STORMWATER MODELING

Diving Into Open-Source SWMM (SWMM2D)

The next-generation SWMM computational engine

Open-Source SWMM Initiative | August 2026

Six Decades of Stormwater Infrastructure Design

1971 — SWMM I: Original EPA Storm Water Management Model

1975 — SWMM II: Major expansion; first widely distributed version

1981 — SWMM III: Major hydraulic and hydrologic expansion, including dynamic-wave routing

1988 — SWMM IV: Continued major expansion, including groundwater, RDII, and more sophisticated hydraulic capabilities

2004 — SWMM 5: Major rewrite with a new computational engine, GUI, and architecture; became the foundation of the modern SWMM ecosystem

2022 — SWMM 5.2: Major modern release with additional modeling capabilities and improvements

2025— SWMM 5.2.x: Most recent EPA generation

LEGACY

Six Decades Supporting Stormwater Infrastructure Planning and Design

Supports design of hundreds of billions in infrastructure

PCSWMM, InfoSWMM and XPSWMM are all built on the EPA SWMM core

Open and transparent, inspectable physics

Expertise is retiring; fewer engineers entering the field

Capital investment in urban drainage modernization is decreasing

More hydrologic and sensor data than ever — and more ways to analyze it

Climate change is weakening the assumption that the past predicts the future

Utilities are shifting from reactive maintenance to model-driven operations

CONTEXT

The Engine Must Evolve in the Open Source

What Modern Practice Can Look Like

Open APIs
Scriptable, reproducible workflows

Fast Engine
Modern high-speed solver

Data-Driven Analytics
Auto-calibration, uncertainty, assimilation, hybrids with data-driven models

Fine-Scale Physics
Hyper-accurate and finer scale process representation

Cloud Scale
Large-scale scenario analysis

Interoperability
Plays well with other tools and model

Multi-Benefit Design
Beyond one-dimensional problem solving

OUR VISION

Digital Public Water Infrastructure

- Communities should not be forced to make infrastructure decisions in the dark. The models, data, and engines behind public infrastructure determine where investments go, whose risks are recognized, and which futures are possible. The profession's foundational tools must therefore be:

Transparent

Inspectable physics and open validation, models that engineers, regulators, and the public can examine and defend.

Accessible

Available to every community, not only those with advanced software, monitoring networks, and specialized modeling teams.

Accountable

Rapid, defensible answers: where flooding may occur, who is exposed, and which interventions are worth scarce public dollars.

Open-Source SWMM Initiative

Community-Born

A group passionate about the continued maintenance and advancement of SWMM, prior to the EPA

Worked at the EPA Managing SWMM

Managed and advanced the software with openness and accessibility in mind

Long-Term Roadmap

Sustainable development and an open-source ecosystem of interoperable tools

Growing Community

Rigorous review, feedback, and acknowledgement of contributions

SUPPORTED BY

ASCE/EWRI • WEF Collection System Technical Practice Group

VISION

Unofficial open-source releases; work with EPA in the open-source towards official releases

A N O U N C E M E N T

We are no longer planning a single v5.3.0 release.

DUAL RELEASE

Mac, Linux, Windows

v5.3.0 + **v6.0.0**

Incremental · Near-term · Backward-compatible

Major architectural leap · New capabilities

Objectives We Set Out to Achieve

Computational Performance

Cache-friendly code, vectorization, OOP engine, Efficient IO

Hydraulic & Water Quality

New solvers, Dynamic Preissmann Slot, full ARD transport

Spatially Explicit Modeling

Finite volume overland and groundwater flow, GIS data model

Data-Driven Modeling

Python API & Ecosystem

Official bindings, MCP server, Gymnasium, real-time control

Rigorous Unit Testing, Validation, Access & Documentation

Multi-platform, live docs, expanded testing coverage

Drawn from the 2018 EWRI Visioning Summit, stakeholder workshops (2024–2026), and ongoing research partnerships.

HydroCouple Component-Based Interfaces

- Monolithic engines bloat as process formulations accrete
- Unwieldy to manage over time
  - Few people are experts in understanding the code
  - Hard for researchers to add new formulations
  - Errors in one part cascade to others
  - Difficult to validate and unit test
- Component-based models
  - Separation of concerns
  - More maintainable and testable
  - Plugins framework facilitating marketplace of components/tools

Component coupling at runtime

Rainfall–Runoff Component

Provides ‣ Runoff (m^3/s)

Types and units match

Hydraulic Routing Component

Expects ‣ Inflow (m^3/s)

Plugin Architecture

A new plugin architecture allows developers to build custom modules for a variety of needs — a precedent for a plugin marketplace and an ecosystem of tools that work seamlessly with SWMM.

Ecosystem Around Four Core Components

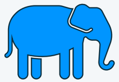
Engine & Python Bindings

Thread-safe, handle-based openswmm.engine.Solver (C++ core) with a full Python API — Nodes, Links, Subcatchments, Controls, Forcing, Inflows, Pollutants, and OutputReader.

Graphical User Interface

Desktop application for building, editing, running, and visualizing SWMM2D models — spatially explicit coupled overland and groundwater flow.

MCP Server

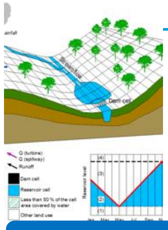


openswmm.mcp exposes the engine to agents over the Model Context Protocol for natural-language model interrogation and control.

Open-Source SWMM Gymnasium

Gymnasium-compatible reinforcement-learning framework for joint CIP design and real-time control, with composable rewards and multi-objective scoring.

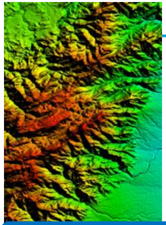
Under Development



IN PROGRESS

Flow Routing

- Better handling of flow transitions between open channel-pressurized conditions
- More stable and conservative routing schemes 1D explicit finite volume model as an alternative to the dynamic wave



IN PROGRESS

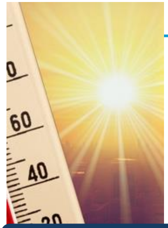
Spatially Explicit Modeling

- Finite-volume 2D flow coupled to 1D network
- Quasi-3d finite-volume subsurface flow
- Native GIS data model for city-scale meshes
- Runs on CPU and GPU

IN PROGRESS

Water Quality Transport

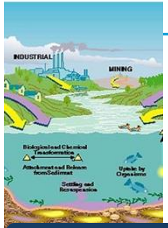
- Advection–reaction–dispersion in pipes, overland, and groundwater under one multispecies framework



PLANNED

Heat Transport

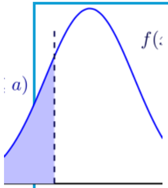
- Energy balance and transport on the ADE solver
- Groundwater baseflow exchange
- Layered stratification in detention basins



PLANNED

Sediment/Reactive Pollutant Transport

- Sediment as species: heteroaggregation, sorption, dissolution
- Fate engine with built-in uncertainty
- Morphodynamics to follow



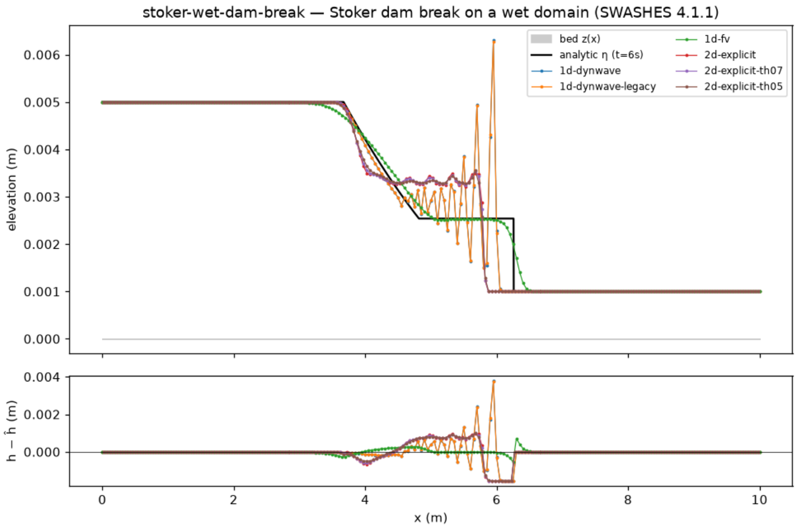
IN PROGRESS

Uncertainty Sidecar

- Topology-derived ensemble deviations at ~1000× Monte Carlo speed
- Registry: roughness, rainfall, inflow, coupling

Explicit 1D Finite Volume Alternative to Dynamic Wave Routing

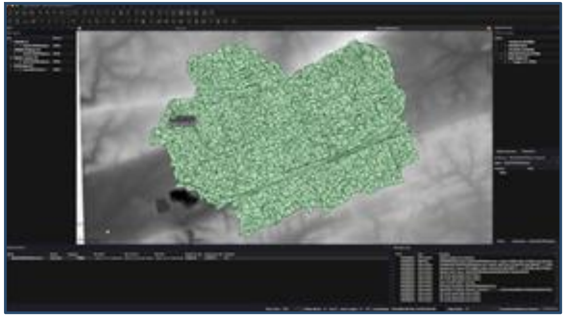
- Explicit conservative finite-volume solver for the 1D Saint-Venant equations — FLOW_ROUTING FV . Mass balance errors a thing of the past
- Local time stepping: stiff cells sub-step at their own rate instead of dragging the whole network down
- Order-of-magnitude accuracy gains on the analytic benchmarks, with machine-level mass conservation
- Performance on par with — and often faster than — DYNWAVE.



Demos: GUI, Engine & MCP Server

Desktop GUI

Build, edit, run, and visualize SWMM2D models — layers, basemaps, object browser, and live simulation status.



Reentrant Engine

Modern C++20 core running 1D/2D hydrology & hydraulics at city scale, from a domain-split C API.

MCP Server

An LLM agent iterating on models, scripts, and maps in natural language through the SWMM2D MCP project.

pip install openswmm

- Build SWMM models
- Iterate the simulation step-by-step or run to completion
- Read and write node, link, and subcatchment state mid-run
- Drive controls and forcing from Python for scripted or RL-based workflows
- Query results after the fact through OutputReader

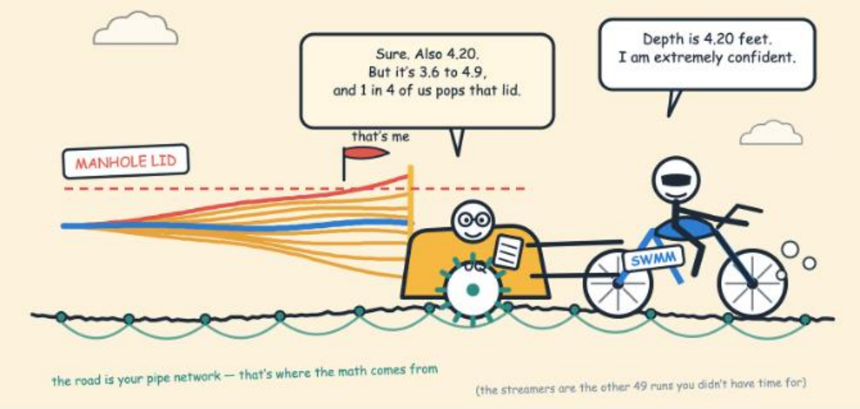
https://github.com/HydroCouple/openswmm.lab

```python
from openswmm.engine import Solver

with Solver("model.inp") as sim:
    for step in sim.run():
        j1 = sim.nodes["J1"]
        depth = j1.depth

    ts = sim.output_reader.get_time_series(
        "J1", "depth")
```

```python
from openswmm.engine import ModelBuilder, NodeType, LinkType, XSectShape

m = ModelBuilder()                                       # no .inp required

j1  = m.add_node("J1",  NodeType.JUNCTION)               # delineate the domain
out = m.add_node("OUT", NodeType.OUTFALL)
c1  = m.add_link("C1",  LinkType.CONDUIT)
m.set_link_nodes(c1, j1, out)

m.set_node_invert(j1, 100.0)                             # parameterize
m.set_node_max_depth(j1, 5.0)
m.set_node_invert(out, 99.0)
m.set_link_length(c1, 300.0)
m.set_link_roughness(c1, 0.013)
m.set_link_xsect(c1, XSectShape.CIRCULAR, 1.0)

m.validate()
m.finalize()
m.write("model.inp")                                     # dump for inspection, or…
solver = m.to_solver()                                   # …run it directly
```

Uncertainty Sidecar

A companion ensemble that rides alongside the real solver. SWMM computes exactly what it does for your run; the sidecar carries 50 perturbed members in parallel and reports a band around every node, every timestep.

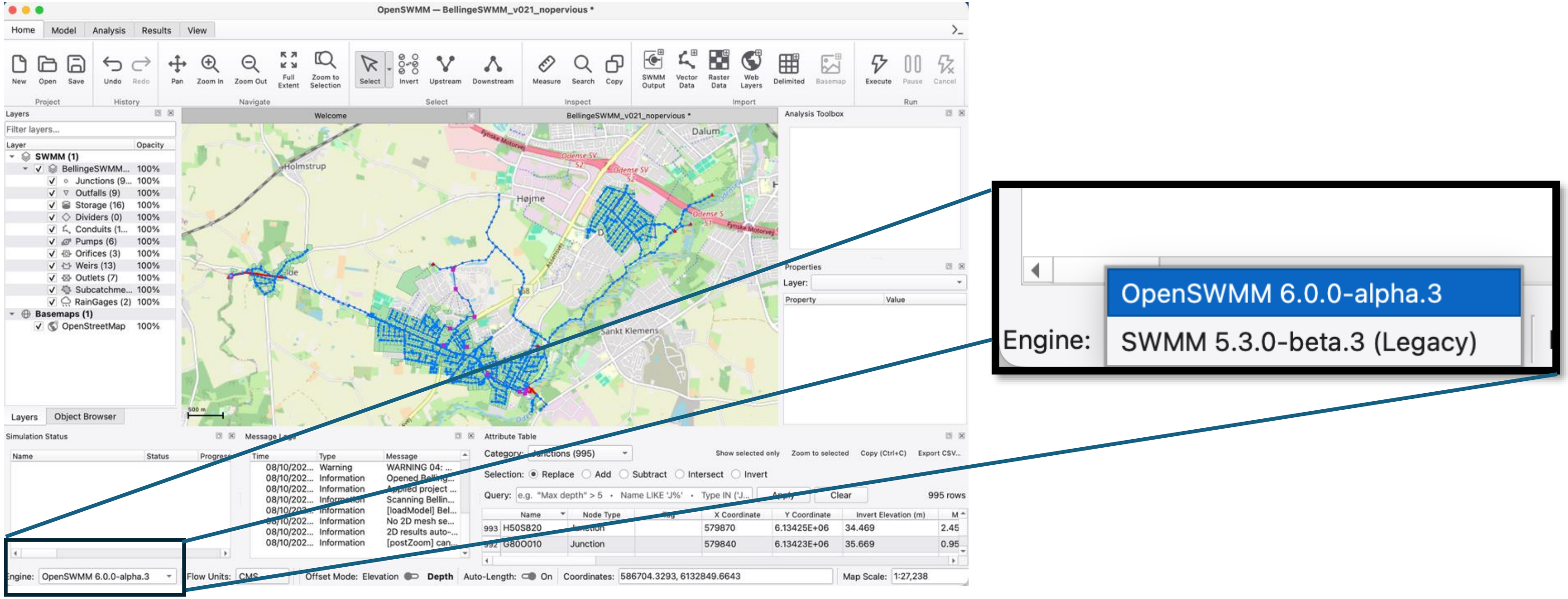


Network with 8,267 nodes using 50 members

~2% overhead.

Compatible with SWMM5

Hydraulic and process improvements are integrated to remain compatible with existing workflows. Check release notes and verification guidance for your specific use case.



Comparing EPA SWMM5 and SWMM6: Use Cases

	EPA SWMM 5.2.4	Open-Source SWMM6
Digital Twins	Run SWMM → exchange files/data → update twin	Embed SWMM directly in the twin
Flood Risk	Generate scenarios → batch-run SWMM → postprocess; No 2D	Programmatically orchestrate scenarios with 2D overland routing
Real-Time Control	External program → call SWMM → read results → repeat	Close the loop around the engine
Data-driven modeling	Treat SWMM largely as a black box	Use SWMM as a composable physics engine

Performance & Scalability

SWMM6 is designed for large networks and high-throughput workflows.

SWMM6 Engine

- Efficient core
- Modern memory
- Re-entrant

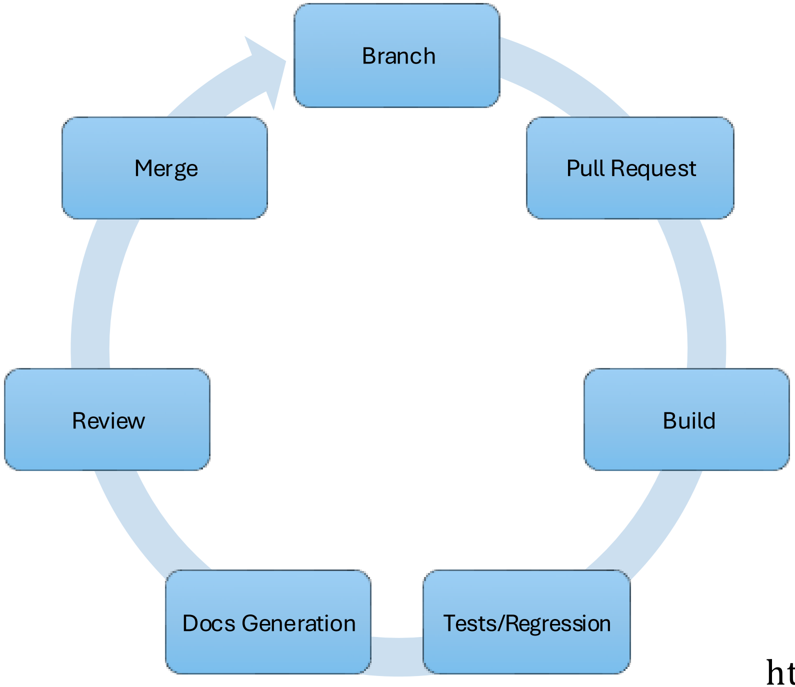
Many Runs

- Calibration
- RL/Gymnasium
- Uncertainty

Decisions

- Optimization
- Calibration
- Other data-driven modeling

Defensible Testing, Validation & Documentation



- Modern software platform
- Continuous integration and development
- Cross-platform builds
- GitHub Copilot reviews
- Docs with releases

https://www.hydrocouple.org/openswmm.engine/index.html

Community Testing and Contributing with Github

Community Testing

1. Real-world case
2. Minimal reproducible example
3. Stability • accuracy • performance
4. GitHub issue / structured feedback
5. Convert finding into a permanent test

Contributing

1. Issue / proposal
2. Branch
3. Implement + test
4. Pull request
5. CI + review
6. Merge

Major Formulation Changes

Discuss → 1 → Experiment → 2 → Analytical validation → 3 → Regression → 4 → Performance → 5 → Document → 6 → Integrate

Timeline & How to Get Involved

v5.3.0.beta
V6.0.0.alpha
Release
2026 Q3

v5.3.0 Release
V6.0.0.beta1
2027 Q1

v6.0.0.beta2
2027 Q2

v6.0.0
Release
2027 Q3

https://github.com/hydrocouple/openswmm.engine

GitHub

Watch, star & file issues on the SWMM repository. While contributions and pull requests not being accepted at this stage, we have a formal process we have created for accepting pull requests after this initial release.

Community

Volunteer as an alpha/beta tester. Post about your work on our open group on LinkedIn. Participate in discussions on Github.

Stakeholder Input

Share your v6.0.0 priorities. Workshops, surveys, and direct collaboration opportunities are available.

Ongoing engagement

Where to access the code for testing

develop

Default branch. All 6.0.0 alpha and beta pre-releases are cut from here.

main

Stable branch. Final releases, 5.3.0 and 6.0.0, are tagged from main.

Get the code

- Public repositories — clone, build, and run today
- Pre-built binaries for Mac, Linux, and Windows on the releases page
- Roadmap, issues, and discussions are open to all

Join the alpha program

Sign up for early release access and give structured feedback that shapes 6.0.0.

Thank you

Questions & Discussion

Join the alpha program

Sign up for early release access and give structured feedback that shapes 6.0.0.

Contact

Caleb Buahin, PhD
caleb.buahin@gmail.com

Corinne Wiesner-Friedman, PhD
cehwiesner@gmail.com