

Integration of SWMM and TELEMAC for 1D/2D coupled modeling

Hailiang Shen, Ph.D., P.Eng.

hailiang@chiwater.com

ICWMM 2026, Toronto, Feb 25th-26th

SWMM-TELEMAC integration objectives

- State of the field
 - EPA SWMM and TELEMAC are mature, well-validated open-source modeling engines
 - Both are supported by strong and active user communities
 - TELEMAC offers powerful numerical solvers but limited native graphical modeling tools
- Motivation
 - Extends SWMM to advanced processes (e.g., sediment transport, water quality, coastal modeling)
 - Improves accessibility through an integrated modeling platform
- Methodology
 - Simulate 1D flow in conveyance systems using EPA SWMM
 - Simulate 2D surface flooding using TELEMAC
 - Develop a robust coupling module supported by an integrated graphic user interface

SWMM engine

- Initial development ~1970
 - Original/classic SWMM written in Fortran
- SWMM 5 rewrite in C (2001-2004)
- Active development since 2004
- Official EPA SWMM engine builds (EXE & DLL)
 - Compiled on Windows using Microsoft Visual C++ (MSVC)
 - Can also be compiled on other operating systems

Note: non-official builds (e.g., on Linux) may produce slightly different numerical results compared with the official EPA SWMM DLL

TELEMAC engine

- 1970s-1980s (Research Phase)
 - Developed by Electricité de France (EDF), Fortran-based
 - Focus on 2D shallow-water modeling for rivers and floods
- 1990s (Operational Adoption)
 - Extended to sediment transport, water quality, estuaries
 - Widely used by European agencies and utilities
- 2010-2013 (Open-Source Transition)
 - Released under GPL
 - Governance transferred to TELEMAC-MASCARET Consortium
- 2013-2017 (HPC & Scalability)
 - MPI parallelization and domain decomposition
 - Enables very large-scale flood and coastal models
- 2017-now (Modernization)
 - Improve stability, documentation, and Python tools

TELEMAC main features

- **High-fidelity hydraulics** – solves 2D/3D shallow-water equations with strong scientific validation
- **Flexible unstructured mesh** – use triangular meshes to represent complex terrain and urban floodplains
- **High-performance computing** – built-in MPI parallelization for large, city-to-national-scale models
- **Multi-physics & coupling platform** – sediment, water quality, ice, waves
- **Open & customizable architecture** – open source, Fortran-based core with strong extensibility and automation tools

Now: SWMM-TELEMAC integration framework

- SWMM interface (1D network)
 - Junctions, storage units, and outfalls support dynamic lateral inflow or outflow
 - Receives surface exchange as external boundary conditions
- TELEMAC interface (2D domain)
 - Exchange handled through designated source points
 - Exchange flow updated using mass source/sink terms
- Hydraulic connection
 - Defined at TELEMAC sources, types can be orifice, weir, pump, outlet, outfall
 - Each connection governed by head-based discharge relationships, using the standard hydraulic equations in SWMM

Integration code updates in Fortran

- 35 Fortran files added/updated
 - Implemented SWMM-TELEMAC exchange logic
 - Added orifice, weir, pump, outlet modules
 - Integrated flow exchange into TELEMAC time loop
- Three companion Python build scripts
 - Automated compilation with gfortran
 - Ensured reproducible engine builds
- AI-assisted development
 - Accelerated understanding of legacy code
 - Reduced development and debugging time

api.cmdf
 telemac2d
 chk_swmm.f
 close_cpl_swmm.f
 cpl_swmm.f
 deall_telemac2d.f
 declarations_telemac2d.f
 exchange_swmm.f
 homere_telemac2d.F
 init_cpl_swmm.f
 interface_swmm_dll.f90
 interface_telemac2d.f
 point_telemac2d.f
 radar_rainfall.f90
 read_swmm_exchange.f90
 runoff_greenampt.f
 runoff_horton.f
 runoff_scs_cn.f
 swmm_calc_q.f
 swmm_curve.f
 swmm_dsce_step.f
 swmm_iface.f
 swmm_iface2.f
 swmm_orifice.f

38 files changed +3105 -74 lines changed

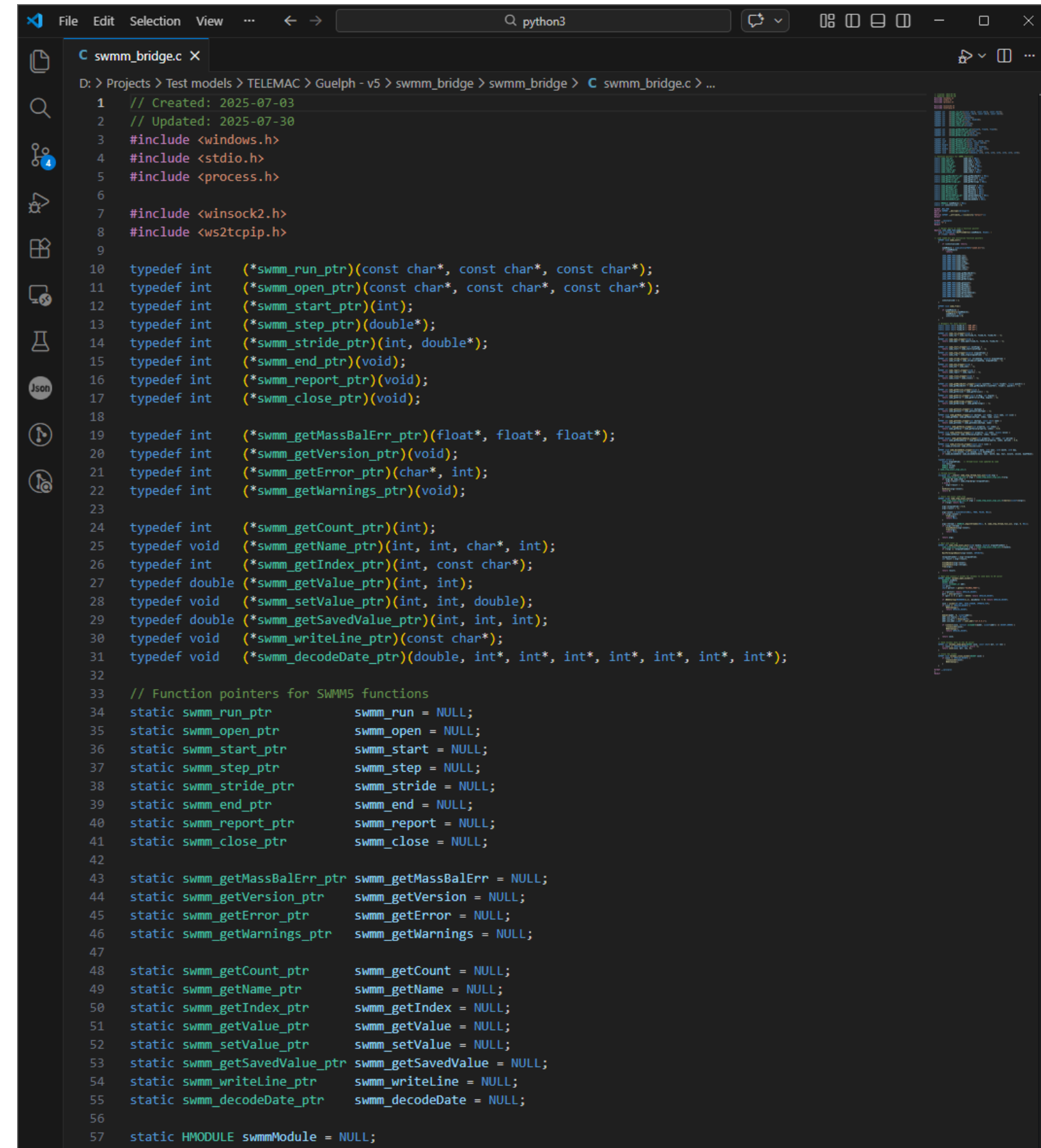
.gitignore
 ... @@ -0,0 +1 @@
 1 + *.pyc

scripts/python3/compilation/compil_tools.py

... @@ -276,6 +276,15 @@ def create_
 276 276 if lib == lname:
 277 277 obj_files = obj_file
 278 278
 279 + # Add swmm_bridge.o for tele
 280 + # Always take swmm_bridge.o
 281 + if (lmdul == 'telemac2d' or
 282 + telemac2d_obj_dir = path
 283 + swmm_obj = path.join(te
 284 + if not path.exists(swmm_
 285 + raise TelemacExcepti
 286 + obj_files = obj_files +
 287 +
 279 288 # ~ is linkage necessary ?
 280 289 if path.exists(lib_file):
 281 290 if 0 < cfg['COMPILER']
 ... @@ -412,6 +412,13 @@ def create_
 412 412 raise TelemacExc
 413 422 obj_files = obj_file
 414 423
 424 + # Add swmm_bridge.o for tele
 425 + if enprog == 'telemac2d':

Key technical highlights

- Direct integration with the official EPA SWMM engine
 - Native access to the 64-bit DLL (no recompiling)
- Custom C bridge
 - Handles calling conventions and memory safety
- Fortran-to-SWMM bridging
 - TELEMAC core calls SWMM DLL through the bridge
- Compatible with user-defined Fortran subroutines



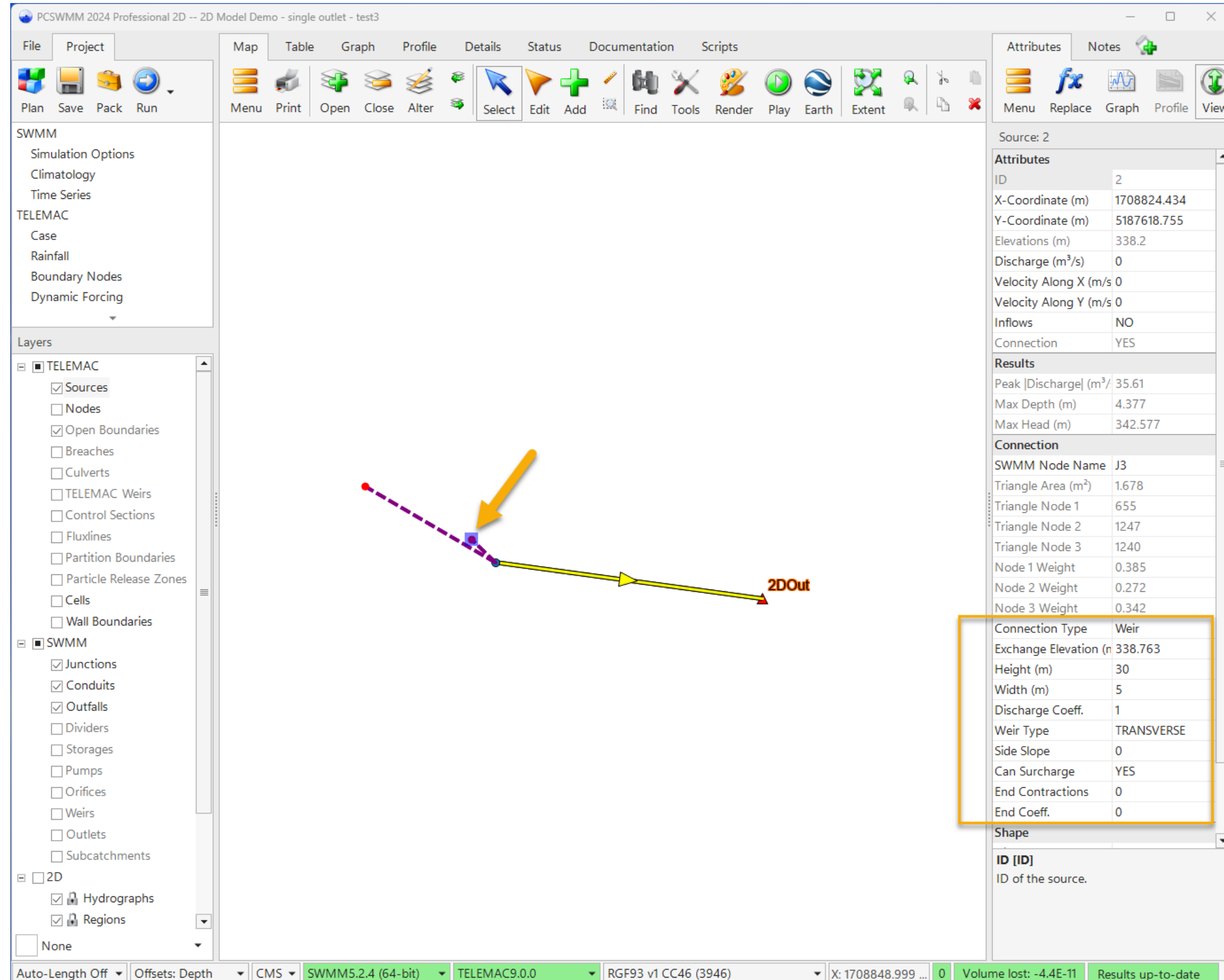
```

1 // Created: 2025-07-03
2 // Updated: 2025-07-30
3 #include <windows.h>
4 #include <stdio.h>
5 #include <process.h>
6
7 #include <winsock2.h>
8 #include <ws2tcpip.h>
9
10 typedef int (*swmm_run_ptr)(const char*, const char*, const char*);
11 typedef int (*swmm_open_ptr)(const char*, const char*, const char*);
12 typedef int (*swmm_start_ptr)(int);
13 typedef int (*swmm_step_ptr)(double*);
14 typedef int (*swmm_stride_ptr)(int, double*);
15 typedef int (*swmm_end_ptr)(void);
16 typedef int (*swmm_report_ptr)(void);
17 typedef int (*swmm_close_ptr)(void);
18
19 typedef int (*swmm_getMassBalErr_ptr)(float*, float*, float*);
20 typedef int (*swmm_getVersion_ptr)(void);
21 typedef int (*swmm_getError_ptr)(char*, int);
22 typedef int (*swmm_getWarnings_ptr)(void);
23
24 typedef int (*swmm_getCount_ptr)(int);
25 typedef void (*swmm_getName_ptr)(int, int, char*, int);
26 typedef int (*swmm_getIndex_ptr)(int, const char*);
27 typedef double (*swmm_getValue_ptr)(int, int);
28 typedef void (*swmm_setValue_ptr)(int, int, double);
29 typedef double (*swmm_getSavedValue_ptr)(int, int, int);
30 typedef void (*swmm_writeln_ptr)(const char*);
31 typedef void (*swmm_decodeDate_ptr)(double, int*, int*, int*, int*, int*, int*, int*);
32
33 // Function pointers for SWMM5 functions
34 static swmm_run_ptr swmm_run = NULL;
35 static swmm_open_ptr swmm_open = NULL;
36 static swmm_start_ptr swmm_start = NULL;
37 static swmm_step_ptr swmm_step = NULL;
38 static swmm_stride_ptr swmm_stride = NULL;
39 static swmm_end_ptr swmm_end = NULL;
40 static swmm_report_ptr swmm_report = NULL;
41 static swmm_close_ptr swmm_close = NULL;
42
43 static swmm_getMassBalErr_ptr swmm_getMassBalErr = NULL;
44 static swmm_getVersion_ptr swmm_getVersion = NULL;
45 static swmm_getError_ptr swmm_getError = NULL;
46 static swmm_getWarnings_ptr swmm_getWarnings = NULL;
47
48 static swmm_getCount_ptr swmm_getCount = NULL;
49 static swmm_getName_ptr swmm_getName = NULL;
50 static swmm_getIndex_ptr swmm_getIndex = NULL;
51 static swmm_getValue_ptr swmm_getValue = NULL;
52 static swmm_setValue_ptr swmm_setValue = NULL;
53 static swmm_getSavedValue_ptr swmm_getSavedValue = NULL;
54 static swmm_writeln_ptr swmm_writeln = NULL;
55 static swmm_decodeDate_ptr swmm_decodeDate = NULL;
56
57 static HMODULE swmmModule = NULL;

```

1D-2D interface configuration

- Defined at TELEMAC source points
- Multiple connection types
 - Weir, orifice, pump, outlet, outfall, and runoff
 - Governed by physically based discharge equations consistent with the SWMM code
- Flexible network-to-surface mapping
 - One SWMM node can connect to multiple TELEMAC sources



Attributes Notes

Menu Replace Graph Profile View

Source: 2

Attributes	
ID	2
X-Coordinate (m)	1708824.434
Y-Coordinate (m)	5187618.755
Elevations (m)	338.2
Discharge (m ³ /s)	0
Velocity Along X (m/s)	0
Velocity Along Y (m/s)	0
Inflows	NO
Connection	YES

Results	
Peak Discharge (m ³ /s)	35.61
Max Depth (m)	4.377
Max Head (m)	342.577

Connection	
SWMM Node Name	J3
Triangle Area (m ²)	1.678
Triangle Node 1	655
Triangle Node 2	1247
Triangle Node 3	1240
Node 1 Weight	0.385
Node 2 Weight	0.272
Node 3 Weight	0.342

Connection Type	Weir
Exchange Elevation (m)	338.763
Height (m)	30
Width (m)	5
Discharge Coeff.	1
Weir Type	TRANSVERSE
Side Slope	0
Can Surcharge	YES
End Contractions	0
End Coeff.	0

Shape

ID [ID]
ID of the source.

Auto-Length Off Offsets: Depth CMS SWMM5.2.4 (64-bit) TELEMAC9.0.0 RGF93 v1 CC46 (3946) X: 1708848.999 0 Volume lost: -4.4E-11 Results up-to-date

Orifice

- Represents inlet openings
 - Street inlets, curb openings, manhole openings
- Automatic transition between free flow and submerged flow
- Bidirectional exchange based on head difference between SWMM node and TELEMAC source
- Required inputs
 - Opening size, exchange elevation, discharge coefficient

Connection Type	Orifice	▼
Exchange Elevation (m)	338.763	
Orifice Type	Side	
Cross-Section	Circular	
Height (m)	30	
Width (m)	5	
Discharge Coeff.	1	

Weir

- Represents overflow structures
 - Spillways, curb overflows, crest-controlled openings
- Supported types
 - Transverse, side-flow, V-notch, and trapezoidal
- Handles surcharge conditions
 - Automatically transitions to orifice behavior when submerged
- Required inputs
 - Dimensions, exchange elevation, and coefficients

Connection Type	Weir
Exchange Elevation (m)	338.763
Height (m)	30
Width (m)	5
Discharge Coeff.	1
Weir Type	TRANSVERSE
Side Slope	0
Can Surcharge	YES
End Contractions	0
End Coeff.	0

Pump

- Represents active flow control
 - Transfers water between 2D surface and 1D network
- Configurable direction
 - To TELEMAC source or to SWMM node
- On/off control
 - Startup and shutdown depths
- Required inputs
 - Pump curve
 - Operational depths

Connection Type	Pump
Curve Name	*
Pump Direction	To TELEMAC
Startup Depth (m)	0
Shutoff Depth (m)	0

Outlet

- Represents predefined discharge control
 - Models regulated outfalls and controlled releases
- Rating-curve based flow
 - Supports bidirectional exchange
- Required inputs
 - Exchange elevation
 - Name of rating curve (tabular head vs. discharge)

Connection Type	Outlet
Exchange Elevation (m)	338.763
Curve Name	*

Outfall

- Uses TELEMAC source water level as SWMM outfall level boundary
 - SWMM node must be an outfall type
- Transfers SWMM outfall total inflow to TELEMAC source
 - Flow added to or removed from TELEMAC source is determined by the outfall's upstream link flow direction
- No hydraulic equation applied

Connection Type	Outfall
Upstream Link Name	*
Exchange Elevation (m)	338.763

Runoff

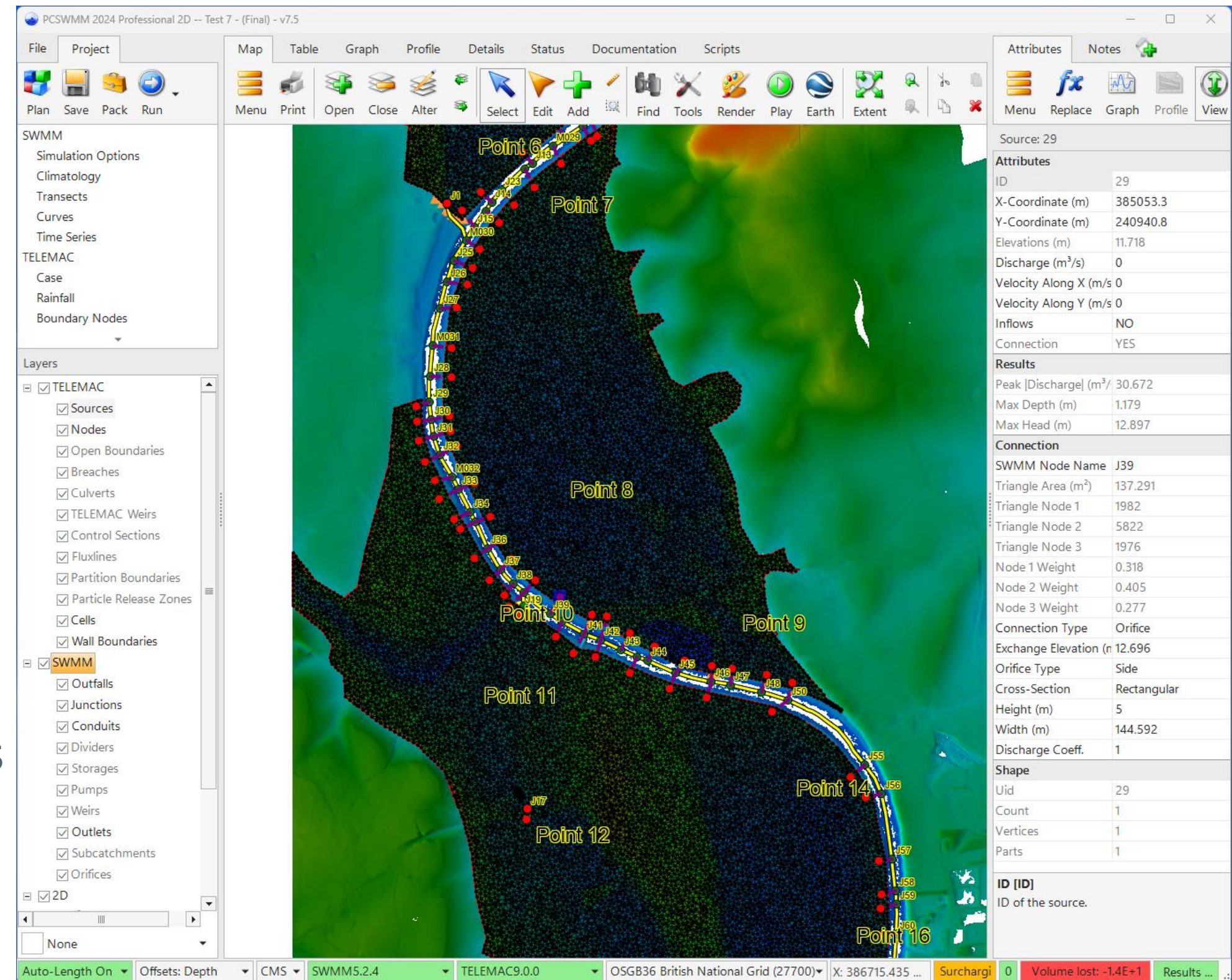
- Direct runoff transfer
 - Works best with SWMM storages and outfalls
- No hydraulic structure required
- One-to-one mass exchange
 - SWMM node total inflow added to TELEMAC source
 - Same flow removed from SWMM node

Connection Type

Runoff

Research tools

- Integration code development
 - Implemented and maintained in Visual Studio Code
- PCSWMM visualization
 - Support TELEMAC-only and integrated projects
- New project type: TELEMAC
- Flexible integrated project creation
 - From TELEMAC models
 - Or extended from existing SWMM models
 - Conversion from existing SWMM1D/2D models



File Map Table Graph Scripts

- Save
- Save As...
- Open
- Close

- Manage

- New


- Import
- Print
- Save & Send
- Layout
- Help


- Alt. Runoff
- Flood Analysis
- Events Analysis
- Erosion
- 2D
- SWMM
- Real-Time
- Reporting


- Defaults
- Preferences
- Exit


New Project


Welcome to PCSWMM 2024 Professional 2D. Please choose the type of project you would like to create:

-  **SWMM5 Project**
Create an EPA SWMM5 project for stormwater, wastewater or watershed modeling.

-  **EPANET2 Project**
Create an EPANET2 project for water distribution modeling.

-  **RAP Project**
Create a Radar Acquisition and Processing (RAP) project for downloading, calibrating, analyzing and preparing radar rainfall for use in SWMM5 and other purposes.

-  **Time Series Project**
Create a time series project for managing, editing, and/or analyzing any type of time series.

-  **TELEMAC2D Project**
Create a TELEMAC2D project for hydrodynamic modeling.



Integration source code

- Currently maintained in a private repository
- Will be made available to the community after testing is completed



Computational Hydraulics International

