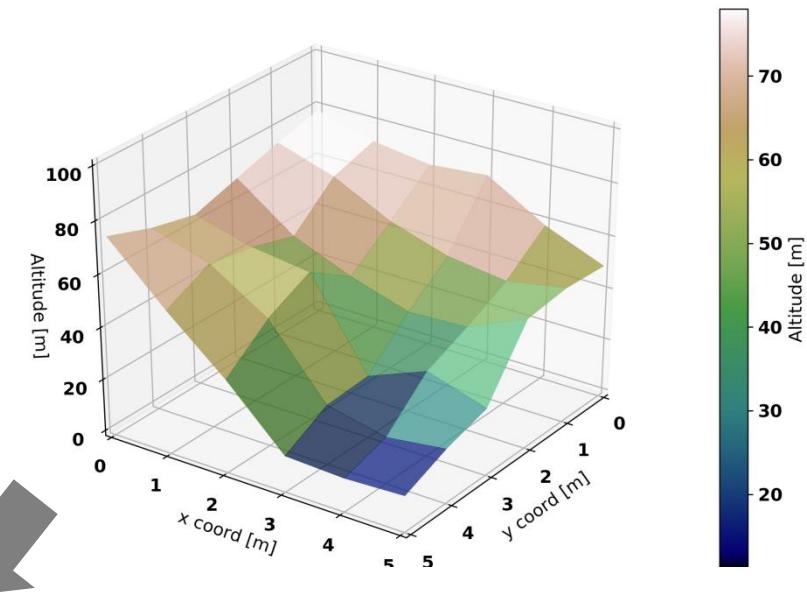
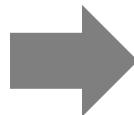
*The coding of the direction of flow*

# FlowDir\_v010: example D8

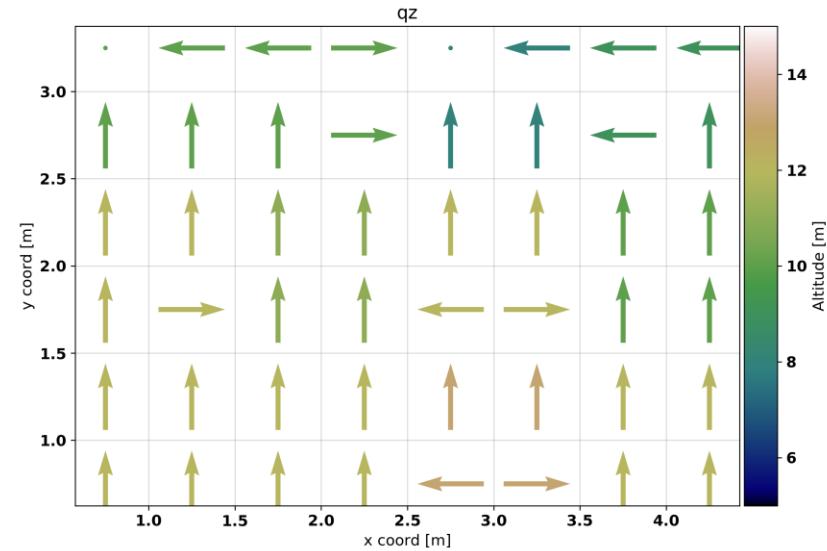
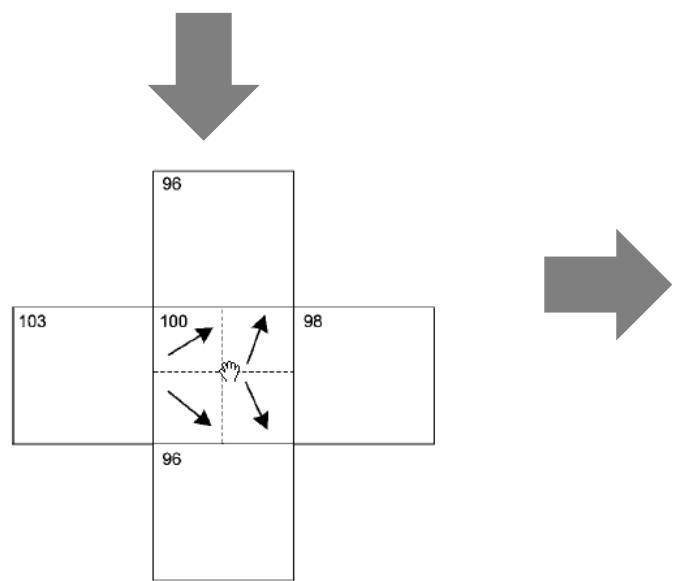
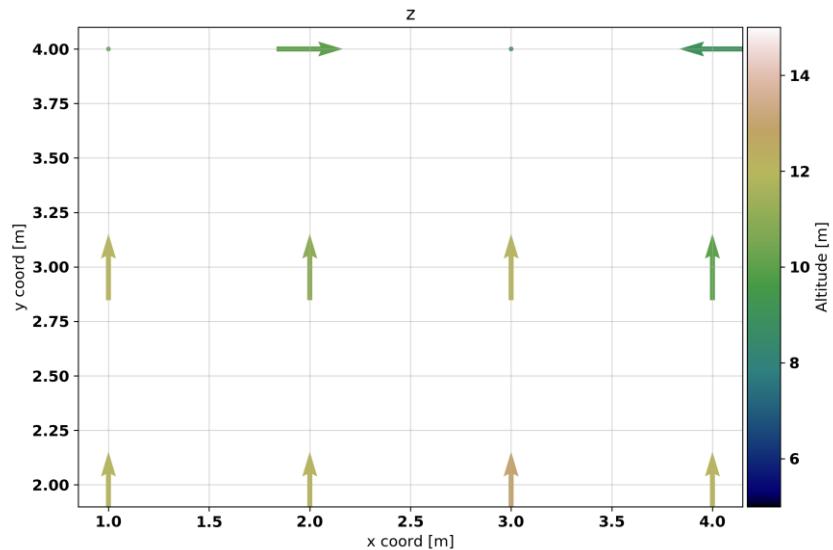


78	72	69	71	58	49
74	67	56	49	46	50
69	53	44	37	38	48
64	58	55	22	31	24
68	61	47	21	16	19
74	53	34	12	11	12

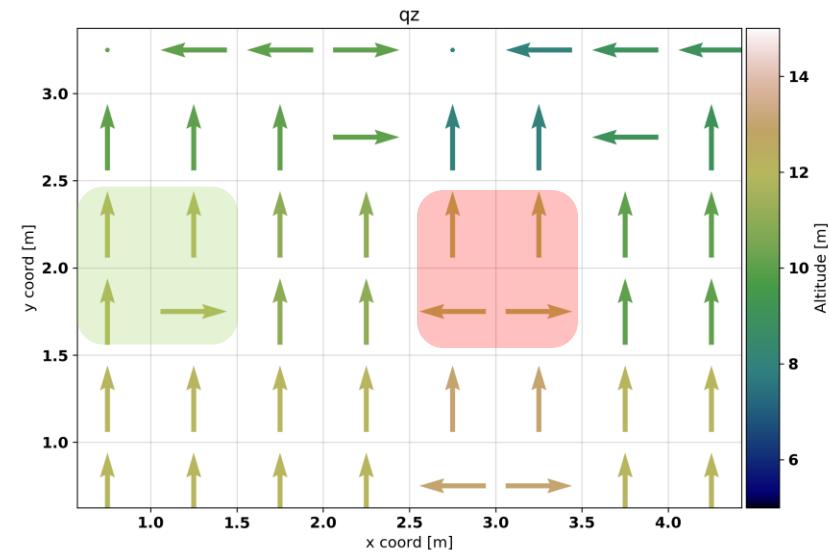
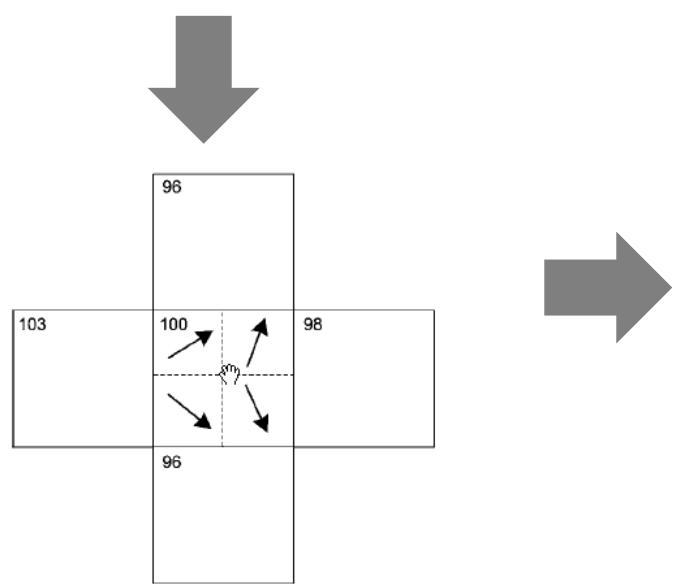
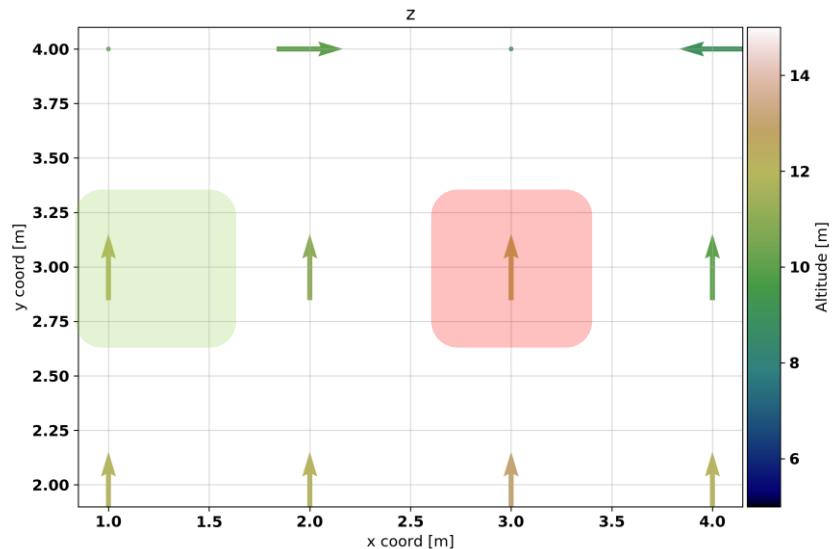
Elevation surface



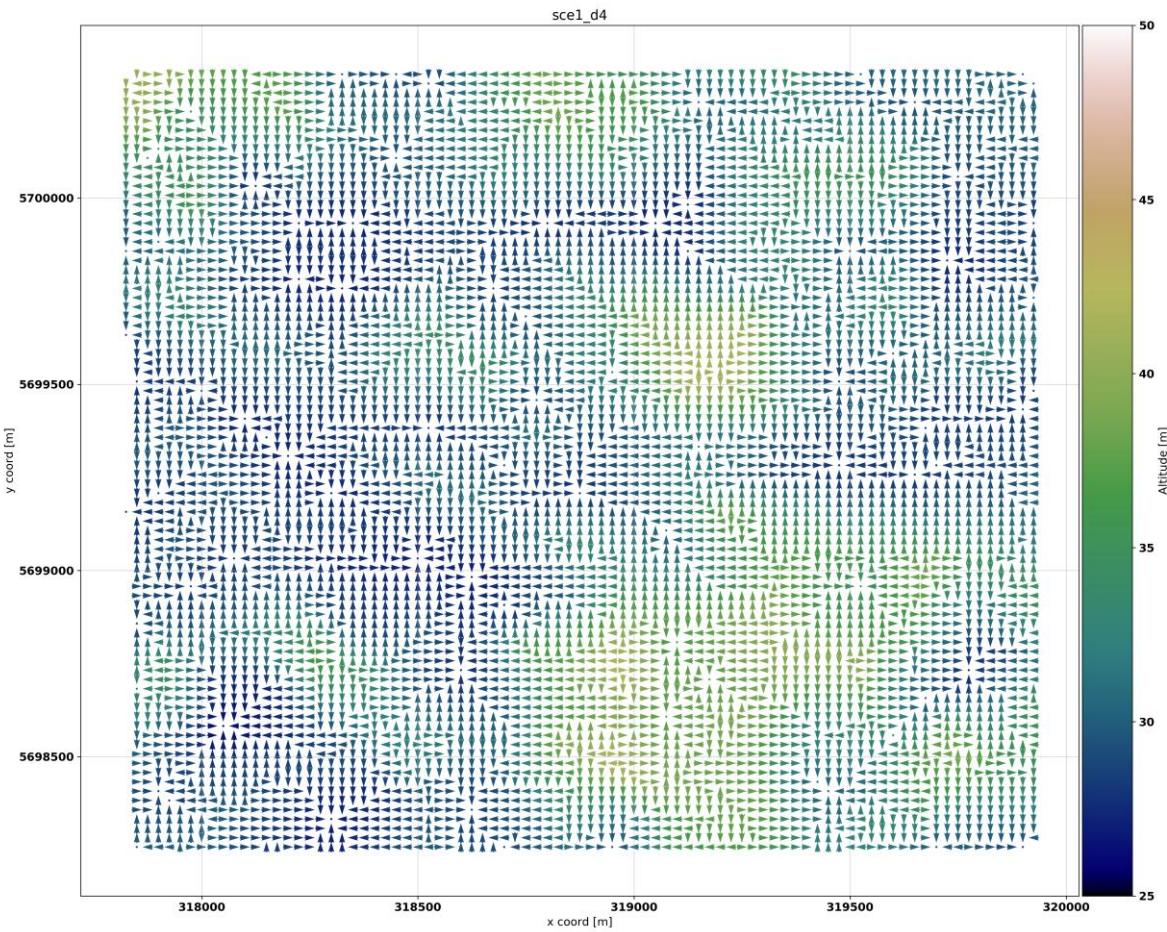
# FlowDir\_v010: example D4q



# FlowDir\_v010: example D4q



# Scenario 1: D4



# Scenario 1: D4q

