
Runtime Comparisons between SWMM4 and SWMM5 using various Continuous Simulation Model Networks

Carl C. Chan, Robert E. Dickinson and Edward H. Burgess

The Stormwater Management Model (SWMM) is a dynamic rainfall-runoff model used for continuous simulation of runoff quantity and quality. SWMM has recently been redeveloped under a Cooperative Research and Development Agreement (CRADA) between CDM Inc. and the U.S. Environmental Protection Agency (USEPA). One of the purposes of the CRADA is to improve the numerical stability of the model without changing the fundamental characteristics of the EXTRAN solution or sacrificing the efficiency of its performance. Extensive QA/QC testing was performed as part of the CRADA to ensure consistency between the SWMM4 and SWMM5 solutions. (Dickinson et al., 2004; Rossman et al., 2003; Schade, 2002; Chan et al., 2003). Another aspect of QA/QC testing was to compare simulation runtimes between SWMM4 and SWMM5.

This chapter presents the computational speed comparison between SWMM4 (Huber & Dickinson, 1988) and SWMM5 (Rossman et al., 2003). Various SWMM4 historical applications were used and runtimes of SWMM4 and SWMM5 were plotted and compared by different calculation methods.

This chapter also describes the approach used to test the simulation speed in SWMM4 and SWMM5. All models were run on the same computer workstation for fairness in the comparison. The test computer was equipped with a commonly available Pentium® P4-1.6 GHz processor with 256Mb RAM. The SWMM4 models were run with SWMM499 and the new SWMM5 (version Beta E). To ensure data consistency, the historical SWMM4 applications were translated to SWMM5 input format, using the converter software distributed with the SWMM5 package. Results generated

Chan, C.C., R.E. Dickinson and E. Burgess. 2005. "Runtime Comparisons between SWMM 4 and SWMM 5 using Continuous Simulation Model Networks." *Journal of Water Management Modeling* R223-11. doi: 10.14796/JWMM.R223-11.

© CHI 2005 www.chijournal.org ISSN: 2292-6062 (Formerly in Effective Modeling of Urban Water Systems. ISBN: 0-9736716-0-2)

from the new SWMM5 were compared to those from the existing SWMM4 model as part of the speed test procedure.

11.1 Methodology

11.1.1 Input Datasets

25 sewer network models were selected for the speed test. These datasets were calibrated SWMM4 continuous simulation models that were selected from various existing project applications. The sizes of the testing models ranged from 10 nodes to 2500 nodes. These selected input files cover all appurtenant elements (i.e. orifices, pumps, storages, weirs, etc.) in SWMM. Different levels of hydraulic surcharging conditions were also well represented in the selected test datasets. Additionally, the characteristics of various runoff methods, mainly surface runoff and rainfall dependent inflow and infiltration, were also covered in these 25 datasets.

11.1.2 SWMM Engines

SWMM499 and SWMM5 Beta E were chosen to be the testing engines for this exercise. SWMM499, an enhanced version of SWMM44H, is the most recent and final member of the SWMM4.x family. The advantage of using SWMM499 is the ability to produce output files that can be used to directly compare each time step in SWMM5. This speeds up the SWMM quantitative comparison. SWMM5 Beta E was the most recent version available during the testing period.

11.1.3 Dynamic Wave Calculation/Integration Methods

Traditionally, SWMM4 has used the modified Euler solution as the sole numerical integration technique in performing dynamic hydraulic analysis. Over the years, two additional variations of this technique were developed (Roesner, Dickinson & Aldrich, 1998).

- Original Modified Euler Solution (ISOL = 0; XROUTE)
- Enhanced Modified Euler Solution (ISOL = 1; YROUTE)
- Iterative Under-Relaxation Solution (ISOL = 2; ZROUTE)

SWMM5 introduces the Picard solution as the new numerical integration method in dynamic hydraulic calculation (Dickinson et al., 2004). The Picard

method appears to be more robust and under some conditions, more efficient than the Euler solution. Another new feature in SWMM5 is the variable time step option. SWMM5 has a built-in algorithm that continuously observes the solution accuracy during the course of the computation and adaptively changes the time step size to maintain a consistent level of accuracy. The time step size may change many times during a simulation. The larger time steps are used when the solution slowly varies (i.e. typically dry-weather) and smaller steps are used when the solution rapidly varies (i.e. typically wet-weather). Users can define the range of the time steps used in the computation. This variable time step option is available for both Euler and Picard integration. Table 11.1 summarizes the computation options available in SWMM4 and SWMM5.

Table 11.1 Computation options in SWMM4 and SWMM5.

SWMM4	SWMM5
1. Original Modified Euler Solution (ISOL = 0; XROUTE)	1. Euler Solution
2. Enhanced Modified Euler Solution (ISOL = 1; YROUTE)	- variable time step
3. Iterative Under-Relaxation Solution (ISOL = 2; ZROUTE)	- constant time step
	2. Picard Solution
	- variable time step
	- constant time step

11.1.4 Testing Procedure

The selected 25 calibrated SWMM4 continuous simulation models were first simulated in SWMM4, producing results and calibration files for SWMM5. Secondly, the input files were translated to SWMM5 input format using the converter software from the SWMM5 package. The next step was to run the simulations in SWMM5 with various dynamic wave integration methods with and without variable time step option. Model results were examined to ensure both engines produced similar quantitative results. The SWMM4 and the SWMM5 produced similar results for the all test models. Runtimes of SWMM4 and SWMM5 were plotted and compared by different computation methods. The same machine was used for all simulations.

There are many variables that can affect the speed of a simulation, e.g. hardware, dataset characteristics, compilers, SWMM solutions. Under certain circumstances, simple comparison of the runtime duration may not fully describe the speed differences of the two different solutions. For example, a SWMM5 simulation could be faster due to the fact that less time steps was required during the course of computation (variable time step = ON) while the

actual calculation time required per time step is slower than SWMM4. In order to avoid this misrepresentation, two measurements of engine speed were developed:

- unit runtime (s/iteration/node)
- total runtime (s)

Unit Runtime was designed to help programmers understand the efficiency of the code. This was used during the engine debugging process. Total runtime was targeted for SWMM users, especially for continuous simulation where total runtime is a primary concern. Therefore, simulation runtime was the main focus in speed testing and comparison. Four tests were developed to isolate different parameters that could affect the engine speed:

- SWMM4 Euler vs SWMM5 Picard (with and without variable time step; non-surcharge condition);
- SWMM4 Euler vs SWMM5 Euler (with and without variable time step; non-surcharge condition);
- SWMM4 Euler vs SWMM5 Optimized solution (surcharge condition); and
- Borland Compiler 5.5 vs Microsoft VC Compiler 6.0 (surcharge and non-surcharge condition).

11.2 Results and Discussions

11.2.1 Test 1: SWMM4 Euler vs SWMM5 Picard

The purpose of this test was to identify the speed difference of SWMM4 Euler solution and SWMM5 Picard solution. Seven calibrated models were used in this test to compare the speed of SWMM4 Euler and SWMM5 Picard. The sizes of the sewer networks ranged from 10 to 240 nodes. All models were run using the SWMM4 Euler (ISO=0) with a 5 s fixed time step, the SWMM5 Picard with a 5 s fixed time step, and with the SWMM5 Picard with a variable time step of a maximum 30 s. The simulation period was 31 days. The models were in a non-surcharge condition. Both engines and all methods produced similar quantitatively results. Figure 11.1 shows the speed difference between engines.

The SWMM5 Euler with a constant time step is slightly faster (10%) than the SWMM4 Euler. The SWMM5 Euler with a variable time step is significantly faster (50-300%) than the SWMM4 Euler. As the size of the

model increases, the relative speed difference between the SWMM5 Euler Variable and the SWMM4 Euler increases.

11.2.2 Test 2: SWMM4 Euler vs SWMM5 Euler

This test was designed to focus on the speed difference between SWMM4 Euler (ISO=0) with a 5 s constant time step, the SWMM5 Euler solution with a 5 s constant time step, and the SWMM5 Euler solution with a variable time step of a maximum 30 s. Seven calibrated non-surcharge models were selected in this test. The sizes of the sewer networks range from 10 to 260 nodes. The simulation period was 31 days.

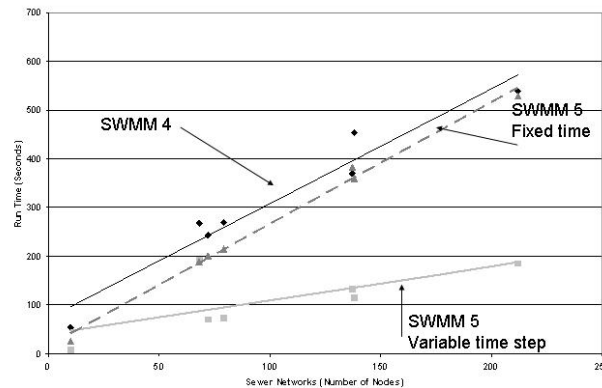


Figure 11.1 Runtime comparisons: SWMM4 vs SWMM5 Picard. (Non-surcharged conditions).

The results from both engines and methods agreed quantitatively. Figure 11.2 shows that the runtime of the SWMM5 Euler with a variable time step is similar to the SWMM4 Euler. However, the fixed time step for the SWMM5 Euler is slower than SWMM4. The causes for the slow Euler fixed solution remains uncertain at the time of testing. It could be a combination of minor algorithm flaws or coding inefficiency.

11.2.3 Test 3: SWMM4 Euler vs SWMM5 optimized solution (surcharge condition)

The optimized solution described in this test can be defined as the fastest computation in SWMM5. The purpose of this test is to compare the speed of

SWMM engines during a surcharge condition. Five calibrated models were used in this test. The sizes of the models were from 100 to 2400 nodes. The simulation length was 96 hours. Both engines shared similar results quantitatively. Figure 11.3 shows the speed test result of this exercise. Under a surcharging condition, SWMM5 is much faster than SWMM4 on average. For small models, the speeds between SWMM4 and SWMM5 are similar. As the size of the models increase, however, the relative speed difference significantly increases in SWMM5.

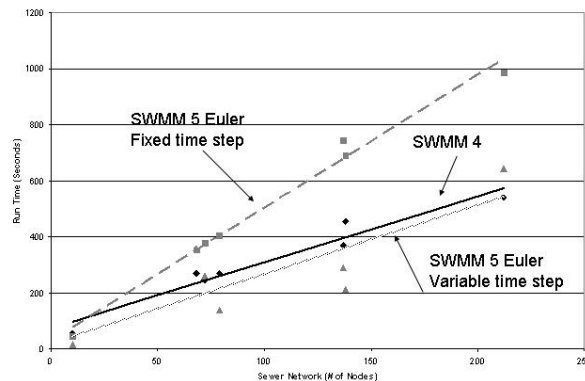


Figure 11.2 Runtime comparisons: SWMM4 vs SWMM5 Euler. (Non-surcharged condition).

The tests above reveal that the Picard solution with a variable time step is the fastest solution in SWMM5 and is significantly faster than SWMM4. The speed benefit dramatically increases when the network size increases. The SWMM5 Picard solution with a fixed time step is slightly faster than SWMM4 although the speed gains *decrease* with increasing network size. The tests show that the SWMM5 Euler solution with a variable time step is about the same speed as SWMM4. However, the SWMM5 Euler fixed time step is slower than SWMM4. The reason could be a combination of code inefficiency, oversight of calculations, or minor algorithm flaws. The SWMM5 Redevelopment Team is committed to the improvement of the engine so that it provides a faster and more accurate solution.

11.2.4 Test 4: Borland C SWMM5 vs Microsoft VC SWMM5

During the course of the investigation of the code efficiency, the SWMM5 Redevelopment Team discovered the choice of the C compiler can impact the

computational speed of the engine. The purpose of this test was to help the SWMM5 Redevelopment Team decide which compiler to use for the official version of SWMM5. Seven models were used in this test, as well as the same datasets used in Test 1. Two compilers were used in this test: Borland C 5.5 (public version) and Microsoft Visual C++ 6.0 (commercial version). Figure 11.4 shows that the Microsoft Visual C++ has a significantly better performance (55% faster) than Borland C 5.5.

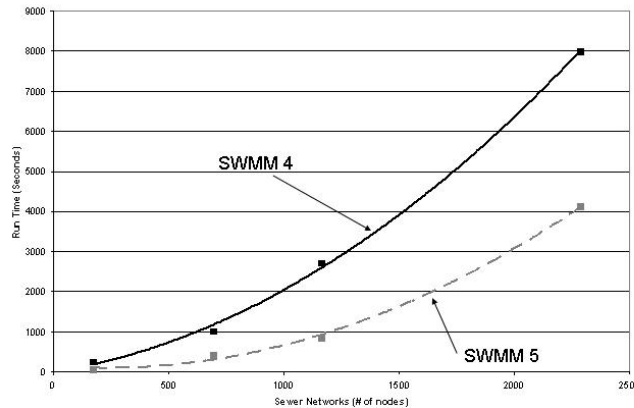


Figure 11.3 Runtime comparisons: SWMM4 vs SWMM5 optimized. (Surcharge condition).

Notice that the Microsoft Visual C++ compiler is two years older than the Borland C 5.5 compiler. The reasons for the different performances remain unknown. The following reasons may explain why Microsoft Visual C++ may be a better choice of compiler for SWMM5:

- The test computer uses Windows XP as its operating system. Microsoft Visual C++ is known to be the better compiler for XP.
- Borland C 5.5 is public domain (free) software. Microsoft Visual C++ is a commercial product. Borland also has a commercial version of a C compiler. It is suspected that the optimization compiling ability in the public domain version is not as proficient as the commercial version.

This test strongly suggests that the SWMM5 Redevelopment Team should further search for a better compiler for the final release version of SWMM5. Microsoft Visual C ++ 7.0 is now on the market. However, the Borland commercial compiler could be faster than Microsoft Visual C++ 6.0.

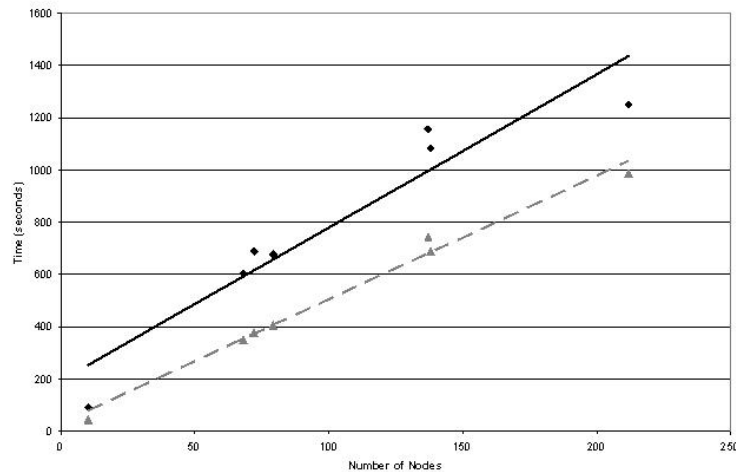


Figure 11.4 SWMM5 compilers comparisons using SWMM5 Euler solution. (Fixed time step).

11.3 Conclusions

In the past ten years, technological advancements in the computer field have been astonishing. As the current trend in the development of high performance desktop computers continues, long term continuous SWMM simulation becomes more feasible in engineering practice. Computation time will become increasingly less of an obstacle to long term continuous hydraulic analysis in the near future. Besides relying on future hardware advancements to improve the computational speed, there are several current ways to increase model efficiency. One is code optimization, in which new algorithms in calculation or data structure may improve computational efficiency. Another possible option to help in gaining speed is to migrate the SWMM program code to the Linux environment. With the SWMM code now entirely written in the C language, migrating to Linux should not be difficult. The new 64-bit microprocessors can also be an option for improving simulation runtimes. Lastly, parallel processing technology should be deployed in SWMM5. CDM has written a SWMM4 engine that can take advantage of parallel processors (Burgess et al., 1999). The SWMM5 Redevelopment Team should review the parallel code and incorporate into SWMM5.

References

- Burgess, E., W.R. Magro, M. Clement, C. Moore and J. Smullen. 2000. "Parallel Processing Enhancement to SWMM/EXTRAN." *Journal of Water Management Modeling* R206-03. doi: 10.14796/JWMM.R206-03.
- Chan, C.C., E. Burgess, T. Schade and R.E. Dickinson. 2003. "Design of the SWMM Data Translation and the Testing of SWMM 5 Engine." *Journal of Water Management Modeling* R220-17. doi: 10.14796/JWMM.R220-17.
- Dickinson R. E., Burgess E. H., Rossman L.A., and Roesner L.A. 2004. The Modified Euler Solution Algorithm and the Picard Solution Algorithm in SWMM5. 2004 Conference on Stormwater and Urban Water Systems Modeling., Toronto, Ontario, Canada. Feb 19-20, 2004.
- Huber, W. C. and R. E. Dickinson (1988). Storm Water Management Model – Version 4: User's Manual; Cooperative Agreement CR-811607; U.S. EPA; Athens, Georgia.
- Roesner, L.A., R.E. Dickinson and J.A. Aldrich (1988) Storm Water Management Model – Version 4: User's Manual – Addendum 1 EXTRAN; Cooperative Agreement CR-Rossman, L.A., Dickinson R.E., Schade T., Chan C., Burgess E.H., Sullivan D., and F. Lai. 2003. "SWMM 5 - the Next Generation of EPA's Storm Water Management Model." *Journal of Water Management Modeling* R220-16. doi: 10.14796/JWMM.R220-16.
- Schade, T. (2002). Quality Assurance Project Plan for SWMM Redevelopment; U.S.EPA Office of Research and Development – Water Supply and Water Resources Division; Cincinnati, Ohio; November 27, 2002.
- U.S. EPA (2002). SWMM Redevelopment Project Plan – Version 5; U.S.EPA Office of Research and Development – Water Supply and Water Resources Division; Cincinnati, Ohio; February 19, 2002.

