An Explicit Conduit Storage Synthesis Algorithm for Solving Decoupled Forcemain Networks

Trent Schade, Chris Baxter, Misgana Muleta, and Paul F. Boulos

A typical collection system may include many gravity mains connected by pump stations pushing flow into a complex network of pressurized conduits or forcemains (ASCE, 1982). The EPA standard SWMM5 engine solves this complex case, but it can be very expensive and time consuming. In most forcemain networks, the solution time improves significantly by separating or “decoupling” the gravity network from the force main network. Decoupling can improve computational speed three to five times. Larger networks have much larger increases in solution time speed for the decoupled network, up to twenty times faster.

Results for the decoupled model may differ from the original model. In particular, wet-well levels in the gravity network influence the pump operations for the forcemain network. Ignoring the storage available in gravity conduits can lead to oversized pump designs, because the overall storage volume is much larger than the volume available in the wet-well alone.

This chapter develops a rigorous dynamic conduit storage synthesizer approach that estimates the conduit volume available for storage and applies it to the wet-well depth-area curve to reduce the impact of decoupling the model on pumping simulations. We discuss the new method and apply it to a decoupled system with and without synthesized storage.
22.1 Introduction

22.1.1 Typical Network Layout

Collection systems in some areas of the United States include small gravity networks connected by pump stations to complex pressurized or forcemain networks (Nicklow et al., 2004). Figure 22.1 shows an example network with this configuration. This network configuration can be problematic to analyze with a dynamic model; an efficient set of solution parameters, in particular the hydraulic routing timestep, for the gravity network can be much different from the parameters in a forcemain network. Consequently, finding an appropriate balance between accurate results and an efficient simulation eventually may lead to a “decoupling” between the gravity network and the forcemain network.

Figure 22.1 Forcemain network layout.

A good candidate location for splitting the network is usually the wet-well feeding the pump. In many cases, the wet well volume is much smaller than the storage available in the conduits upstream from the wet-well. This presents a challenge when reviewing pump sizes or controls in a decoupled network, and can result in undersized pumping solutions or incorrect controls from ignoring storage in the conduits. Simple analytical techniques
can yield estimates of the conduit storage relationship to stage, but they require hand calculations under the assumption that conduits in the network are horizontal because a volume calculation on a partially full and sloped pipe is a complicated task.

In contrast to complex volume approximation procedures, this chapter proposes a Conduit Storage Synthesizer (CSS) algorithm utilizing the dynamic computations inherent in the SWMM engine. CSS computes and returns the gravity-system conduit volume upstream from the wet-well as additional volume—resulting in an accurate and effortless model decoupling process. The module builds a depth-storage curve for the gravity portion of the collection system, integrates it with the original depth-volume relationship of the wet well, and constructs aggregated depth-area curve for the wet well. The proposed method can also be effectively used to calculate the depth-volume relationship for natural channels and flood plains.

22.1.2 Wet Well Storage Relationship to Depth

SWMM5 expresses the basic relationship between stage and wet well volume as a function of depth versus storage area (Rossman, 2005). This function takes one of two forms, either a function as described in Equation 4.1 or as a user-defined tabular relationship.

\[ \text{Area} = A \cdot (\text{Depth})^B + C \]  

(4.1)

where:

- \( A \) = Depth coefficient,
- \( B \) = Depth exponent, and
- \( C \) = Area constant.

The most common configuration is for vertical-walled wet wells. In this case, the area is a constant, so the coefficient, \( A \) and the exponent, \( B \), are both set to zero, and \( C \) would represent area of the wet well as seen in a plan view of the wet well. The functional relationship provides a monotonic function of the depth-area relationship, whereas the tabular depth-area relationship is not necessarily linear—allowing for abrupt changes to the area with changes in depth. The CSS algorithm will operate under either type of configuration of wet well, it does not require a linear relationship.
22.1.3 Conduit Storage Relationship to Stage

Of course, the relationship of conduit storage to stage is very unlikely to be linear; rather, it is a function of network configuration. Figure 22.2 shows a group of conduits connected to a storage node with horizontal markers (1-9) showing each point where the relationship between conduit storage volume and depth changes. Obviously, there is no simple function to describe the conduit volume relationship to the wet well depth. Therefore, an efficient and reliable means to express this additional storage is as a tabular depth-area curve.

![Figure 22.2 Conduit storage concept.](image)

22.1.4 Geometric Approach

It is possible to compute the geometric volumes at each of the points (1-9) shown in Figure 22.2. Furthermore, it is possible to compute the depth area relationship between these points. However, it is a daunting task to derive the relationship between head and storage volume using a geometric approach as the integral calculus required is cumbersome. Conduit Y shows the simplest case where a conduit is horizontal. In this case, the computation of the volume based on the depth is simply the cross-sectional area multiplied by the length. Considering that conduit networks are very complex with potentially unlimited configurations, the geometric solution quickly becomes intractable.
22.2 Methodology

The proposed CSS algorithm avoids the complex geometry problem. Rather we use an analytical technique that measures system volume changes as computed in the SWMM5 engine at regular intervals. For each wet well location, CSS performs the following four steps to obtain estimates of conduit storage volumes at predefined regular intervals for a given wet well.

Step 1. Trace Upstream Network

Given the wet well location and network topology, CSS applies an advanced tracing algorithm that can identify gravity conduits contributing flow to the wet well. The backward tracing algorithm stops when it encounters another pump station or when it finds all network components draining to the wet well. CSS then activates this portion of the network only—inactivating other network elements. Figure 22.3 illustrates how this trace appears outside of the CSS algorithm. Highlighted conduits are gravity sewers flowing into a wet well at the pump station location.

![Figure 22.3](image) Illustrative example of a Trace Upstream Network operation.
Step 2. Create a Boundary Condition Time Series

With the pressurized conduits downstream of the wet well inactive, CSS creates an artificial outfall, and connects the outfall to the wet well with an artificial circular conduit. The outfall uses stage time series data that acts as a boundary condition for the decoupled gravity system. The time-series is constructed so that water depth in the system sequentially drops from full depth of the wet well to its bottom within the CSS simulation duration. Two input parameters, Acting Storage Area and Stage Area Data Points, define how to construct the time-series and ensure that the effect of the boundary condition will fully reflect the stage-storage relationship. The input values given below apply to all wet wells selected for analysis.

![Figure 22.4 Change in Boundary Condition in Time.](image)

*Acting Stage Area* - Water-level fluctuation in a wet well depends on its size. If the size of the wet well is too large, fluctuation of water level at the outfall (i.e., boundary condition) may not result in equivalent water level fluctuation in the wet well, limiting examination of stage-storage relationship to the higher water levels only. On the contrary, if the wet well is too small the model may experience numerical instability and may yield unreliable results. To avoid these problems, CSS lets the modeler specify a wet well area that could be different from the actual wet well area. This area, known as *Acting Storage Area*, will be used only to compute storage in the gravity conduits and will not affect the original (i.e. actual) size of the wet well. CSS assumes that the storage-node has a cylindrical shape. Therefore, *Acting Storage Area* represents cross-sectional area of the cylindrical wet well. CSS uses a default value of 1000 ft² for acting storage area.
**Stage Area Data Points** - This input data determines the construction of the stage time-series data that is used as the simulation boundary condition. Once the simulation duration and height of the wet well are known, CSS internally constructs a stage-time series that has the specified number of data points (i.e., time series rows) in such a way that the wet well level drops linearly from the top to the bottom of the wet well within the simulation duration. The plot on the right shows, as an example, the stage time-series CSS constructs when the specified number of data points is 100; the wet well height is 18 ft; and the simulation duration is 18 h. CSS uses a default value of 100 for stage-area data points.

The boundary condition time series decreases from the elevation of the wet well top to the elevation of the wet well invert. The user chooses the number of points to sample between the top of the wet well and the invert elevation. These points represent a dropping boundary condition, controlling the depth of water throughout the simulation.

Typically, the sampling would result in a reasonable depth change for each hour. Values between 50 and 200 about 0.1-0.5 ft (0.04-0.2 m.) will work for most systems. CSS sets this time series as a time varying boundary condition at the outfall node, effectively controlling the water level in the simulation.

**Step 3. Initial Run**

CSS modifies the existing network, adding both an outfall and a conduit connecting the wet well to the outfall. CSS initializes the network by setting all nodes to their maximum depth. With the boundary condition time-series controlling the water level in the storage unit, CSS runs the model. Each hour, the boundary condition drops allowing some volume of water to drain from the system. At the end of the run, CSS produces a system volume time-series as shown in Figure 22.5.

**Step 4. Second Run**

The system volume time-series from the first run includes a volume for the conduit leaving the system. CSS makes a second run where the only elements in the model are the storage area, the outfall and the conduit connecting the two. The results from this run are then simply subtracted from the results for the first run, leaving only the upstream gravity network volume.
Figure 22.5 System Storage from two runs.

Figure 22.6 Depth area results.
This procedure yields two stage volume curves. The first is for the original configuration, including both the gravity network and the storage node, and the second is for just the storage node and the artificial conduit. CSS computes the stage-area relationship from these data by computing the difference between the system storage volumes for both runs. At some point, the network is empty and any differences in volume are relatively small. CSS allows the user to filter these data based on a relative volume threshold or maximum and minimum storage levels. Figure 22.6 shows the final result expressed as a stage-area.

22.3 Comparison Network

To compare modeling results for a coupled, decoupled, and decoupled with CSS, we use a sample network with a complex system of wet-wells to estimate the actual improvements from running the coupled and decoupled networks. Clearly, actual networks have some interesting depth-volume relationships. Figure 22.7 shows some of these relationships in the example network.

Figure 22.7  Depth vs. Network Storage in sample network.
We made four model runs with four different configurations of the system. The original model includes all of the connected system as originally configured. A gravity system model includes only the top reaches of the gravity system, and excludes gravity system portions downstream from pump stations. The gravity system supplies loading for the decoupled model and the CSS model. Both the decoupled model and the CSS include the conduits excluded from the gravity model. The only difference between these two is that the conduit storage areas are included in the CSS model.

All of these systems simulated a 7-day period with dry-weather loading conditions. Table 22.2 gives the Continuity results for each simulation. Notice the CSS run seems to overestimate the storage volumes. Figure 22.8 shows a clear example of how the CSS algorithm creates a more realistic pump cycling result.

![Graphs showing pump cycling example](image-url)
Table 22.2 Continuity for simulations.

<table>
<thead>
<tr>
<th>Item</th>
<th>Original</th>
<th>Gravity</th>
<th>Decoupled</th>
<th>CSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dry Weather Inflow</td>
<td>1.04</td>
<td>0.957</td>
<td>0.13</td>
<td>0.13</td>
</tr>
<tr>
<td>External Inflow</td>
<td>0</td>
<td>0</td>
<td>0.956</td>
<td>0.956</td>
</tr>
<tr>
<td>External Outflow</td>
<td>1.114</td>
<td>0.957</td>
<td>1.157</td>
<td>1.141</td>
</tr>
<tr>
<td>Initial Stored Volume</td>
<td>0.132</td>
<td>0.007</td>
<td>0.096</td>
<td>0.168</td>
</tr>
<tr>
<td>Final Stored Volume</td>
<td>0.147</td>
<td>0.007</td>
<td>0.112</td>
<td>0.191</td>
</tr>
<tr>
<td>Run Time</td>
<td>0:04:42</td>
<td>0:00:10</td>
<td>0:00:50</td>
<td>0:00:53</td>
</tr>
<tr>
<td>Continuity Error (%)</td>
<td>-7.671</td>
<td>0.039</td>
<td>-7.399</td>
<td>-6.158</td>
</tr>
</tbody>
</table>

22.4 Conclusions

Conduit storage applied to forcemain networks provides a clear computational advantage with limited impact on hydraulic results. Like all modeling tools, its applications should be very carefully considered before they are implemented. Results should be evaluated and weighed against a full coupled model. However, the distinct time savings will significantly improve the runtime for many forcemain networks. The decision of when to use CSS is solely dependent on the modeler’s judgment and the particular sewer system and situation.

Further research into the conduit storage synthesis approach will yield more algorithm refinements for enhancing the accuracy of detailed simulation results. Refinements to the CSS methodology may include estimations proper discretization of the boundary condition, adjustments to the hydraulic time step between estimation steps, boundary estimates of conduit storage based on simple geometric assumptions (assume flat conduit), and error analysis of the an estimate of stage-area relationship.

References

American Society of Civil Engineers Manuals and Reports on Engineering Practice No. 60 (1982). Gravity Sanitary Sewer Design and Construction. Reston, VA.

