

# Modular Designs, Fall 2015

Stephanie Sun

December 11, 2015

## Table of Contents

[Table of Contents](#)

[Part I: Problem Definition](#)

[Introduction](#)

[Design Details](#)

[Part II: Documentation](#)

[Initial Steps - Box](#)

[Sedimentation Tank & Testing](#)

[Beta Server & Design Tool](#)

[LFOM \(Linear Flow Orifice Meter\)](#)

[Flocculator & Entrance Tank](#)

[Future Goals](#)

## Part I: Problem Definition

### Introduction

The Fall 2015 Modular Designs task is a continuation of previous efforts to provide AguaClara clients the ability to request individual component designs on the AguaClara website. In Fall 2012, design team member Heidi Rausch began working on creating upper level MathCAD files that the design tool will use to create individual component designs in addition to the already available full scale plant design. The goal of the Fall 2015 Modular Designs team is to continue working on the modular designs by updating already existing component files as well as creating new upper level files for other parts of the plant.

### Design Details

The main challenge of this design task is to create component design files that do not rely on other components of the plant. The full plant designs, which contain rapid mix, flocculation, sedimentation, and filtration unit processes, can be drawn fully through existing code and are also accessible to the public. Specific components of the plant, however, are not

able to be drawn individually due to reliance on other parts of the plant. For plant components to be drawn individually, upper level individual component design files which contain enough information to draw the specified component need to be created. These component design files do not affect other parts of the code as they draw from already existing MathCAD functions. While some inputs and functions may need to be created or redefined in the MathCAD component file, they will not alter the pre-existing files as they only build on what is already present. All changes are made in the individual component files found in the ADT Designs folder, not the original design files. After creating the component designs, they are then uploaded onto the beta server for testing. The final goal is for these designs to be integrated into the actual design server so clients can easily request individual component designs.

## Part II: Documentation

### Initial Steps - Box

To approach the design task, the team familiarized themselves with the modular designs challenge by reviewing past documentation and existing files. The component that was initially examined was the Box as it was the most basic example of an individual component and was used to understand how the upper level design files work.

Each component has a MathCAD design file, a BetaMethod text file, and a Method text file. They can be found in folders named after each component which is located in the ADT Designs folder. The BetaMethod file is used for the beta server and contains sample values of input variables as well as other information that is grabbed to upload the designs on the beta server. The Method file is used for the same purpose but for the actual design server instead. Currently, only the full-scale AguaClara plant can be requested on the actual design server so these component Method files are not yet utilized on the public server. The goal of the modular designs challenge is to eventually provide component designs on the actual design server. The beta server is used for testing updates before finally committing them to the public server. To determine and change what components can be requested on each server, there is a BetaMethod and a Method text file in the ADT Designs folder which control which components can be accessed on the server.

## Sedimentation Tank & Testing

The first component worked on was the sedimentation tank. Building on the existing SedTank MathCAD file, the first step was to replace the old sedimentation tank display variables with updated ones which were found in the EtFlocSedFi file. The file references were also updated to reflect changes and new user input variables were defined. These changes in the input variables were also reflected in the SedTank BetaMethod as old input variables were replaced with new ones and their default values were revised.

$T_{SedWall} :=$	outer wall thickness
$T_{SedDividingWall} :=$	dividing wall thickness
$T_{SedChannelWall} :=$	channel wall thickness
$H_{SedSlopesToFlocWeirMin} :=$	floc blanket depth (slope peak to floc weir)
$T_{FiBoxWall} := 15\text{cm}$	Note: arbitrary input, needed for ACoutput to work

Figure 1: New user input variables for the Sedimentation Tank

To test the code initially, the user input variables were filled in with default values and the AC.SedTank variable was checked to see if it was undefined. If it was undefined, the references and inputs were copied and pasted into the SedimentationTankAC file (found in AutoCAD Scripts) to debug and try to figure out what variables were undefined, finding the cause of the problem. Once it seemed to be working in the MathCAD files, the modular code was committed and the overall BetaMethod was updated to include the Sedimentation Tank and it could now be requested on the beta server.

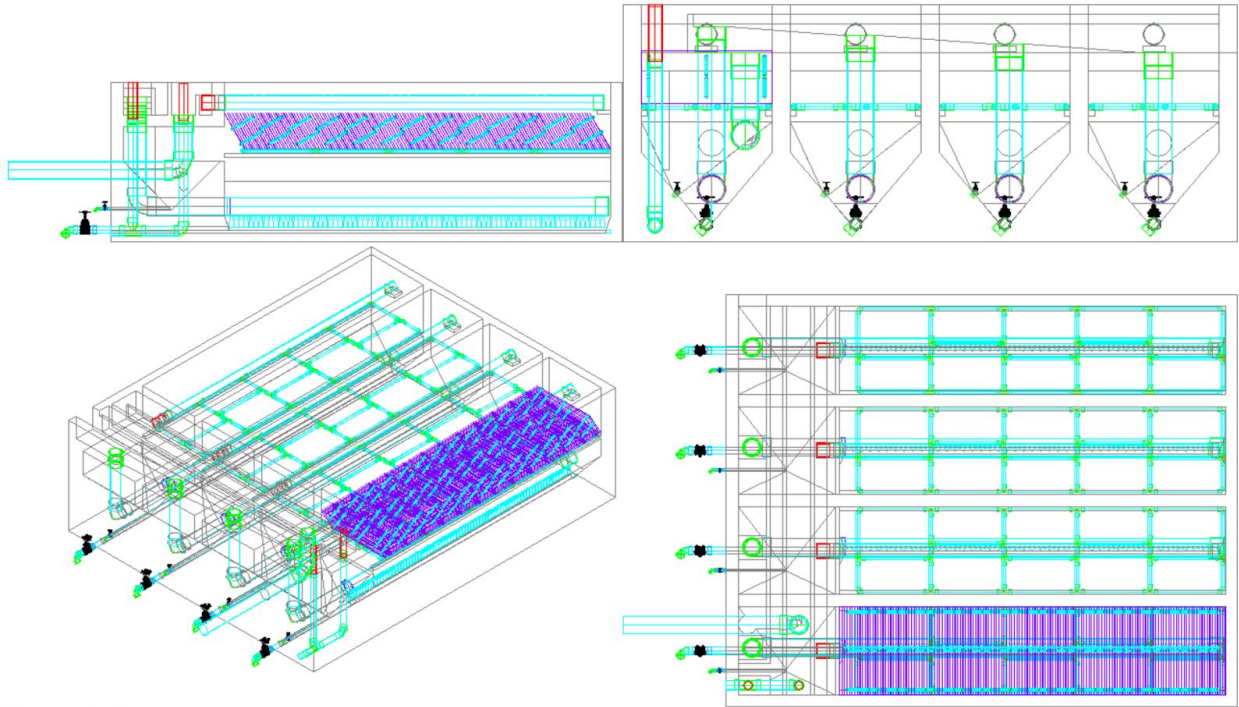


Figure 2: 20 Lps sedimentation tank from beta server

<http://designbeta.cee.cornell.edu/Designs/SedTank/7239/20Lps/Index.html>

There were also three available files for download:

- [About.html](#)
- [Sedimentation Tank.dwg](#)
- [Sedimentation Tank.txt](#)

The html file contains all the input and display variables, the dwg file contains the AutoCAD drawing, and the txt file contains the MathCAD output that can be copied and pasted into AutoCAD to draw the component.

It did not work initially on the beta server because Monroe had changed the way the server grabbed information so additional lines of text were needed in the BetaMethod in order for it to work properly. These four lines are now necessary at the end of every method file in order for the information to be properly grabbed and displayed on the server:

```
Publish files
.html .pdf .dwg .doc .txt
Display images
.png
```

## Beta Server & Design Tool

The beta server is a continuously updating server, using the most recently committed version of the code at all times. Anytime a design team member commits a file, the beta server restarts and begins to redraw and remake all the designs from the beginning. The content that is on the beta server is controlled by the BetaMethod files in each component folder and the one in the ADT Designs folder. The BetaMethod located in the ADT Designs folder controls which components are drawn on the beta server as well as which version of the code is used and the order they are drawn. If you want a certain component to be drawn first, you just change the number ordering in the right column and re-sort the numbers (done in Excel). This can be useful as some designs take longer to draw on the beta server than others and the process can be time consuming. The BetaMethod files located in each component folder control the input variables, their default values, what output variables are grabbed, and what kind of files are outputted.

When committing a component designs file, it is important to remember to leave input variables undefined for it to work properly on the beta server. They need to be left blank so users can input their own values for these variables on the server.

## LFOM (Linear Flow Orifice Meter)

The next component worked on was the LFOM. After examination, it was realized that the existing LFOMAC file was outdated. Since that file was last updated, many changes had been made to the plant design such as the location of the entrance tank in which the LFOM is located as well as input variables for functions such as PS.default. There were two options moving forward.

The first option was to just reference the EntranceTankCompactAC file in the LFOM modular design file instead of the LFOMAC. While the LFOM was drawn with the Rapid Mix as a part of the entrance tank, the LFOM/Rapid Mix drawing script could be separated into two. Then, just the LFOM portion could be referenced in the modular design file. However, when this was implemented and the EntranceTankCompactAC was referenced in the LFOM modular design file, the LFOM pipe was not drawn correctly. Because only the LFOM portion of the code was extracted from the EntranceTankCompactAC file, the rest of the plant (such as the walls) was not being drawn and therefore, the PipeSUBF function was not working correctly, hindering the rest of the code from running as desired. To fix this problem, the PipeSUBF function was

changed to the PipeF function. From this process, it was realized that the LFOM modular designs file could not just reference EntranceTankCompactAC file because of the discrepancy in code. Therefore, a new LFOMAC file that just drew the LFOM would need to be created and referenced instead.

A new LFOMAC file was created as the existing one was outdated. To create the drawing functions in the new LFOMAC file, the team examined existing drawing functions that were used in the EntranceTankCompactAC to draw the LFOM and based the LFOM code off them. However, after removing the sample input values from the LFOM file to prepare it for use on the beta server, it was realized that in addition to the Q.plant (flow rate) input, other inputs that were not directly related to the LFOM were needed for the functions to be defined and work correctly. These additional inputs were:

```

HL_FlocMax := 40cm
HSedSlopesToFlocWeirMin := 25cm
TFlocSlab := 15cm
TSedWall := 15cm
TSedDividingWall := 15cm
TSedChannelWall := 15cm
TFlocWall := 15cm
TFlocDividingWall := 15cm

```

Figure 3: LFOM potential input variables

The reason for this was discovered to be because the flocculator origin was used to calculate the entrance tank origin, which was used in the functions for the LFOM.

To alleviate this issue the MathCAD code was revised. Instead of basing it completely off the EntranceTankAC file, an origin of (0, 0, 0) was picked for simplicity. Analyzing the old LFOMAC file, some variables could no longer be found so the pipe drawing method was edited and re-written.

$$\begin{aligned}
 L_{RMVertPipe} &:= L_{RMVertPipeTop} + L_{LfomPipe} + L_{RMVertPipeBottom} \\
 RMVertPipe &:= PipeF \left[ \left( \begin{array}{c} LFOM_{Origin_0} \\ LFOM_{Origin_1} \\ LFOM_{Origin_2} - L_{RMVertPipeBottom} \end{array} \right), L_{RMVertPipe}, \left( \begin{array}{c} 0 \\ 90 \\ 0 \end{array} \right) deg, ND_{RMPipe}, PS_{Default} \right]
 \end{aligned}$$

Figure 4: Original LFOM MathCAD code

$$EtRMPipe := PipeF \left[ LFOM_{Origin}, H_{LfomOrifices_{last}}(H_{LfomOrifices}) + H_{PlantFreeboard}, \begin{pmatrix} 0 \\ 90 \end{pmatrix} deg, ND_{RMPipe}, SDR_{41} \right]$$

Figure 5: Updated LFOM MathCAD code

However, the orifices of the LFOM were very close to the bottom of the pipe:



Figure 6: AutoCAD output of LFOM

To take that into account, H.LfomFreefall was added for when the pipe is drawn, and also when the orifice heights are calculated.

$$EtRMPipe := PipeF \left[ LFOM_{Origin}, H_{LfomOrifices_{last}}(H_{LfomOrifices}) + H_{PlantFreeboard} + H_{LfomFreefall}, \begin{pmatrix} 0 \\ 90 \end{pmatrix} deg, ND_{RMPipe}, SDR_{41} \right]$$

```

LfomOrifices :=
| Local ← LfomFrz
| for h ∈ 0..last(H_LfomOrifices)
|   Local ← stack Local, CylinderF
|     [
|       LFOM_Origin_0
|       LFOM_Origin_1
|       LFOM_Origin_2 + H_LfomFreefall + H_LfomOrifices_h
|     ],
|     D_LfomOrifices / 2, OR(ND_RMPipe) + πc,
|     [
|       90deg + (asin(0.5 * S_LfomOrifices_Min / OD(ND_RMPipe)) + asin(D_LfomOrifices / OD(ND_RMPipe))) · (mod(h, 2) = 0)
|       0deg
|     ]
|     if N_LfomOrifices_h = 0
|   Local ← stack Local, ArrayLastPolar
|     [
|       LFOM_Origin_0
|       LFOM_Origin_1
|       LFOM_Origin_2 + H_LfomFreefall + H_LfomOrifices_h
|     ],
|     N_LfomOrifices_h · (N_LfomOrifices_h - 1) · (2 · asin(D_LfomOrifices / OD(ND_RMPipe)) + 2 · asin(S_LfomOrifices_Min / OD(ND_RMPipe)))
|     if N_LfomOrifices_h > 1
| return Local

```

Figure 7: Updated LFOM MathCAD code

However, although the LFOM can be drawn when manually outputting it in the file, it continued to fail on the beta server. Examining the server output more closely, it was realized that it

seemed to be grabbing the AI.LFOM variable, but not the AC.LFOM variable. Therefore, nothing was being drawn. This should be further investigated by future challenge team members.

## Flocculator & Entrance Tank

The final component worked on was the Flocculator. Similar to previous approaches, the Display Variables in the ADT Designs Flocculator file were updated to account for changes in the MathCAD code in the past few years. However, in contrast to previous component designs, it was decided that this design should include both the flocculator and entrance tank. Currently, the entrance tank and the flocculator are combined, so in order to draw them separately the actual algorithm would need to be changed. After discussing with the team, it was concluded that for now, the best plan of action would be to draw the flocculator and entrance tank as one unit. Perhaps as a future challenge, the algorithm could be revised so that there is the option for the flocculator and entrance tank to be drawn separately.

Therefore, the entrance tank display variables were included as well. The Flocculator reference was replaced with the FlocculatorEntranceTank reference and the variables AI.Et and AC.EntranceTank were added to the beta method (those for the flocculator were already in the component file). These User Input variables were added to the Flocculator/EntranceTank file:

```
HLFlocMax := 40cm  
HSedSlopesToFlocWeirMin := 25cm  
TFlocSlab := 15cm  
TSedWall := 15cm  
TSedDividingWall := 15cm  
TSedChannelWall := 15cm  
TFlocWall := 15cm  
TFlocDividingWall := 15cm
```

Figure 8: Additional user input variables

And these User Input and Expert Input variables were deleted:



$L_{Sed} := \blacksquare$	$CP_{Floc} := CP_{FlocUser}$
$H_{Sed} := \blacksquare$	$ED_{Floc} := ED_{FlocUser}$
$CP_{FlocUser1} := \blacksquare$	$HW_{Sed} := HW_{FlocEnd1}$
$ED_{FlocUser} := \blacksquare$	$Z_{SedWeirExit} := HW_{FlocEnd2}$
$HW_{FlocEnd1} := \blacksquare$	$W_{SedInletChannelPreWeir} := 0.5m$
$HW_{FlocEnd2} := \blacksquare$	$Z_{SedChannelBottom} := Z_{WaterExit}$
$Z_{WaterExit} := \blacksquare$	$ND_{RMPipe} := ND_{WaterStart}$
$ND_{WaterStart} := \blacksquare$	$ND_{SedManifold} := 6in$
	$N_{SedTanks} := 1$
$CP_{FlocUser} := CP_{FlocUser1} \cdot \frac{m^3}{s}$	$W_{Sed} := 2m$

Figure 9: Deleted input and display variables

They did not seem to make a difference whether they were present or not as they seemed to be outdated variables. The Flocculator/Entrance Tank BetaMethod was updated to reflect these changes in User Inputs.

To draw the Flocculator/Entrance Tank, these input variables are required:

$T_{SedWall} := 0.15m$
$T_{SedDividingWall} := 0.15m$
$T_{SedChannelWall} := 0.15m$

Figure 10: Input variables

However, these wall thicknesses involved the sedimentation tank which were not part of the modular design for the Flocculator/Entrance Tank unit. Two potential options were to either define them as part of Expert Inputs (if the wall thickness does not really matter in the actual design calculations as they seem to be defaulted at 0.15 m no matter the flow rate), or test and see what variables in the flocculator they do affect and try to make those the user inputs instead. After discussing with the team, it was decided that assuming design requests will be reasonable, it is acceptable to define them in the file as Expert Inputs. Therefore, the flocculator-entrance tank unit was successfully integrated into the beta server.

## Future Goals

The goal of future Modular Designs teams are to continue to create and update the component designs so that they are up-to-date with current AguaClara technology and plants. The development of modular designs is still in progress and is currently being tested on the beta server. The final goal is that they will be fully integrated into the actual design server. Therefore, clients and other people who are interested in AguaClara technology can easily request plant component designs in addition to the already available full-scale plant design. This semester, progress has been made on the sedimentation tank, LFOM (which does not yet work on the beta server and needs to be further investigated), and flocculator-entrance tank unit but they have not been uploaded onto the actual server yet. The amount of information to be provided has also yet to be fully decided and both should be discussed with Monroe.

Currently the beta server works when selecting sample plant flow rate designs, but when trying to input your own user inputs to request a plant design, a “500 Internal Server Error” appears. This is not only an issue for the modular designs, but for the full plant design as well. However, the function does work on the regular AguaClara design server so it appears to be a problem with the beta server specifically. Monroe and Cameron had identified the problem to involve the NSlookup and ipconfig commands but the error still has not been solved and needs to be investigated further. Once this function is working on the beta server, the min and max constraints should be examined, what happens when an extremely small or large value is entered?