Reformatting and Post Processing Tools for SWMM5

Rajesh Rajan and Kim Artita, Jaclyn Marge and James Smullen
(Rajesh Rajan and Kim Artita are equivalent contributors to this chapter)

In its effort to meet regulatory requirements outlined in the United States Environmental Protection Agency's (USEPA) *Combined Sewer Overflow (CSO) Control Policy*, the City of Philadelphia characterizes its combined sewer system (CSS) using the USEPA's storm water management model (SWMM). Philadelphia's CSS is divided into three separate drainage districts each ending at its own wastewater treatment plant. The first hydraulic models of the CSS were developed in 1994, using the U.S. Army Corps of Engineers STORM and Camp Dresser McKee's NetSTORM. During 1995–1998, the city expanded these models in SWMM4 to include all major trunks, interceptors, and regulating structures. Due to the complexity of these models (~8 000 junctions and conduits) and resulting computational expense (model run times of 14 h to 16 h), the models underwent a simplification process (~4 000 junctions and conduits) to achieve model run times on the order of 2 h to 4 h. This simplification process was performed cautiously to avoid losing any of the hydraulic characteristics of the systems. Planning level estimates included in the city's long term control plan update (LTCP) submitted in fall 2009 were generated using the SWMM4 simplified models. In late 2009, the city began converting its simplified SWMM4 models to SWMM5. At the time of writing (April 2011), the conversion process is 95% complete.

We note here some of the major advantages of SWMM5 over its predecessors:

- dynamic memory allocation—there are no limits on the number of model elements;

- better solution techniques—Saint-Venant equations (for dynamic waves) are solved by a successive approximation technique that results in faster convergence;
- better simulation of orifices and weirs through use of the classical orifice and weir equations instead of using equivalent pipe approximations;
- a variable time step option for simulations;
- a pipe lengthening option—helps with convergence issues in short pipes; and
- better simulation of force mains.

However, SWMM5 has some disadvantages:

- the engine has bugs (that are being addressed);
- elements like bridges have not yet been included;
- the output format makes post processing more cumbersome; and
- without care, result files can be extremely large and difficult to post-process.

As part of the SWMM4 to SWMM5 model conversion, it was necessary to update the post processing tools originally developed in SAS. SWMM4's output format made it simple to import the data into SAS or Excel in a single dataset as a space delimited or fixed width format file. Conversely, SWMM5's output format is stacked (i.e. the time series output for each model element is written below the previous one) and must be read in sequentially to be post processed. For smaller datasets, sequential reads are not too time consuming; but for larger datasets, sequential reads are inefficient.

In this chapter, we describe several of the post processing tools that were created or updated and modified as part of the SWMM4 to SWMM5 model conversion:

- a SWMM5 output reformatting tool;
- a hydrograph comparison tool;
- a percent capture calculation tool; and
- a calibration tool.

19.1 Open Source Languages Python and R

Historically, most of the city's post processing was executed in the statistical analysis package SAS. Although SAS is a proven, widely used, and well respected software package, its cost and availability make it difficult for simultaneous use. The post processing programs described in this chapter were developed or updated in the two open source computing languages Python and R.

19.1.1 Python

Python is a free, high level programming language capable of supporting object oriented programming. Python is used because it is portable across multiple operating systems (Windows, Mac OS, and Linux/Unix), easily integrated with other programming languages such as C++ and Fortran through extension modules such as SWIG and f2py (see Table 19.1 for sources of Python and R modules and information). It is very simple and quick to code. In addition to text manipulation and data handling, Python is well suited for advanced scientific and engineering computing. The modules scipy (Scientific Python) and numpy (Numerical Python) include tools for statistics, integration, linear algebra, optimization, and more. Matplotlib, a plotting library with syntax similar to MatLab, makes graphing very straightforward. Users may edit or browse Python scripts using a simple text editor, free integrated development environments such as Komodo, or through Python's own graphical user interface, IDLE. Entire scripts can also be run or tested interactively in IDLE.

Table 19.1 Information and availability of Python and R modules.

Module	Information
Python	http://www.python.org/
SWIG	http://www.swig.org/
F2py	http://www.f2py.org/
Scipy	http://sourceforge.net/projects/scipy/
numpy	http://sourceforge.net/projects/numpy/
Matplotlib	http://sourceforge.net/projects/matplotlib/
Komodo	http://www.activestate.com/komodo/
GNU license	http://www.gnu.org/licenses/gpl.html
R, CRAN	http://www.r-project.org/index.html
Tinn-R	https://sourceforge.net/projects/tinn-r
Rpy	http://rpy.sourceforge.net/
Quick-R	http://www.statmethods.net/

19.1.2 R

R is a programming language, and statistical computing and graphic software freely available under the GNU general public license. Like Python, R is portable across multiple operating systems and can be downloaded through multiple CRAN (Comprehensive R Archive Network) mirrors. R's statistical and graphing functionality can be extended through packages or libraries, which can also be obtained through CRAN. R scripts can be edited using a simple text editor or integrated development environment such as Tinn-R that is capable of connecting to the R GUI. R can be called by Python through the module rpy. SAS users may find the website Quick-R useful in converting from SAS to R.

19.2 Reformatting Tool

Currently, SWMM5 text output files created using the command line executable are not formatted so that the flow, level and water quality information can be easily imported or post processed using programs like R, SAS, or Excel. For example, time series information for links and nodes are stacked (Figure 19.1). A reformatting tool was developed in Python that reduces the output file to just the subcatchment, link or node time series of interest (e.g. flow, velocity or depth) and places the information into adjacent columns as a comma delimited file (Figure 19.2). Resulting output files are much smaller (by a factor of 10) and can be quickly imported into the post processing tools described below. A previously written SAS program took several hours (up to a day) to achieve the same results. With Python, a single file (e.g. a 1 y run with ~400 conduits at 15 min output intervals) can be processed in <10 min. This is because Python reads each line of data sequentially, but does not store all lines into memory.

OCT-09-2005 23:53:00	1.484	11.265	0.191	12.8
OCT-09-2005 23:54:00	1.484	11.265	0.191	12.8
OCT-09-2005 23:55:00	1.484	11.265	0.191	12.8
OCT-09-2005 23:56:00	1.484	11.265	0.191	12.8
OCT-09-2005 23:57:00	1.484	11.265	0.191	12.8
OCT-09-2005 23:58:00	1.484	11.265	0.191	12.8
OCT-09-2005 23:59:00	1.484	11.265	0.191	12.8
OCT-10-2005 00:00:00	1.484	11.265	0.191	12.8

<<< Link CD13-000 >>>					
Date	Time	Flow CFS	Velocity ft/sec	Depth feet	Percent Full
OCT-07-2005	00:00:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:01:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:02:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:03:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:04:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:05:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:06:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:07:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:08:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:09:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:10:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:11:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:12:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:13:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:14:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:15:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:16:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:17:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:18:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:19:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:20:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:21:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:22:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:23:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:24:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:25:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:26:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:27:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:28:00	0.168	2.085	0.146	11.7
OCT-07-2005	00:29:00	0.168	2.085	0.146	11.7

Figure 19.1 Example of stacked SWMM5 time series outputs.

Inputs

Outputs

Figure 19.2 Example of inputs and outputs of the Python SWMM5 reformatting tool.

Although SWMM5 does not limit the number of elements that can be simulated, there is a limit to the size of the results file. To overcome this limitation, the models can be run for shorter periods and then the results can be joined together as a single output file using the reformatting program (see Figure 19.2 above). The reformatting tool can string together time series information from multiple SWMM5 output files into a single comma delimited output file and extract mass balance information to a separate text file. The tool makes use of the numpy function *array* to create an array from a list of time series (a list of lists). The array is then transposed and written or appended to file.

19.3 Hydrograph Comparison Tool

Differences in solution techniques between SWMM4 and SWMM5 (e.g. flow splitting between pipes, orifice equations) can lead to differences in model output (Rossman, 2006). By default, SWMM5 reports the instantaneous value for flow or depth at the specified reporting time step; in contrast, SWMM4 reports the average flow. To visualize and help quantify differences in instantaneous peak flows and total volumes at each regulating structure between SWMM4 and SWMM5, a hydrograph comparison tool was developed in R. This was an essential task because the LTCPU was submitted using SWMM4 simulations; future reporting will include estimates from SWMM5 simulations.

To check system performance, we examined two events at 1 min outputs: one short duration high intensity event (June 6) and one long duration medium intensity event (October 8) during a representative rainfall year for the Philadelphia area (2005). The tool first imports SWMM4 and Python converted SWMM5 time series, an input file relating all conduits, orifices and weirs to its respective regulator, and a date–time file describing the events of interest. For each regulator, the program plots hydrographs for each conduit and event of interest (Figure 19.3). For each conduit, the program computes the peak flow and total volume for the specified event period. Once all hydrograph plots are generated, the plotting program generates simple linear regressions on the peak flows and total volumes (Figures 19.4 and 19.5). Statistics for each hydrograph are printed on the hydrograph plot and written to a summary file.

Figure 19.3 Example of SWMM4 vs SWMM5 hydrograph comparison plot.

Figure 19.4 Example SWMM4 vs SWMM5 peak flow regression plot.

Figure 19.5 Example SWMM4 vs SWMM5 total volume regression plot.

The hydrograph comparison tool helped identify pipes and structures that needed modification beyond the simple SWMM4 to SWMM5 conversion. These include pipes and orifices with instabilities in dry and wet weather; differences in base wastewater flows at flow splits and regulating structures; and significant differences in total volumes. Once modifications were made to the SWMM5 model, the simulations and hydrograph comparison tool were re-run. The process was repeated until there was an acceptable match between SWMM4 and SWMM5 visually and numerically.

19.4 Percent Capture Tool

One indicator used to characterize the performance of a combined sewer collection system is *percent capture*. Generally it is the fraction of the average annual wet weather combined sewer volume that is captured and treated, but there are many different ways to compute percent capture (for examples see An and Gianvito, 2011). Higher percent capture indicates better system performance with fewer CSOs to receiving waters. In Philadelphia, percent capture is defined as the ratio of combined volume captured by the connector pipes to total volume in connector and overflow pipes during wet weather (Figure 19.6). Wet weather is defined as flow in the connector pipe above a stated threshold (e.g. 5%) compared to dry weather conditions, or when there is flow in the overflow pipe. Flow during any inter-event dry periods (e.g. 6 h) are not counted; however, base flow (that is the sum of the base waste water and the groundwater infiltration) during wet weather is considered captured.

Figure 19.6 Definition of capture used in Philadelphia.

The current version of the percent capture program operates on binary output files generated by SWMM4. The capture program computes the percent capture for each event and every regulator. Outputs from the capture program were then run through a SAS program to summarize percent capture by regula-

tor, by interceptor system and by drainage district. Because SWMM5 binary outputs could not be read by the same capture program, a new capture program was written in R to operate on the SWMM5 ASCII outputs converted with the Python reformatting tool described above. In addition to calculating percent capture by event, the new capture program provides percent capture by regulator and interceptor system. For each interceptor system, the following statistics are also calculated:

- total volume overflowed;
- peak overflow rate;
- minimum, mean, median and maximum number of overflow events;
- standard deviation of overflow events;
- minimum, mean, median and maximum overflow duration;
- percentage of sanitary contribution overflowed (assuming complete mixing);
- total volume captured and treated;
- percentage of sanitary contribution treated;
- minimum, mean, median and maximum number of wet weather events;
- standard deviation of wet weather events; and
- minimum, mean, median and maximum wet weather duration.

The R capture program produces a fourth output file to assist users in identifying problematic wet weather events and to troubleshoot the program upon premature termination. Any missing time series needed to run the capture program are identified and written to file before the program self-terminates. If there are negative flows in any connector or overflow conduits, these conduits and the date and time of occurrence are written to this output file. These times are either instabilities in the model or instances of the interceptor system backing up and causing negative capture.

19.5 Calibration Tool

Any computer model used for decision making purposes should be calibrated against an observed dataset and verified on an independent dataset. A previous calibration tool that assisted in comparing SWMM model simulations to observed rainfall and monitored flows was written in SAS. This tool was translated to R, and a few minor modifications and additional statistical calculations added. The raw flow monitoring and rainfall data is put through an intensive quality assurance (QA) process designed to identify any systematic errors or other inconsistencies in the data. These data errors are corrected if and when possible.

After the rainfall and flow monitoring data passes QA, the data is further analyzed and flow components are separated using the sanitary sewer overflow analysis and planning (SSOAP) tool box (Vallabhaneni and Burgess, 2007). The wet weather component of the flow that results from SSOAP analysis is compared to the wet weather response from the model. This comparison helps in deciding the adjustments that need to be made to the calibration parameters in the model. The calibration tool provides a quick qualitative and quantitative comparison between the observed and simulated data. This helps in deciding which model parameters need to be adjusted to match the model response as close as possible to the monitor response.

For all events of interest, the program reads in rainfall and modeled and monitored flow data then performs the following calculations:

- rain intensity;
- time of peak rainfall for each event;
- rainfall deviation between rain gauges;
- peak rainfall intensity;
- rainfall statistics by event (mean, standard deviation and variance);
- time to peak for modeled and monitored flows by event;
- ratio between modeled time to peak and monitored time to peak;
- whole event volumes;
- mean rainfall of all rain gauges;
- difference of mean rainfall to a selected rain gauge; and
- cumulative volumes: model, monitor, selected rain gauge and mean rainfall from all rain gauges.

The calibration tool then produces the following plots and associated statistics:

1. Rainfall plots:
 - a rainfall plot for each event for each rain gauge;
 - double mass regressions plots using cumulative rain gauge totals of the selected rain gauge and the mean of all rain gauges with (i) y-intercept set to 0 and (ii) without setting y-intercept; and
 - a scatter plot of the residuals from the linear regression of the cumulative rain gauge totals overlaid with a plot of the 80% residual ellipse (for more information see Martens and Smullen, 2004);
2. Event volume plots:
 - a scatter plot of modeled versus monitored event volumes for all *good* events, where a good event is one that is absent of snowpack, where the monitor shows a good response to

- rainfall (i.e. peak monitor response lags after peak rainfall), and the event is not an extreme event marked by flooding and volume loss. A linear regression (and all associated statistics) is calculated and plotted for all good events. Plots are created with (i) upper and lower 95% confidence intervals, (ii) upper and lower 95% prediction intervals, and (iii) a line of equal fit;
 - double mass regressions plots using cumulative model and monitor event volumes with (i) y-intercept set to 0 and (ii) without setting y-intercept;
 - a scatter plot of the residuals from the linear regression of the cumulative volumes overlaid with a plot of the 80% residual ellipse; and
 - a cumulative frequency distribution for modeled and monitored event volumes;
3. Peak flow plots:
- time series plots of modeled versus monitored flows for each event with associated rainfall histogram on upper x-axis. The time to peak for both modeled and monitored flows is calculated and written to file;
 - a scatter plot of modeled versus monitored peak flows for all good events. A linear regression (and all associated statistics) is calculated and plotted for all good events. Plots are created with (i) upper and lower 95% confidence intervals (Figure 19.7 below), (ii) upper and lower 95% prediction intervals, and (iii) a line of equal fit;
 - double mass regressions plots using cumulative modeled and monitored peak flows with (i) y-intercept set to 0 and (ii) without setting y-intercept; and
 - a scatter plot of the residuals from the linear regression of the cumulative peak flows overlaid with a plot of the 80% residual ellipse.

There are numerous reasons why a rainfall event can be considered bad and excluded from calibration statistics (Figure 19.8). These include unusually low or high rainfall derived inflow and infiltration immediately preceding or following a rainfall event (determined using USEPA's SSOAP) and observed or modeled flows that are not correlated to observed rainfall. Limiting statistics to just good events is a better method of performing calibration. Additional events can be included or excluded, and the calibration tool re-run. The tool is typically run several times with adjustments to the events before actual calibration adjustments are made to the model.

Figure 19.7 Example calibration scatter plot with linear regression and 95% confidence intervals for good events; bad events are shown as gray cross-hairs.

Figure 19.8 Example time series plot for a bad event generated using the R calibration tool; note how there is an observed response in dry weather and the peak observed flow (denoted by an asterisk) occurs before the peak rainfall.

19.6 Summary

There are many free and widely available open source resources to help post process SWMM5 outputs. Two well known cross-platform computer and statistical languages are Python and R. In this chapter, we described several useful post processing tools either developed or translated using these languages. Creation and translation of these tools were initiated by the demands of the city's CSS model conversion from SWMM4 to SWMM5 as well as future needs. SWMM5 users may find the descriptions of these tools useful for the development of similar tools to meet their respective modeling problems and needs.

References

- An, W. and J.M. Gianvito. 2011. "Comparison of Different Methods in Calculating CSS Percent Capture." *Journal of Water Management Modeling* R245-18. doi: 10.14796/JWMM.R245-18.
- City of Philadelphia. 2009. Green City, Clean Waters – Combined Sewer Overflow Long Term Control Plan Update.
- Martens, G. and J. Smullen. 2004. "Normalizing Rain Gauge Network Biases with Calibrated Radar Rainfall Estimates." *Journal of Water Management Modeling* R223-16. doi: 10.14796/JWMM.R223-16.
- Rossman, L.A. 2006. Storm Water Management Model Quality Assurance Report: Dynamic Wave Flow Routing. USEPA EPA/600/R-06/097.
- Vallabhaneni, S. and Burgess, E.H. 2007. Computer Tools for Sanitary Sewer System Capacity Analysis and Planning. EPA/600/R-07/111.