

# Take your Stormwater Modelling to next level

with PySWMM and the Open Water Analytics initiative

Gonzalo Andrés Peña-Castellanos  
@goanpeca

PyCon Colombia  
EAFIT, Medellín  
February 9<sup>th</sup>, 2018

# Content

- About
- The (Urban) Water Cycle
- On Modelling... (How, What, Why, How)
- On [Hydrology](#), [Hydraulics](#) and [Water Quality](#)

*Intermission: on the perils of CFD*

- EPA - SWMM

*Intermission: networking, lying and coding*

- OWA
- SWMM API
- PySWMM API
- Expanded PySWMM API (Under Construction)
- Roadmap and future work



# About @goanpeca

## Civil Engineer



- MSc Hydroinformatics (2010)
- MSc Sanitary Engineering (2012)
  - PhD dropout (2015)



# About @goanpeca



## Civil Engineer



- MSc Hydroinformatics (2010)
- MSc Sanitary Engineering (2012)
  - PhD dropout (2015)

## 'Software Engineer'



- Python developer since 2009



Spyder IDE Core Developer since 2014 <http://spyder-ide.org>



PyBee Project Area leader since 2017 <https://pybee.org>



# About @goanpeca



## Civil Engineer



- MSc Hydroinformatics (2010)
- MSc Sanitary Engineering (2012)
  - PhD dropout (2015)

## 'Software Engineer'



- Python developer since 2009



Spyder IDE Core Developer since 2014 <http://spyder-ide.org>

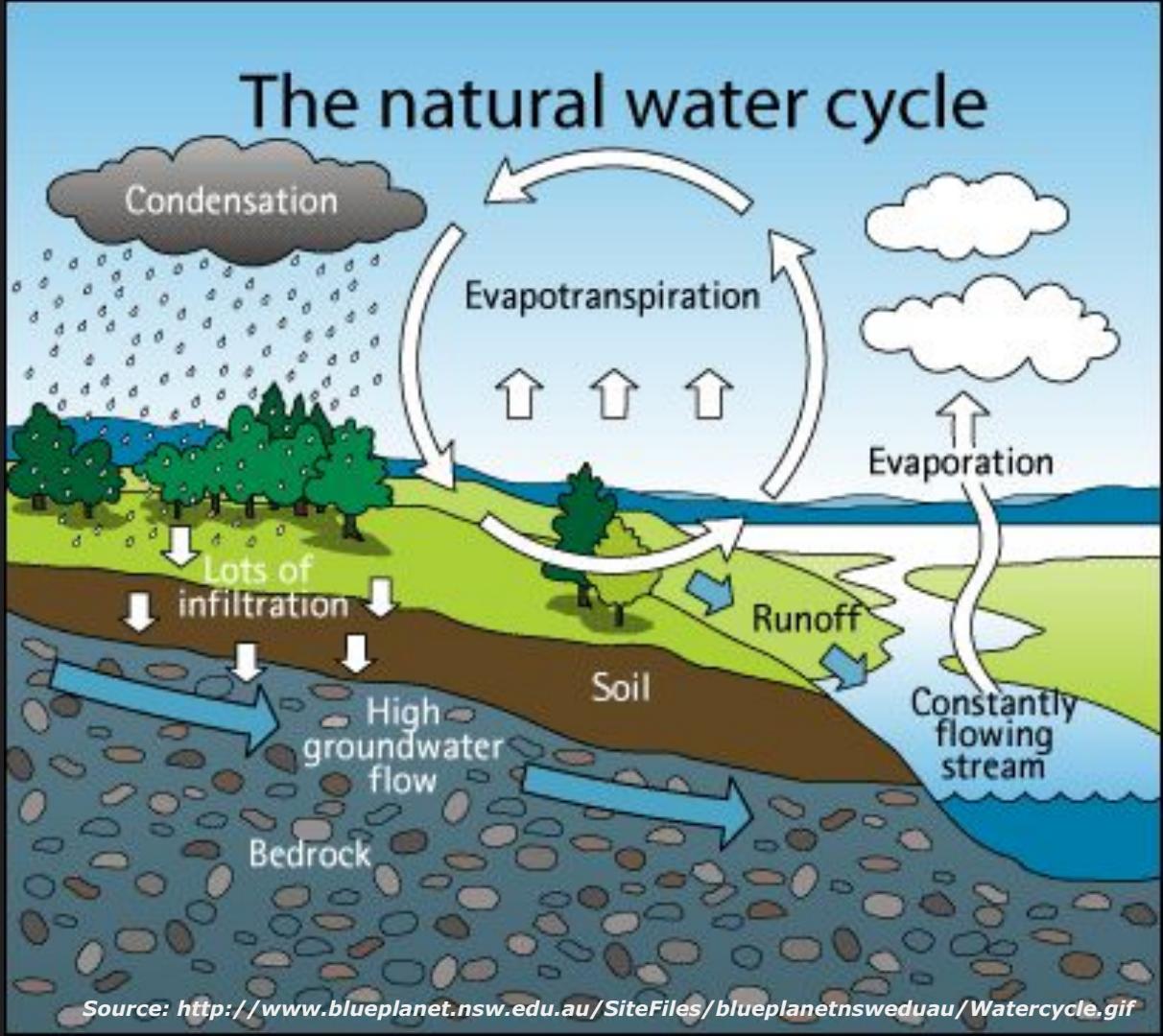


PyBee Project Area leader since 2017 <https://pybee.org>

**PySWMM core developer since 2017**

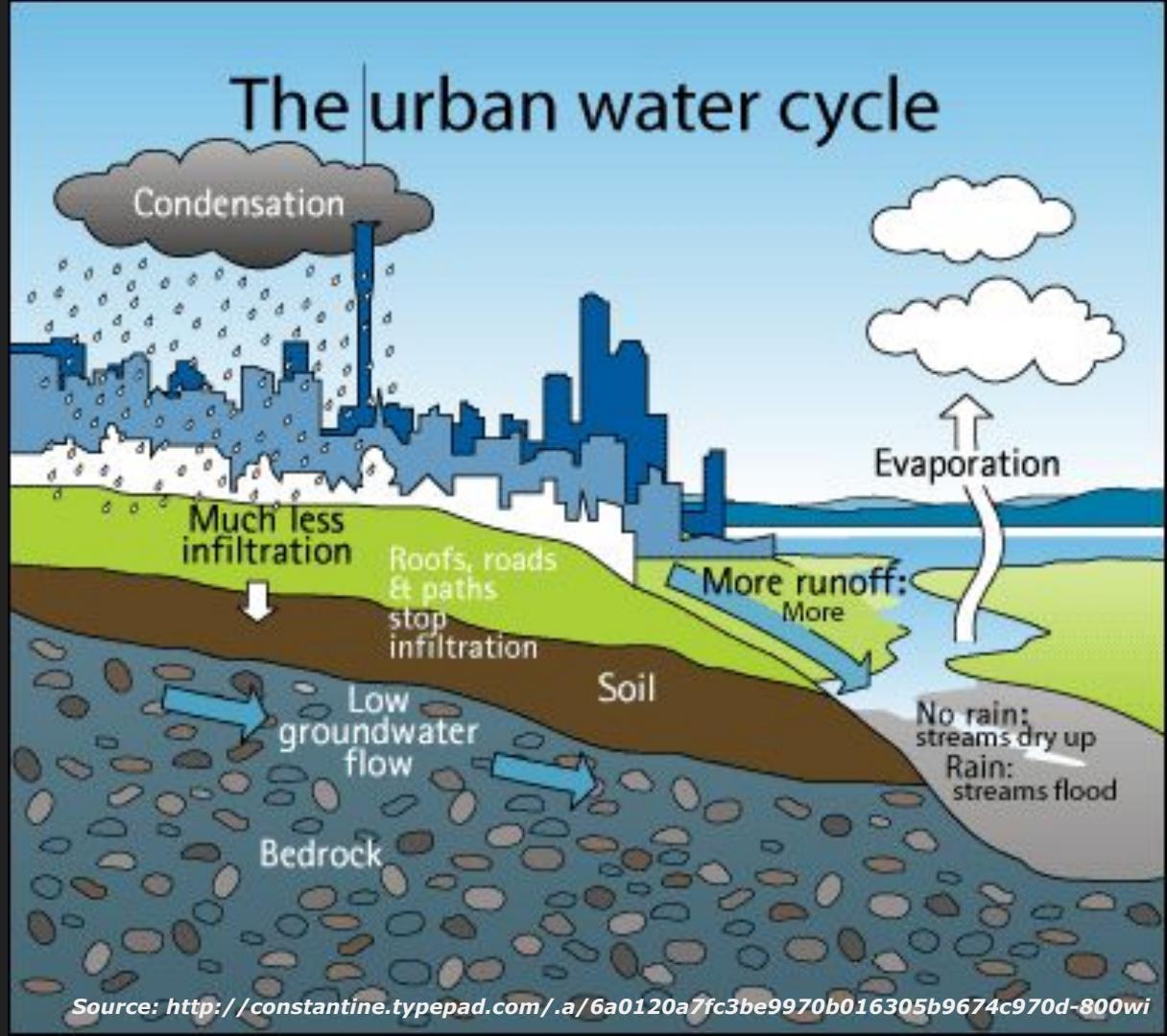
# The Water Cycle

- Different processes
- Different flows and interactions



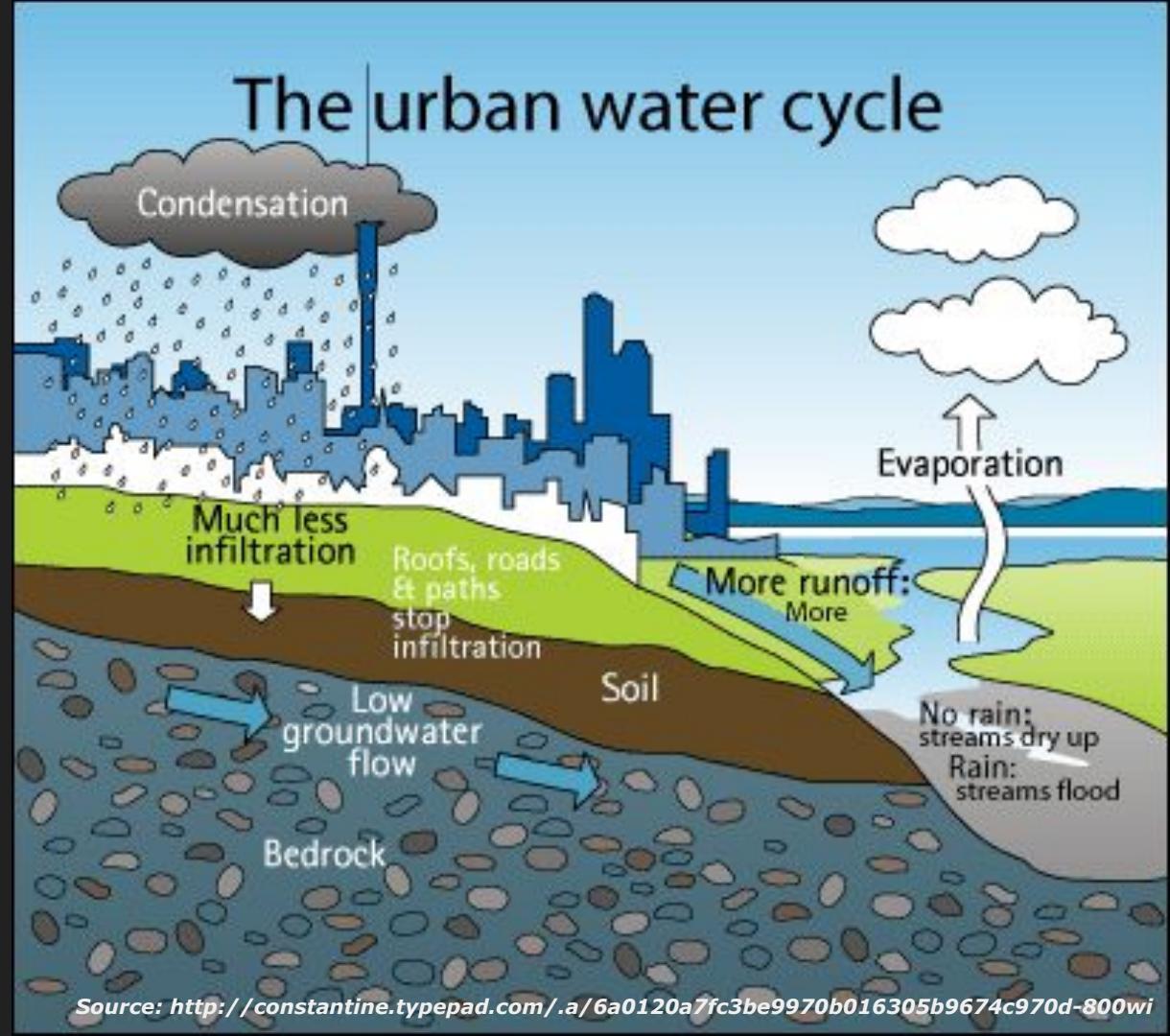
# The Urban Water Cycle

Similar as the water cycle we were taught at school, but with humans messing things up with **dams**, **pipes**, **roads**, **crops** and varying loadings of

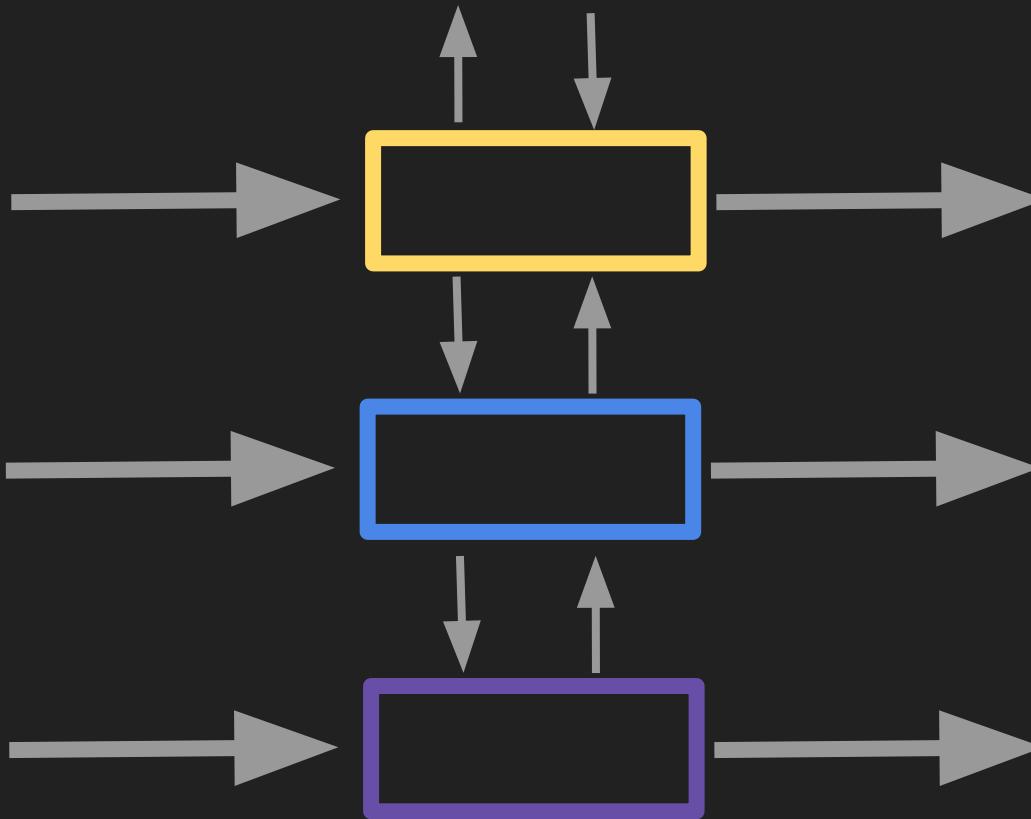


# The Urban Water Cycle

Similar as the water cycle we were taught at school, but with humans messing things up with **dams**, **pipes**, **roads**, **crops** and varying loadings of



# How do we model this system?



# What is a model?

Is a way to understand,  
define, quantify, visualize  
reality by referencing to  
existing and usually commonly  
accepted knowledge.

<Insert Pipe Here>

# What is a model?

Is a way to understand,  
define, quantify, visualize  
reality by referencing to  
existing and usually commonly  
accepted knowledge.



# What is a model?

Is a way to understand,  
define, quantify, visualize  
reality by referencing to  
existing and usually commonly  
accepted knowledge.

Source: <https://en.wikipedia.org/wiki/File:MagrittePipe.jpg>



**The Treachery of Images**  
**(This is not a pipe)**  
René Magritte

# Why model urban water cycle?

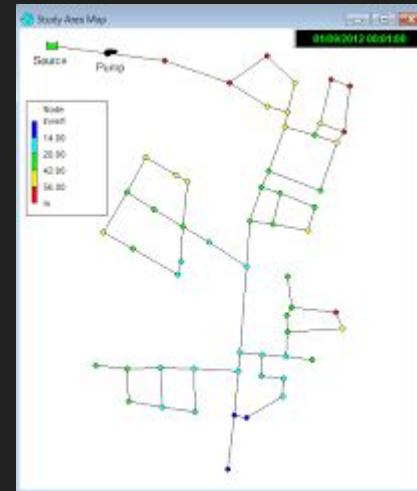
- Evaluate influence of climate change



*Source:* <https://www.epa.gov/water-research/storm-water-management-model-swmm>  
<https://news.nationalgeographic.com/content/dam/news/2017/02/12/conservative-climate-change/>

# Why model urban water cycle?

- Evaluate influence of climate change
- Designing and sizing of drainage system components



# Why model urban water cycle?

- Evaluate influence of climate change
- Designing and sizing of drainage system components
- Sizing detention facilities



# Why model urban water cycle?

- Evaluate influence of climate change
- Designing and sizing of drainage system components
- Sizing detention facilities
- Mapping flood plains of natural channel systems

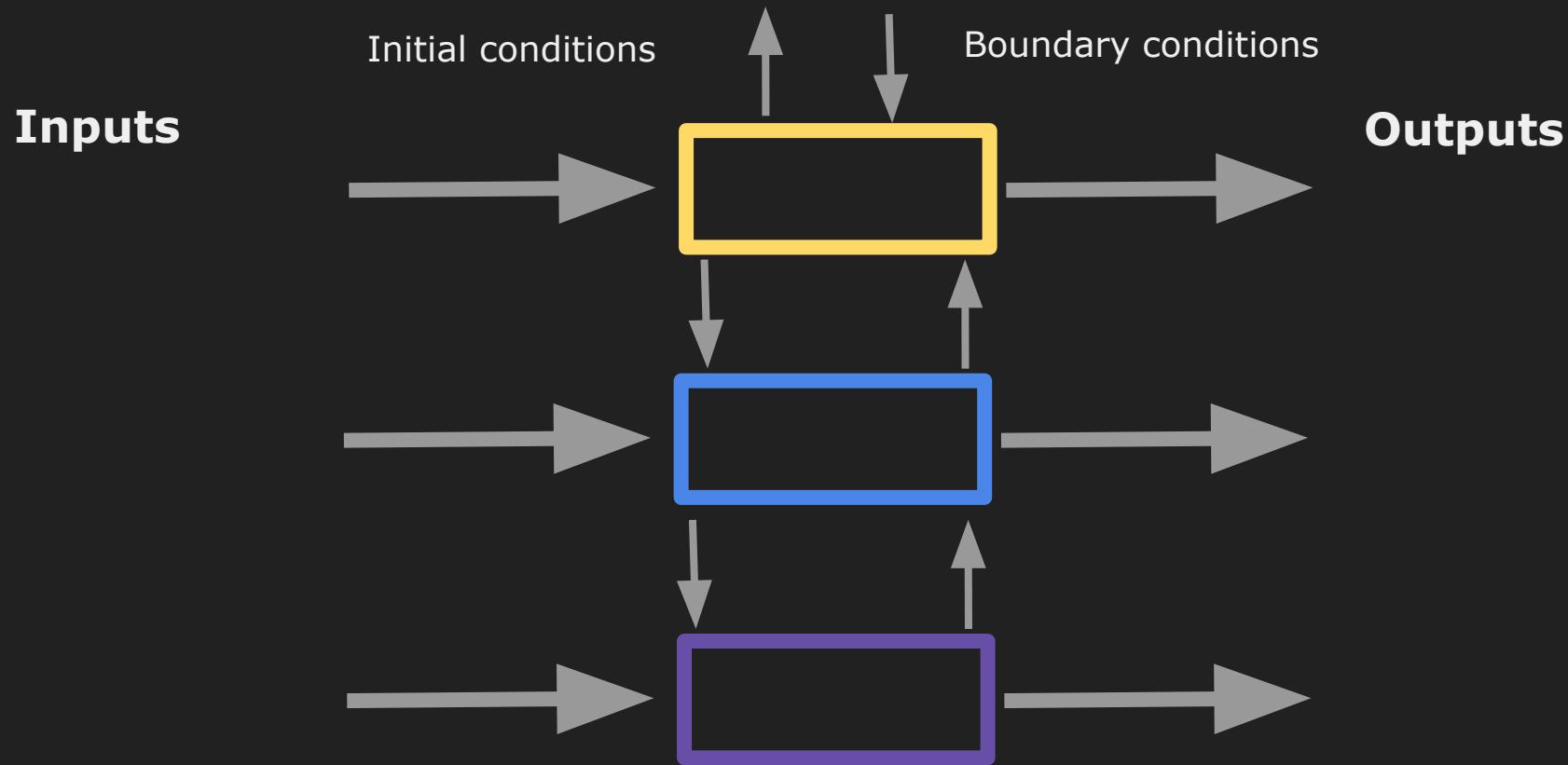


# Why model urban water cycle?

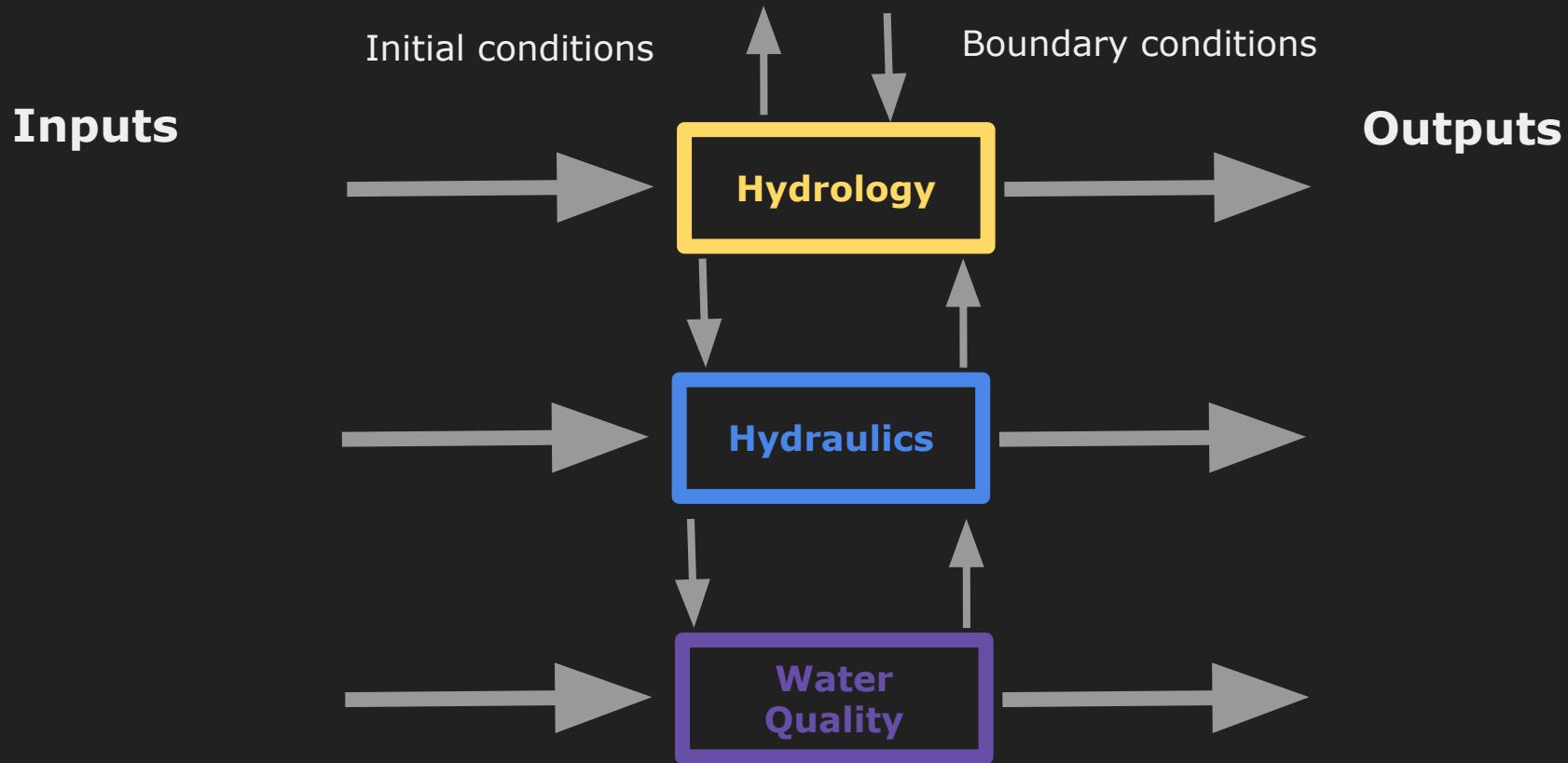
- Evaluate influence of climate change
- Designing and sizing of drainage system components
- Sizing detention facilities
- Mapping flood plains of natural channel systems
- Designing control strategies for minimizing combined sewer overflows.
- Evaluating the impact of inflow and infiltration on sanitary sewer overflows.



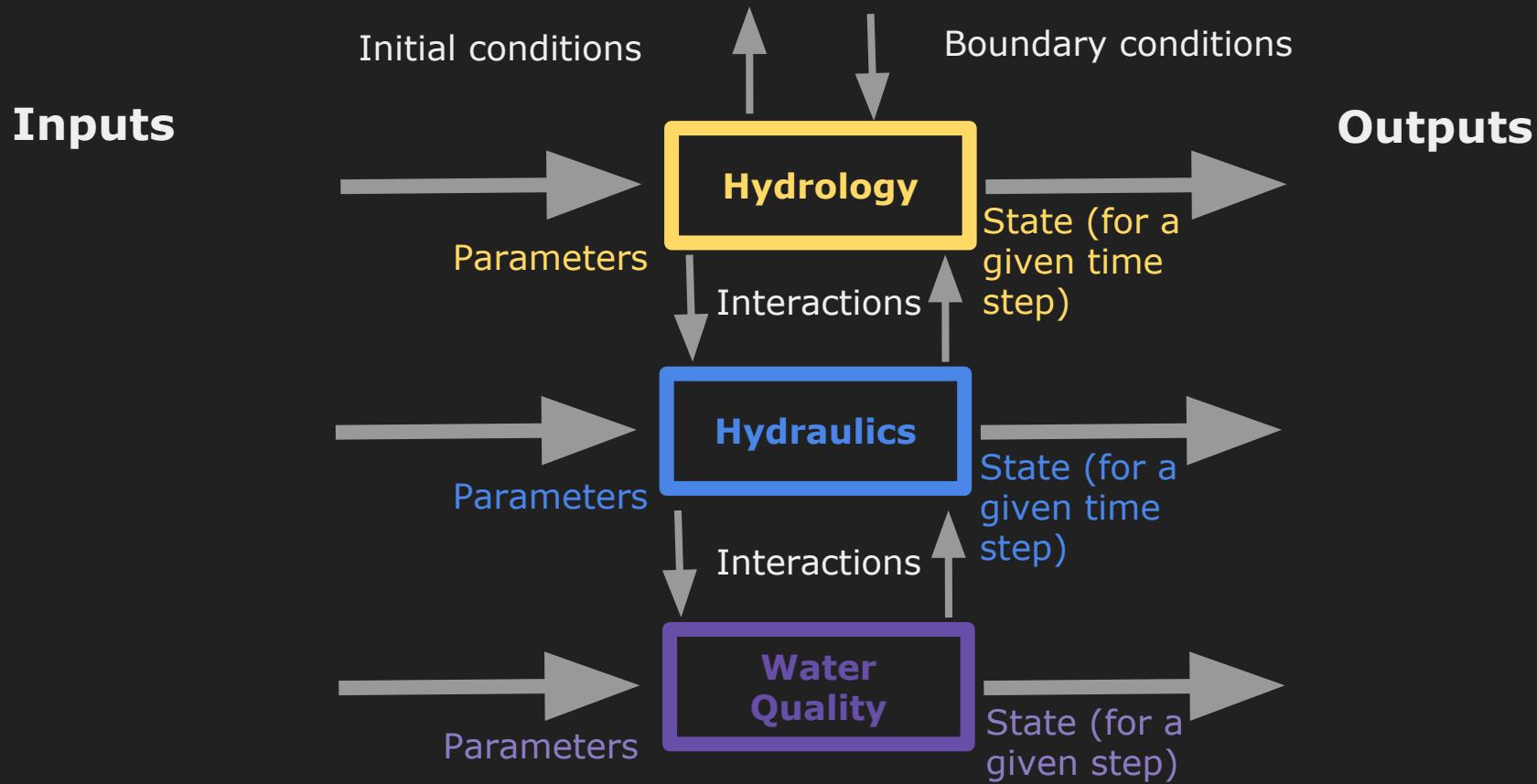
# How do we model this system?



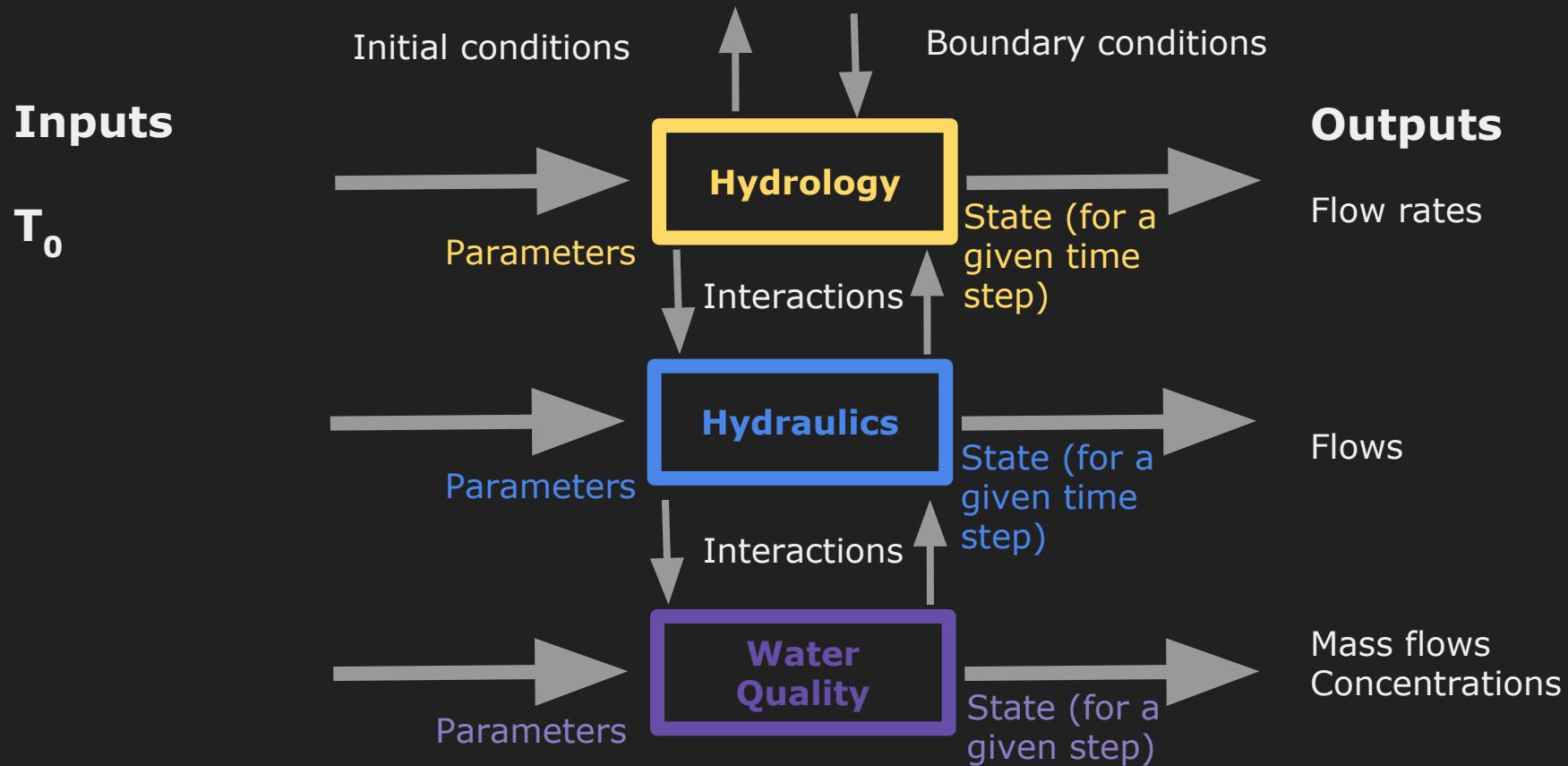
# How do we model this system?



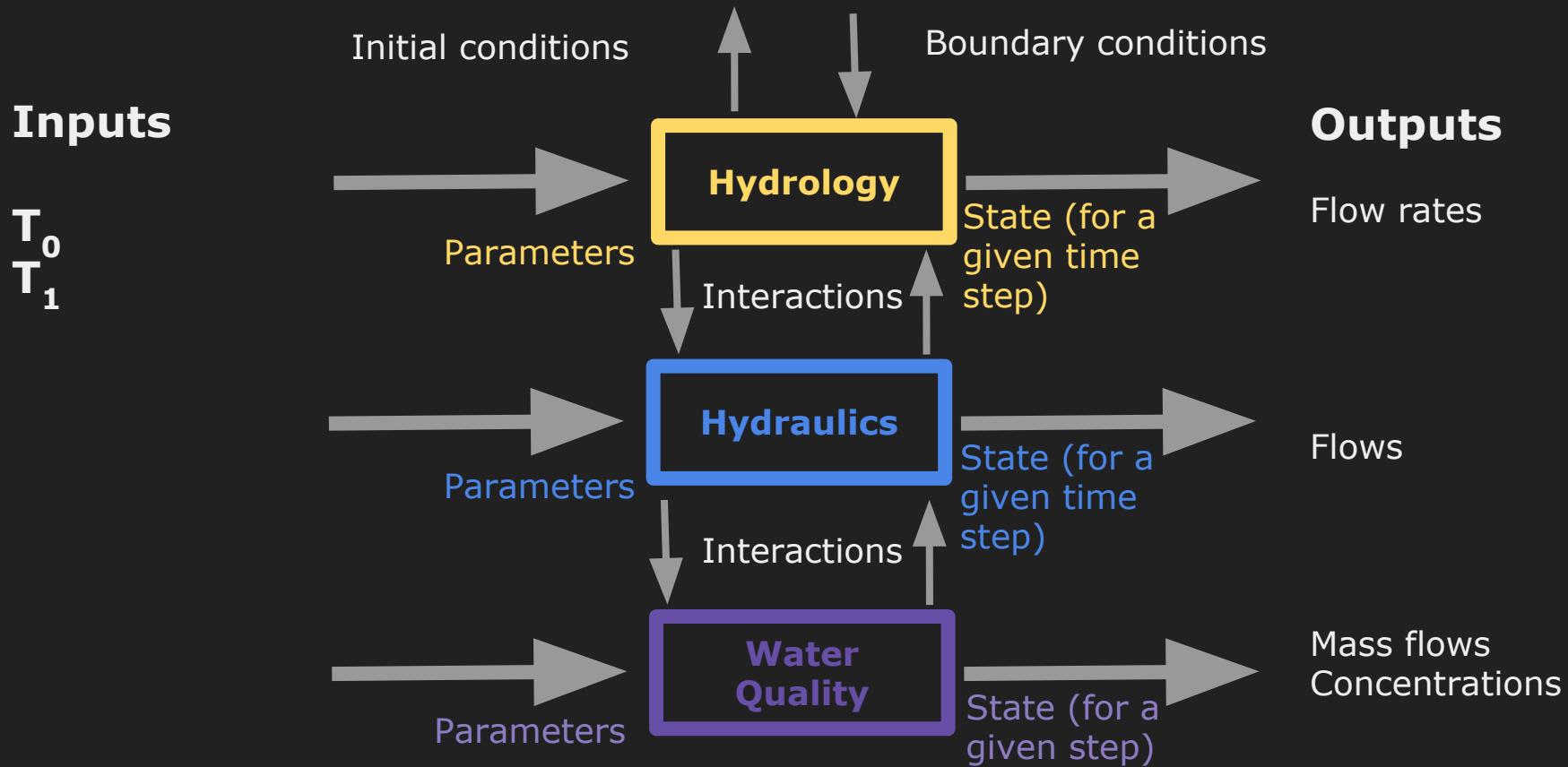
# How do we model this system?



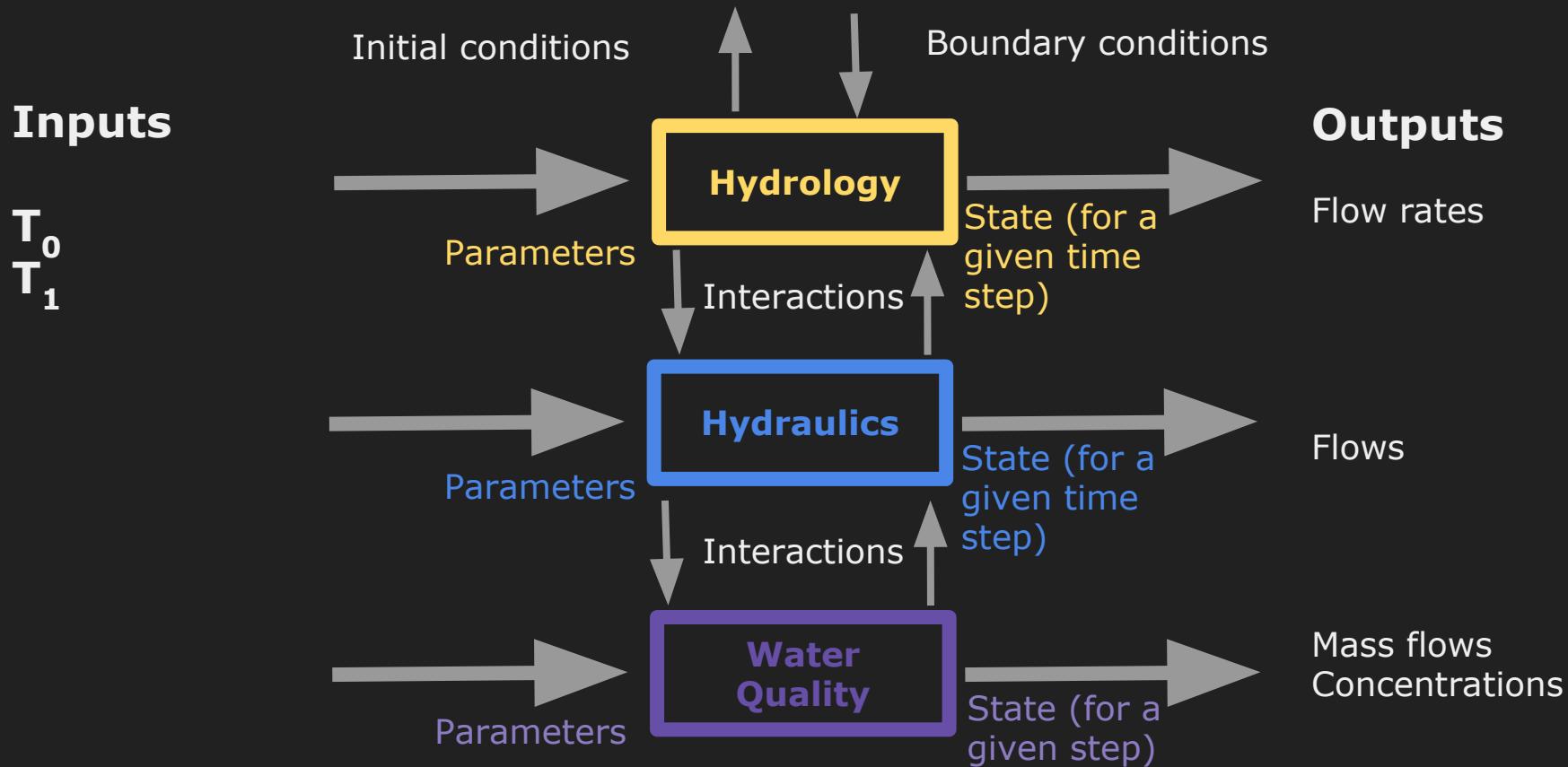
# How do we model this system?



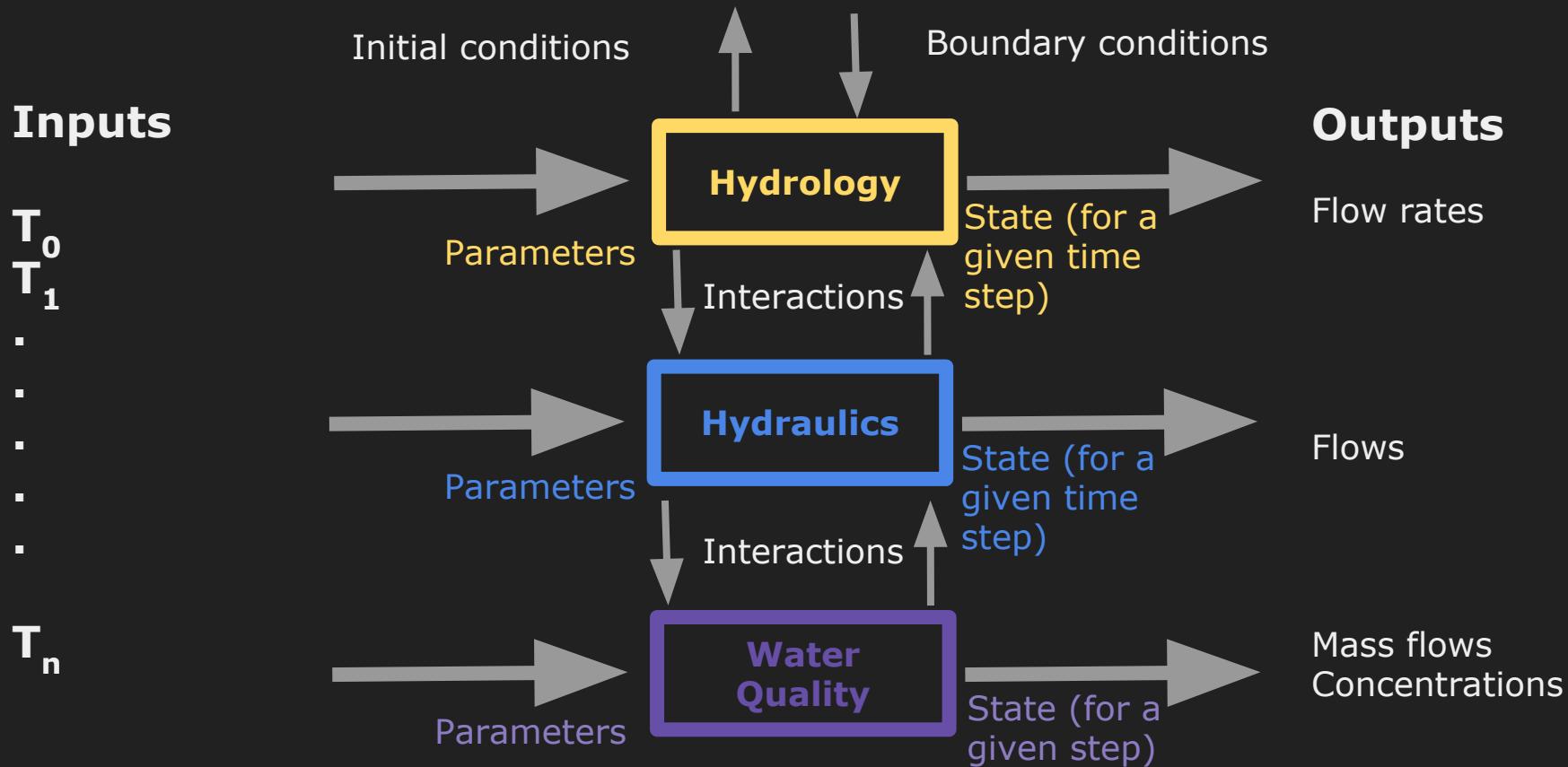
# How do we model this system?



# How do we model this system?



# How do we model this system?



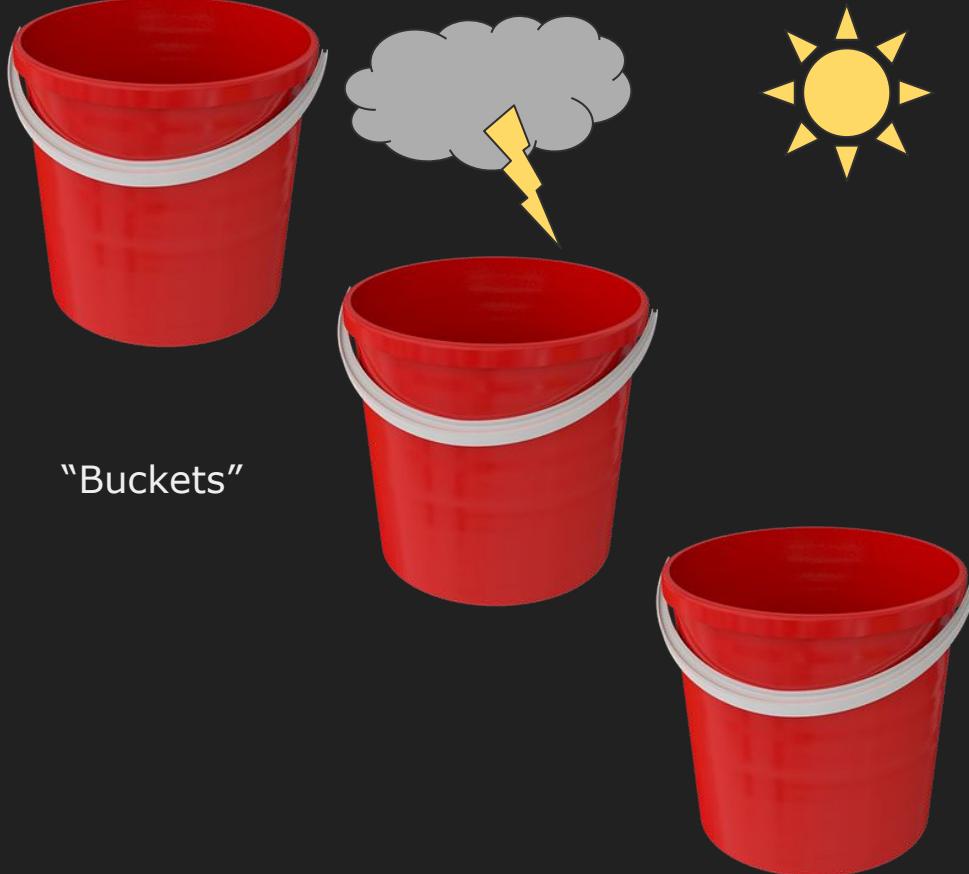
# Hydrological modelling

- **Rainfall**
- **Evaporation**
- **Snow** accumulation
- **Storage**
- **Infiltration**
- **Percolation**
- **Interflow**
- **Overland flow**



# Hydrological modelling

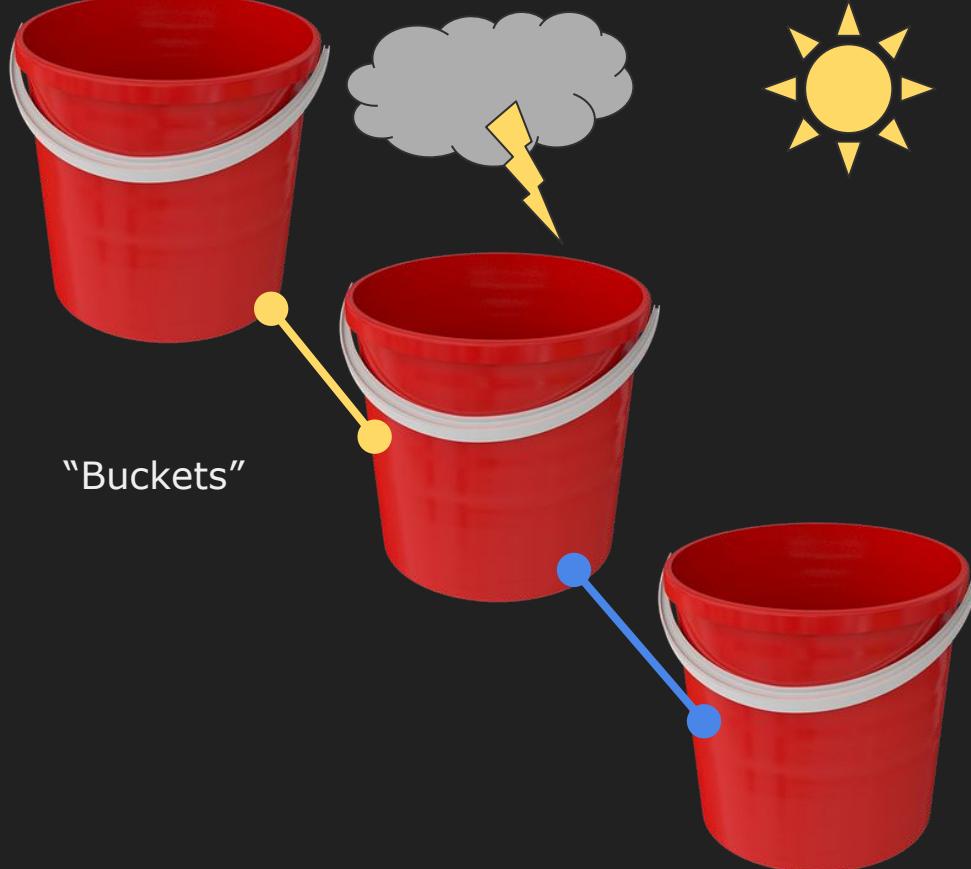
- Rainfall
- Evaporation
- Snow accumulation
- Storage
- Infiltration
- Percolation
- Interflow
- Overland flow



Source: [https://cdn.pixabay.com/photo/2016/09/04/07/02/bucket-1643406\\_960\\_720.png](https://cdn.pixabay.com/photo/2016/09/04/07/02/bucket-1643406_960_720.png)  
<https://www.epa.gov/water-research/storm-water-management-model-swmm>

# Hydrological modelling

- Rainfall
- Evaporation
- Snow accumulation
- Storage
- Infiltration
- Percolation
- Interflow
- Overland flow



Source: [https://cdn.pixabay.com/photo/2016/09/04/07/02/bucket-1643406\\_960\\_720.png](https://cdn.pixabay.com/photo/2016/09/04/07/02/bucket-1643406_960_720.png)  
<https://www.epa.gov/water-research/storm-water-management-model-swmm>

# Hydraulic modelling

- **Drainage** networks
- Natural channel **flows**
- Model special elements, such as:
  - **Storage**
  - **Flow dividers**
  - **Pumps, weirs, and orifices.**

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0$$

$$\frac{\partial Q}{\partial t} + \frac{\partial(Q^2/A)}{\partial x} + gA \frac{\partial H}{\partial x} + gAS_f = 0$$

*The 1D Saint-Venant Equations*

*(Derived from the Navier Stokes Equations)*

*Source: <https://www.epa.gov/water-research/storm-water-management-model-swmm>*

# Hydraulic modelling

- **Drainage** networks
- Natural channel **flows**
- Model special elements, such as:
  - **Storage**
  - **Flow dividers**
  - **Pumps, weirs, and orifices.**

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0$$

$$\frac{\partial Q}{\partial t} + \frac{\partial(Q^2/A)}{\partial x} + gA \frac{\partial H}{\partial x} + gAS_f = 0$$

*The 1D Saint-Venant Equations  
(Derived from the Navier Stokes Equations)*

*Source: <https://www.epa.gov/water-research/storm-water-management-model-swmm>*

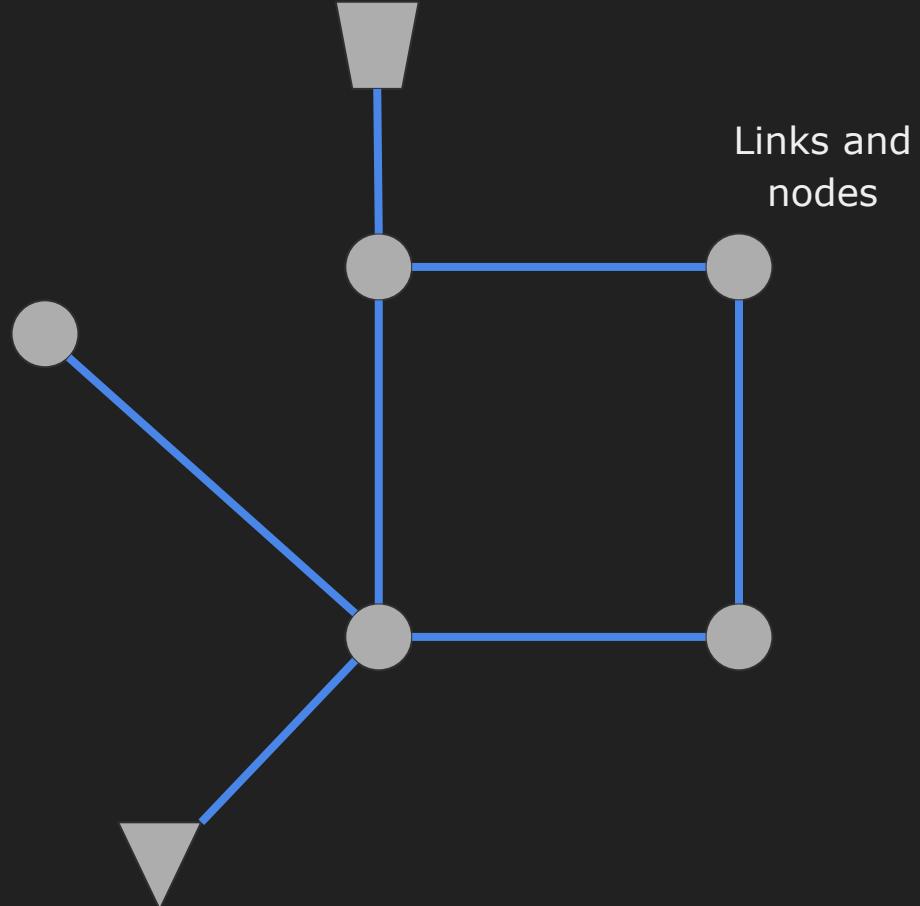
Links and nodes

# Hydraulic modelling

- **Drainage** networks
- Natural channel **flows**
- Model special elements, such as:
  - **Storage**
  - **Flow dividers**
  - **Pumps, weirs, and orifices.**

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0$$

$$\frac{\partial Q}{\partial t} + \frac{\partial(Q^2/A)}{\partial x} + gA \frac{\partial H}{\partial x} + gAS_f = 0$$



*The 1D Saint-Venant Equations*

*(Derived from the Navier Stokes Equations)*

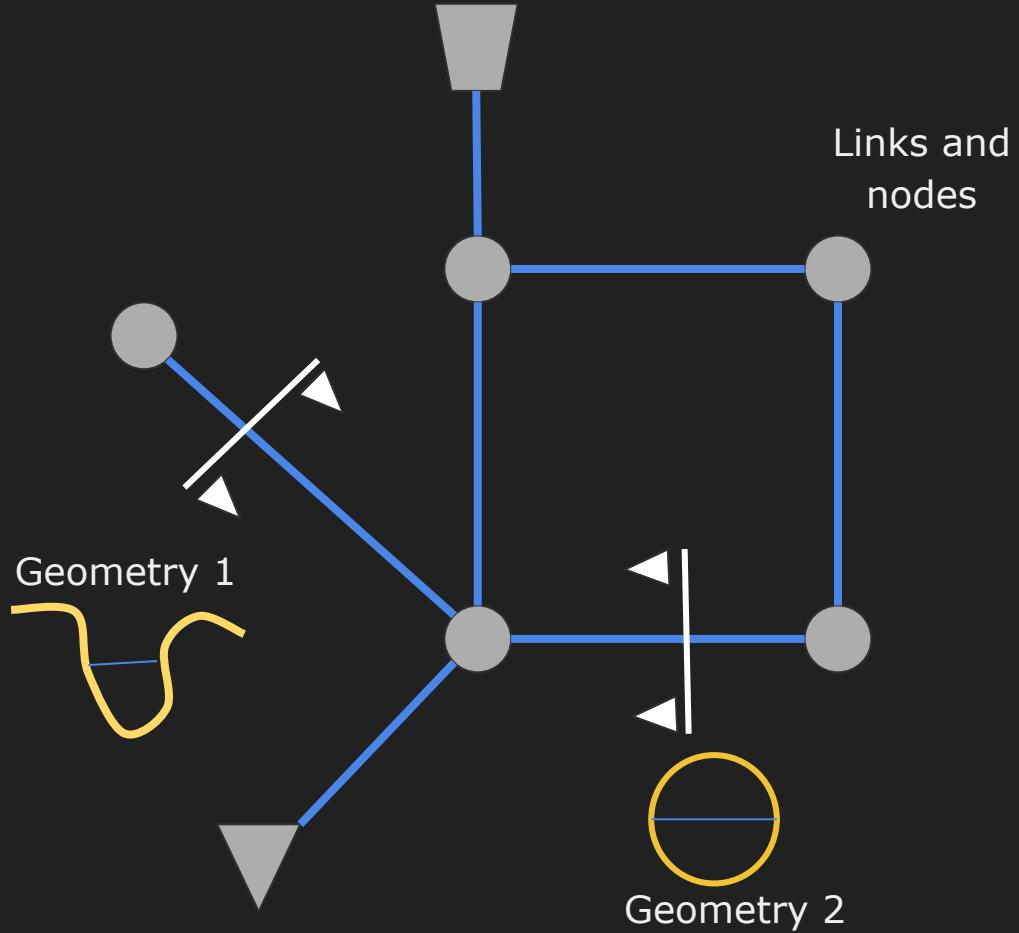
*Source: <https://www.epa.gov/water-research/storm-water-management-model-swmm>*

# Hydraulic modelling

- **Drainage** networks
- Natural channel **flows**
- Model special elements, such as:
  - **Storage**
  - **Flow dividers**
  - **Pumps, weirs, and orifices.**

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0$$

$$\frac{\partial Q}{\partial t} + \frac{\partial(Q^2/A)}{\partial x} + gA \frac{\partial H}{\partial x} + gAS_f = 0$$



*The 1D Saint-Venant Equations*

*(Derived from the Navier Stokes Equations)*

*Source: <https://www.epa.gov/water-research/storm-water-management-model-swmm>*

# Water quality modelling

*The 1-D Advection Dispersion Equation*

$$\frac{\partial c}{\partial t} = -\frac{\partial(uc)}{\partial x} + \frac{\partial}{\partial x} \left( D \frac{\partial c}{\partial x} \right) + r(c)$$

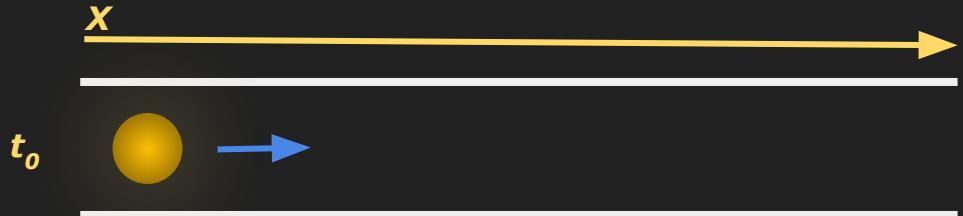
- **Pollutant build up**
- **Pollutant wash-off**
- **Routing** of water quality constituents through the system.
- **Changes in constituent concentration** through treatment in storage units or by natural processes in pipes and channels.

# Water quality modelling

- **Pollutant build up**
- **Pollutant wash-off**
- **Routing** of water quality constituents through the system.
- **Changes in constituent concentration** through treatment in storage units or by natural processes in pipes and channels.

*The 1-D Advection Dispersion Equation*

$$\frac{\partial c}{\partial t} = -\frac{\partial(uc)}{\partial x} + \frac{\partial}{\partial x} \left( D \frac{\partial c}{\partial x} \right) + r(c)$$

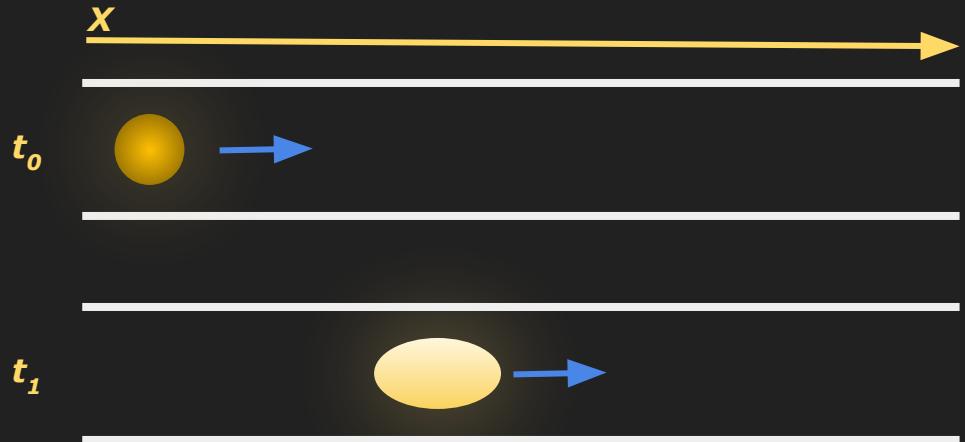


# Water quality modelling

- **Pollutant build up**
- **Pollutant wash-off**
- **Routing** of water quality constituents through the system.
- **Changes in constituent concentration** through treatment in storage units or by natural processes in pipes and channels.

*The 1-D Advection Dispersion Equation*

$$\frac{\partial c}{\partial t} = -\frac{\partial(uc)}{\partial x} + \frac{\partial}{\partial x} \left( D \frac{\partial c}{\partial x} \right) + r(c)$$

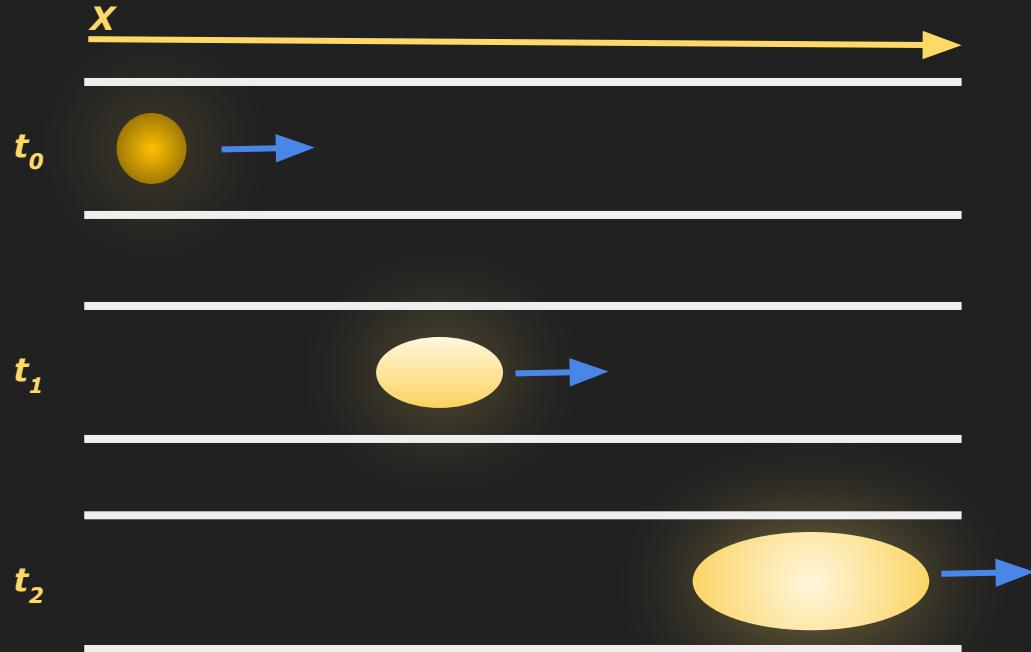


# Water quality modelling

- **Pollutant build up**
- **Pollutant wash-off**
- **Routing** of water quality constituents through the system.
- **Changes in constituent concentration** through treatment in storage units or by natural processes in pipes and channels.

*The 1-D Advection Dispersion Equation*

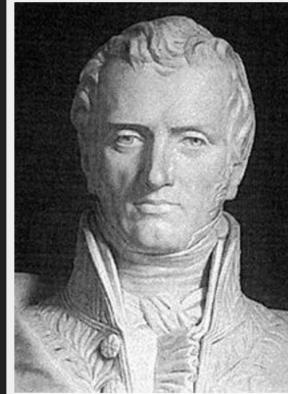
$$\frac{\partial c}{\partial t} = -\frac{\partial(uc)}{\partial x} + \frac{\partial}{\partial x} \left( D \frac{\partial c}{\partial x} \right) + r(c)$$



<intermission name="navier-stokes">

# Hydraulic modelling

In Computational Fluid Dynamics (CFD) we have a deity:



*Claude-Louis Navier*  
1785-1836

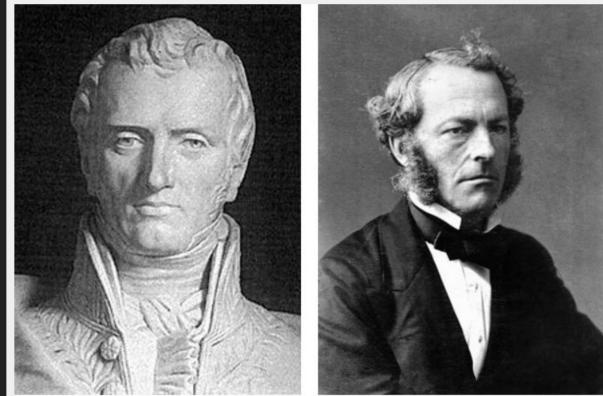


*Sir George Stokes*  
1819-1903

## The Navier-Stokes Equations

# Hydraulic modelling

In Computational Fluid Dynamics (CFD) we have a deity:



Claude-Louis Navier

1785-1836

Sir George Stokes

1819-1903

## The Navier-Stokes Equations

Describe viscous flows



Coordinates: $(x,y,z)$	Time : $t$	Pressure: $p$	Heat Flux: $q$
Velocity Components: $(u,v,w)$	Density: $\rho$	Stress: $\tau$	Reynolds Number: $Re$
		Total Energy: $E_t$	Prandtl Number: $Pr$

$$\text{Continuity: } \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0$$
$$\text{X - Momentum: } \frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} + \frac{\partial(\rho uw)}{\partial z} = - \frac{\partial p}{\partial x} + \frac{1}{Re_r} \left[ \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} \right]$$
$$\text{Y - Momentum: } \frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho v^2)}{\partial y} + \frac{\partial(\rho vw)}{\partial z} = - \frac{\partial p}{\partial y} + \frac{1}{Re_r} \left[ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} \right]$$
$$\text{Z - Momentum } \frac{\partial(\rho w)}{\partial t} + \frac{\partial(\rho uw)}{\partial x} + \frac{\partial(\rho vw)}{\partial y} + \frac{\partial(\rho w^2)}{\partial z} = - \frac{\partial p}{\partial z} + \frac{1}{Re_r} \left[ \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} \right]$$
$$\text{Energy: } \frac{\partial(E_t)}{\partial t} + \frac{\partial(uE_t)}{\partial x} + \frac{\partial(vE_t)}{\partial y} + \frac{\partial(wE_t)}{\partial z} = - \frac{\partial(ue)}{\partial x} - \frac{\partial(ve)}{\partial y} - \frac{\partial(we)}{\partial z} - \frac{1}{Re_r Pr_r} \left[ \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z} \right] \\ + \frac{1}{Re_r} \left[ \frac{\partial}{\partial x} (u\tau_{xx} + v\tau_{xy} + w\tau_{xz}) + \frac{\partial}{\partial y} (u\tau_{xy} + v\tau_{yy} + w\tau_{yz}) + \frac{\partial}{\partial z} (u\tau_{xz} + v\tau_{yz} + w\tau_{zz}) \right]$$

# Hydraulic modelling

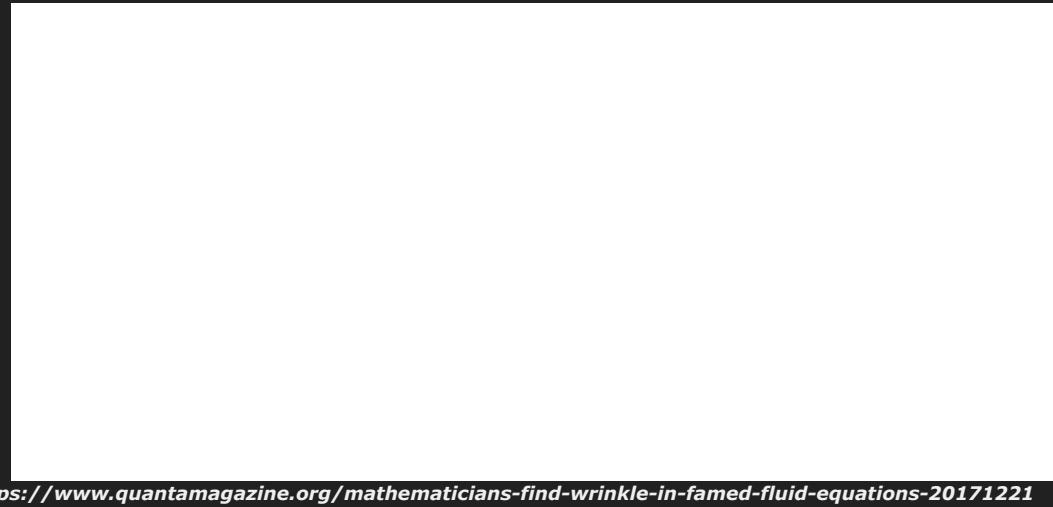
In Computational Fluid Dynamics (CFD) we have a deity:



*Source: <https://www.metoffice.gov.uk>*

The **Kelvin–Helmholtz instability**

## The Navier-Stokes Equations



*Source: <https://www.quantamagazine.org/mathematicians-find-wrinkle-in-famed-fluid-equations-20171221>*

# Hydraulic modelling

In Computational Fluid Dynamics (CFD) we have a deity:

## The Navier-Stokes Equations



### Millennium Problems

#### Yang–Mills and Mass Gap

Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.

#### Riemann Hypothesis

The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.

#### P vs NP Problem

If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

#### Navier–Stokes Equation

This is the equation which governs the flow of fluids such as water and air. However, there is no proof for the most basic questions one can ask: do solutions exist, and are they unique? Why ask for a proof? Because a proof gives not only certitude, but also understanding.

#### Hodge Conjecture

The answer to this conjecture determines how much of the topology of the solution set of a system of algebraic equations can be defined in terms of further algebraic equations. The Hodge conjecture is known in certain special cases, e.g., when the solution set has dimension less than four. But in dimension four it is unknown.

#### Poincaré Conjecture

In 1904 the French mathematician Henri Poincaré asked if the three dimensional sphere is characterized as the unique simply connected three manifold. This question, the Poincaré conjecture, was a special case of Thurston's geometrization conjecture. Perelman's proof tells us that every three manifold is built from a set of standard pieces, each with one of eight well-understood geometries.

#### Birch and Swinnerton-Dyer Conjecture

Supported by much experimental evidence, this conjecture relates the number of points on an elliptic curve mod p to the rank of the group of rational points. Elliptic curves, defined by cubic equations in two variables, are fundamental mathematical objects that arise in many areas: Wiles' proof of the Fermat Conjecture, factorization of numbers into primes, and cryptography, to name three.

# Hydraulic modelling

In Computational Fluid Dynamics (CFD) we have a deity:

## The Navier-Stokes Equations



### Millennium Problems

#### Yang–Mills and Mass Gap

Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.

#### Riemann Hypothesis

The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.

#### P vs NP Problem

If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

#### Navier–Stokes Equation

This is the equation which governs the flow of fluids such as water and air. However, there is no proof for the most basic questions one can ask: do solutions exist, and are they unique? Why ask for a proof? Because a proof gives not only certitude, but also understanding.

**Do solutions exist?  
Are they unique?**

# Hydraulic modelling

But last September / October

## Nonuniqueness of weak solutions to the Navier–Stokes equation

Tristan Buckmaster, Vlad Vicol

(Submitted on 28 Sep 2017 ([v1](#)), last revised 5 Oct 2017 (this version, v2))

For initial datum of finite kinetic energy, Leray has proven in 1934 that there exists at least one global in time finite energy weak solution of the 3D Navier–Stokes equations. In this paper we prove that weak solutions of the 3D Navier–Stokes equations are not unique in the class of weak solutions with finite kinetic energy. Moreover, we prove that Holder continuous dissipative weak solutions of the 3D Euler equations may be obtained as a strong vanishing viscosity limit of a sequence of finite energy weak solutions of the 3D Navier–Stokes equations.

Comments: 34 pages, added comments regarding Oseen solutions

Subjects: Analysis of PDEs (math.AP); Mathematical Physics (math-ph)

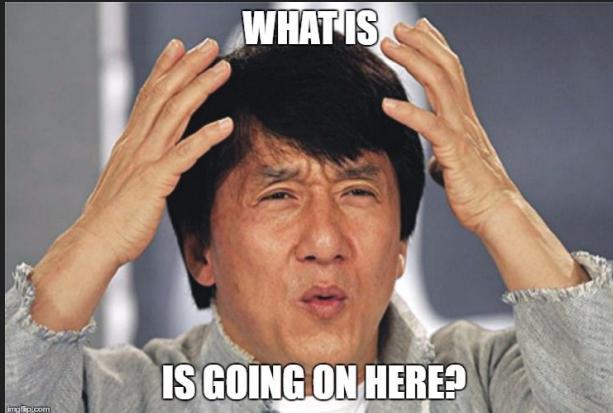
Cite as: [arXiv:1709.10033](#) [math.AP]

(or [arXiv:1709.10033v2](#) [math.AP] for this version)

Source: <https://arxiv.org/abs/1709.10033>

# Hydraulic modelling

But last September / October



## Nonuniqueness of weak solutions to the Navier–Stokes equation

Tristan Buckmaster, Vlad Vicol

(Submitted on 28 Sep 2017 ([v1](#)), last revised 5 Oct 2017 (this version, v2))

For initial datum of finite kinetic energy, Leray has proven in 1934 that there exists at least one global in time finite energy weak solution of the 3D Navier–Stokes equations. In this paper we prove that weak solutions of the 3D Navier–Stokes equations are not unique in the class of weak solutions with finite kinetic energy. Moreover, we prove that Holder continuous dissipative weak solutions of the 3D Euler equations may be obtained as a strong vanishing viscosity limit of a sequence of finite energy weak solutions of the 3D Navier–Stokes equations.

Comments: 34 pages, added comments regarding Oseen solutions

Subjects: Analysis of PDEs (math.AP); Mathematical Physics (math-ph)

Cite as: [arXiv:1709.10033](#) [math.AP]

(or [arXiv:1709.10033v2](#) [math.AP] for this version)

Source: <https://arxiv.org/abs/1709.10033>

# Hydraulic modelling

## Nonuniqueness of weak solutions to the Navier–Stokes equation

Tristan Buckmaster, Vlad Vicol

(Submitted on 28 Sep 2017 (v1), last revised 5 Oct 2017 (this version, v2))

For initial datum of finite kinetic energy, Leray has proven in 1934 that there exists at least one global in time finite energy weak solution of the 3D Navier–Stokes equations. In this paper we prove that weak solutions of the 3D Navier–Stokes equations are not unique in the class of weak solutions with finite kinetic energy. Moreover, we prove that Holder continuous dissipative weak solutions of the 3D Euler equations may be obtained as a strong vanishing viscosity limit of a sequence of finite energy weak solutions of the 3D Navier–Stokes equations.

Comments: 34 pages, added comments regarding Oseen solutions

Subjects: Analysis of PDEs (math.AP); Mathematical Physics (math-ph)

Cite as: arXiv:1709.10033 [math.AP]

(or arXiv:1709.10033v2 [math.AP] for this version)

Source: <https://arxiv.org/abs/1709.10033>

**Deobfuscating the title:**

FLUID DYNAMICS

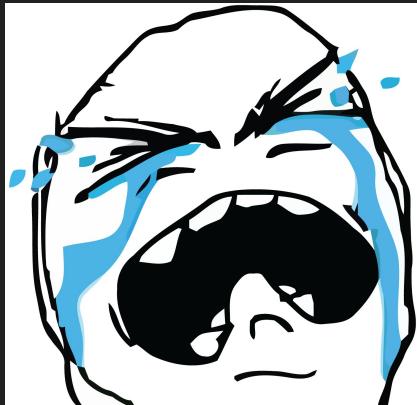
## Mathematicians Find Wrinkle in Famed Fluid Equations



Two mathematicians prove that under certain extreme conditions, the Navier–Stokes equations output nonsense.

Source: <https://www.quantamagazine.org/mathematicians-find-wrinkle-in-famed-fluid-equations-20171221>

# Hydraulic modelling



## Nonuniqueness of weak solutions to the Navier–Stokes equation

Tristan Buckmaster, Vlad Vicol

(Submitted on 28 Sep 2017 (v1), last revised 5 Oct 2017 (this version, v2))

For initial datum of finite kinetic energy, Leray has proven in 1934 that there exists at least one global in time finite energy weak solution of the 3D Navier–Stokes equations. In this paper we prove that weak solutions of the 3D Navier–Stokes equations are not unique in the class of weak solutions with finite kinetic energy. Moreover, we prove that Holder continuous dissipative weak solutions of the 3D Euler equations may be obtained as a strong vanishing viscosity limit of a sequence of finite energy weak solutions of the 3D Navier–Stokes equations.

Comments: 34 pages, added comments regarding Oseen solutions

Subjects: Analysis of PDEs (math.AP); Mathematical Physics (math-ph)

Cite as: arXiv:1709.10033 [math.AP]

(or arXiv:1709.10033v2 [math.AP] for this version)

Source: <https://arxiv.org/abs/1709.10033>

### Deobfuscating the title:

FLUID DYNAMICS

## Mathematicians Find Wrinkle in Famed Fluid Equations

34 |

Two mathematicians prove that under certain extreme conditions, the Navier–Stokes equations output nonsense.

**Output nonsense**

Source: <https://www.quantamagazine.org/mathematicians-find-wrinkle-in-famed-fluid-equations-20171221>

</intermission>

# US EPA: United States Environmental Protection Agency



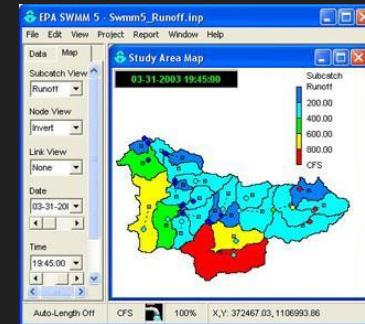
*Source:* <https://www.epa.gov/water-research/storm-water-management-model-swmm>  
<https://www.epa.gov/water-research/epanet>

# US EPA: United States Environmental Protection Agency



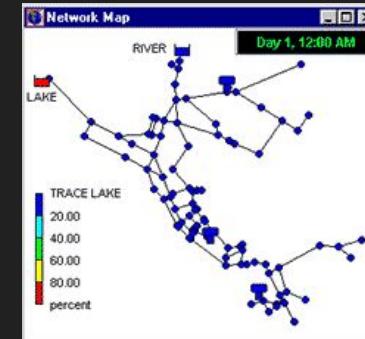
## SWMM (Sewers)

First developed between 1969–1971

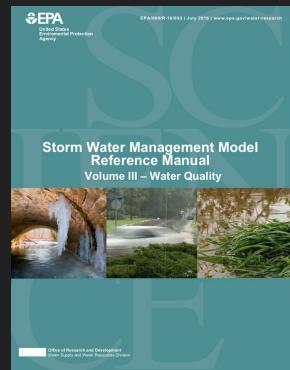
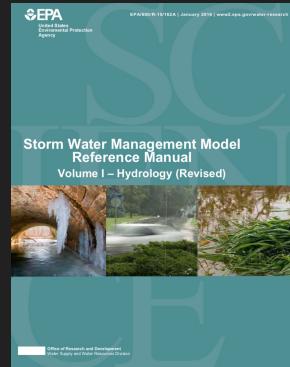
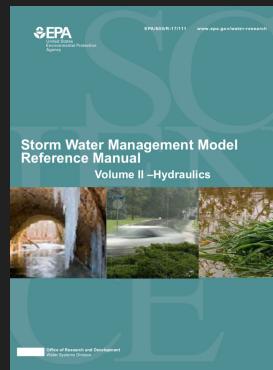
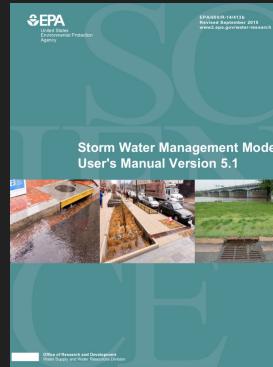
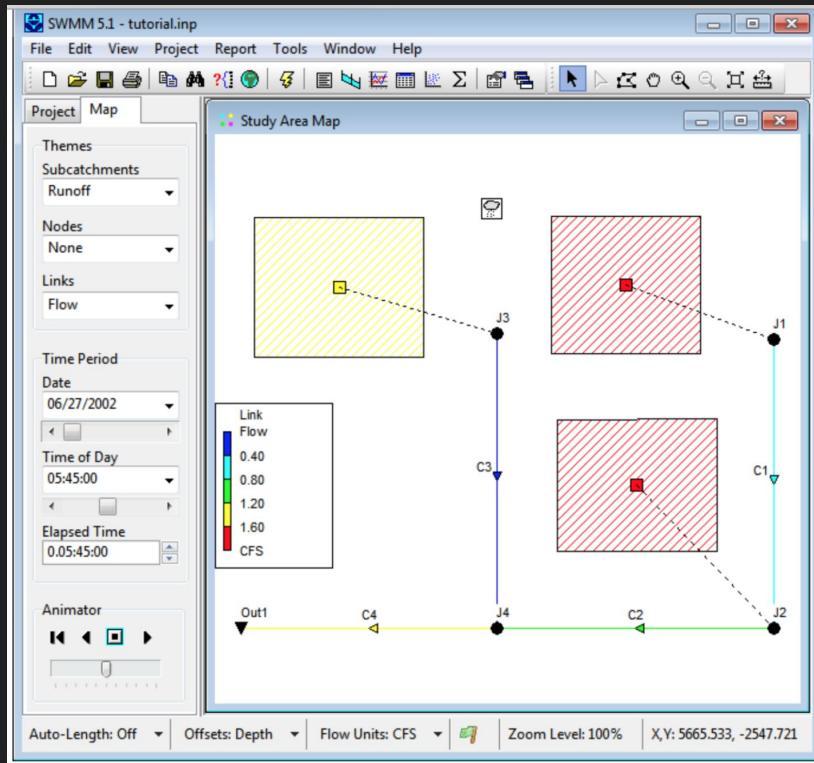


## EPANET (Water distribution)

EPANET first appeared in 1993



# SWMM: StormWater Management Model



# SWMM: So where is the API?

How to automate  
modeling?

## SWMM Knowledge Base

20-Mar-2007

### API for accessing SWMM

 Dr. Darko Joksimovic

One of our research students is interested in coupling SWMM5 to an optimization engine in order to investigate the potential for performing reliability analysis on sewer systems.

Unlike EPANET, which has a mature API capable of straightforward interfacing to an optimizer, SWMM appears to lack this facility. As it stands, the only way to manipulate a



Lew Rossman

Providing an API (or programmer's toolkit) for SWMM 5 is on our to-do list. It is, however, competing with the need to get a comprehensive Reference Manual published, so it probably won't happen until several months from now.

# (EPA) SWMM: Is on GitHub

For some years  
already...

This organization Search Pull requests Issues Marketplace Explore

U.S. Environmental Protection Agency <http://www.epa.gov>

Repositories 103 People 17

Storm Type: All ▾ Language: All ▾

1 result for repositories matching Storm

**Stormwater-Management-Model**  
ORD Stormwater Management Model repository

● C ★ 19 45 Updated on May 2, 2017

Top languages

- JavaScript
- R
- HTML
- Python
- Jupyter Notebook

People 17 >

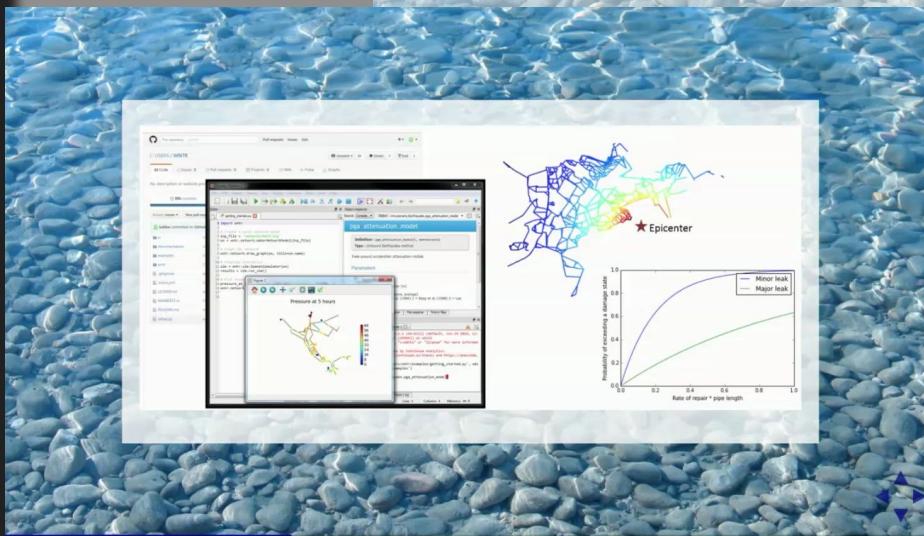
Source: <https://www.github.com/USEPA>

# Plus a bunch of other tools!

Go check them out!

<https://github.com/usepa>

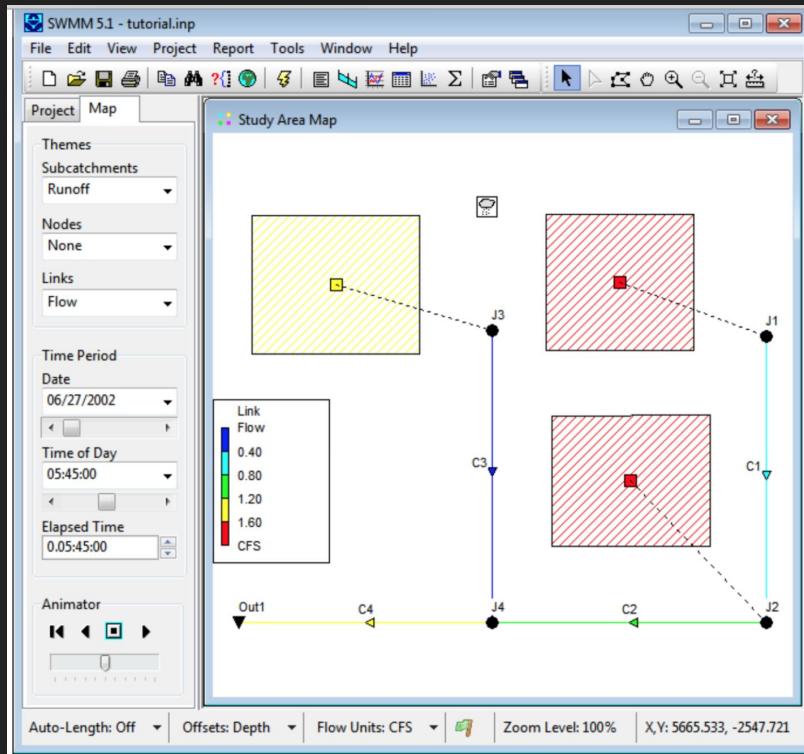
US EPA WNTR: Water Network Tool for Resilience



Katy Huff Keynote PyCon US 2017

<https://www.youtube.com/watch?v=kaGS4YXwciQ>

# SWMM: StormWater Management Model



## Limitations:

- Originally Windows only
- (until 2015) no API to interact programmatically with models
- No bindings for a higher level language
- This GUI is showing its age...
- Control Language not flexible enough for fast pace iteration

<intermission name="networking-and-lying">

# Back in 2012



Assela PATHIRANA

Associate Professor of Integrated Urban  
Water Cycle Management

- [https://pypi.python.org/pypi/SWMM5 \(SWIG\)](https://pypi.python.org/pypi/SWMM5)
- [https://pypi.python.org/pypi/SWMM5\\_EA](https://pypi.python.org/pypi/SWMM5_EA)



Gonzalo

Doing MSc. thesis “Evaluating the impact of climate change on urban scale extreme rainfall events”

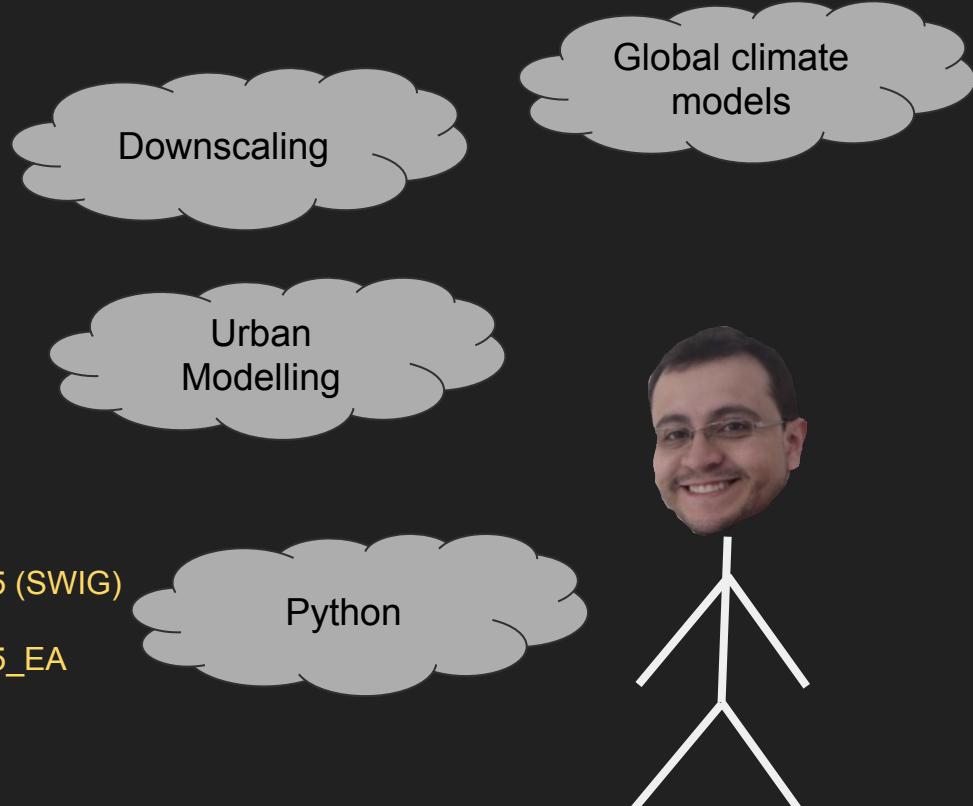
# Back in 2012



Assela PATHIRANA

Associate Professor of Integrated Urban  
Water Cycle Management

- [https://pypi.python.org/pypi/SWMM5 \(SWIG\)](https://pypi.python.org/pypi/SWMM5)
- [https://pypi.python.org/pypi/SWMM5\\_EA](https://pypi.python.org/pypi/SWMM5_EA)



Gonzalo

Doing MSc. thesis “Evaluating the impact of climate change on urban scale extreme rainfall events”

# Back in 2012



Assela PATHIRANA

Associate Professor of Integrated Urban  
Water Cycle Management



Gonzalo

Doing MSc. thesis “Evaluating the impact of climate change on urban scale extreme rainfall events: Coupling of multiple global circulation models with a stochastic rainfall generator”

# Back in 2012

I like the pyswmm  
name, let's register  
that on PyPI



<https://pypi.python.org/pypi/PySWMM>

# Back in 2016: networking...



9/12/2016



Hey Gonzalo, I am looking to push this project to Python Package Index and I ran into a problem:

<https://pypi.python.org/pypi/pyswmm/0.1.0>

It looks like you have registered the pyswmm project name. Do you have plans to submit a pyswmm to PyPI? If not, would you mind removing it so then this pyswmm project could live there? I really appreciate it!

Thanks

Bryant

10:44



# Back in 2016: networking... and lying



9/12/2016

 Hey Gonzalo, I am looking to push this project to Python Package Index and I ran into a problem:

<https://pypi.python.org/pypi/pyswmm/0.1.0>

It looks like you have registered the pyswmm project name. Do you have plans to submit a pyswmm to PyPI? If not, would you mind removing it so then this pyswmm project could live there? I really appreciate it!

Thanks

Bryant

10:44

Hi Bryant, yes I have plans o upload something next year most likely.

Cheers

12:57



*It's my  
preciousssss*

# Back in 2017: networking... and coding

<https://pypi.python.org/pypi/PySWMM>

» Package Index > pyswmm > 0.4.7

**pyswmm 0.4.7**

*Python Wrapper for SWMM5 API*

*Python Wrapper for Stormwater Management Model (SWMM5)*

Documentation  
<http://pyswmm.readthedocs.io/en/latest/>

Development  
<https://github.com/OpenWaterAnalytics/pyswmm/>

PySWMM Wiki  
<https://github.com/OpenWaterAnalytics/pyswmm/wiki/>

**Build status**

build passing circleci passing build error Scrutinizer 5.74

**Project information**

docs passing

**YouTube Examples**

Stream Results and Adjust Weir Setting

</intermission>

# Open Water Analytics (2015)



Bryant McDonnell



Sam Hatchet



Open Water Analytics [?](#)  
<http://wateranalytics.org>

Repositories 19 People 7

Pinned repositories

**EPANET**  
Forked from USEPA/Water-Distribution-Network-Model  
The Water Distribution System Hydraulic and Water Quality Analysis Toolkit  
● C ★ 37 ⚡ 51

**Stormwater-Management-Model**  
Forked from USEPA/Stormwater-Management-Model  
Open Water Analytics Stormwater Management Model repository  
● C ★ 16 ⚡ 25

**pyswmm**  
Python Wrappers for SWMM  
● Python ★ 34 ⚡ 28

**WNTR**  
Forked from USEPA/WNTR  
An EPANET compatible python package to simulate and analyze water distribution networks under disaster scenarios.  
● Python ★ 6 ⚡ 4

**epanet-dev**  
Development Repository for the Next Version of EPANET  
● C++ ★ 28 ⚡ 22

PySWMM Type: All ▾ Language: All ▾

1 result for repositories matching PySWMM  Clear filter

**pyswmm**  
Python Wrappers for SWMM  
python stormwater hydrology-stormwater-analysis  
● Python ★ 34 ⚡ 28 Updated 10 days ago

Top languages

- Python
- C
- Matlab
- C++
- CSS

Most used topics

- hydraulics
- stormwater
- swmm5

Source: <https://github.com/OpenWaterAnalytics/>

# Open Water Analytics (2015)



Bryant McDonnell



Sam Hatchet



2018/02/06



Attendees: 7 of 26 (max)	
Names - Alphabetically	
Sam Hatchett - Organizer	▼
Adam Erisapha	▼
Attendee 6	▼
Bryant McDonnell - Presenter	▼
Gonzalo Pena-Castellanos - Me	▼
Laurent County - Web	▼
Michael Tryby	▼

This organization Search Pull requests Issues Marketplace Explore

## Open Water Analytics 1

<http://wateranalytics.org>

Repositories 19 People 7

Pinned repositories

**EPANET**  
Forked from USEPA/Water-Distribution-Network-Model  
  
The Water Distribution System Hydraulic and Water Quality Analysis Toolkit  
  
● C ★ 37 ⚡ 51

**Stormwater-Management-Model**  
Forked from USEPA/Stormwater-Management-Model  
  
Open Water Analytics Stormwater Management Model repository  
  
● C ★ 16 ⚡ 25

**pyswmm**  
Python Wrappers for SWMM  
  
● Python ★ 34 ⚡ 28

PySWMM Type: All ▾ Language: All ▾

1 result for repositories matching PySWMM Clear filter

**pyswmm**  
Python Wrappers for SWMM  
  
python stormwater hydrology-stormwater-analysis  
● Python ★ 34 ⚡ 28 Updated 10 days ago

Top languages

- Python ● C ● Matlab ● C++
- CSS

Most used topics

- hydraulics stormwater
- swmm5

Source: <https://github.com/OpenWaterAnalytics/>

# Open Water Analytics (2015)

*Why should we care about  
Open Source?*

- Let's move our community away from a single developer's vision
- Else, Research Projects Grow and Die
- Combine Industry with Academia and Open Source developers

The screenshot shows the GitHub organization page for 'Open Water Analytics'. At the top, there are links for 'This organization', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The user profile icon of a person with a blue checkmark is visible in the top right corner.

The organization's logo features a stylized water droplet icon above the text 'Open Water Analytics' and a link to 'http://wateranalytics.org'.

Below the header, there are two main sections: 'Repositories 19' and 'People 7'. Under 'Pinned repositories', there are five boxes:

- EPANET**: Forked from USEPA/Water-Distribution-Network-Model. Description: The Water Distribution System Hydraulic and Water Quality Analysis Toolkit. Stats: 0 C, 37 stars, 51 forks.
- Stormwater-Management-Model**: Forked from USEPA/Stormwater-Management-Model. Description: Open Water Analytics Stormwater Management Model repository. Stats: 0 C, 16 stars, 25 forks.
- pyswmm**: Python Wrappers for SWMM. Stats: Python, 34 stars, 28 forks.
- WNTR**: Forked from USEPA/WNTR. Description: An EPANET compatible python package to simulate and analyze water distribution networks under disaster scenarios. Stats: Python, 6 stars, 4 forks.
- epanet-dev**: Development Repository for the Next Version of EPANET. Stats: C++, 28 stars, 22 forks.

At the bottom of the pinned repositories section, there is a search bar containing 'PySWMM', a 'Type: All' dropdown, and a 'Language: All' dropdown.

The search results for 'PySWMM' show one result: 'pyswmm' (Python Wrappers for SWMM). The repository has 34 stars, 28 forks, and was updated 10 days ago. It is associated with the tags 'python', 'stormwater', and 'hydrology-stormwater-analysis'. A green line graph icon is shown next to the repository name.

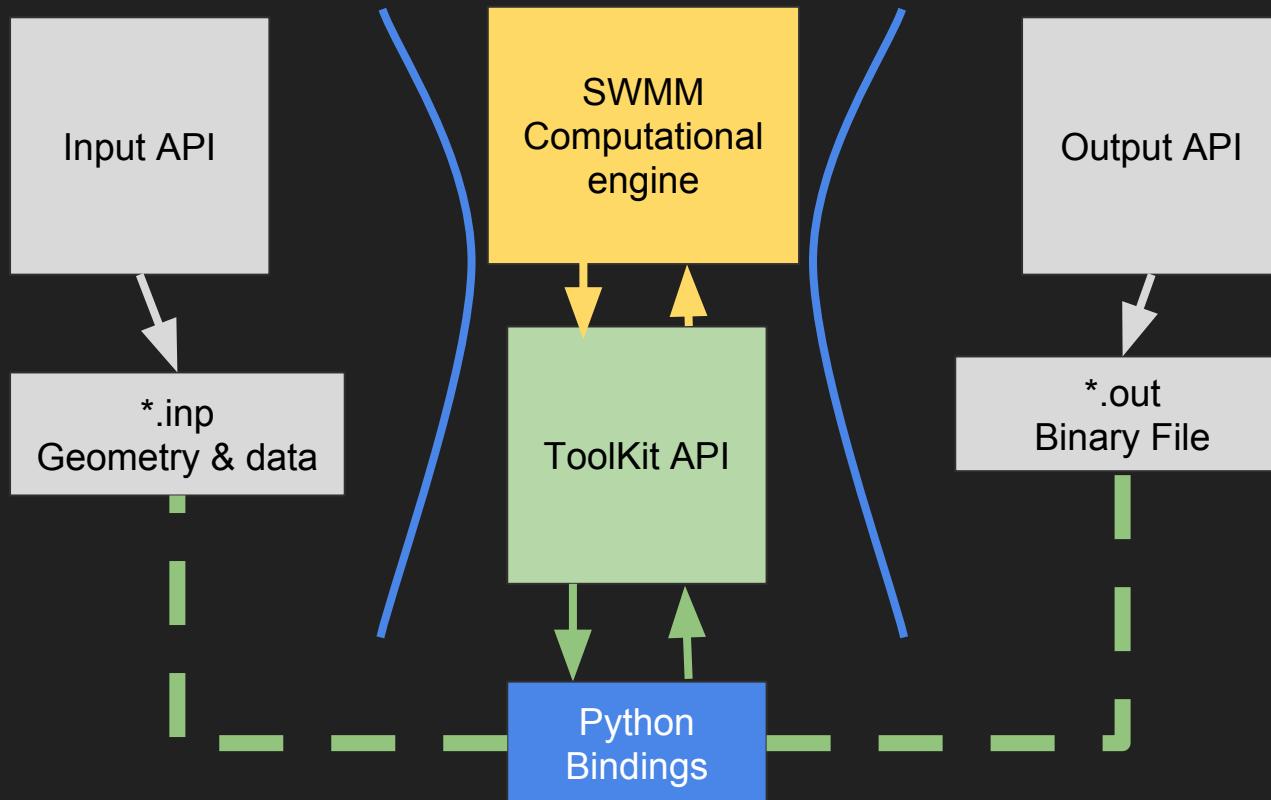
On the right side of the search results, there are two sidebar boxes: 'Top languages' (Python, C, Matlab, C++, CSS) and 'Most used topics' (hydraulics, stormwater, swmm).

At the very bottom, the source of the screenshot is cited as 'Source: <https://github.com/OpenWaterAnalytics/>'.

# SWMM API:

- Exposing the Data Model
- Observe Simulated Results During Run

# SWMM API: General Framework



# PySWMM API:

- Run Simulation
- Link Settings
- Control Rules in Python
- Set Node Inflows

# SWMM API: Run a simulation

```
>>> from pyswmm import Simulation  
>>>  
>>> with Simulation('./testmodel.inp') as sim:  
...     for step in sim:  
...         pass  
...     sim.report()
```

# SWMM API: Adjust Link Settings

```
>>> from pyswmm import Simulation, Links
>>>
>>> with Simulation('./testmodel.inp') as sim:
...     link_object = Links(sim)
...
...
...     #C1:C2 link instantiation
...     c1c2 = link_object["C1:C2"]
...     print(c1c2.flow_limit)
...     print(c1c2.is_conduit())
...
...
...     #Step through a simulation
...     for step in sim:
...         print(c1c2.flow)
...         if c1c2.flow > 10.0:
...             c1c2.target_setting = 0.5
...
...
...     sim.report()
```

# SWMM API: Adjust Link Settings

```
>>> from pyswmm import Simulation, Links  
>>>  
>>> with Simulation('./testmodel.inp') as sim:  
...     link_object = Links(sim)  
...  
...     #C1:C2 Link instantiation  
...     c1c2 = link_object["C1:C2"]  
...     print(c1c2.flow_limit)  
...     print(c1c2.is_conduit())  
...  
...     #Step through a simulation  
...     for step in sim:  
...         print(c1c2.flow)  
...         if c1c2.flow > 10.0:  
...             c1c2.target_setting = 0.5  
...  
...     sim.report()
```

Query information for every time step

Control based on varying conditions

# SWMM API: Build Custom Control Rules

```
>>> from pyswmm import Simulation, Links, Nodes  
  
>>>  
  
>>> def TestDepth(node, node2):  
>>>     if node > node2:  
>>>         return True  
>>>     else:  
>>>         return False  
  
>>>
```

```
>>> with Simulation('./testmodel.inp') as sim:  
...     link_object = Links(sim)  
  
...  
...     #C1:C2 link instantiation  
...     c1c2 = link_object["C1:C2"]  
  
...  
...     node_object = Nodes(sim)  
...     #J1 node instantiation  
...     j1 = node_object["J1"]  
...     #J2 node instantiation  
...     j2 = node_object["J2"]  
  
...  
...     #Step through a simulation  
...     for step in sim:  
...         if TestDepth(j1.depth, j2.depth):  
...             c1c2.target_setting = 0.5  
  
...  
...     sim.report()
```

# Expanded PySWMM API:

- No API (yet) to construct a network programmatically
- Needs to generate an input file “manually” or using the GUI.

# Expanded PySWMM API:

- No API (yet) to construct a network programmatically
- Needs to generate an input file “manually” or using the GUI.



Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

**ScienceDirect**

Procedia Engineering 119 (2015) 710 – 718

**Procedia  
Engineering**

[www.elsevier.com/locate/procedia](http://www.elsevier.com/locate/procedia)

13th Computer Control for Water Industry Conference, CCWI 2015

## OOPNET: An object-oriented EPANET in Python

D. Steffelbauer<sup>a\*</sup> and D. Fuchs-Hanusch<sup>a</sup>

<sup>a</sup>*Institute of Urban Water Management and Landscape Water Engineering  
Graz University of Technology, Stremayrgasse 10/I, A-8010 Graz, Austria*

### Abstract

Several attempts of the past aimed to convert EPANET into a bigger open-source project by rewriting EPANET in an object-oriented way. We introduce a Python based object-oriented EPANET (OOPNET) with the purpose to address water engineers that might be not so familiar with complex programming languages like C++ or Java. EPANET input files are translated into the object oriented structure of OOPNET and manipulated and simulated with EPANET’s command-line interface through Python. The replacement of EPANET by a hydraulic solver in Python is still ongoing and has the purpose to provide a solution completely written in one programming language.

© 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the Scientific Committee of CCWI 2015

**Keywords:** Water Distribution, Parallel Computing, Open Source Project, Simulation Software, Genetic Algorithms, Optimization

**Source:** <https://www.sciencedirect.com/science/article/pii/S1877705815025941>

# Example input file \*.inp

[CONDUITS]								
;; ;;Name	Inlet Node	Outlet Node	Length	Manning N	Inlet Offset	Outlet Offset	Init. Flow	Max. Flow
<hr/>								
C1	J1	J5	185.00	0.05	0	0	0	0
C2	J2	J11	526.00	0.016	0	4	0	0
C3	J3	J4	109.00	0.016	0	0	0	0
C4	J4	J5	133.00	0.05	0	0	0	0
C5	J5	J6	207.00	0.05	0	0	0	0
C6	J7	J6	140.00	0.05	0	0	0	0
C7	J6	J8	95.00	0.016	0	0	0	0
C8	J8	J9	166.00	0.05	0	0	0	0
C9	J9	J10	320.00	0.05	0	0	0	0
C10	J10	J11	145.00	0.05	0	0	0	0
C11	J11	SU1	89.00	0.016	0	1	0	0
C_out	J_out	02	100	0.01	0	0	0	0

# API Proof of Concept

```
# Create Model
model = Model(title='Example 3', description='Detention Pond')

# Create Timeseries (Notice dt_range was passed as a copy of the range, we could do this internally?)
ts_2y = model.create_timeseries(index=dt_range, values=raindata_2y5min, relative=True)
ts_5y = model.create_timeseries(index=dt_range, values=raindata_5y5min, relative=True)
ts_10y = model.create_timeseries(index=dt_range, values=raindata_10y5min, relative=True)

# Create Raingages
rg = model.create_raingage('RainGage', x=-148.485, y=1207.602, timeseries=ts_2y)

# Create Cross Sections/Shapes to be used with conduits
xs_pipe1 = Circular(diameter=2.25)
xs_pipe2 = Circular(diameter=3.5)
xs_pipe3 = Circular(diameter=4.75)
xs_channel1 = Trapezoidal(max_height=3, width=5, left_slope=5, right_slope=5)
xs_channel2 = Trapezoidal(max_height=1, width=0, left_slope=0.0001, right_slope=25)

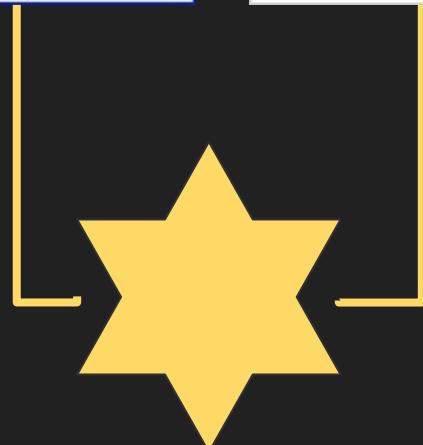
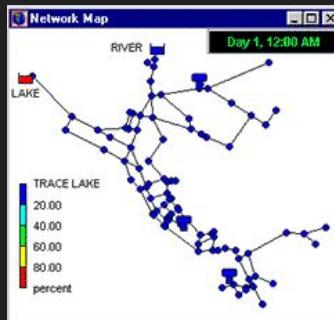
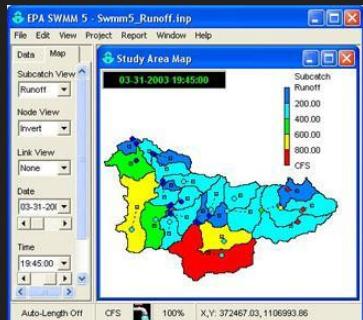
# Create Junctions/Nodes
j1 = model.create_junction('J1', x=648.532, y=1043.713, invert_elevation=4973)
j2 = model.create_junction('J2', x=648.532, y=1043.713, invert_elevation=4969)
j11 = model.create_junction('J3', x=648.532, y=1043.713, invert_elevation=4973)

# Create Conduits/Pipes
c1 = model.create_conduit('C1', from_node=j1, to_node=j2, length=185, tag='Swale')
c2 = model.create_conduit('C2', from_node=j2, to_node=j11, length=526, tag='Gutter')
```

# Why should you care about the API?

- Gives you a window to your model
- Optimization packages
- New Inflow algorithms
- Control Rules in Python
- Get more out of your model
- Watch simulated results while running
- *Something we haven't thought of yet*

# Roadmap and future work - Devs



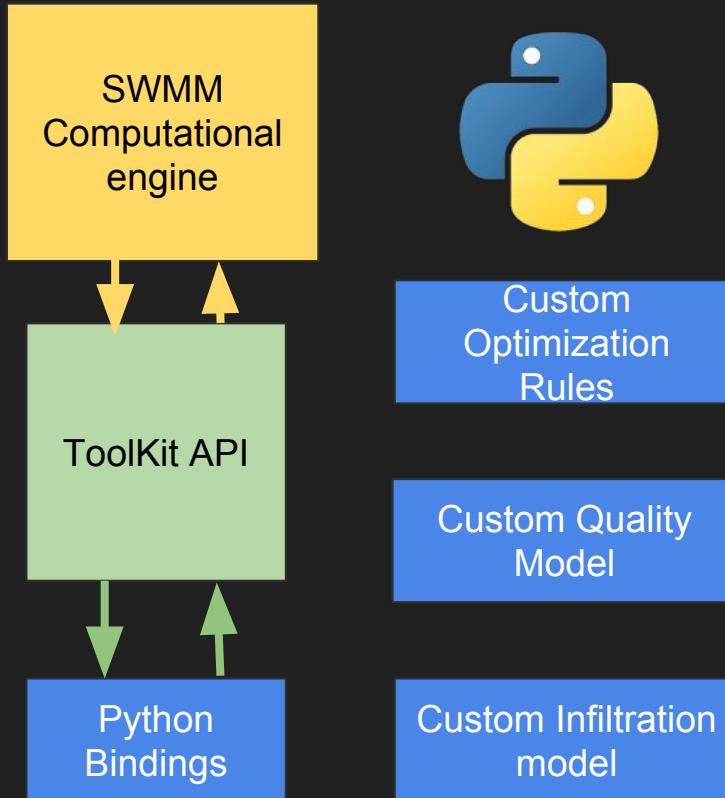
**Create a common core  
for EpaNet and SWMM**

**C -> C++**

# Roadmap and future work - Devs

- 1. Unit tests**
- 2. Create Conda Packages**
- 3. Continuous Integration**
- 4. Official cross platform support**
  - a. Linux**
  - b. Mac**

# Roadmap and future work - Academia



**Create a pythonic interface to allow so that new models can be created in Python and plugged to the numerical engine.**

# We are **NOT** hiring (yet!)

But if you...

- Like C?
- Like C.I.?
- Like C++?
- Like SWIG?
- Like Python?
- Like Bindings?
- Like Hydrology?
- Like Hydraulics?
- Like Unit Testing?
- Like 💩 Modelling?
- Like Open Source?
- Like Water Quality?

... let's talk!



# Questions & (hopefully) Answers

@goanpeca

goanpeca@gmail.com



Source: <http://www.revistalaocaloca.com/2017/04/el-poporo-quimbaya>