

EPANET-MATLAB Toolkit: An Open-Source Software for Interfacing EPANET with MATLAB

Demetrios G. Eliades¹, Marios Kyriakou¹, Stelios Vrachimis¹, and Marios M. Polycarpou¹

¹KIOS Research Center for Intelligent Systems and Networks, Dept. of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus

Correspondence to: Demetrios Eliades (eldemet@ucy.ac.cy)

Abstract. In this work we introduce the EPANET-MATLAB Toolkit, an open-source software for interfacing a drinking water distribution system simulation library, EPANET, with the MATLAB technical computing language. The basic functionalities of the Toolkit are presented, and in the case-study a contamination simulator based on EPANET and EPANET-MSX is described and demonstrated on a benchmark network.

5 1 Introduction

In 1994, the US Environmental Protection Agency (EPA) released EPANET (version 1.1), an open source software developed by Lew Rossman, for modeling water distribution systems hydraulic and quality dynamics. EPANET was designed as a research tool to better understand the dynamics of drinking water constituents, taking into account bulk flow and pipe wall reactions (Rossman, 1994). EPANET considers a geometric representation of the pipe network, along with a set of initial conditions (e.g. water levels in tanks), rules of how the system is operated, and uses this information to compute flows, pressures and water quality (e.g. disinfection concentrations and water age) throughout the network, for a certain period of time. EPANET was originally developed in the C programming language and simulations can be executed through a stand-alone compiled application using a text file which contained the model description. EPANET utilizes the “gradient algorithm” for solving the hydraulic state-estimation at each time step (Todini and Pilati, 1987). For water quality the Finite Volume Method was originally utilized (Rossman et al., 1993), however, a Lagrangian approach (Rossman and Boulos, 1996) was adopted in the following release of EPANET (version 2.0). This new version allowed the dynamic linking of EPANET with external software through its shared object library. In 2015, the *Open Source EPANET Initiative* was established, comprised of various academic, industrial and other stakeholders, to manage the further development of EPANET. An updated version of EPANET (version 2.1)¹ was released 2016, and the next major release (Rossman, 2011) is currently under development.

EPANET has been established as the *de facto* standard tool for both the industry and academia: for water operators to simulate “what-if” scenarios, for researchers to evaluate novel algorithms for a variety of research challenges using realistic water distribution network benchmarks, and for the industry to create new products and services while benefiting from EPANET’s public-domain software license. During the past 20 years, EPANET was extensively used as a tool to facilitate

¹<https://github.com/OpenWaterAnalytics/EPANET>

research in topics such as network design optimization (Eusuff and Lansey, 2003; Savic and Walters, 1997), operational optimization (van Zyl et al., 2004) and sensor placement (Ostfeld et al., 2008).

A number of extensions have been released by the water community to expand EPANET's capabilities. For instance, as EPANET (version 2.0) was not able to simulate the interaction of multiple chemical agents, the EPANET-MSX extension was developed which allowed the simulation of the reaction and transport dynamics of multiple physical/chemical/biological parameters within a distribution network (Shang et al., 2008). Another example is EPANET-BAM, which allows incomplete mixing in pipe junctions (Ho and Khalsa, 2009), as well as EPANET-PDX for pressure-driven hydraulic state estimation (Siew and Tanyimboh, 2012).

From a software engineering point-of-view, EPANET has been used within procedural programs through a series of direct calls to its library. This requires the user to be aware of all the different functions offered by EPANET, as well as the sequence of function calls in order to successfully implement a simulation cycle. Moreover, there was not a common EPANET data structure in order to share data between different function modules and applications. These challenges can be addressed by adopting an Object-Oriented Programming approach. Recently, effort was given in creating software for using EPANET through Object-Oriented Programming interfaces, in different programming languages, such as R (Eck, 2016) and Python (Steffelbauer and Fuchs-Hanusch, 2015). A significant effort in utilizing Object-Oriented Programming to expand EPANET's capabilities was by van Zyl et al. (2003), who introduced OOTEN. OOTEN is comprised of different classes with associated methods (for instance the Class which describes water pipes provides functions to return pipe parameters, such as the diameter and length).

The contribution of this work is the introduction and demonstration of operation of the EPANET-MATLAB Toolkit² (version 2.1), an open-source software released under the European Union Public License (EUPL), developed at the *KIOS Research Center for Intelligent Systems and Networks* of the *University of Cyprus*. The Toolkit interfaces EPANET with MATLAB®, a widely-used programming environment and allows the user to have access to all the network information through a data structure, to execute direct calls to the EPANET library, to modify and to create EPANET networks, to run multi-species simulations through EPANET-MSX, as well as to visualize the network.

This paper is organized as follows: In Section 2, the EPANET-MATLAB Toolkit is introduced, and in Section 3, a case study is presented illustrating the use of the Toolkit. Section 4 concludes the paper and discusses future work.

2 The EPANET-MATLAB Toolkit

2.1 Interfacing with EPANET in MATLAB

EPANET can be used in two ways: 1) as a standalone executable software, or 2) as a shared object library. As a standalone executable software, EPANET can be called through a standard shell (e.g. *Command Line* in Windows). As a shared object, e.g. Dynamic Link Library (DLL) for Windows, EPANET can be called through a programming interface by external software

²<https://github.com/OpenWaterAnalytics/EPANET-Matlab-Toolkit>

written in different programming languages (such as C/C++, Python, MATLAB and Visual Basic). The external software can make calls to specific EPANET functions which modify system parameters, the time series and the simulation configuration.

Programming languages such as MATLAB® are used by researchers to design and evaluate new methodologies and tools for analyzing water distribution networks. MATLAB is a high-level programming environment used for data processing and analysis. It allows the development of applications in different platforms, and has build-in a large number of sophisticated applications for optimization, control, signal processing and others. MATLAB is also able to connect to external software libraries, which allows researchers to use tools and simulators developed originally in a different language, such as C or C++. There are three methods of interfacing EPANET with MATLAB:

1. The first method is to make direct calls to the EPANET library, through the build-in function of the programming tool, which requires using MATLAB's build-in methods for loading and calling library functions (i.e. using the *loadlibrary* and *calllib* functions).
2. The second method is to use “wrappers”; MATLAB methods which follow similar naming conventions as the EPANET functions, that handle the communication with the library internally³. This is a higher-level of interfacing with the library, however, it requires the user to design custom data structures. For each EPANET function, a corresponding MATLAB function is required, and new algorithms need to be designed using those functions.
3. The third method is to use an Object-Oriented approach, by defining a MATLAB Class, which provides a standardized way to handle the network structure, to call all functions as well as procedures using multiple functions, to simulate and in general to perform different types of analysis in the network, through the corresponding object. The EPANET-MATLAB Toolkit utilizes this approach.

2.2 How to use the EPANET-MATLAB Toolkit

The Toolkit is based on a MATLAB Class, **epanet**, which is composed of the following elements: 1) properties of the input network model, 2) static properties, 3) public methods (functions) that the user can use directly in MATLAB, 4) local (internal) functions the Toolkit uses to make direct calls to EPANET/EPANET-MSX. An **epanet** object is a specific instance of the **epanet** class. To create a new **epanet** object in MATLAB for a specific network (e.g. 'Net2.inp'), the following command is used⁴:

```
G = epanet('Net2.inp') % Load network and use the EPANET library
```

The element G is an object which can be defined mathematically as the set comprised of the network topology, structural parameters and functions. This **epanet** object can be shared between different MATLAB functions.

³<https://github.com/OpenWaterAnalytics/epanet-matlab>

⁴The code in this section is provided at <https://github.com/eldemet/ccwi2016>.

When the object is constructed, the Toolkit reads the input file and populates more than 300 object parameters (including pipe diameters and node elevations). The Toolkit can update these parameters when there is a change in the network model. Examples on how to retrieve parameter values are provided below:

```
5 properties(G) % Lists all available parameters
diameters = G.LinkDiameter % Link diameters from parameters
elevations = G.NodeElevations % Node elevations from parameters
```

After the construction of the object, it is possible to call the Toolkit functions. The Toolkit provides an extensive set of methods which allow the user to retrieve data and to simulate hydraulic and quality dynamics using the EPANET libraries. Some examples are provided below:

```
methods(G) % Lists all available methods
elevations = G.getNodeElevations([2 5]) % Node elevations for Nodes 2 & 5
15 diameters = G.getLinkDiameter % Link diameters from library
diameters(2)=18 % Change Link 2 diameter from 14 to 18
G.setLinkDiameter(diameters) % Set new link diameter
G.getLinkDiameter(2) % Confirms that Link 2 diameter is 18
G.plot % Plots the network in a MATLAB figure
20 A = G.getConnectivityMatrix % Construct connectivity matrix
func_list = G.getENfunctionsImpemented % EPANET functions implemented
```

To simulate the system dynamics, such as flows/pressures and water quality, various methods have been implemented to solve and retrieve the data: 1) solve using the EPANET's shared object library and get the desired results from memory (step-wise mode); 2) solve using the EPANET's shared object library and create a Binary output file, which is then read to retrieve all the results (batch mode); solve using the EPANET's executable and create a Binary output file, which is then read to retrieve all the results (batch mode). The hydraulic and quality dynamics can be computed using the following instructions:

```
H = G.getComputedHydraulicTimeSeries % Solve hydraulics in library
30 Q = G.getComputedQualityTimeSeries % Solve quality dynamics in library
B = G.getBinComputedAllParameters % Solve in library, create Binary file
```

To use EPANET-MSX, the user must first load an EPANET network, and then load an MSX file which describes the chemical dynamics. The Toolkit can be used to 'get/set' different MSX variables, to solve the multi-species dynamics, to plot concentrations and to write new MSX files. An example is provided below:

```
G.loadMSXFile('net2-cl2.msx') % Load MSX file with reactions
Q_msx = G.getMSXComputedQualityNode % Compute water quality using MSX
40 G.plotMSXSpeciesNodeConcentration(3,1) % Plot MSX species in MATLAB
```

Finally, to unload the libraries from MATLAB memory, the user should make the following calls:

```
G.unloadMSX % Unload EPANET-MSX library
5 G.unload % Unload EPANET library
```

3 Case Study

In this section, we illustrate through a case study some of the Toolkit's key features. In specific, we consider the case where we want to model Arsenite contamination events of various magnitudes occurring at a single location within a water distribution system. Here we consider the case of creating multiple Arsenite contamination scenarios, affecting a drinking water distribu-
10 tion system which uses Chlorine for disinfection. As Chlorine reacts with Arsenite, its concentration is reduced; details on the reaction dynamics can be found in (Umberg et al., 2008; Dodd et al., 2006). For this case study, we consider the benchmark 'Network 1' from the Battle of the Water Sensor Placement (Ostfeld et al., 2008; Eliades and Polycarpou, 2010), and further-
more we assume that chlorine sensors have been installed at 'optimal' locations, at the nodes '17', '83', '122', '31' and '45'. The network's demands and roughness coefficients are partially known.

15 Typically when implementing such as software module using low-level calls to the library, a large number of commands need to be written in order to achieve specific results, such as extracting the pipe roughness coefficients or specifying a new demand pattern. Through the use of the EPANET-MATLAB Toolkit, however, a large part of the repetitive code is already included in the Toolkit functions, and can be used directly. Moreover, the use of the Toolkit facilitates the creation of modules which can be integrated in other software. In addition to that, it is possible to write the software in such as way so that it
20 supports distributed computing, which is useful when developing cloud services.

Below, we provide a template solution on how to design an Arsenite contamination simulator based on EPANET and EPANET-MSX using the EPANET-MATLAB Toolkit. To reproduce the case study, the complete source code is provided at <https://github.com/eldemet/ccwi2016>.

The EPANET Input and MSX files are loaded as follows, constructing the **epanet** object *G*:

```
25 G = epanet('BWSN_Network_1.inp'); % Load EPANET Input file
G.loadMSXFile('Arsenite.msx'); % Load MSX file
```

The locations of the chlorine sensors are declared as:

```
30 sensor_id = {'JUNCTION-17', 'JUNCTION-83', 'JUNCTION-122', 'JUNCTION-31', 'JUNCTION-45'};
sensor_index = G.getNodeIndex(sensor_id);
```

The duration of the case-study is set to 5 days. The network parameters (demand patterns and roughness coefficients) are
35 retrieved, to be used for randomizing the hydraulics parameters.

```

t_d = 5; % days
G.setTimeSimulationDuration(t_d*24*60*60); % Set simulation duration
demand_pattern = G.getPattern;
roughness_coeff = G.getLinkRoughnessCoeff;
5 node_id = G.getNodeNameID;

```

A scenario matrix is constructed, which is comprised of all the contamination injection parameters (location, magnitude of concentration, start-time, duration). In addition, the uncertainty in demands and roughness coefficients is defined.

```

10 Ns = 100; % Number of scenarios to simulate
u_p = 0.20; u_r = 0.20; % pattern/roughness uncertainty
max_inj_conc = 2.0; %maximum Arsenic source concentration
inj_start_time = 2*48; % after day 2 (Dt = 30min)
inj_duration = 24; % maximum duration of 12 hours
15 inj_sc=[randi(G.NodeCount,Ns,1), max_inj_conc*rand(Ns,1), randi(48,Ns,1)+inj_start_time,
randi(inj_duration,Ns,1)]; % Scenarios

```

The main part of the algorithm, is composed of the algorithms for adding uncertainty to the demand patterns and roughness coefficients, as well as for adding contaminant sources and solving the quality dynamics.

```

20 for i = 1:Ns
G.setPatternMatrix(add_unc(demand_pattern, r_p)); % Randomize patterns
G.setLinkRoughnessCoeff(add_unc(roughness_coeff, r_r)); % roughness
G.setMSXSources(node_id(inj_sc(i,1)), 'AsIII', 'Setpoint', inj_sc(i,2), 'AS3PAT'); %
25 Specify Arsenite injection source
G.setMSXPattern('AS3PAT',create_pat(t_d, inj_sc)); % Injection pattern
Q{i} = G.getMSXComputedQualityNode(sensor_index); % Solve dynamics
G.setMSXSources(node_id(inj_sc(i,1)), 'AsIII', 'Setpoint', 0, 'AS3PAT'); % Reset
injection source
30 end

```

The results of this case study are depicted in Fig. 1, in which the estimated chlorine concentrations for 100 scenarios are overlaid on the same graph. This illustrates the variability of chlorine concentrations during normal operation; the drops in concentration correspond to contamination events, after the second day.

35 The collection of these time-series can be used as a benchmark for evaluating the effectiveness of various Contamination Event Diagnosis algorithms *in silico*, as in (Eliades et al., 2015).

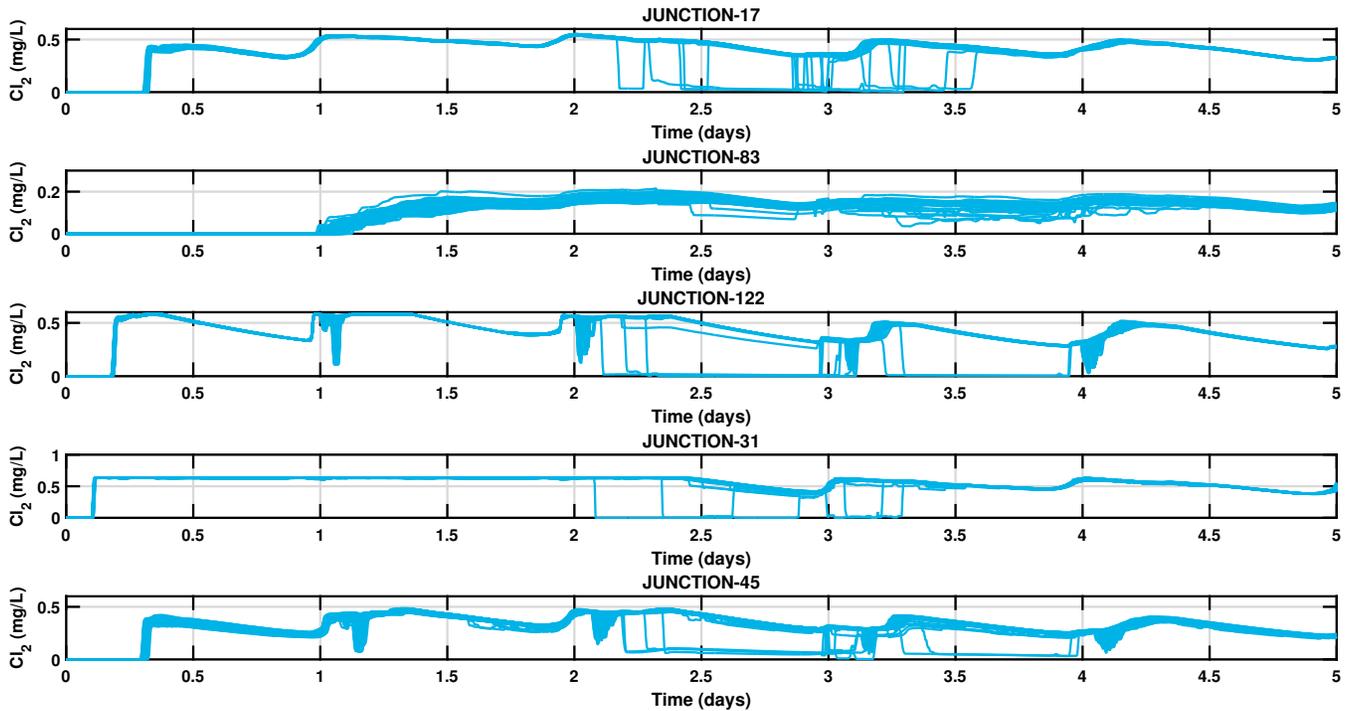


Figure 1. Overlay of the estimated chlorine concentration at 5 nodes from the benchmark network, from 100 random simulation scenarios.

4 Conclusions

In this work we have presented the EPANET-MATLAB Toolkit, an open-source software for interfacing MATLAB with EPANET, in an intuitive and an easy-to-use way. The Toolkit allows the user to access EPANET and EPANET-MSX through their shared object libraries, as well as their executables. The potential uses of the Toolkit expand to a wide range of applications; for instance, the Toolkit can be used as a framework to design new monitoring, control and fault/event diagnosis algorithms, as well as to formulate optimization problems which are based on water distribution network models. The modular architecture of the Toolkit allows the user to easily expand its capabilities by adding new functions. Furthermore, as a data structure, the Toolkit can facilitate exchange of data between function modules. The EPANET-MATLAB Toolkit is released under an open-source license (EURL), where any user can contribute, report issues or make suggestions for improvement, following the best practices of the EPANET Open Source Initiative. Future work will expand the Toolkit's capabilities, so that it is cross-platform and can be used directly in distributed algorithms which are suitable for cloud-based big-data analysis.

Acknowledgements. This research is funded by the European Research Council under the ERC Advanced Grant ERC-2011-ADG-291508 (FAULT-ADAPTIVE).

References

- Dodd, M. C., Vu, N. D., Ammann, A., Kissner, R., Pham, H. V., Berg, M., and von Gunten, U.: Kinetics and mechanistic aspects of As (III) oxidation by aqueous chlorine, chloramines, and ozone: Relevance to drinking water treatment, *Environ. Sci. Technol.*, 40, 3285–3292, 2006.
- 5 Eck, B. J.: An R package for reading EPANET files, *Environ. Model. Softw.*, 84, 149–154, 2016.
- Eliades, D., Stavrou, D., Vrachimis, S., Panayiotou, C., and Polycarpou, M.: Contamination event detection using multi-level thresholds, in: *Proc. Comput. Control Water Ind.*, vol. 119, 2015.
- Eliades, D. G. and Polycarpou, M. M.: A Fault Diagnosis and Security Framework for Water Systems, *IEEE Trans. Control Syst. Technol.*, 18, 1254–1265, 2010.
- 10 Eusuff, M. M. and Lansey, K. E.: Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm, *J. Water Resour. Plan. Manag.*, 129, 210–225, 2003.
- Ho, C. K. and Khalsa, S. S.: EPANET-BAM: Water Quality Modeling with Incomplete Mixing in Pipe Junctions, *Proc. Water Distrib. Syst. Anal. Symp.*, p. 11, 2009.
- Ostfeld, A., Uber, J. G., Salomons, E., Berry, J. W., Hart, W. E., Phillips, C. A., Watson, J.-P., Dorini, G., Jonkergouw, P., Kapelan, Z., di Pierro, F., Khu, S.-T., Savic, D., Eliades, D., Polycarpou, M., Ghimire, S. R., Barkdoll, B. D., Gueli, R., Huang, J. J., McBean, E. A., James, W., Krause, A., Leskovec, J., Isovitsch, S., Xu, J., Guestrin, C., VanBriesen, J., Small, M., Fischbeck, P., Preis, A., Propato, M., Piller, O., Trachtman, G. B., Wu, Z. Y., and Walski, T.: The Battle of the Water Sensor Networks (BWSN): A Design Challenge for Engineers and Algorithms, *J. Water Resour. Plan. Manag.*, 134, 556–568, 2008.
- Rossman, L. A.: EPANET Users Manual, Tech. rep., US EPA, Cincinnati, OH, 1994.
- 20 Rossman, L. A.: An Overview of EPANET Version 3.0, in: *Proc. Water Distrib. Syst. Anal. Symp.*, pp. 14–18, 2011.
- Rossman, L. A. and Boulos, P. F.: Numerical Methods for Modeling Water Quality in Distribution Systems: A Comparison, *J. Water Resour. Plan. Manag.*, 122, 137–146, 1996.
- Rossman, L. A., Boulos, P. F., and Altman, T.: Discrete Volume-Element Method for Network Water-Quality Models, *J. Water Resour. Plan. Manag.*, 119, 505–517, 1993.
- 25 Savic, D. A. and Walters, G. A.: Genetic Algorithms for Least-Cost Design of Water Distribution Networks, *J. Water Resour. Plan. Manag.*, 123, 67–77, 1997.
- Shang, F., Uber, J. G., and Rossman, L. A.: EPANET Multi-Species Extension User's Manual, 2008.
- Siew, C. and Tanyimboh, T. T.: Pressure-Dependent EPANET Extension, *Water Resour. Manag.*, 26, 1477–1498, 2012.
- Steffelbauer, D. and Fuchs-Hanusch, D.: OOPNET: An object-oriented EPANET in Python, in: *Proc. Comput. Control Water Ind.*, vol. 119, pp. 710–718, 2015.
- 30 Todini, E. and Pilati, S.: A gradient method for the analysis of pipe networks, in: *Proc. of Comput. Appl. Water Supply Distrib.*, 1987.
- Umberg, K., Uber, J. G., and Murray, R.: Performance Evaluation of Real-Time Event Detection Algorithms, in: *Proc. Water Distrib. Syst. Anal. Symp.*, p. 14, American Society of Civil Engineers, Reston, VA, 2008.
- van Zyl, J. E., Borthwick, J., and Hardy, A.: OOTEN: An object-oriented programmers toolkit for epanet, in: *Proc. Comput. Control Water*
- 35 *Ind.*, p. 8, 2003.
- van Zyl, J. E., Savic, D. A., and Walters, G. A.: Operational Optimization of Water Distribution Systems Using a Hybrid Genetic Algorithm, *J. Water Resour. Plan. Manag.*, 130, 160–170, 2004.