

# Progress Report: Design Updates

Fletcher Chapin ftc22 and Nandini Nayar nn269

December 9, 2016

## 1 Introduction

AguaClara is a project team at Cornell University which researches and designs water treatment technologies for water treatment plants in Honduras and India. The conventional water treatment technologies used in the United States are often not feasible globally, so it's necessary for AguaClara to design new technologies so that as many people as possible have access to reliable, clean water. The team is made up of smaller subteams which each work on a specific part of the project, such as the Ram Pump, EStARS, and CDC teams.

The Design Team serves as an intermediary between the research teams and the outside world. By taking the results of AguaClara's research and translating it into Mathcad code that draws designs of the treatment plants which are published on the AguaClara server, the Design Team allows field engineers and other interested parties access to the water treatment technologies researched by AguaClara. This semester, the team is tackling a number of challenges, including automating the creation of section cuts using Mathcad, adding a backwash lagoon to the design plans, and resolving a variety of bugs and issues with the current code.

## 2 Section Cuts

To make it easier for field engineers to execute the AguaClara designs, section cuts of key components of the treatment plant are necessary. Currently, creating section cuts in AutoCAD using Mathcad code is not feasible. The Design Team will work with a team at AutoDesk to develop this capability in AutoCAD. The main focus of the Design Team's work will be communicating with the AutoDesk team what features the AguaClara team as whole would like to see implemented, progress meetings with AutoDesk team, and providing feedback once the AutoDesk team begins to produce results. The communication phase will involve sending AutoDesk a 3D model of a plant along with the current design tool output and the ideal output. Then AutoDesk will work on optimizing the tool so that it produces the ideal output. Finally, the Design Team will test it on the server to ensure it works.

### 2.a Previous Work

Meghan previously worked on Section Cuts to understand why it is difficult to implement code for section cuts in MathCAD and convert it to command line code. The report that summarizes previous work mentions that the gap in implementation is, in part, because AutoCAD requires values to be chosen through a dialogue box, which cannot be achieved through the command line. A function was created to draw a section plane, see Figure 2, and then create a 2D view from the perspective of this section plane, a section cut. As can be seen in Figure 3, the color of components and the locations of the section cuts are marked on the final product, but there was also difficulty obtaining these results using the old section cut command.

### 2.b Current Work

The team started by making a conference call with Kean and Jeremy, the AutoDesk employees who volunteered to help us with this challenge. The conference call introduced the new team members to the

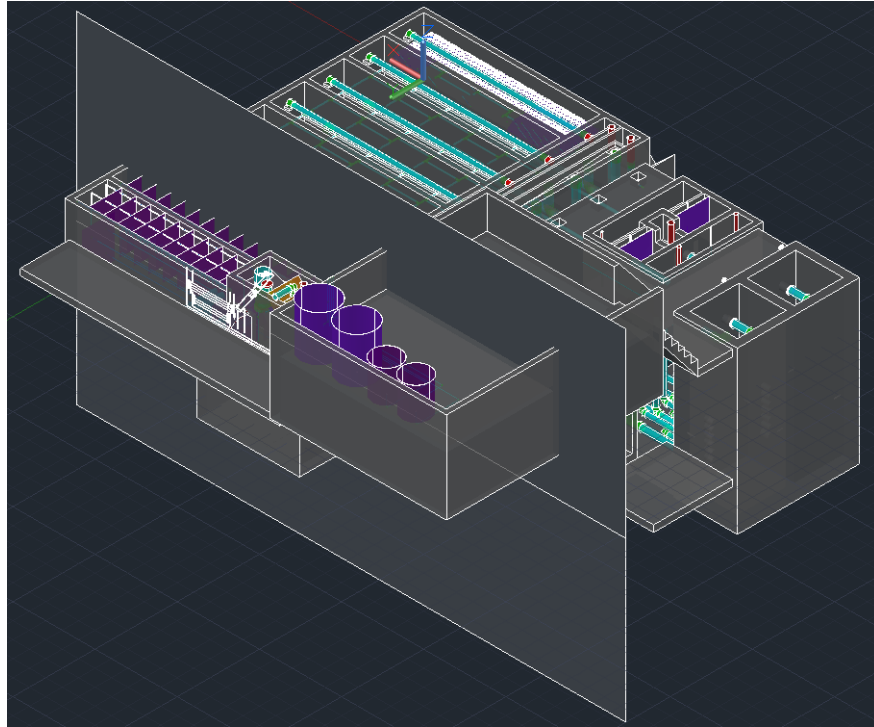


Figure 1: An example of a section plane that could be used to make a section cut of the plant

challenges and served as a refresher for team members who've worked on it in the past. The main points from the conference call were that the Design Team should provide AutoDesk with sample section cuts to serve as an ideal final product and an explanation of why this final product is unattainable through the command line, so that AutoDesk would know what features are needed to create a sufficient section cut command line tool.

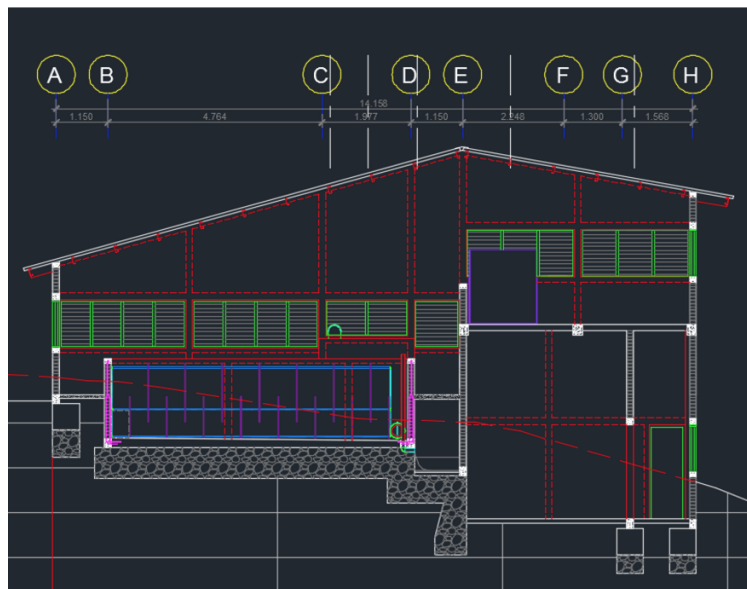


Figure 2: Honduras flocculator section cut with colors maintained and dimensions labeled

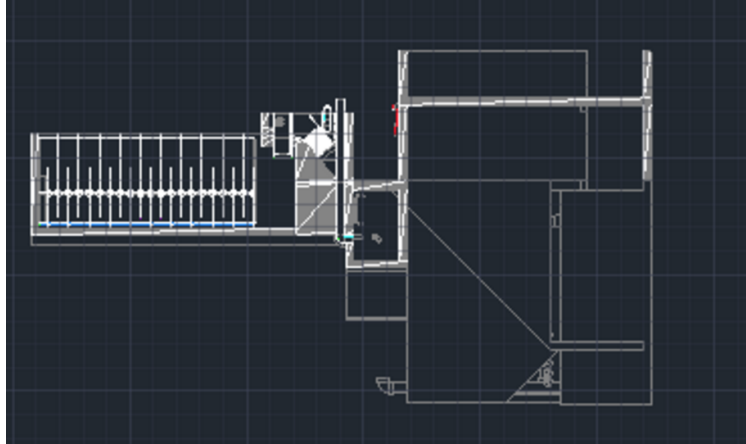


Figure 3: Design Team flocculator section cut created from dialogue box

### 2.b.1 Creating Section Cuts in AutoCAD

First, the Honduras engineers shared with us copies of the AutoCAD drawings that were made manually (see Figure 3). The team sent these drawings to AutoDesk as an example of an ideal product then tried to replicate these drawings using the AutoCAD tool. A couple of important features of the sections from Honduras are the layers maintaining their color and the labeling of dimensions. Replicating these features in AutoCAD was not as simple as was hoped because there are a multitude of different settings which can be selected before sectioning.

Creating the section plane is a simple step using the SectionPlane tool which can be easily automated, but in order to maintain color of the layers and ensure the desired results, a number of settings must be correct in the dialogue box while using the section tool. These settings are remove hidden lines, but maintain background lines, and changing Layer to \*LayerByObject\* intersection fill and intersection boundary. Unfortunately, even with these settings the white boundary lines cover up most of the color so that it's difficult to differentiate between layers.

### 2.b.2 Sending the Final Product to AutoDesk

After practicing with the tool and becoming proficient, the team communicated with AutoDesk to ask what was needed in order to move forward with the project. Kean, the AutoDesk lead on the project, asked for a single .dwg file that contained a 3D model of the plant, section objects for the various views you want to generate (one should be enough as an example), 2D section geometry created by the existing tool for the section objects, and the ideal 2D section geometry the team would like the new tool to create. The ideal final 2D sections that the team had obtained from Honduras were not accompanied by a 3D model, so the team asked Carlos, the architect in Honduras, if he could create another ideal section cut in the same file as a 3D model of the plant.

After receiving the finished product from Carlos, the team used the old command line section plane to block tool Kean had developed. In order to use the tool, the team first made sure the AutoCAD module was downloaded, then used the "netload" command to load the module. Once the module was loaded, the "SPTB" command was entered into the command line and the same section object that Carlos used for his section cut was selected so that a section cut using the current tool was made. As shown in Figure 6, the current automated section cut tool produced poor results, not maintaining color or displaying background lines as desired in the ideal final product.

Once the section cut produced by the current automated tool was produced, the entire .dwg file was sent to AutoDesk for them to try to replicate the manual results with a command line tool.

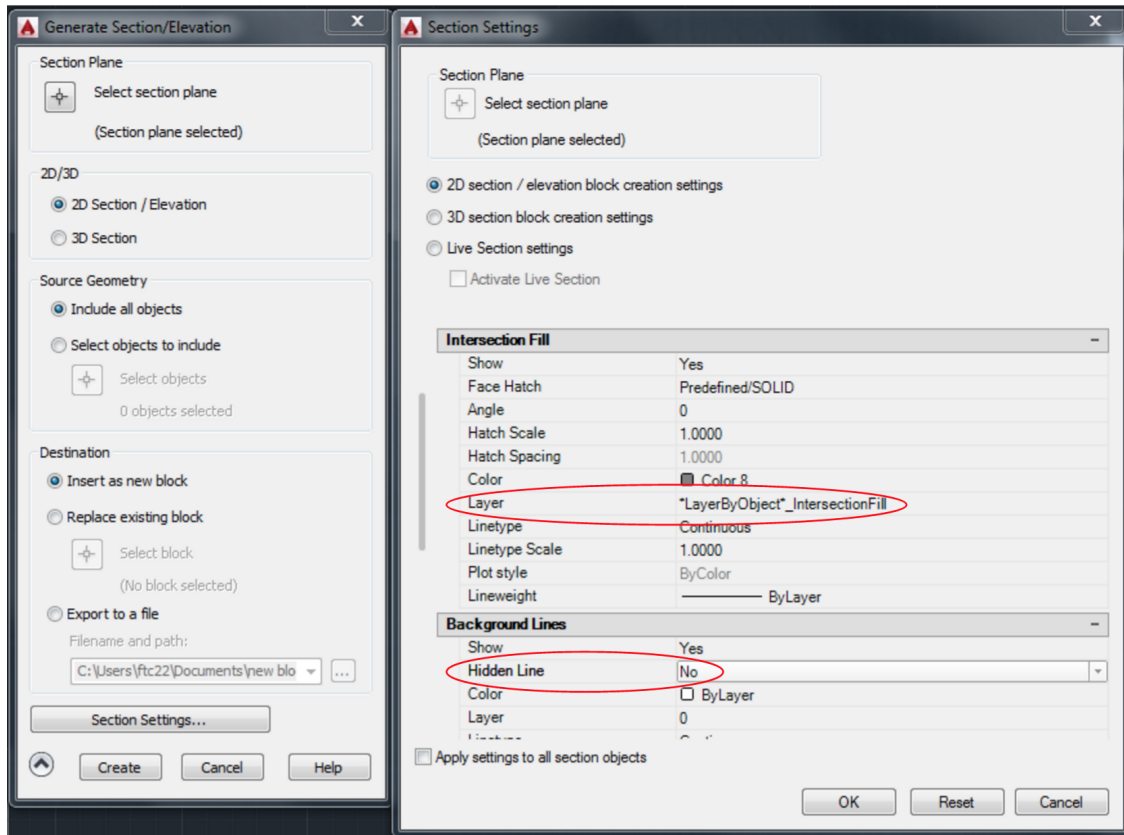


Figure 4: Section cut dialogue box with important commands highlighted

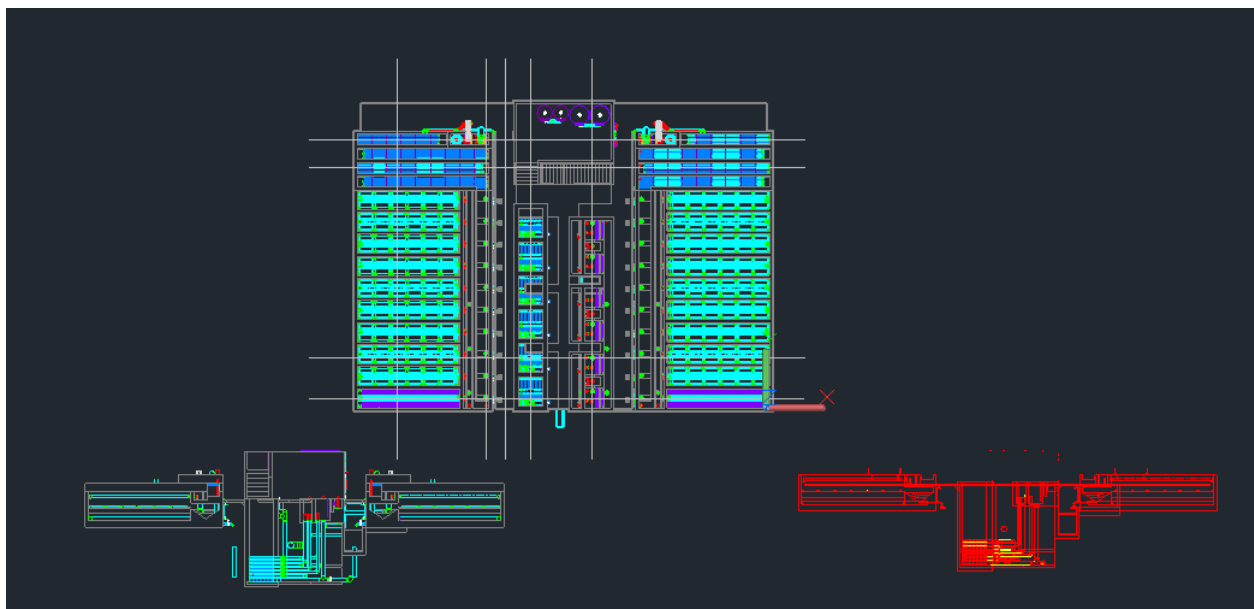


Figure 5: The ideal section cut final product (left) and the final product using the current automated tool (right); note that all the lines created by the current tool are red and no background lines are displayed

### 3 Flocculator Max Head Loss

The flocculator code had a maximum head loss constraint as an input that's used when designing the flocculator. The issue was that this code was producing outputs which had a larger head loss than the maximum head loss that was inputted by the user. In order to fix this the team took a look at the flocculator code.

First, calculations for head loss were scanned and all the variables contributing to the calculation were documented and examined. Each variable was then looked up individually to look for opportunities to tweak values. The equation for HL.Floc, the actual head loss in the designed plant, is shown in figure 7. The variables used to calculate HL.Floc were: K.FlocBaffle, the minor loss coefficient, V.Floc, the fluid velocity in the flocculator, and N.FlocExpansions, the number of flocculator expansions. K.FlocBaffle was an expert input and V.Floc was dependent on Q.plant, the flow rate and the cross sectional area of the flocculator, so neither were affected by the constraint. Initially, no links between the input variable for max head loss and the measurement calculations for the flocculator were found, suggesting that the max head loss constraint was not being implemented in the calculations. However, upon closer examination the number of flocculator expansions was dependent on the geometry of the baffles, which was dependent on the maximum head loss parameter, and therefore the head loss constraint was having an effect. Even with this effect, the final head loss still exceeded the maximum head loss parameter regardless of the inputs and constraints used.

$$HL_{FlocMax} := 40\text{cm}$$

$$HL_{Floc} := K_{FlocBaffle} \cdot \frac{V_{Floc}^2}{2g} \cdot N_{FlocExpansions} = 0.422\text{m}$$

Figure 6: The equation that is used to derive the head loss in the sedimentation tank as well as the constraint, which as shown is not being followed properly

After discussing this with Monroe, it was decided that this was acceptable, since certain calculations involving head loss require whole numbers rather than decimal numbers. The whole numbers are required because since each expansion is the same size, there cannot be a decimal number of spaces. This rounding off inevitably leads to a slightly larger head loss value, but within a reasonable margin. In summary, the challenge has been completed, and there is no more work required on the flocculator head loss.

### 4 Plate Settler Support in the Sedimentation Tank

A new challenge was taken up that involved making changes to the MathCAD code for the sedimentation tank. The original design for the plate settler in the sedimentation tank involved numerous elbows, tees, and pipes (see Figure 8 below) and so required extremely lengthy code to make each of these smaller pipes and the elbows connecting them separately.

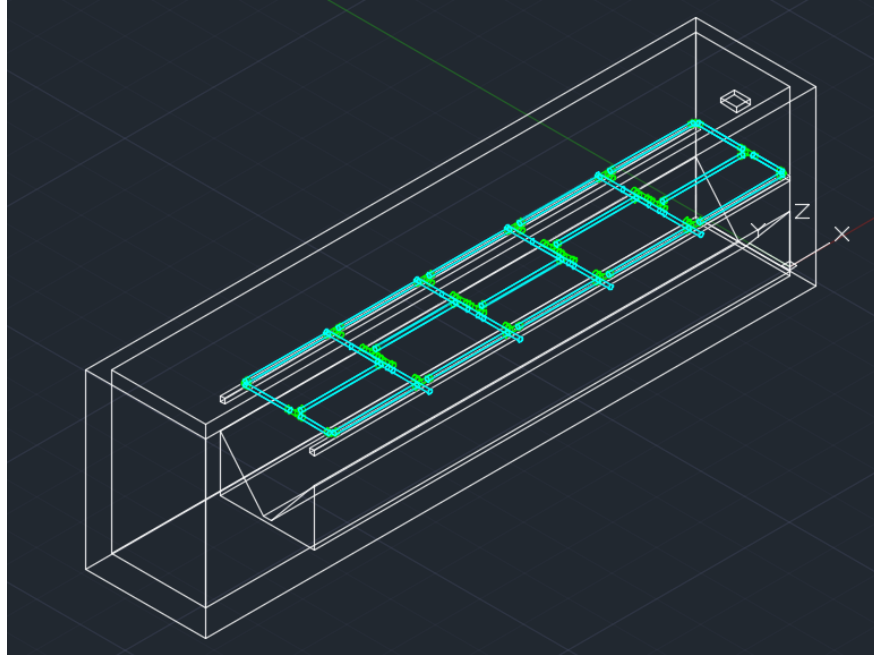


Figure 7: Original plate settler support design in the sedimentation tank, which included a number of tees and elbows

This design was replaced with a simplified design composed of 6 parallel pipes running across the width of the sedimentation tank, with a single central length-wise pipe placed over these cross pipes (see Figure 9 below).

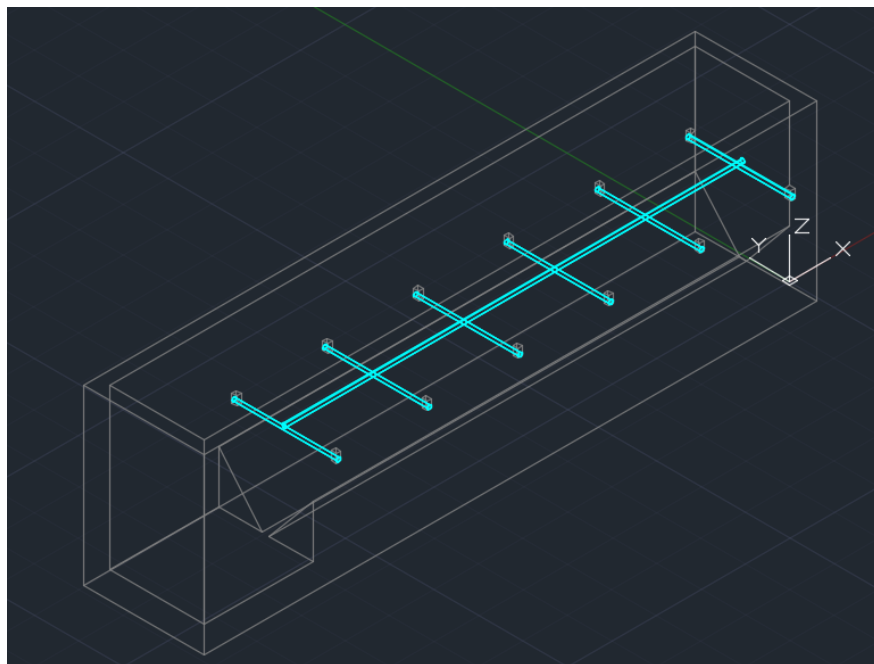


Figure 8: Simplified plate settler support design in the sedimentation tank without tees or elbows and consists of Only 7 Pipes

The purpose of this change in design was to simplify the code by reducing the number of components

that are drawn, and also to simplify the construction of the plate settler support. First, the current design was examined and its code and variables were documented in order to understand the origin points, length, width etc. of the components to be made. Then, 12 notches were drawn in the walls of the Sedimentation Tank. These were made wide enough so the cross pipes can just fit into them, and their height was made double the diameter of the cross pipes, so that they were level with the pipe on top. The dimensions of the notches with respect to the cross pipe can be seen in Figure 10 below.

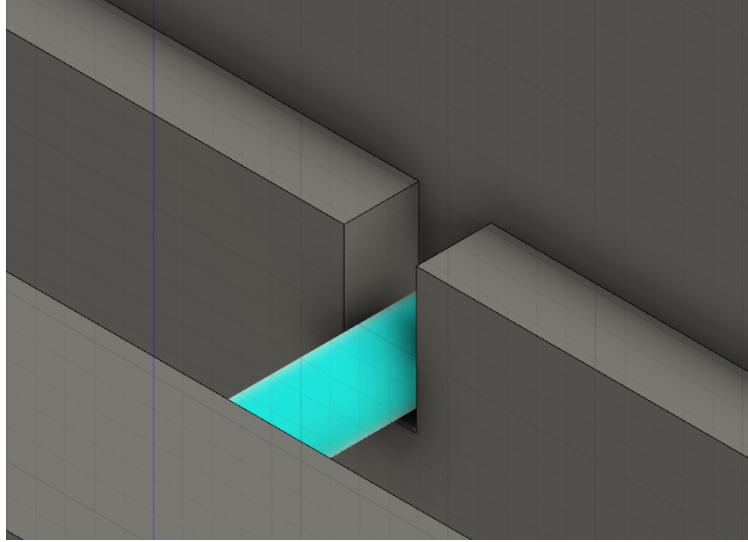


Figure 9: An individual notch from a Fusion 360 drawing of the sedimentation tank

The notches were made using a for loop and the origin points of the notch were changed within each run of the loop. Each notch was made as a box which was then subtracted from the walls of the sedimentation tank. The code for the notches can be seen in Figure 11 below.

```

Notches :=
  NotchStack ← BoxF PlantOrigin +  $\begin{pmatrix} 0m \\ T_{ConcreteMin} \\ Z_{SedLamellaSupportLedge} \end{pmatrix}$ , PlantOrigin +  $\begin{pmatrix} -OD(ND_{SedPlateFrame}) \\ 0m \\ Z_{SedLamellaSupportLedge} - 2OD(ND_{SedPlateFrame}) \end{pmatrix}$ ,  $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$  deg
  for i ∈ 1..5
    Notch1 ← BoxF PlantOrigin +  $\begin{pmatrix} 0m - iB_{SedPlateFrameCross} \\ T_{ConcreteMin} \\ Z_{SedLamellaSupportLedge} \end{pmatrix}$ , PlantOrigin +  $\begin{pmatrix} -iB_{SedPlateFrameCross} - OD(ND_{SedPlateFrame}) \\ 0m \\ Z_{SedLamellaSupportLedge} - 2OD(ND_{SedPlateFrame}) \end{pmatrix}$ ,  $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$  deg
    NotchStack ← stack(NotchStack, Notch1)
  for i ∈ 0..5
    Notch2 ← BoxF PlantOrigin +  $\begin{pmatrix} 0m - iB_{SedPlateFrameCross} \\ W_{Sed} - T_{ConcreteMin} \\ Z_{SedLamellaSupportLedge} \end{pmatrix}$ , PlantOrigin +  $\begin{pmatrix} -iB_{SedPlateFrameCross} - OD(ND_{SedPlateFrame}) \\ W_{Sed} \\ Z_{SedLamellaSupportLedge} - 2OD(ND_{SedPlateFrame}) \end{pmatrix}$ ,  $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$  deg
    NotchStack ← stack(NotchStack, Notch2)
  return NotchStack

SubNotches := SubtractF("SedTankCon"), "DeleteMe"

```

Figure 10: MathCAD code for 2 rows of 6 notches each in sedimentation tank, with the origin being shifted by a scale of  $i$  within the for loop. First the notches are created and then subtracted from the lamella support ledge.

Then, pipes were drawn and modified so that they neatly fit into the notches. The code for the six parallel

pipes or the cross pipes was made using a for loop, where the origin for each cross pipe was modified within the loop. This can be seen in Figure 12 below and the code for the Central Pipe can be seen in Figure 13 following it.

$$\begin{array}{l}
 \text{CrossPipes} := \left[ \begin{array}{l}
 \text{PipeStack} \leftarrow \text{PipeF} \left[ \begin{array}{c}
 -\text{OR}(\text{ND}_{\text{SedPlateFrame}}) \\
 0\text{m} \\
 Z_{\text{SedLamellaSupportLedge}} - 3\text{OR}(\text{ND}_{\text{SedPlateFrame}})
 \end{array} \right], \text{L}_{\text{CrossPipe}}, \begin{pmatrix} -90 \\ 0 \end{pmatrix} \text{deg}, \text{ND}_{\text{SedPlateFrame}} \cdot \text{PS}_{\text{SedPlateFrame}} \\
 \text{for } i \in 1..5 \\
 \quad \text{Pipe} \leftarrow \text{PipeF} \left[ \begin{array}{c}
 -\text{OR}(\text{ND}_{\text{SedPlateFrame}}) - i \cdot \text{B}_{\text{SedPlateFrameCross}} \\
 0\text{m} \\
 Z_{\text{SedLamellaSupportLedge}} - 3\text{OR}(\text{ND}_{\text{SedPlateFrame}})
 \end{array} \right], \text{L}_{\text{CrossPipe}}, \begin{pmatrix} -90 \\ 0 \end{pmatrix} \text{deg}, \text{ND}_{\text{SedPlateFrame}} \cdot \text{PS}_{\text{SedPlateFrame}} \\
 \quad \text{PipeStack} \leftarrow \text{stack}(\text{PipeStack}, \text{Pipe}) \\
 \text{PipeStack} \leftarrow \text{stack}(\text{PipeStack}, \text{sp}) \\
 \text{return PipeStack}
 \end{array} \right]
 \end{array}$$

Figure 11: MathCAD code for 6 cross pipes across the width of the sedimentation tank with the coordinates of the origin shifted by a scale of  $i$  within each loop

$$\text{CentralPipe} := \text{PipeF} \left[ \begin{array}{c}
 0\text{m} \\
 \frac{W_{\text{Sed}}}{2} \\
 Z_{\text{SedLamellaSupportLedge}} - \text{OR}(\text{ND}_{\text{SedPlateFrame}})
 \end{array} \right], 5 \cdot \text{B}_{\text{SedPlateFrameCross}} + \text{OD}(\text{ND}_{\text{SedPlateFrame}}), \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{deg}, \text{ND}_{\text{SedPlateFrame}} \cdot \text{PS}_{\text{SedPlateFrame}}$$

Figure 12: MathCAD code for the central Pipe across the length of the sedimentation tank

Once these changes were shown to Monroe and Skyler, there were certain additions that were suggested by them. Bolts were added at the ends of the central pipe so the central pipe and cross pipes were secured together so that the central pipe would not roll from its position. The bolts were drawn by inserting the URL of an AutoCAD drawing of a bolt and modifying it until its orientation and dimensions matched the requirement. The code and picture of the bolts has been posted below in Figure 14 and 15 respectively.

$$\begin{array}{l}
 \text{CentralPipeBolt1} := \text{insertdwg} \left[ \text{SupportBoltURL}, \begin{array}{c}
 -1\text{in} \\
 \frac{W_{\text{Sed}}}{2} \\
 Z_{\text{SedLamellaSupportLedge}} - 4\text{in} + .002\text{m}
 \end{array} \right], \frac{1}{450}, \frac{1}{450}, 0\text{deg} \\
 \text{CentralPipeBolt2} := \text{insertdwg} \left[ \text{SupportBoltURL}, \begin{array}{c}
 -5 \cdot \text{B}_{\text{SedPlateFrameCross}} + -\text{OD}(\text{ND}_{\text{SedPlateFrame}}) + 1\text{in} \\
 \frac{W_{\text{Sed}}}{2} \\
 Z_{\text{SedLamellaSupportLedge}} - 4\text{in} + .002\text{m}
 \end{array} \right], \frac{1}{450}, \frac{1}{450}, 0\text{deg}
 \end{array}$$

Figure 13: MathCAD code for inserting the bolts that were uploaded to the wiki and modifying their size, origin, and orientation



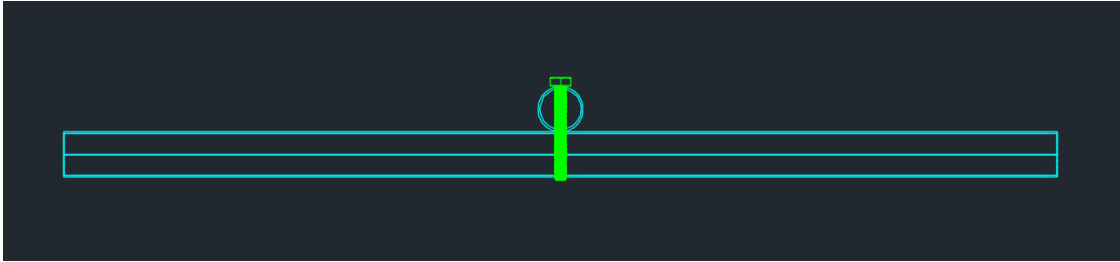


Figure 14: Image of the bolts at the ends of the central pipe

## 5 Future Work

Both the plate settler support challenge and flocculator max head loss challenge are complete, so no future work is necessary. For the section cuts challenge, for now the team will wait to hear from AutoDesk on their progress. AutoDesk will continue working on the tool and in future semesters the team will provide feedback to AutoDesk after testing it rigorously. Through this testing and feedback process, the tool will be perfected and then Mathcad code for automated section cuts can be written so that the design server displays section cuts for all the hydraulic components whenever a design is created.

## 6 Task Map

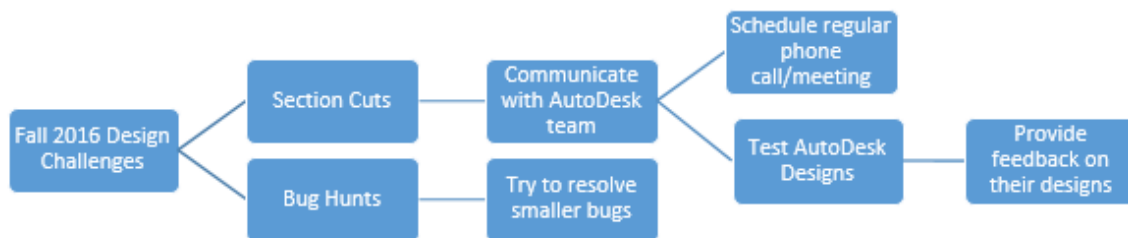


Figure 15: Task Map

### 6.a Task Map Details

- Section Cuts
  - work with AutoDesk to implement and publish section cuts which show important parts of the plant
  - through communication with AutoDesk get their solutions, test them, and return feedback
- Bug Hunts
  - with any free time outside of section cut work, work on some of the smaller bug hunts listed in Design Team challenges