

# Latte Art: Running a Model and Extracting Basics

---

Project Link: <https://www.pyswmm.org>

Date: January 10, 2023

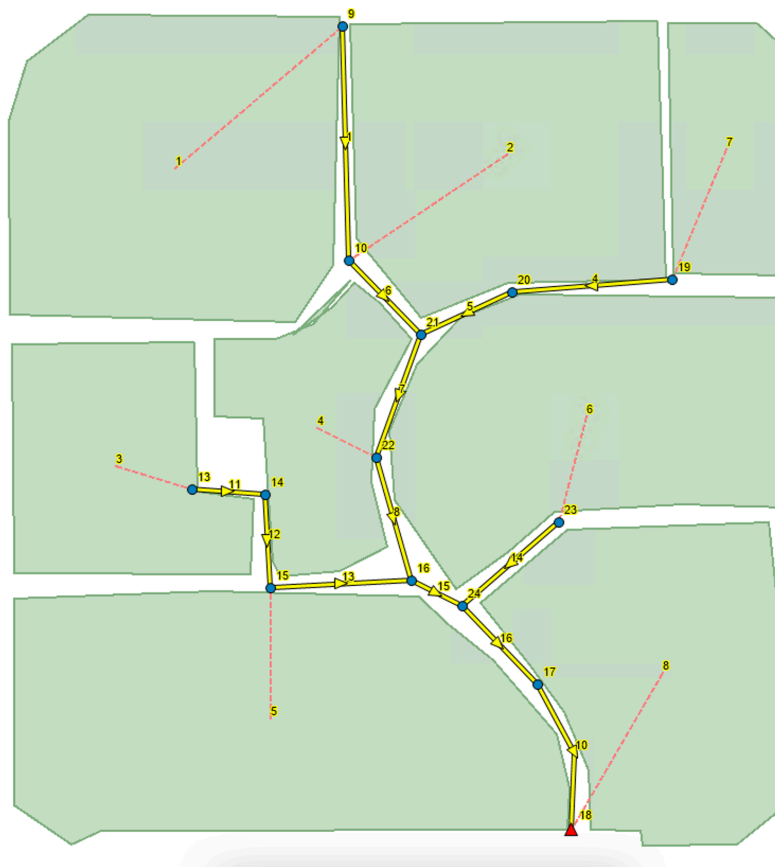
Author: Bryant McDonnell

Version: 2.0

Files: `Latte.py`, `Example1.inp`

## Background

This example walks the user through writing the code and running a simulation and extracting simple pieces of data pertaining to the model/simulation using `Example 1` data set. The PNG file is simply to provide visual context of where we are reading information out of the model.



## Code Example

---

This code builds off of the `Espresso` example:

```
'''
PySWMM Latte Code
Author: Bryant McDonnell
Version: 1
Date: Oct 23, 2022
'''

from pyswmm import Simulation, Nodes, Links

with Simulation(r'Example1.inp') as sim:
    #####
    # Simulation information
    print("Simulation info")
    flow_units = sim.flow_units
    print("Flow Units: {}".format(flow_units))
    system_units = sim.system_units
    print("System Units: {}".format(system_units))
    print("Start Time: {}".format(sim.start_time))
    print("End Time: {}".format(sim.end_time))

    #####
    # Node Information
    Node21 = Nodes(sim)["21"]
    print("Node 21 info")
    print("Invert Elevation: {}".format(Node21.invert_elevation))
    print("Physical Depth: {}".format(Node21.full_depth))
    print("Is it a Junction?: {}".format(Node21.is_junction()))

    #####
    # Link Information
    Link15 = Links(sim)['15']
    print("Link 15 info")
    print("Inlet Node ID: {}".format(Link15.inlet_node))
    print("Outlet Node ID: {}".format(Link15.outlet_node))

    # Launch a simulation!
    for ind, step in enumerate(sim):
        if ind % 100 == 0:
            print(sim.current_time, ", ", round(sim.percent_complete*100), "%", \
                  Node21.depth, Link15.flow)

    node21_stat_out = Node21.statistics
    print(node21_stat_out)
    print("Max Node 15 Depth: {}".format(node21_stat_out['max_depth']))
```

```
link15_stat_out = Link15.conduit_statistics
print("Link 15 Peak Velocity: {}".format(link15_stat_out["peak_velocity"]))
```

Let's decompose some of these things now. Just like the `Espresso` example, we are opening the simulation using the `with` context manager. That means we can now fetch data out of the open model! Before running the simulation, let's pull some basic data out of the model about the simulation we are about to run AND some of the hydraulic network data. We are in a "pre simulation" state. We have access to many pieces of data for things like `nodes`, `links`, `subcatchments`, and `raingages` and etc.

The example shows some simple examples of what can be scraped out of a simulation.

**Simulation References can be found here:**

<https://pyswmm.readthedocs.io/en/stable/reference/simulation.html#pyswmm.simulation.Simulation>

**Node References can be found here**

<https://pyswmm.readthedocs.io/en/stable/reference/nodes.html#pyswmm.nodes.Node>

**Links References can be found here**

<https://pyswmm.readthedocs.io/en/stable/reference/links.html#pyswmm.links.Link>

To run this the you just have to hit `F5` if using `Idle`.

## Output

```
Simulation info
Flow Units: CFS
System Units: US
Start Time: 1998-01-01 00:00:00
Start Time: 1998-01-02 12:00:00
Node 21 info
Invert Elevation: 990.0
Physical Depth: 3.0
Is it a Junction?: True
Link 15 info
Inlet Node ID: 16
Outlet Node ID: 24
1998-01-01 00:01:00 , 0 % 1.0 0.0
1998-01-01 01:41:00 , 5 % 1.4754735008659758 4.750870622247299
1998-01-01 03:21:00 , 9 % 2.0 15.293390925313071
1998-01-01 05:01:00 , 14 % 2.0 11.558165916025786
1998-01-01 06:41:00 , 19 % 1.1834333339168235 0.6817178789300413
1998-01-01 08:21:00 , 23 % 1.045903877135518 0.04702572258935772
1998-01-01 10:01:00 , 28 % 1.0269102308696727 0.015101711883805378
1998-01-01 11:41:00 , 32 % 1.018688119653874 0.00692885528294457
```

```

1998-01-01 13:21:00 , 37 % 1.0141707231522317 0.0038260211832854187
1998-01-01 15:01:00 , 42 % 1.0113381121522798 0.002368478260504914
1998-01-01 16:41:00 , 46 % 1.0094039988922354 0.0015830939469498415
1998-01-01 18:21:00 , 51 % 1.0080060887299167 0.0011188837318096452
1998-01-01 20:01:00 , 56 % 1.006951569444992 0.00082493589897352
1998-01-01 21:41:00 , 60 % 1.006129178771194 0.000628535086290998
1998-01-01 23:21:00 , 65 % 1.0054714035190844 0.0004918203599983807
1998-01-02 01:01:00 , 69 % 1.004934088607198 0.00038245469942075065
1998-01-02 02:41:00 , 74 % 1.0044873296022006 0.0003078288242553105
1998-01-02 04:21:00 , 79 % 1.5836465793110421 6.698889196508224
1998-01-02 06:01:00 , 83 % 1.4626345064550967 4.6424187262796615
1998-01-02 07:41:00 , 88 % 1.067603930830037 0.10673140272057194
1998-01-02 09:21:00 , 93 % 1.0337232774043619 0.02461545433719277
1998-01-02 11:01:00 , 97 % 1.0219502364513948 0.00982409238269685
{'average_depth': 1.160688401480764, 'max_depth': 2.0, 'max_depth_date':
35796.12152778935, 'peak_total_inflow': 5.416681580890388, 'max_inflow_date':
35796.16805556713, 'peak_lateral_inflowrate': 0.0, 'peak_flooding_rate': 0.0,
'max_flooding_date': 35796.0, 'max_ponded_volume': 0.0, 'max_report_depth': 2.0,
'courant_crit_duration': 0.0, 'flooding_duration': 0.0, 'surcharge_duration': 0.0,
'lateral_infow_vol': 0.0, 'flooding_volume': 0.0}
Max Node 15 Depth: 2.0
Link 15 Peak Velocity: 9.670543803890657

```

The units of the various data streams are always a function of how the user defined the `[OPTIONS]` section of the INP file. The following table provides the look up unit information based on the USEPA SWMM Reference manual.

**Table 1-4 Units of expression used by SWMM**

Variable or Parameter	US Customary Units	SI Metric Units
Area (subcatchment)	acres	hectares
Area (storage surface area)	square feet	square meters
Depression Storage	inches	millimeters
Depth	feet	meters
Elevation	feet	meters
Evaporation	inches/day	millimeters/day
Flow Rate	cubic feet/sec (cfs) gallons/min (gpm) 10 <sup>6</sup> gallons/day (mgd)	cubic meters/sec (cms) liters/sec (lps) 10 <sup>6</sup> liters/day (mld)
Hydraulic Conductivity	inches/hour	millimeters/hour
Hydraulic Head	feet	meters
Infiltration Rate	inches/hour	millimeters/hour
Length	feet	meters
Manning's n	seconds/meter <sup>1/3</sup>	seconds/meter <sup>1/3</sup>
Pollutant Buildup	mass/acre	mass/hectare
Pollutant Concentration	milligrams/liter (mg/L) micrograms/liter (µg/L) organism counts/liter	milligrams/liter (mg/L) micrograms/liter (µg/L) organism counts/liter
Rainfall Intensity	inches/hour	millimeters/hour
Rainfall Volume	inches	millimeters
Storage Volume	cubic feet	cubic meters
Temperature	degrees Fahrenheit	degrees Celsius
Velocity	feet/second	meters/second
Width	feet	meters
Wind Speed	miles/hour	kilometers/hour

## Helpful Info

For those who are newer to Python, the following screenshot provides some Python-specific syntax requirements. The code has been divided into three regions. The first region is tabbed in 1 time (4 spaces). This is the region where the models is open (`swmm_open`). The user has access to many values around the physical network and system info. The yellow region is tabbed in 2 times (8 spaces). This region is where the user can interact with the running simulation (`swmm_start`, `swmm_step`-iterated until, `swmm_end`). Finally, the green region is 1 tab (4 spaces) where the user has access to the final simulated results (such as accumulations and things that would be calculated for the RPT file.) After this, 0 tabs (0 spaces) `swmm_close` is called which frees the memory and closes SWMM.

```

PySWMM Latte Code
Author: Bryant McDonnell
Version: 2
Date: Oct 23, 2022
'''
from pyswmm import Simulation, Nodes, Links

with Simulation(r'Example1.inp') as sim:
    #####
    # Simulation information
    # remaining references are available here:
    # https://pyswmm.readthedocs.io/en/stable/reference/simulation.html#pyswmm.simulation.Simulation
    print("Simulation info")
    flow_units = sim.flow_units
    print("Flow Units: {}".format(flow_units))
    system_units = sim.system_units
    print("System Units: {}".format(system_units))
    print("Start Time: {}".format(sim.start_time))
    print("End Time: {}".format(sim.end_time))

    #####
    # Node Information
    # remaining references are available here:
    # https://pyswmm.readthedocs.io/en/stable/reference/nodes.html#pyswmm.nodes.Node
    Node21 = Nodes(sim)["21"]
    print("Node 21 info")
    print("Invert Elevation: {}".format(Node21.invert_elevation))
    print("Physical Depth: {}".format(Node21.full_depth))
    print("Is it a Junction?: {}".format(Node21.is_junction()))

    #####
    # Link Information
    # remaining references are available here:
    # https://pyswmm.readthedocs.io/en/stable/reference/links.html#pyswmm.links.Link
    Link15 = Links(sim)['15']
    print("Link 15 info")
    print("Inlet Node ID: {}".format(Link15.inlet_node))
    print("Outlet Node ID: {}".format(Link15.outlet_node))

    # Launch a simulation!
    for ind, step in enumerate(sim):
        if ind % 100 == 0:
            print(sim.current_time, ",", round(sim.percent_complete*100), "%", \
                  Node21.depth, Link15.flow)

    node21_stat_out = Node21.statistics
    print(node21_stat_out)
    print("Max Node 15 Depth: {}".format(node21_stat_out['max_depth']))

    link15_stat_out = Link15.conduit_statistics
    print("Link 15 Peak Velocity: {}".format(link15_stat_out["peak_velocity"]))

```

## Follow up

---

If you have run into problems, try posting your questions on Stack Overflow and tag it with `pyswmm`. The development team is very active on there and will for sure follow up!

## References

---

Rossman, L. A. (2015). Storm Water Management Model User's Manual Version 5.1 (No. EPA-600/R-14/413b). U.S. Environmental Protection Agency.