```
# subgrid hydrological response units (colors indicate data belonging to the same grid cell.
land_use_type = [ 0, 1 , 3, 3, 1 , 3] # 0: forest, 1: grassland, 3: irrigated cropland
land_owner = [ -1, -1, 1, 2, -1, 2] # -1: no owner, 1: owned by farmer #1, 2: owned by farmer #2.
land_use_ratio = [ .3, .7 , .1, .2, .7 , 1. ] # HRUs of one grid cell always sum to 1
cell_boundaries = [0, 2, 5, 6] # 1st grid cell contains the 0th up to the 2nd HRU, the 2nd grid cell the 2nd up to the 5th, etc.

# set runoff from natural sources
runoff_m_HRU = [1, 2 , 4, 1, 1 , 0]

# farmer #2 has a groundwater pump running, creating additional runoff from irrigation return flow.
for i, land_owner in enumerate(land_owners):
        if land_owner == 2: # increase runoff by 1 in fields owned by farmer #2
                runoff_m_HRU[i] += 1

runoff_m_HRU
>> [1, 2 , 4, 2, 1 , 1]

# translate subgrid to grid – unit in meters in the example.
runoff_m_grid = []
for left, right in zip(cell_boundaries[:-1], cell_boundaries[1:]):
    runoff_m_grid.append(sum([
            runoff * land_size
            for runoff, land_size in zip(runoff_m_HRU[left:right], land_use_ratio[left:right])
    ]))

runoff_m_grid
>> [1.7, 1.5, 1]

# translate grid to subgrid – unit in meters in the example.
precipitation_m_grid = [ 1, 3, 2]
precipitation_m_HRU = []
for left, right, precipitation_HRU_m in zip(cell_boundaries[:-1], cell_boundaries[1:], precipitation_m_grid):
    number_of_HRUs_in_grid_cell = right – left
    for i in range(number_of_HRUs_in_grid_cell):
        precipitation_m_HRU.append(precipitation_HRU_m)

precipitation_m_HRU
>> [1, 1 , 3, 3, 3 , 2]
```