# The soil moisture and water conservation on the SIRTA observatory

## Introduction

A set of observations on the SIRTA site allow to try and close the water conservation at the surface to 1 meter depth in the soil. The data used in this exercise are the result of a re-analysis of the long term observations carried out on campus. They are quality controlled before being aggregated to the hourly scale. Missing or corrupted data are replaced by NaN values and this needs to be taken into account during the exercises by using the Numpy functions which skip these values.


Soil moisture was measured through the dielectric permitivity of the soils using the ThetaProbe displayed above. From this measurement and knwowing the fraction of mineral and organic matter in the soil the volumetric soil moisture can be derived. 6 probes are placed in the soil (at 5, 10, 20, 30, 50 and 100 cm) and the data are provided hourly.

The high frequency at which the data are provided allows to explore time scales from the annual to rainy episode and to understand how the soil moisture react. The aim of the exercises is to close the water conservation at the surface to 1 meter depth in the soil and deduce the drainage below 1 meter from the continuity equation :

$$\frac{\partial W}{\partial t} = \int_S (P-E)dS - Q - I$$

$\frac{\partial W}{\partial t}$ can be deduce from soil moisture measurements, and assuming $Q = 0$, $I$ can be determinate.

The full documentation of these re-analysed data are available here. On this page some quicklooks are also provided and will allow to verify that your results are correct. For the purpose of the exercise only precipitation, evaporation and soil moisture at 6 levels are provided over the 2015-2022 period.

For this exercise xarray module will be used. The xarray module (documented here) simplifies the data processing by keeping in one object the data, dimensions, coordinates and attributes of meta-data.

# Initialization of Notebook

The first step of the exercise will be to work on soil moisture response to precipitation and evaporation at annual and rainy episode time scales.

In [3]:
```
# Numerical analysis module
import numpy as np
```

```python
# Plot module
import matplotlib.pyplot as plt
import pandas as pd
# read netCDF files
import xarray as xr
# Add Root directory to search path
import sys
import datetime
DataPath="../../Data/"
# Set data directory
filename = DataPath+'ReObs_SIRTA_Soil_moisture_2015-2022.nc'

# load the data
ds = xr.open_dataset(filename)
ds
```

Out[3]: xarray.Dataset

► Dimensions: (**time**: 61368)

▼ Coordinates:

| | | | |
|---|---|---|---|
| **time** | (time) | datetime64[ns] | 2015-09-01 ... 2022-08-31... |

▼ Data variables:

| | | | |
|---|---|---|---|
| precipitation | (time) | float32 | ... |
| evaporation | (time) | float32 | ... |
| ms_0m05 | (time) | float32 | ... |
| ms_0m10 | (time) | float32 | ... |
| ms_0m20 | (time) | float32 | ... |
| ms_0m30 | (time) | float32 | ... |
| ms_0m50 | (time) | float32 | ... |
| ms_0m100 | (time) | float32 | ... |

▼ Attributes:

| | |
|---|---|
| Title : | SIRTA ReOBS dataset soil moisture |
| Description : | Simplified dataset from SIRTA ReObs for MEC558 Cours |

You can see the details of each variables (units, name, comments, ...) by clicking on the file symbol "Show / Hide attributes"

# Q1 : The precipitation, evaporation and soil moisture

To start the analysis select the year you wish to process. The choice is between 2016 and 2022.

In [4]:
```python
Year = 2019
```

```
SDate=datetime.datetime(Year-1, 9, 1, 0, 0, 0)
EDate=datetime.datetime(Year, 8, 31, 23, 59, 59)

Precip = ds['precipitation'].sel(time=slice(SDate,EDate))
```

# Coding tip :

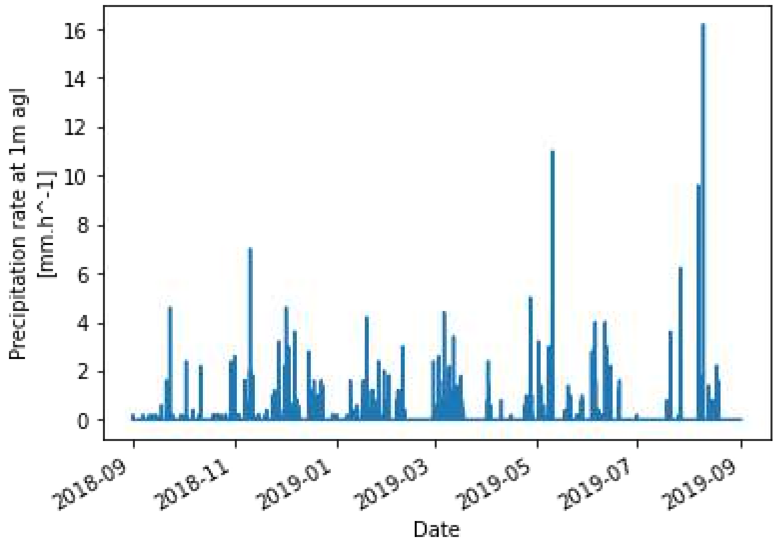
- ['variable'] select a variable in the Dataset ds.
- .sel select a part of the data
- The object obtained is a DataArray (as Precip).
- We can observe the data contained in a DataArray with .plot() method.

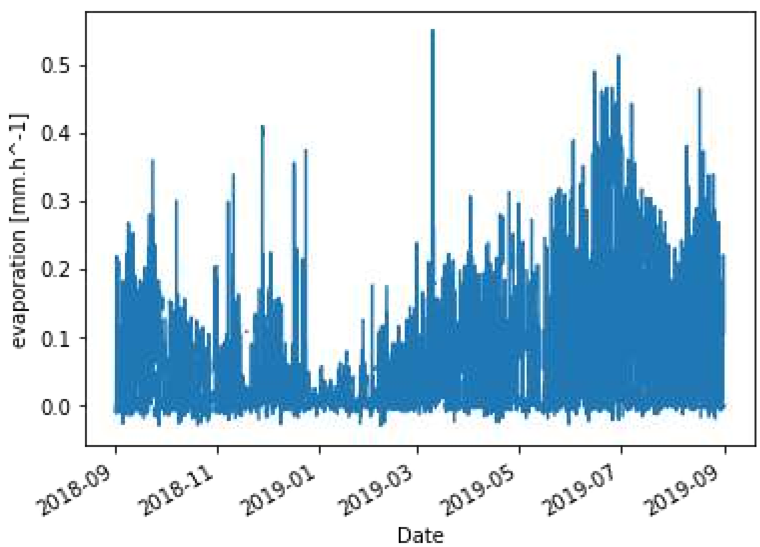
**Extract DataArray from Dataset for evaporation and each level of soil moisture.**

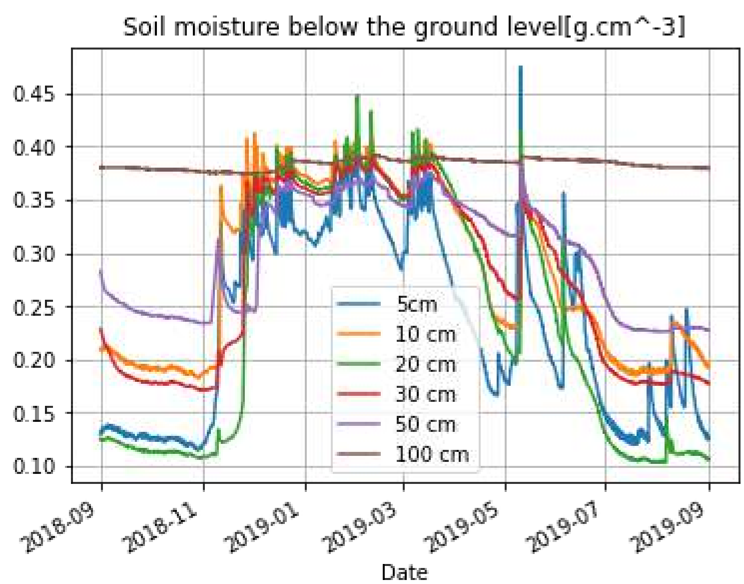
In [5]:
```
# answer cell
Evap = ds['evaporation'].sel(time=slice(SDate,EDate))
ms_05 = ds['ms_0m05'].sel(time=slice(SDate,EDate))
ms_10 = ds['ms_0m10'].sel(time=slice(SDate,EDate))
ms_20 = ds['ms_0m20'].sel(time=slice(SDate,EDate))
ms_30 = ds['ms_0m30'].sel(time=slice(SDate,EDate))
ms_50 = ds['ms_0m50'].sel(time=slice(SDate,EDate))
ms_100 = ds['ms_0m100'].sel(time=slice(SDate,EDate))
```

**Plot on different figures precipitation, evaporation and soil moisture.**

In [6]:
```
# answer cell
Precip.plot()
plt.figure()
Evap.plot()
plt.figure()
ms_05.plot(label="5cm")
ms_10.plot(label="10 cm")
ms_20.plot(label="20 cm")
ms_30.plot(label="30 cm")
ms_50.plot(label="50 cm")
ms_100.plot(label="100 cm")
plt.title("Soil moisture below the ground level[g.cm^-3]")
plt.grid(True)
plt.ylabel("")
plt.legend()
plt.show()
```




# Q1 : Analysis of soil moisture variability over one year

- What is the most variable factor between precipitation and evaporation ?
- What is the impact on the first levels of soil moisture ?
- What is the soil moisture variability at each level ? Why ?
- The low variability at 100cm is a good thing for water balance ? Comment the choice of an hydrologic year.

## Answer 1

- Precipitation has the strongest temporal variability. Evaporation displays a lower variability as it is mostly driven by energy availability.
- Precipitation impacts directly soil moisture at 5cm and then to a depth of about 20cm. Then the variability becomes much more low frequency.
- The diffusion process in the soil slows down the propagation of the moisture brought by rainfall.
- The fact that soil moisture at 100cm is nearly constant allows to assume that at the annual cycle the soil below 1m does not contribute actively to the water balance. So this is a reasonable lower limit for the active soil to be considered.

## Q2 : Soil moisture response to rainy episodes

Based on the last plot, zoom on the first rainy episodes which recharge each level of the soil (at least until 50cm) between october and december. Select about 2 weeks.

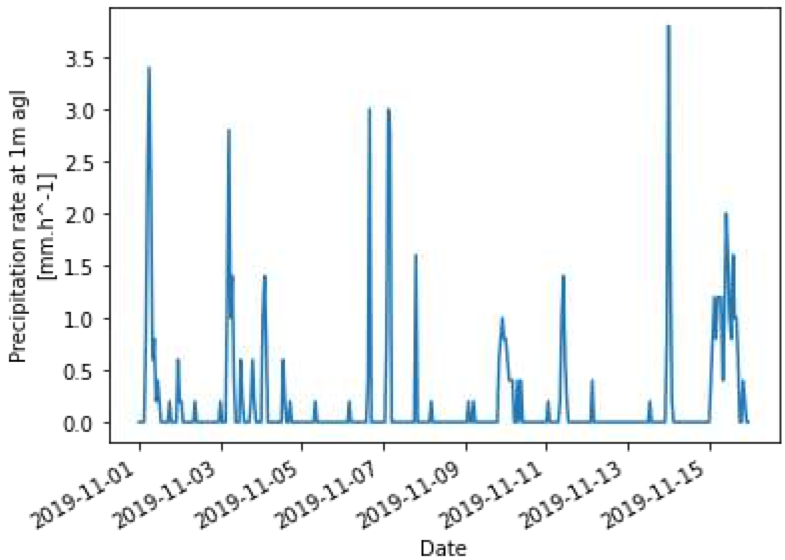
In [7]:
```
year = 2019
month_start = 11
month_end = 11
day_start = 1
day_end = 15
```

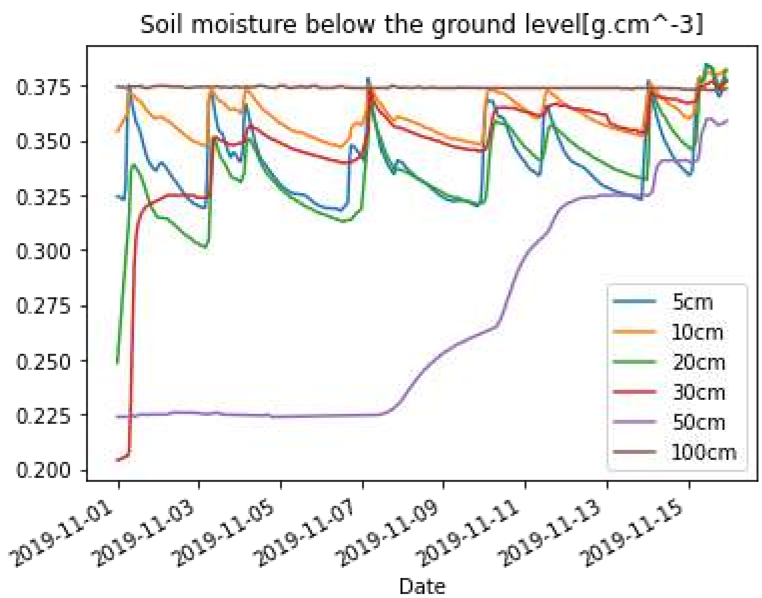
In [8]:
```
precip_zoom = ds['precipitation'].sel(time=slice(str(year)+"-"+str(month_sta
ms_05_zoom = ds['ms_0m05'].sel(time=slice(str(year)+"-"+str(month_start)+"-"
ms_10_zoom = ds['ms_0m10'].sel(time=slice(str(year)+"-"+str(month_start)+"-"
ms_20_zoom = ds['ms_0m20'].sel(time=slice(str(year)+"-"+str(month_start)+"-"
ms_30_zoom = ds['ms_0m30'].sel(time=slice(str(year)+"-"+str(month_start)+"-"
ms_50_zoom = ds['ms_0m50'].sel(time=slice(str(year)+"-"+str(month_start)+"-"
ms_100_zoom = ds['ms_0m100'].sel(time=slice(str(year)+"-"+str(month_start)+"
```

**Plot on different figures precipitation and soil moisture.**

In [9]:
```
# answer cell
plt.figure()
precip_zoom.plot()

plt.figure()
```

```
ms_05_zoom.plot(label="5cm")
ms_10_zoom.plot(label="10cm")
ms_20_zoom.plot(label="20cm")
ms_30_zoom.plot(label="30cm")
ms_50_zoom.plot(label="50cm")
ms_100_zoom.plot(label="100cm")
plt.title("Soil moisture below the ground level[g.cm^-3]")
plt.ylabel("")
plt.legend()
plt.show()
```




# Q2 : Analysis of soil moisture variability after rainy episodes

- What is the time decay between the rainy episode and soil moisture response in function of depth ?
- What is the decreasing exponential rate for each level ? Why ?

- What is the soil moisture variability at each level ? Why ?

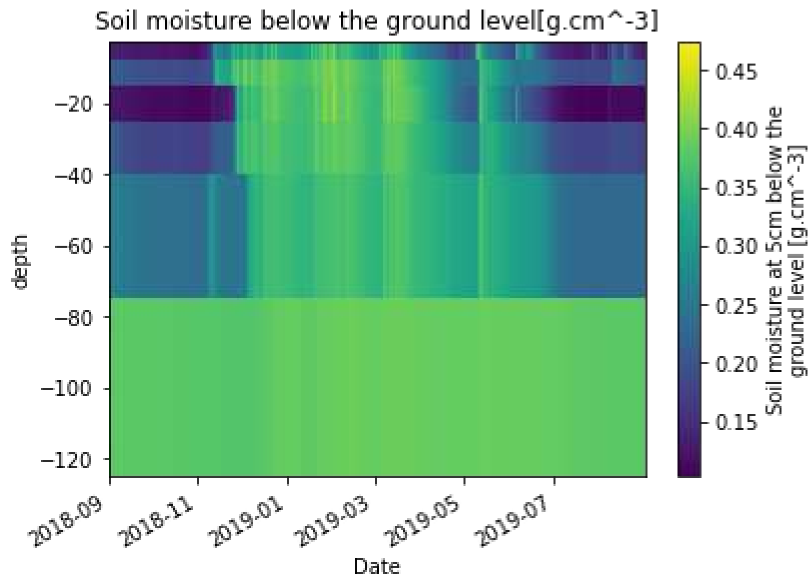
# Answer 2

- The decay of soil moisture in the upper levels (above 30cm) is caused by evaporation and the diffusion to lower levels. It can be seen that during this period in early November the levels below 30cm systematically gain moisture.
- The exponential decrease is explained by the increasing difficulty for moisture to evaporation and diffuse as the soil dries and the capilarity and adsorbption forces play an important role.
- The amplitude of variation during this november period is largest in the lower level as they recharge. The variations at the surface are of smaller amplitude as evaporation is not as strong any more.

# Q3 : Interpolation of soil moisture within the top meter of the soil

To compute $W$, the water content within the soil, we need to group DataArrays representing soil moisture at different levels together first and then interpolate between these levels.

In [10]:
```
ms_profil = xr.concat([ms_05, ms_10, ms_20, ms_30, ms_50, ms_100], dim='dept

depth = -np.array([5,10,20,30,50,100])
ms_profil = ms_profil.assign_coords(depth=("depth", depth))

ms_profil.plot(x='time',y="depth")
plt.title("Soil moisture below the ground level[g.cm^-3]")
plt.show()
```




The cell below compute a list of DataArrays which are the results of the interpolation on a "new_depth", a numpy array in centimeter. The "depth" is the precedent vertical coordinate based on measurements. "ms_profil" is the 2D DataArray of the soil moisture (time and depth).

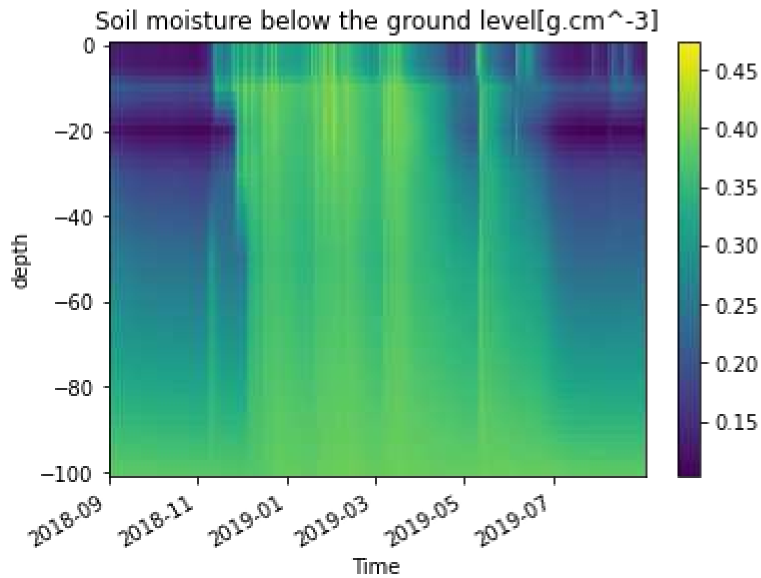
In [11]:
```
def interp_depth(new_depth,depth,ms_profil,ms=ms_05):
    coeff = np.zeros((len(new_depth),len(depth)))
    for i in range(len(new_depth)):
        j=0
        while depth[j]>new_depth[i]:
            j+=1
        if j==0:
            coeff[i,j] = 1
        else:
            coeff[i,j-1] = (-new_depth[i] + depth[j]) / (depth[j] - depth[j-
            coeff[i,j] = (new_depth[i] - depth[j-1]) / (depth[j] - depth[j-1

    new_values = coeff @ ms_profil.values

    return new_values
```

# Coding tip :

- Use xr.DataArray(data, dims=('depth', 'time')) to create a new DataArray with two dimensions depth and time.
- Use the previous cells to understand assign_coords

**On a new depth coordinate, compute interpolation with interp_depth function and assign the result in a new DataArray that we will call "ms_profil_total". Plot the result and compare it with the previous plot without interpolation.**

In [12]:
```
# answer cell
new_depth = -np.linspace(0,100,51)
new_values = interp_depth(new_depth,depth,ms_profil)
ms_profil_total = xr.DataArray(new_values, dims=('depth', 'time'))
ms_profil_total = ms_profil_total.assign_coords(depth=("depth", new_depth))
ms_profil_total = ms_profil_total.assign_coords(Time=("time", ms_05.coords["
ms_profil_total.plot(x='Time',y="depth")
plt.title("Soil moisture below the ground level[g.cm^-3]")
plt.show()
```




# Q3 : Analysis of soil moisture evolution and interpolation effect

- What is the annual cycle of the soil moisture ?
- Where the influence of the interpolation is the most important ? What time ? What depth ? How can we improve that ?
- A constant soil moisture hypothesis have been done on the first 5cm of the soil. Could you discuss this hypothesis ? Is it a relevant issue for the water balance ?

# Answer 3

- We clearly see the downward propagation of soil moisture in autumn and the drying from the top downward in summer.
- The interpolation of soil moisture is most critical in the upper layer as the gradients are strongest there. But we also have sounds which are closer together.
- At the bottom the profile is more homogeneous and thus the interpolation should work better. But as more moisture is stored there for the total soil moisture it is also more critical.

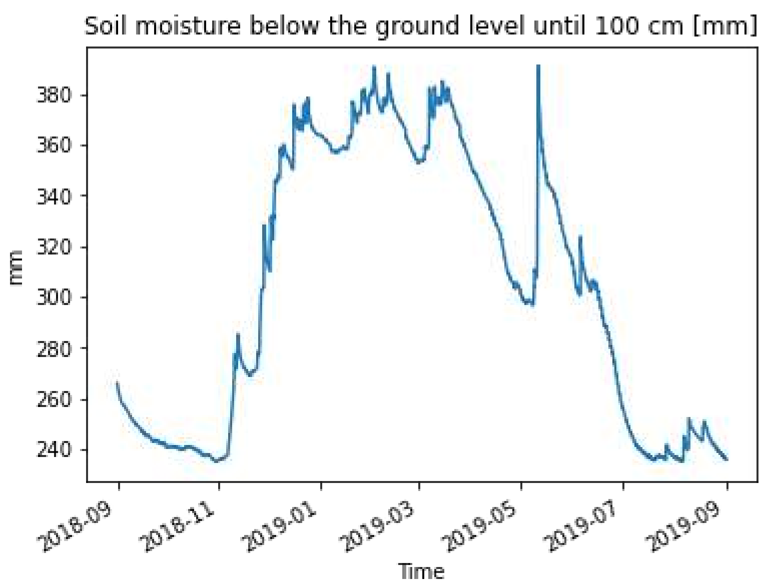
# Q4 : Total water content within the top meter of soil

# Coding tip :

- Use .mean(dim = ['dims']) to compute the mean on a dimension "dims" of a DataArray.
- Use .sum(dim = ['dims']) to compute the sum on a dimension "dims" of a DataArray.
- Use the relevant method to code the following instruction.

**Now, the total water content within the top meter of soil can be easily compute. Did it, compute the result in mm and plot it. The name of the new DataArray compute need to be "ms".**

In [13]:
```
# answer cell
ms = ms_profil_total.mean(dim = ['depth'])*10*100 # en mm
ms.plot(x='Time')
plt.title("Soil moisture below the ground level until 100 cm [mm]")
plt.ylabel("mm")
plt.show()
```



# Q4 : Analysis of the total water content within the top meter of soil

- What is the water content within the top meter of soil in average ? What does it represent ?
- Why the decreasing rate changes over time ?

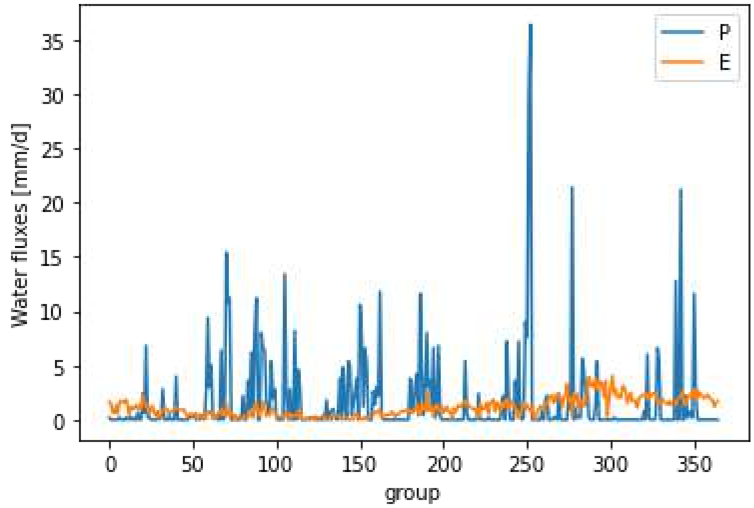
# Answer 4

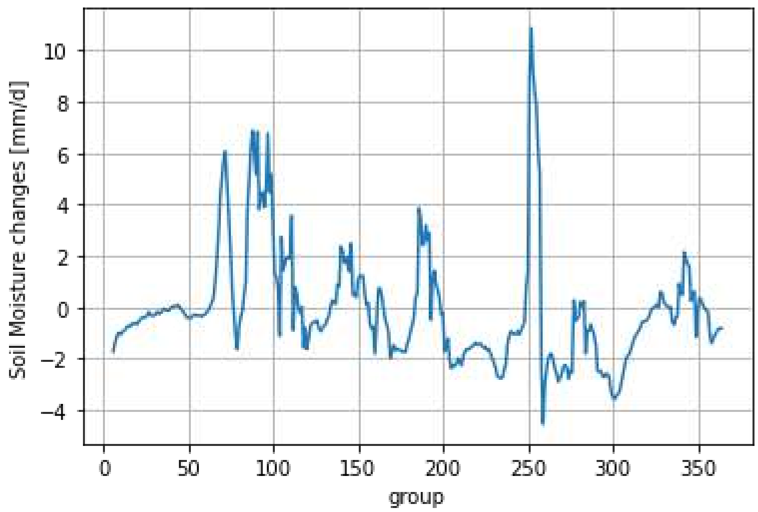
- The annual cycle of soil moisture at SIRTA is about 150mm over the year. This corresponds well to the accessible soil moisture we derived in the Notebook on the pedotransfer functions.
- In March the soil moisture starts to dry as evaporation becomes the dominant term and dries the soils.

# Q5 : Water balance and drainage at the bottom of the first meter of soil

A daily mean is compute for precipitation, evaporation and soil water content variation which is just a derivate of the last figure. A rolling is done on one week to smooth dW.

In [14]:
```
dayofyear = (Precip.time.dt.year - Precip.time.dt.year.min()) * 365 + Precip
Pdaily=Precip.groupby(dayofyear - 244).sum()
Pdaily.attrs["units"]="mm/d"
Pdaily.attrs["long_name"]="Daily Precipitation"

Edaily=Evap.groupby(dayofyear - 244).sum()
Edaily.attrs["units"]="mm/d"
Edaily.attrs["long_name"]="Daily Evaporation"

dW = ms.differentiate("time",datetime_unit="h").groupby(dayofyear - 244).sum
dW.attrs["units"]="mm/d"
dW.attrs["long_name"]="Soil Moisture changes"

Pdaily.plot(label="P")
Edaily.plot(label="E")
plt.ylabel("Water fluxes [mm/d]")
plt.legend()
plt.show()

dW.rolling(group=7).mean().plot()
plt.grid(True)
plt.show()
print("Total dW : ", dW.sum().values, 'mm')
```




```
Total dW :  -30.382245457639954 mm
```

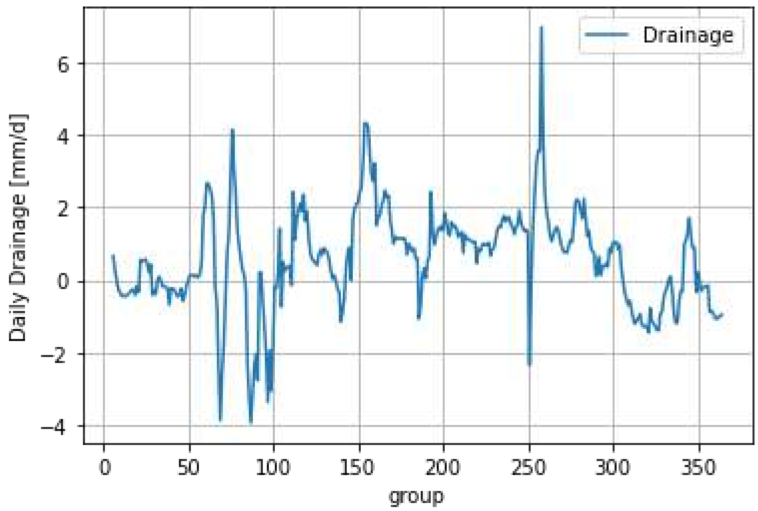
# Coding tip :

- Use .attrs["units"]= to assign a unit to a DataArray
- Use .attrs["units"]= to assign a name to a DataArray

**From continuity equation, compute the drainage at the bottom of the first meter of soil. Plot the result. Calculate the total precipitation, evaporation and drainage over the year.**

In [15]:
```
# answer cell

drainage = Pdaily - Edaily - dW
drainage.attrs["units"]="mm/d"
drainage.attrs["long_name"]="Daily Drainage"
drainage.rolling(group=7).mean().plot(label="Drainage")
plt.legend()
plt.grid(True)
plt.show()

print("Total precipitation : ", Pdaily.sum().values, 'mm')
```

```
print("Total Evaporation : ", Edaily.sum().values, 'mm')
print("Total drainage : ", drainage.sum().values, 'mm')
```




```
Total precipitation :  582.8 mm
Total Evaporation :  409.899 mm
Total drainage :  203.28321505282264 mm
```

# Q5 : Analysis of the drainage at the bottom of the first meter of soil

- What is the water balance at the SIRTA for the considered year ? Which part of precipitation leaves the soil as drainage, as evaporation ?
- How can we explain the extreme daily drainage values that we obtain without rolling, considering the lack of data ?
- A constant soil moisture hypothesis have been done on the first 5cm of the soil. Could you discuss this hypothesis ? Is it a relevant issue for the water balance ?
- What is the period of the year when we observe a drainage ?

# Answer 5

- The precipitation, Evaporation and soil moisture change balance to within 30kg/m^2. This is quite a good balance given that the total rainfall over the period is 500 kg/m^2.
- The estimated drainage is thus quite realistic and correspond to a little less than half of the evaporation in 2019.
- The drainage estimate suffers from the errors in the estimate of the soil moisture changes. It is affected by the missing data, the vertical interpolation and in general the lack of data within the soil.
- The high frequency variations in the top 5cm will affect the daily derivative of soil moisture as this is the fastest part of the reservoir.

- Drainage is strongest from winter to summer. In autumn when the deeper soils are dry it is less important.

## Q6 : Cumulative variables

In the following, cumulative variables are compute over the year for precipitation, evaporation, soil water content and drainage. These quantities give us a different understanding of how the soil reacts throughout the year

In [16]:

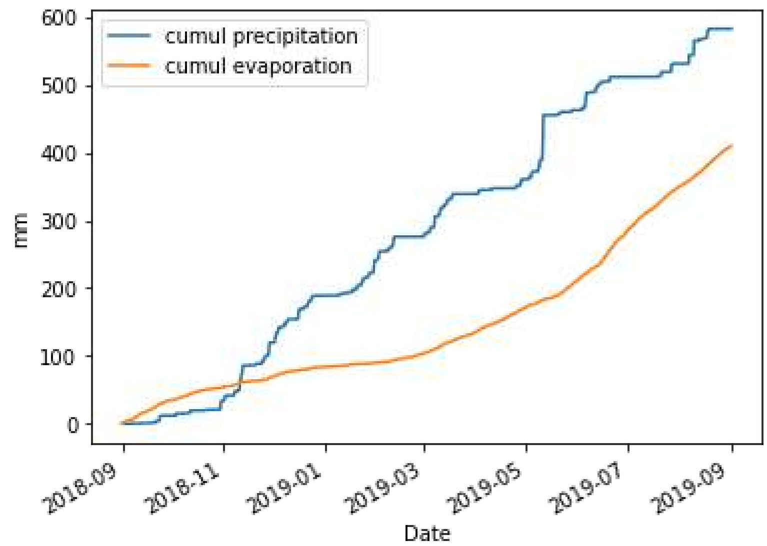
```
sum_partiel_p = Precip.copy()
sum_partiel_e = Evap.copy()
sum_partiel_pe = Precip.copy()
tlen=Precip.coords['time'].size
for i in range(tlen):
    sum_partiel_p.values[i] = np.nansum(Precip.values[:i+1])
    sum_partiel_e.values[i] = np.nansum(Evap.values[:i+1])
    sum_partiel_pe.values[i] = sum_partiel_p.values[i]-sum_partiel_e.values[
```

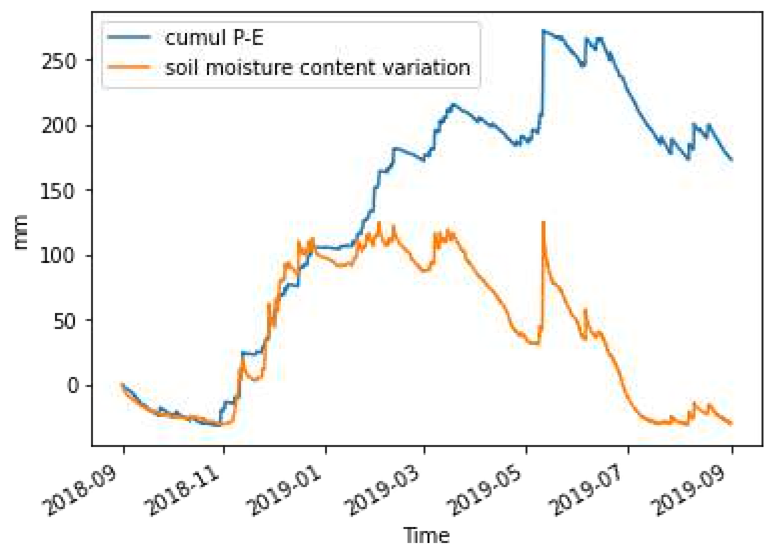
**Plot on different figures :**

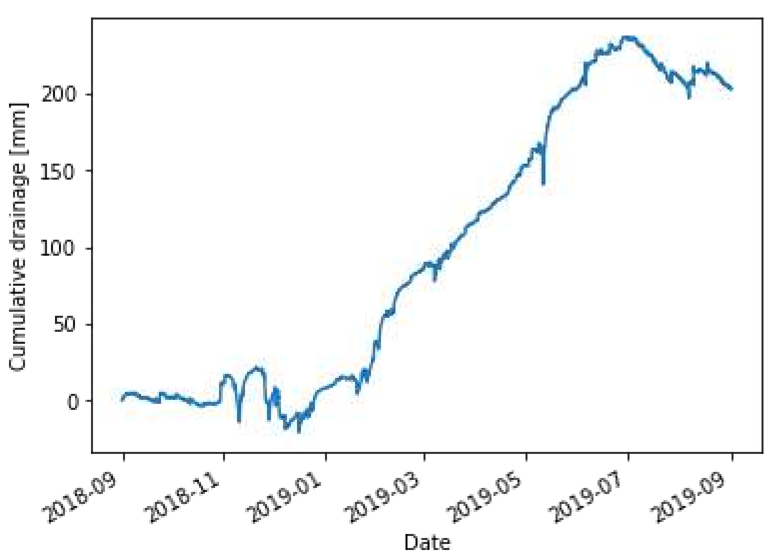
- Cumulative precipitation, cumulative evaporation.
- Cumulative precip-evap, soil water content difference from the beginning of the year and cumulative drainage.

In [17]:

```
# answer cell

sum_partiel_p.plot(label='cumul precipitation')
sum_partiel_e.plot(label='cumul evaporation')
plt.legend()
plt.ylabel('mm')
plt.show()

sum_partiel_pe.plot(label='cumul P-E')
ms.values = ms.values - ms.values[0]
ms.plot(x="Time",label='soil moisture content variation')
plt.legend()
plt.ylabel('mm')
plt.show()

drainage = sum_partiel_pe - ms
drainage.plot()
plt.ylabel('Cumulative drainage [mm]')
plt.show()
```




# Q6 : Analysis of cumulative plots

- What is the confidence of our balance considering the difference between soil moisture content variation and cumulative P-E ?
- Where does the excess precipitation go ?

# Answer 6

- The soil moisture change between the begining and end of the year is small compared to the total P-E.
- Soil moisture starts to decrease as soon as P-E flattens out and thus E dominates.
- The cumulated P-E corresponds well to the cumulated drainage. We can thus conclude that this is where the excess water goes.

In [ ]:

```

```