Progress Report: Kevin

Kevin Juan kj89

December 10, 2016

1 Introduction

1.a AguaClara Overview

AguaClara is an engineering student project team at Cornell University started in 2005 with the goal of bringing sustainable water treatment to communities in Honduras and India. Through connections with local communities, AguaClara has been able to implement many economic, gravity-powered water treatment plants in Honduras and India. Members on the team carry out the research to improve the technologies and help generate designs for treatment plants and components. All of this is made possible through the communication between engineers at the water treatment sites and students on campus.

1.b Design Team Overview

The Design Team works on various projects which involve automating the design process for AguaClara's water treatment technology. Designs are coded using MathCAD and drawn out via AutoCAD and then published on the AguaClara server for partners to use. The overarching aim of this team is to allow users to enter various parameters about a plant and be presented with detailed AutoCAD designs, a materials list, and a report describing any information needed to build the water treatment plants or its components. Furthermore, the Design Team is responsible for updating the MathCAD scripts to reflect the most current designs and technologies developed by the research and fabrication teams and from feedback provided by on-site engineers.

2 New AutoCAD Designs

Because AguaClara is constantly making innovations to the water treatment methods and designs, new components and modifications are always being created. These novel creations or modifications are often developed in the lab, but often the field engineers in the communities also make changes to the plants. Since these changes can be consistent between plants of varying specifications, hand-drawing the change in AutoCAD is preferred so as to bypass the need to write Mathcad script for the component to vary by user inputs. To achieve this, communication between students in the lab and the field engineers is crucial. The field engineers provide us with sketches of the modifications made to a plant along with the dimensions to be reproduced in AutoCAD. Doing this allows AguaClara to have a detailed basis to go off of in case a request to fabricate that exact piece is requested in the future for the plant it originated from or for another plant that needs the modification.

2.a 1 L/s Entrance Tank Challenge Details

This past semester, AguaClara shipped its first 1 L/s water treatment system to be implemented in Honduras in Winter 2016-17. The 1 L/s plant has been a culmination of multiple semesters' worth of research, fabrication, and testing in order to determine the best operating design.

Prior to the shipment of the first 1 L/s plant, Monroe requested a rough AutoCAD drawing of a new entrance tank design to be used in the plant being shipped. His original request was to have a design of the entrance tank box with the proper dimensions for fabrication team member and adviser, Juan Guzman,

to fabricate. The first design proposal was to have a box that would be 35cm long and 18.5cm wide. Five PVC pieces in total would be cut out to make the box, each with 1/4 inch thickness. The two lengthwise pieces would be around 34cm in length, while one width piece would be 18.5cm and the other 23.6cm. One width piece was made longer than the other in order to accommodate for an 8.9cm x 8.9cm wooden support beam. The bottom piece would take a length of 34cm and a width of 17.24cm. The total depth of the box would then be 16cm. Another L-shaped piece of PVC would be attached to the box to support the other side of the wooden beam. In one side of the entrance tank, there was a 6in x 6in x 1in CDC float, which was attached to a lever arm, and would be 1cm from each of the 3 walls surrounding it. On the opposite side of the float was a 1in x 1in thick PVC rod for the lever arm to be attached to. This rod was placed so that the lever arm would be aligned centrally with the CDC float.

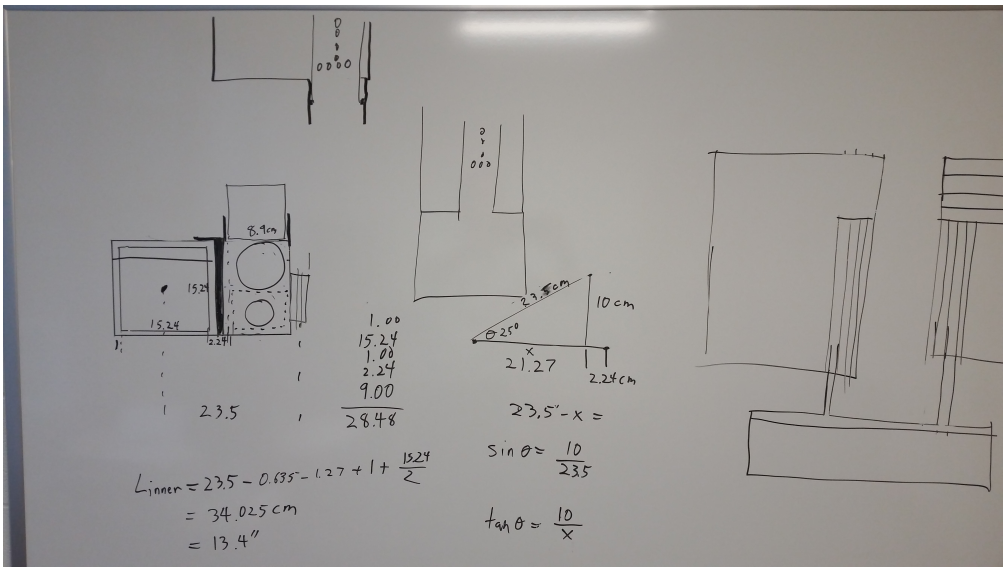

Figure 1: Monroe's original rough sketches for the design of the entrance tank. The far left drawing shows a proposed top view of the entrance tank.

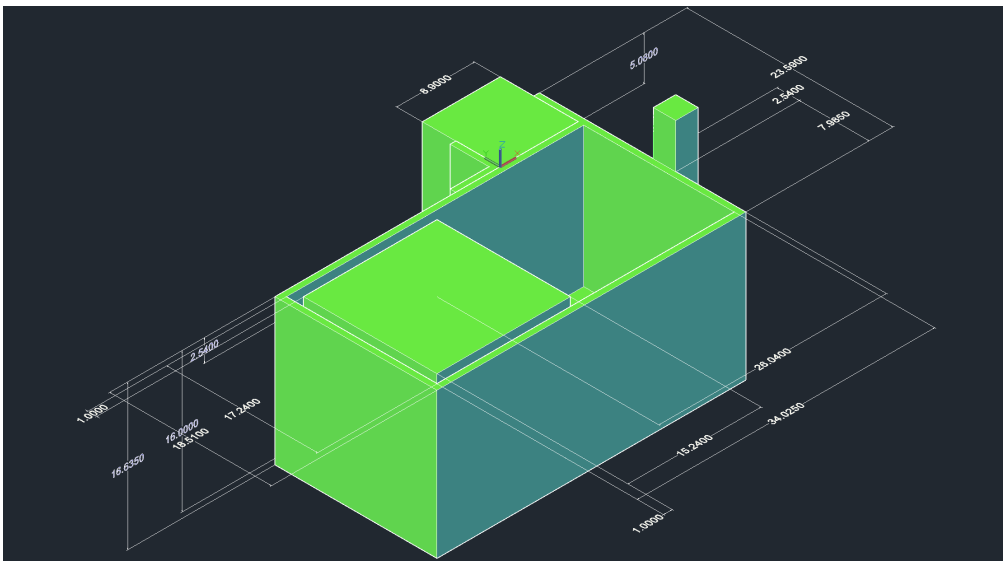

Figure 2: First iteration of the 1 L/s entrance tank with dimensions.

In the first draft of the design, cutouts in the bottom for two PVC pipe couplings were omitted. These openings would have diameters of 3in and 2in, and would serve as openings for an LFOM and trash rack, both of which were also omitted from the design. In addition, the lever arm and connection with the float were also left out.

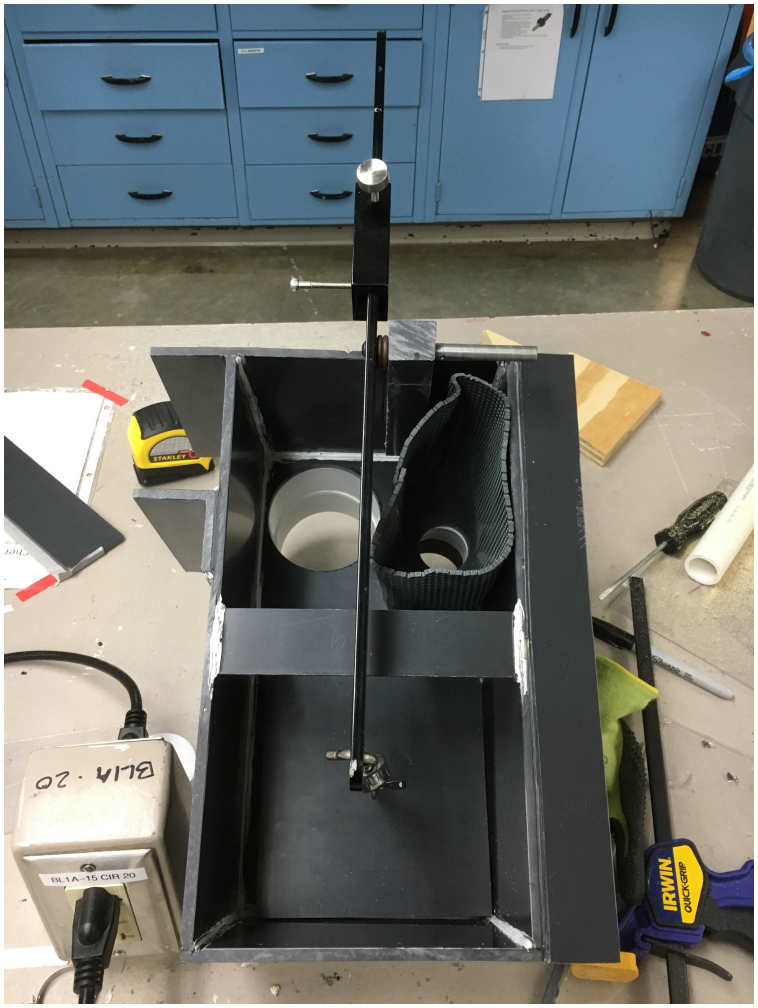

Figure 3: The actual entrance tank shipped to Honduras using the dimensions provided in the AutoCAD file, and fabricated by Juan Guzman.

From the first fabricated design shown in Figures 3 and 4, there were issues with the first set of proposed dimensions, which resulted in the PVC bar supporting the lever arm to be placed within the entrance tank rather than outside it as originally planned. In order to accommodate for this, the entire design had to be reduced in length by 1 inch.

In addition to the reduction in length, the second iteration of the design included details not originally accounted for in the first design. One addition was the couplings and cutouts for the PVC pipes leading to the LFOM and trash rack. These were added so that they were centered with the middle of the wooden support beam. The second iteration also accounted for the two PVC strips shown in Figure 3. The last major addition to the design was the lever arm and the change of placement of the CDC float. The new design showed a rough schematic of where the lever arm should be placed, and also showed how the connection between the lever arm and the float would be centered about the float.

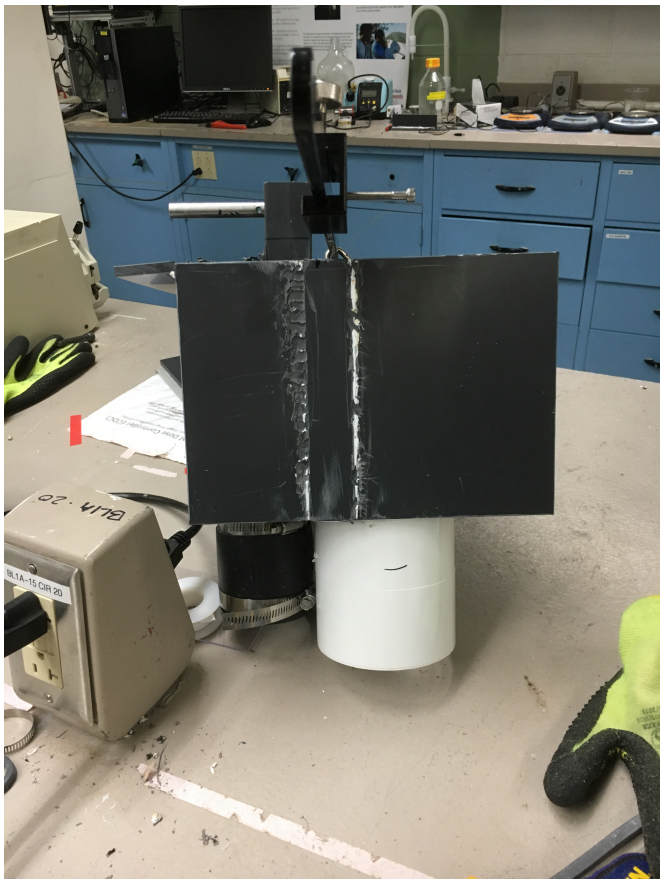

Figure 4: Another view of the entrance tank that shipped with the 1 L/s plant.

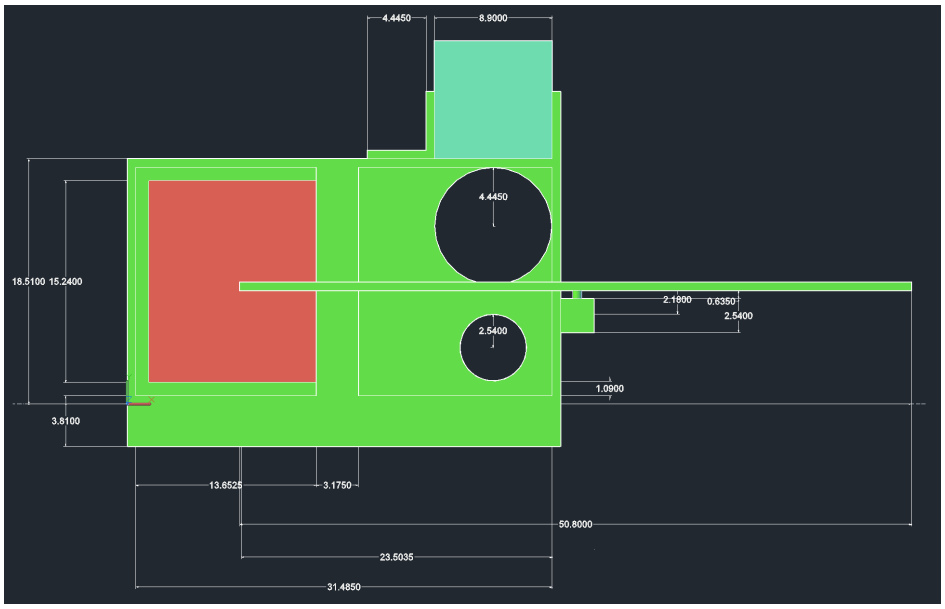

Figure 5: A top view of the second iteration of the 1 L/s plant entrance tank, showing the additions mentioned previously.

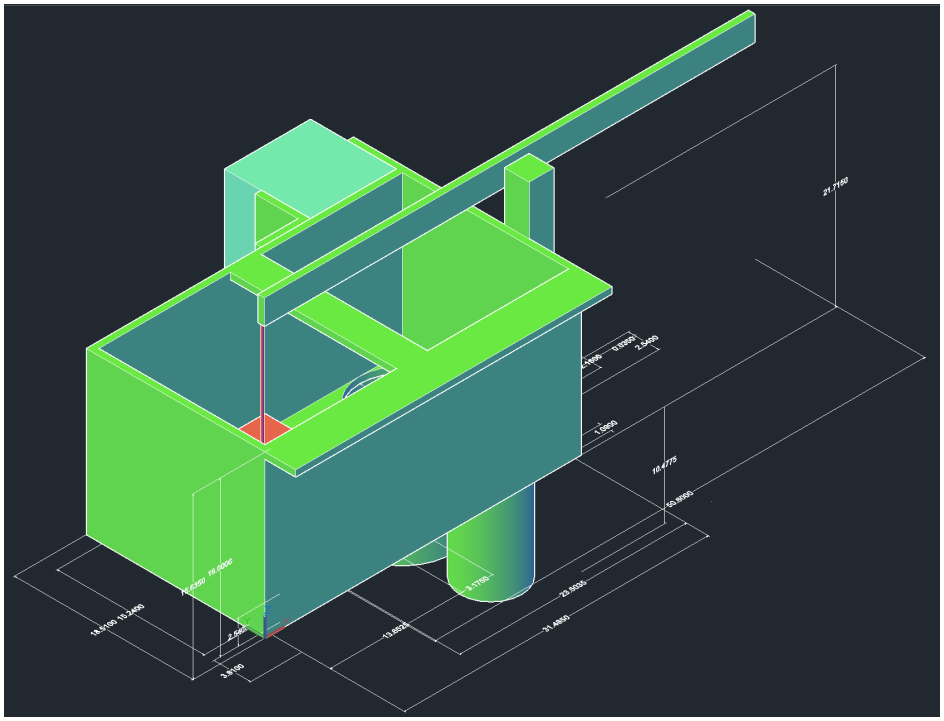

Figure 6: An oblique view of the second iteration of the 1 L/s plant entrance tank, showing the additions mentioned previously.

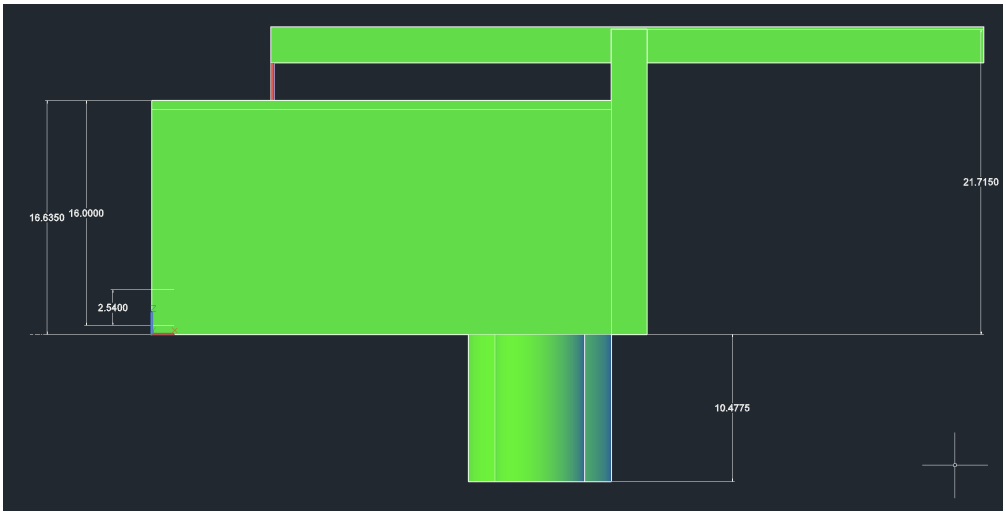

Figure 7: A side view of the second iteration of the 1 L/s plant entrance tank, showing the additions mentioned previously.

3 Section Cuts

A project that has been pursued on and off over the past few years by the Design Team has been to automate designs of section cuts of the treatment plants. These section cuts allow the field engineers to see inside the models generated by the automated design tool rather than just a model that only shows the exterior. Furthermore, 2D section cuts are more suited towards use in construction, and the process of obtaining

these view is extremely intensive in a computing sense, which can cost a significant amount of time if field engineers were to do them on their own. In addition, creating section cuts requires user navigation of a dialogue box to enter commands to allow it to generate a section cut. This is a clear issue because as it is, this functionality is not fully automated. In order to get it to be fully automated, AguaClara has worked with programmers at Autodesk in the past to get specialized files and code that do this process automatically. Though Autodesk has created code to automate section cuts, the state that it's currently in doesn't make it fully functional. Firstly, the given code only works with the 2013 version of AutoCAD, which needs to be updated to work with the versions being used by the computers in the lab. The script also doesn't preserve layer colors, plotting everything in red, which makes it difficult to distinguish between the different components. Furthermore, the final design doesn't include a legend to show users what view their looking at with regards to the overall plant.

The overall objective for this semester is to collaborate with Autodesk to fix these issues and implement the new additions. The Autodesk programmers will develop the code to reflect the changes, and will have the designers at AguaClara test the code to determine if it fits the team's needs. Through communication between Autodesk and AguaClara, feedback will be provided the programmers to get the script in its finished state.

3.a Challenge Details

The project official started this semester after communication was made in September with Kellan Hays, who was responsible for setting up pro bono projects at Autodesk. During the first meeting with Kellan, ground rules were set to make sure that the project would meet AguaClara's demands in addition to meeting the expectations of the Autodesk programmers in terms of commitment. Once that was established, contact was made with Kean Walmsley, a software architect at Autodesk who previously worked on section cuts with AguaClara, to establish a meeting date to outline the project goals and schedule.

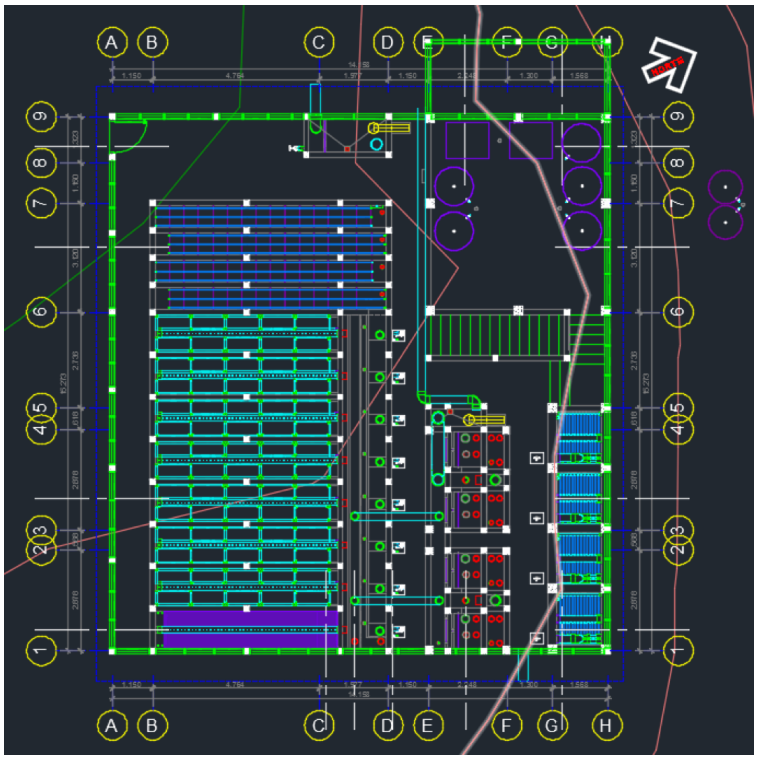

Figure 8: A top view of the plant in Las Vegas that details the locations of section cuts

During this meeting, the two teams decided that the first step to take was for the AguaClara designers to get acquainted with the built-in section cuts command by manually creating section cuts of a plant. To

understand the technical aspects of how section cuts work, it is recommended that Fletcher Chapin and Nandini Nayar's report be read. Once the team became familiar with the process of manually creating a section cut, Skyler Erickson and Ethan Keller, AguaClara engineers in Honduras, were contacted to obtain an AutoCAD file containing the locations of section cuts needed for construction. This file was sent to the Autodesk team to show what an ideal section cut file should look like .

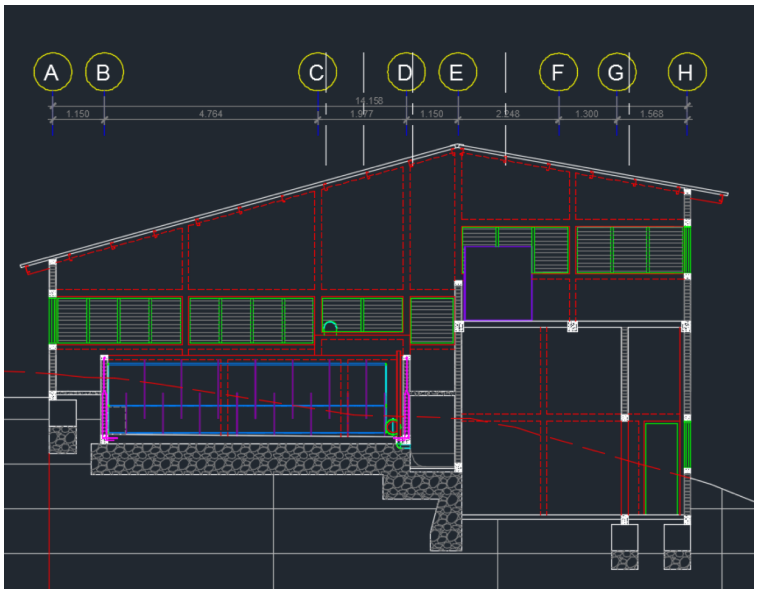

Figure 9: A side view of the plant in Las Vegas that details the locations of section cuts

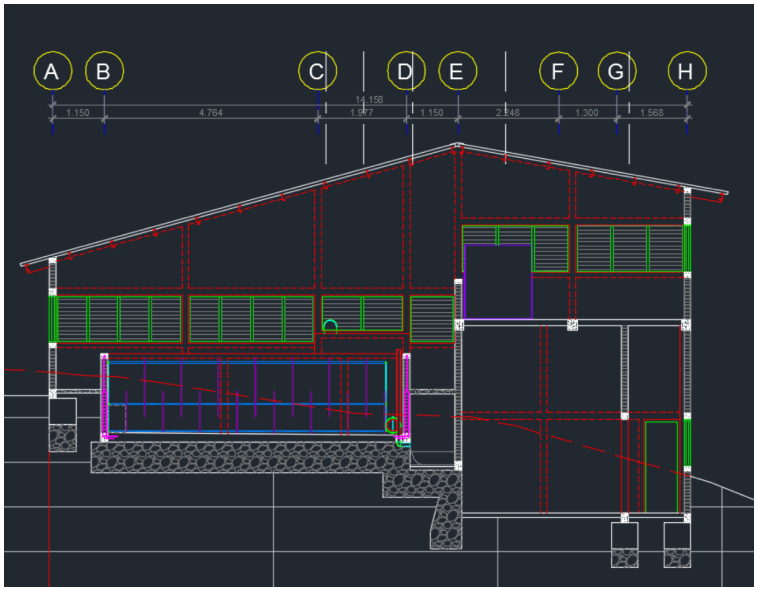

Figure 10: A side view of the plant in Las Vegas that details the locations of section cuts

4 LFOM Modular Code

One of the challenges completed last semester was the development of a fully functional script that generates a paper template for the LFOM. The paper template was designed for fabricators and engineers to conveniently

print out a 1:1 scaled template showing the orifice placement for the LFOM on conventional letter paper using a standard printer. The new template also reflected aesthetic and functionality changes to the LFOM orifices including centering of the cluster of orifices and spacing the orifices to make the LFOM structurally sound. Despite these changes shown on the template, they weren't reflected on the 3D model of the LFOM. Because of this, the objective of the challenge is to implement the changes to the orifice placement so that the model also shows the new design.

4.a Challenge Details

At the end of the Spring 2016 semester, Mathcad code was completed for a 2D template of the LFOM orifices for engineers to print out and use as a cut out template for drilling. The modifications included a change in the orifice spacing that made the center of the orifices line up with the halfway point between the two orifices in the row below. Before this implementation, the space between an orifice and the orifice to its bottom left would be noticeably smaller than the space between the same orifice and the one to its bottom right. Figure 12 illustrates this in the 3D model. This would create an issue with the structural integrity of the LFOM as one side of the orifice cut out would contain less material than the other, making it slightly weaker. It also served as an aesthetic fix to make it look even. Another major fix to the old LFOM design made on the template was moving the cluster of orifices that sit above the first line of orifices towards the center, shown in Figure 11 rather than the left as seen in Figure 12. This was mostly an aesthetic fix, and was intended to better reflect the actual orifice placement in real life.

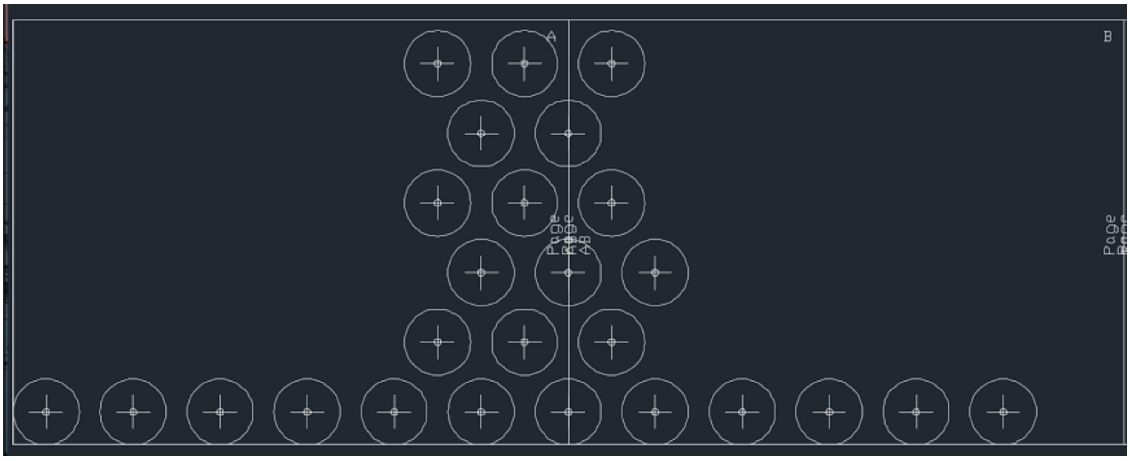

Figure 11: An LFOM orifice Template generated by the code implemented last spring

Despite the fact that both the template and the model for the LFOM were coded within the same Mathcad file, the orifice origins are generated by completely separate code. This meant that none of the changes made to the template were reflected on the model, which was left looking like Figure 12.

The first fix implemented into the model was the spacing issue between orifices on different rows. Within the LFOM code, there is a variable named LfomOrifices, which is defined by a loop dictated by the number of rows of orifices in the LFOM. The for-loop creates each row of orifices one at a time by creating the farthest left orifice first. The process starts by creating a cylinder, which is later subtracted as a group of cylinders from the pipe to get the holes, using the CylinderF function. The cylinders have a radius equal to half the diameter of an orifice and origins where the x and y coordinates are zero and the z coordinate corresponds to the height of the row above the bottom. The length of the orifices are slightly longer than the radius of the LFOM pipe to make sure that they'll be able to be subtracted out.

The issue with the spacing resided within the angle vector that determines the rotation of the cylinder to get it into place. The old code relied on two separate inverse sine functions. One function was the inverse sine of half the spacing between orifices over the outer diameter of the LFOM pipe, while the other was dependent on the diameter of the orifice over the outer diameter of the LFOM pipe. From basic trigonometric identities, it is known that adding the two inverse sine functions would not produce the same angle as the inverse sine

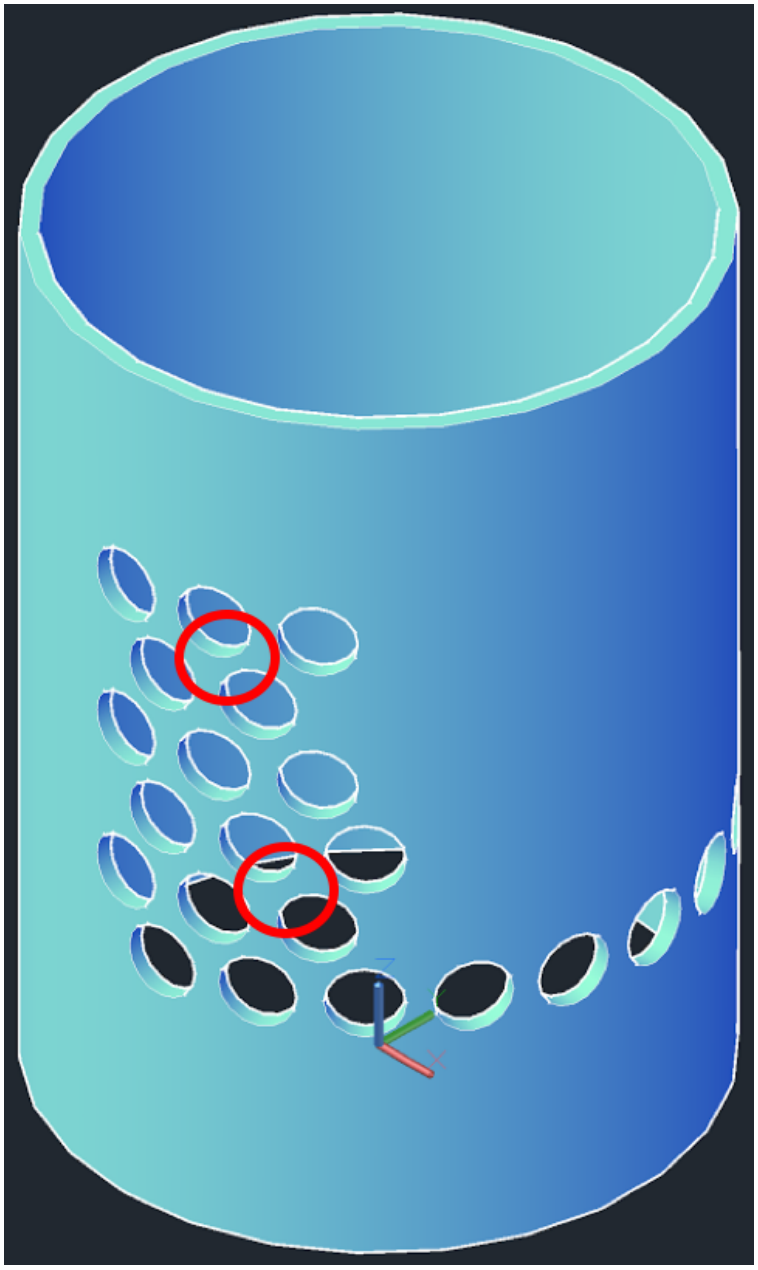

Figure 12: The LFOM model created by the old code where the red circles indicate the different sizes of spacing between orifices

of the spacing and the diameter added together. This improper addition of inverse sine functions meant that the first cylinders created in each even row were not placed correctly as the modulus condition would only add the extra angle if the row being worked on was even numbered. By looking at Figure 12, it's evident that the spacing between the first orifice in each row was too small. To fix this, everything was combined into one inverse sine function that contained OrificeCenterDist over the outer diameter of the LFOM pipe as shown in Figure 14. OrificeCenterDist was used because it is the variable that represents the center-to-center distance between orifices.

Once the cylinders representing the first orifice in each row were placed, the ArrayLastPolar function was used. This function took the first cylinder drawn and created n-1, where n is the number of orifices in the row being worked on, cylinders of identical dimensions. It would then take the array of identical cylinders

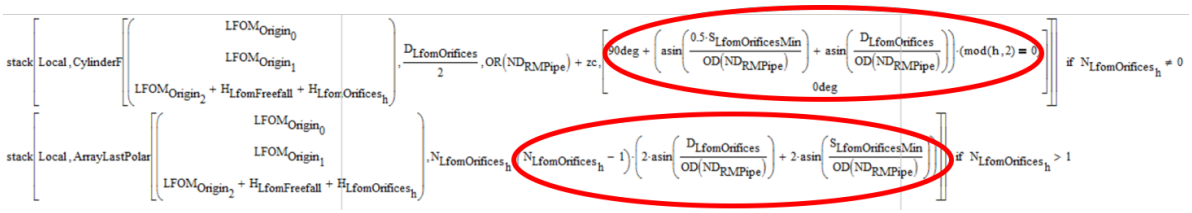

Figure 13: The old code showing the CylinderF function and ArrayLastPolar function with improper angles

and rotate them to their desired position. However, similar to creating the first orifice in each row, the angle calculation was incorrect. The improper addition of inverse sine functions created a rotation angle that was inconsistent with the rotation angle defined in the creation of the first orifice. The inconsistency arose from the fact that the angle of rotation of the array of orifices in the same row was different from the angle of rotation of the first orifice in even numbered rows. This meant that the bottom right spacing between an orifice on an even row and an orifice on the row below would be too big. Conversely, the bottom right spacing between an orifice on an odd row and an orifice on the row below would be too small. Case 1 can be seen in the bottom circle in Figure 12 while case 2 can be seen in the top circle in Figure 12. The fix to this issue was similar to the last fix. By having only one inverse sine term, the incorrect angle calculation was fixed. In order to make sure the angle would be correct, the variable OrificeCenterDist was used in the inverse sine function. Figure 14 shows the ArrayLastPolar function with the corrected angle calculations, and Figure 15 shows the LFOM with the new spacing.

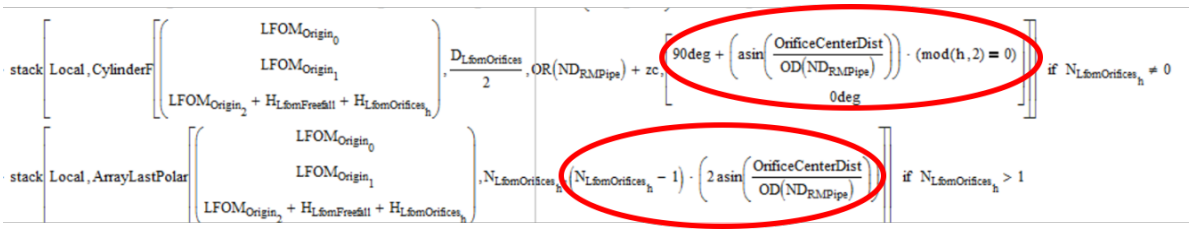

Figure 14: Code that uses the ratio of OrificeCenterDist to the outer diameter of the LFOM in an inverse sine function to calculate rotation for spacing between orifices. This was an intermediate step towards a simpler calculation.

Even though the spacing was corrected, the spacing was still not technically correct. The angle calculation used in the code was more of an approximation. This is because the variable OrificeCenterDist is actually an arc length of the LFOM pipe. The inverse sine functions would only get the actual angle if OrificeCenterDist represented the straight line distance between orifices centers on a curvature. A more precise angle measurement would be calculated from the arc length formula. Since OrificeCenterDist is relatively small compared to the size of the LFOM pipe, the calculation results in an angle that is only a few hundredths to a few tenths of a degree different from the precise calculation. Since the LFOM template is the primary means of creating an LFOM, the purpose of the 3D model is to simply be in the plant designs to show its placement. The difference between the rotation angle calculated in the code and what it should be in real life would likely make little impact to the overall plant design and has no impact on the template. Despite this, the code was modified to reflect the slightly better calculation for angle of rotation. The newest code modifies the ArrayLastPolar function with an angle calculation of the ratio of OrificeCenterDist to the radius of the LFOM pipe, or half of OD(ND.RMPipe). Only the ArrayLastPolar function was modified because it is the function responsible for duplicating the orifices and rotating them to create the spacing. Figure 16 shows this modification.

The next issue with the model that had to be resolved was the fact that the cluster of orifices sitting above the bottom row was positioned to the far left rather than the center of the bottom row. Current MtoA translators do not allow for simply adding a rotation function into the for loop that creates the cylinders that become orifices. Adding some of the various rotations functions would either result in different rows

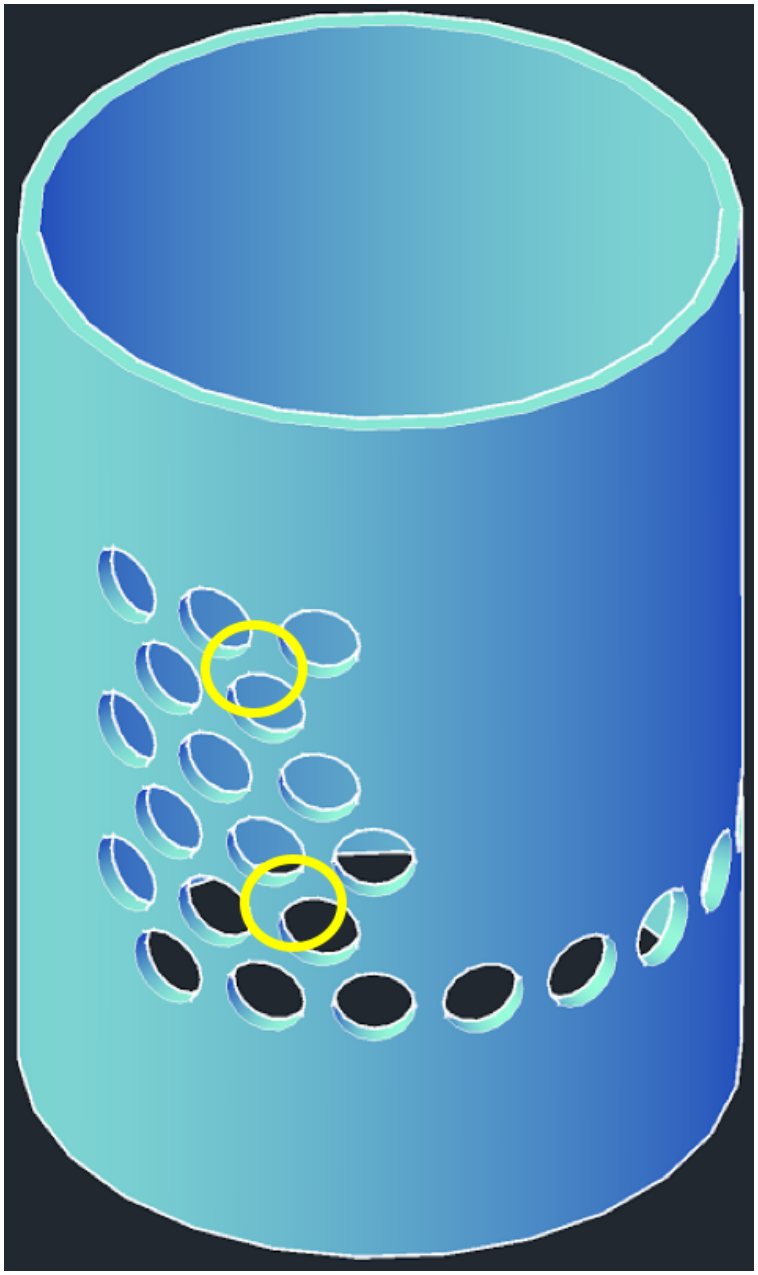

Figure 15: The new LFOM with the proper spacing as indicated by the yellow circles

$$\text{Local} \leftarrow \text{stack}\left(\left[\text{Local}, \text{ArrayLastPolar}\left(\begin{pmatrix} \text{LFOM}_{\text{Origin}_0} \\ \text{LFOM}_{\text{Origin}_1} \\ \text{LFOM}_{\text{Origin}_2} + H_{\text{LfomFreefall}} + H_{\text{LfomOrifices}_h} \end{pmatrix}, N_{\text{LfomOrifices}_h}, \left(N_{\text{LfomOrifices}_h} - 1\right) \cdot \frac{\text{OrificeCenterDist}}{0.5 \cdot \text{OD}\left(\text{ND}_{\text{RMPipe}}\right)}\right)\right] \text{ if } N_{\text{LfomOrifices}_h} > 1\right)$$

Figure 16: Code that uses the ratio of OrificeCenterDist to the outer radius of the LFOM to calculate the central angle of the arc formed between the center-to-center distance between orifices. This was the more accurate way of calculating the spacing.

being rotated different amounts or only the last orifice drawn in each row being rotated.

One option that was tested, and suggested by Meghan Furton, was to create code that started by creating

a script that involved producing a new layer for each row of orifices. The idea behind this was that it would allow for each row to be created and rotated independently of each other. In theory this would be a versatile and elegant fix that would bypass the issue with current rotation functions either rotating all rows on each iteration or only the last orifice drawn. However, when attempting to implement it, there was no simple way to create the layers, draw the orifices and rotate them, and then freeze each layer within a for-loop and produce an output where all rows were present in their proper location. The reason was because within the loop, a layer would be created and named OrifN, where N was the current index of the loop, and within the same iteration, the layer would have to be frozen so that subsequent iterations wouldn't affect the placement of the row when using rotation functions. This meant a layer freeze function was implemented within the loop, which would pose problems when trying to produce a return at the end of the for-loop. The problem with freezing each row after it was made was that the return call at the end of the loop would consistently return the last row made, meaning all intermediate rows in the design were omitted, and not drawn. The loop produced this sort of output because the object that was desired to be returned was dependent on the loop index, and since the loop ends at the index for the highest row on the LFOM, only that one would be drawn.

The next proposed solution was to split the code for drawing orifices into two loops. One loop would bear the responsibility of drawing and rotating the cluster of orifices while the other was only responsible for drawing, but not rotating, the base row of orifices. The idea behind using this algorithm was that having a separate loop for the cluster would allow more freedom as to where the first orifice is positioned on the LFOM pipe. When the two part of the orifices were in one loop, the orifices would have to be drawn with all the same origin point, which was flush with the left of the bottom row. The separation gives the ability to start the cluster at a new point, and the base row at the original point by rotating the first drawn orifice in each row within the cluster.

The first thing that was done was to create a new layer called OrifClusterLayer. After that, a new variable called OrifCluster was defined as a for-loop starting from the first row above the base and ending at the top row. Within the loop, cylinders to be group deleted were first drawn in the same manner, and in the same specifications as the old code did when all rows were drawn in one loop. Similarly, the loop also contained an ArrayLastPolar function with the same specifications as the prior code. The only modification to the loop other than the new layer and indexing was the addition of a Rotate3DLast function between the CylinderF function and the ArrayLastPolar function. This was used in that location because within each iteration, the cylinder is drawn at the original origin. At this moment, only one orifice has been drawn in the current row, which means the Rotate3DLast function will act on the first orifice drawn since it was also the last item created. Doing this allows for the single cylinder to be rotated by the amount desired to place it in the correct location. After the cylinder is created and rotated, the ArrayLastPolar makes identical copies of the cylinder and rotates them by the specified amount from the new location to get the correct spacing. Once all of the rows have been completed, the layer is frozen. Figure 17 shows this process.

After creating and freezing the cluster layer, the base is drawn in. The variable for the base takes on the name LfomOrifices. The code for this variable is identical to the code that drew all the rows within one loop because it maintains only the CylinderF and ArrayLastPolar functions with their original specifications. The difference from before was that there was no longer a for-loop, and the index variable, h, used as the looping index was before was instead locally defined to be 0 because it is the starting index number for Mathcad, and represents anything associated with the bottom row. Figure 18 shows the code for the base row.

All of the new additions to the LFOM model code meant that the final stack for AC.Lfom had to be modified to include the new variables. After the LfomOrifices variable, came OrifClusterLayer, OrifCluster, and ClusterFrz in that order. This meant that the sequence following the drawing of the base was to create the new layer for the cluster, draw the cluster in the correct position, and finally freeze the cluster layer in place.

When attempting to generate a model for an LFOM, an issue with the placement of the orifices occurred. The new model successfully rotated the cluster while maintaining the original position of the base, but the rotation of the cluster was not correct. The angle of rotation was set to be locally defined as Xorigin as shown in Figure 17, which was the exact same calculation for the x starting point of the cluster rows for the 2D template. The reason this was chosen was because this value represented the distance between the first orifice in each row from the original starting point, which was flush with the leftmost orifice of the base row. When projecting the flat template to a cylinder, this distance would also represent an arc length. Knowing

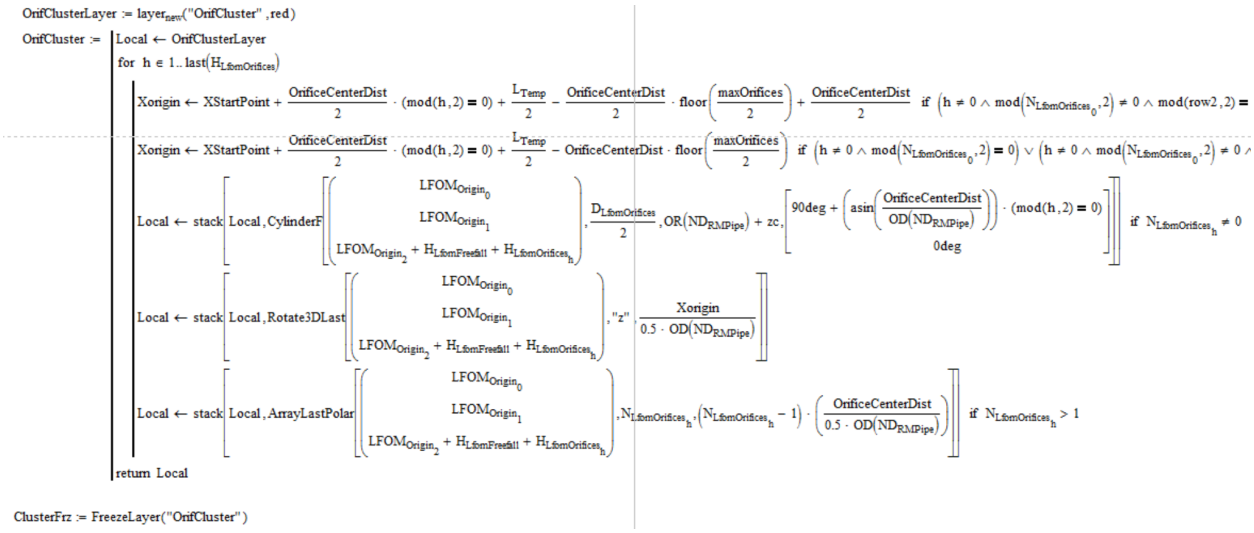

Figure 17: The process of drawing the cluster of orifices. First the cluster layer is made, and within that layer, cylinders are drawn and rotated into the proper starting position. Once in the correct position, the cylinders are copied and rotated from the new position to have the proper spacing between them. The final step is to freeze the layer.

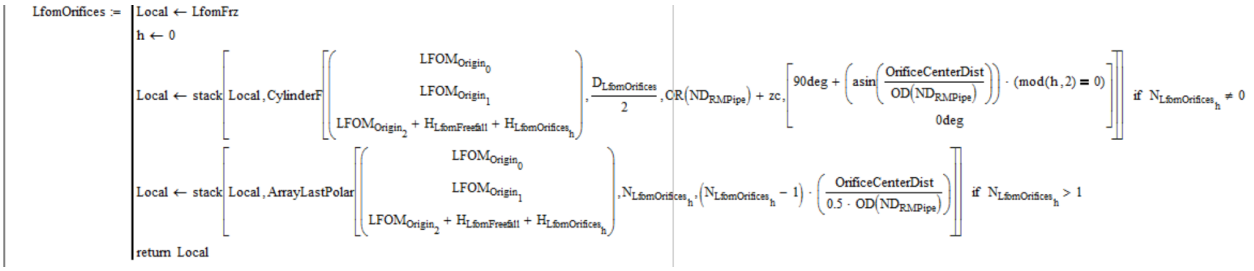

Figure 18: The base row only needs one cylinder drawn at the original start point. That cylinder is then copied and each copy is rotated to get the spacing between orifices. The indexing number h, is locally defined as zero.

Figure 19: The AC.Lfom variable with correctly ordered variables to carry out the complete drawing of the model.

this arc length along with the radius of the LFOM pipe would provide a simple calculation of the central angle of the arc. This central angle would be the exact angle of rotation needed to set the first orifice in each row to its proper place. Despite this, the cluster was consistently rotated to the wrong place, and the spacing would also be negatively influenced.

The underlying problem with the rotation had to do with the conditions that determine the rotation amount, and the calculation of the rotation amount. The original proposed solution used in the code in Figure 17 was to simply copy the same formula used to calculate the displaced origin for orifices in the template version. The logic behind this was that the calculation being done not only determined the new x-origin for the first orifice in each row, but would produce a displacement from the original start point as a length. This linear distance would then be interpreted as an arc length when creating the model because the template represented an LFOM pipe as flat. These distances were dependent on the conditions of whether

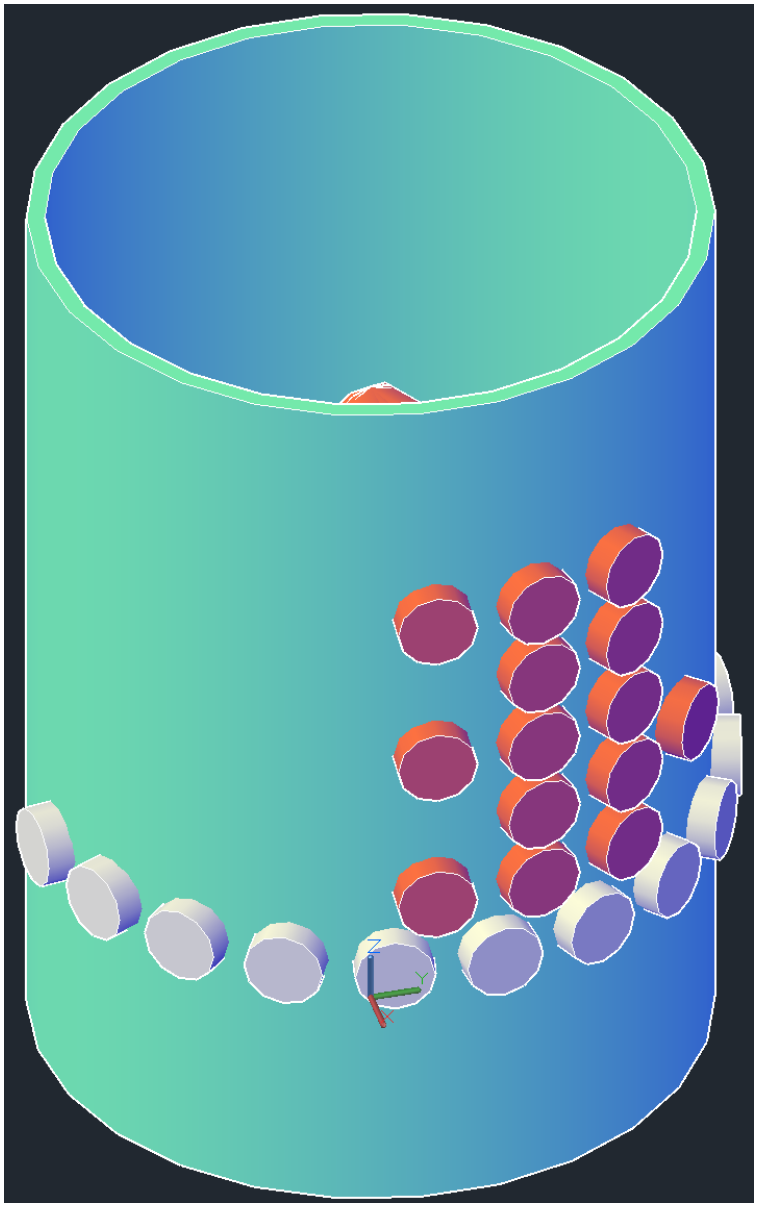

Figure 20: The LFOM generated by the new code that separates the drawing of the base row and the cluster. The cluster layer is shown by its red coloring, which was defined in the creation of the layer in Figure 17. The cluster is rotated, but not correctly rotated, an issue apparent in all flow rates.

the number of orifices in the base row and the row above it was odd or even, and whether the maximum number of orifices in a row was even or odd. To understand how the formula was derived for these new origins, please read Kevin Juan's Spring 2016 Final Report.

On the first attempt at implementing this, the formulas were copied and pasted from the section of code for the template origins. The conditions for each calculation, however, were mixed up, which resulted in only having two formulas copied. This was an oversight issue as there were five uniquely calculated origins, each with small changes that were not observed initially. Figure 17 shows only two Xorigin calculations. For the second attempt, all five unique formulas were added, but an issue with all of the even rows showed up where they were not rotated correctly, but rather too far to the right. This was a result of the origins of the orifices before any rotation between the template and the 3D model. In the model, the orifices start flush left with the proper spacing as noted in Figure 15, which wasn't the case in the template. The template

$$Xorigin \leftarrow LFOM_{Origin_0} + \frac{L_{Temp}}{2} - OrificeCenterDist \cdot floor\left(\frac{maxOrifices}{2}\right) + \frac{S_{LfomOrificesMin}}{2} \text{ if } mod\left(N_{LfomOrifices_0}, 2\right) = 0$$

$$Xorigin \leftarrow LFOM_{Origin_0} + \frac{L_{Temp}}{2} - OrificeCenterDist \cdot floor\left(\frac{maxOrifices}{2}\right) + \frac{S_{LfomOrificesMin}}{2} - \frac{OrificeCenterDist}{2} \text{ if } \left(mod\left(N_{LfomOrifices_0}, 2\right) \neq 0 \wedge mod(maxOrifices, 2) \neq 0\right)$$

$$Xorigin \leftarrow LFOM_{Origin_0} + \frac{L_{Temp}}{2} - OrificeCenterDist \cdot floor\left(\frac{maxOrifices}{2}\right) + \frac{S_{LfomOrificesMin}}{2} + \frac{OrificeCenterDist}{2} \text{ if } \left[mod\left(N_{LfomOrifices_0}, 2\right) \neq 0 \wedge mod(row2, 2) \neq 0 \wedge (mod(maxOrifices, 2) = 0)\right]$$

$$Xorigin \leftarrow LFOM_{Origin_0} + \frac{L_{Temp}}{2} - OrificeCenterDist \cdot floor\left(\frac{maxOrifices}{2}\right) + \frac{S_{LfomOrificesMin}}{2} + \frac{3OrificeCenterDist}{2} \text{ if } \left(mod\left(N_{LfomOrifices_0}, 2\right) \neq 0 \wedge mod(row2, 2) = 0 \wedge mod(maxOrifices, 2) = 0\right)$$

Figure 21: The four different calculations for rotation amount, which depend on whether the number of orifices in the base row and the row above it is even or odd and whether the maximum number of orifices in a row is even or odd.

origins needed a value of half of OrificeCenterDist dependent on whether the current row being worked on was even or odd, which is seen in Figure 17. This dependence was unnecessary. Instead, it was assumed that mod(h,2) in the original calculation would always be 1, meaning that value of half of the OrificeCenterDist would be added regardless of the row. This was done because the total distance rotated would be the same for each row, meaning the rotation angle was the same for each row, which was desired to start with. Letting mod(h,2) equal 1 was chosen because fewer changes would be needed to calculate the distance rotated.

When the code was tested again, it was found that all the flow rates tested produced models that were off by half the value of S.LfomOrificesMin, so that value had to be added back in each formula for Xorigin. Another change made, though it had no difference, was that XStartPoint was s changed to the variable LFOM.Origin0, the x-origin for the 3D model. Both values take on 0, but this was changed for consistency. Finally, the code was checked for robustness. Since the were three different conditions to be met as noted above, with each condition having two possible outcomes (1 or 0), there were 8 different combinations total. However, from the LFOM template code, it was reduced to five since some designs did not rely on all 3. Further testing found that only four combinations were needed. The combination where all conditions produced a value of 1 was unnecessary, and was better off having the same design as the design produced with a combination where mod(N.LfomOrifices0,2)=1 and mod(maxOrifices,2)=1. This change was also made in the calculations done in the template code.

Testing flow rates that matched the four different combinations showed that the model was reflecting the correct design as determined by the template code. The code was then submitted to the beta server for testing. The first run did not produce an output, which was a result of not leaving out the L/s units in the input variable for Q.Plant, a nuisance discovered last semester when testing the LFOM modular code. After this was changed, the code produced all the correct stock models with the corresponding template that reflected the model in 2D paper space. Submission to the full server had not been completed yet, and the design had not been reflected in treatment plant designs that use the LFOM.

5 Autodesk Fusion 360 Implementation

During the fall 2016 semester, AguaClara was invited to take part in a week long design event hosted by Autodesk Fusion 360 on the Cornell campus. The StaRS filter theory team accepted the invite and participated in the event. During the event, the Fusion 360 team on campus garnered interest in the design work that AguaClara does with AutoCAD. A meeting was set up between AguaClara and Fusion 360 to discuss the software and how it could be worked into the Design team. During the meeting, design members learned about the capabilities of Fusion 360 with regard to creating realistic and interactive 3D models.

Although Fusion 360 was much more capable of making 3D models with ease, it could not be used as a direct replacement to AutoCAD because it's construction drawings capabilities doesn't match that of AutoCAD. Furthermore, it wouldn't be able to interact with the design server in it's current state because Fusion 360 doesn't have a native command line like AutoCAD. The lack of a command line like AutoCAD prohibits the design team from being able to simply reuse Mathcad code. Despite this, Fusion 360 is able to interact with Python, Java, or C++ script. Because of this, it was decided to use Fusion 360 as a learning tool for the time being, and once more was understood about the software, it could be implemented into the design server.

Figure 22: The correct 3D model of an LFOM for a 20L/s plant, which has the correct spacing and rotation displayed in the template in Figure 11.

5.a Challenge Details

The first part of this challenge was to get acquainted with the Fusion 360 interface and capabilities. One of the main concerns when first approaching Fusion 360 was how the team would convert already created AutoCAD design files to workable Fusion 360 files. For the purpose of learning the program and producing a model for educational use, the 20 L/s plant AutoCAD plant was exported as an .iges file. This file type was preferable to other file types supported between Fusion 360 and AutoCAD because it maintained the drawings for many of the valves and miscellaneous parts that are not produced by code created by AguaClara. This ultimately results in a larger file. Once the AutoCAD file was exported as an .iges file, it was imported into Fusion 360.

The user interface in Fusion 360 is completely different from that of AutoCAD. One of the most obvious

differences is the sidebar in the workspace. The sidebar shows a long list of named items with light bulbs next to each one. These items are organized into components and bodies, which are analogous to layers and groups in AutoCAD. The component is the top level item, which is composed of smaller bodies. The component for the 20 L/s plant would be the entire plant, and its constituent bodies would be all of the different pipes, valve, concrete, tubing, etc. The sidebar provides users the ability to turn on or off bodies or components at their will by simply clicking on the light bulb. An illuminated bulb means the body or component is on and visible while an off light bulb means the component of body is not visible. Turning off a component will cause all bodies to turn off while turning off a body has no effect on the other bodies.

Another helpful aspect of Fusion 360 is its ability to create much more realistic designs than AutoCAD. Current designs produced by the ADT come out as colored lines that have little bearing to their real-life properties. In Fusion 360, users have the ability to choose the type of material that a certain part is made from and its physical appearance. These are accessible from the toolbar, and applying physical material or physical properties requires the user to select the objects and drag the property or material onto the selected objects. When rendered, this aspect gives the designs a more realistic look, and enables users to understand what they're looking at compared to the colored lines produced by AutoCAD.

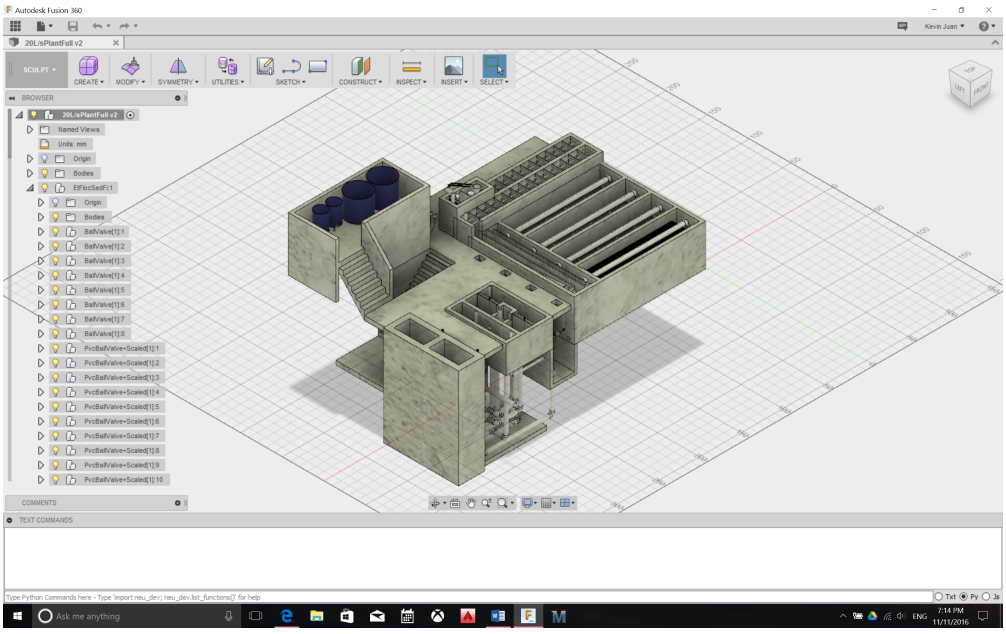

Figure 23: The user interface shows a sidebar with options to turn on and off bodies and components.

A unique part of Fusion 360 is that all of the models are saved through Autodesk's A360 cloud service. From there, users can share links to their designs, or generate HTML code to embed a design in a viewer for users to interact with in web pages. So far this semester, a full model of a 20 L/s plant was completed on Fusion 360 and published on the AguaClara wiki. A similar model of the 20 L/s plant was also placed on the wiki except the model lacked the concrete so users could see the inner components of the plant. The embedded viewer provides very nice access to the designs because it only requires adequate internet access as all of the designs are rendered by Autodesk's servers. This means less time is spent by users to download large files that require heavy graphics and memory usage. In addition, the viewer enables users to rotate, zoom, and pan the object to get an infinite number of views. Lastly, the viewer also enables users to produce live section cuts, which is essentially the user selecting the plane in which to do the cut and dragging the cut plane back and forth to get section cuts that change as the user drags the plane.

Currently, there are no established ways to automate the entire process of changing AutoCAD models into Fusion 360 models, and the entire process could take multiple days for a single design. Each design requires manually exporting an AutoCAD .dwg file to either an .iges, .igs, or .sat file. Once exported, the files can be opened in Fusion 360, but the designs will appear in a monotone gray color or in the colors set

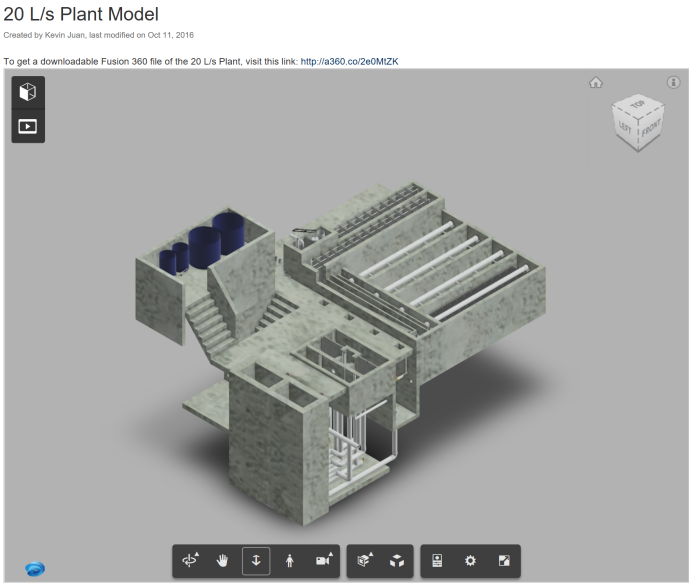

Figure 24: The 20 L/s plant with concrete currently published on AguaClara's wiki.

by the layers in the AutoCAD file depending on the file type. Because of this, the physical materials and appearances all parts need to be altered by hand, which is a tedious task considering the number the small components drawn in each model. Furthermore, current restrictions on knowledge of Fusion 360 does not allow the design team to create Fusion 360 models from scratch. Doing so will require a broad knowledge of Fusion 360's APIs, Python, Java, or C++, and guidance from Autodesk and its community to create code tailored to AguaClara's needs.

Work for the remainder of the semester, and in the future, would be to learn more about how Fusion 360 is able to interact with Python, Java, or C++ code, and use one of the languages to automate the process much like Mathcad and AutoCAD. Team members working on the transitional project are learning about Fusion 360 APIs through tutorial videos, and hope to begin testing soon.

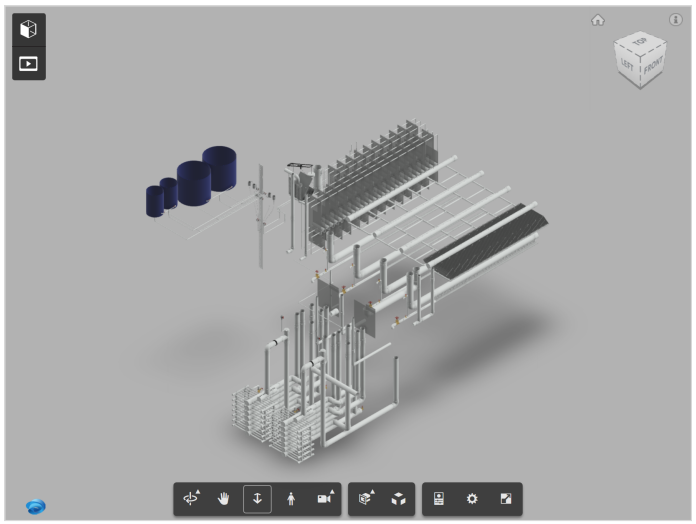

Figure 25: The 20 L/s plant without concrete currently published on AguaClara's wiki, which enables users to see the internals of the plant. This was done by simply turning off all the concrete bodies in the model.

6 Future Work

As of the end of the Fall 2016 semester, there is still much work to be done on many of the challenges taken on this semester. All of these challenges are expected to be continued either next semester, or in future semesters.

One challenge that wasn't fully completed this semester was the LFOM. Though the modular code for the LFOM has been completed, the changes made this semester have not been implemented in any of the designs that rely on an LFOM, namely the standard and low flow water treatment plant designs. This is because the LFOM is a component of the entrance tank, and is drawn within the entrance tank drawing code. In order to reflect the changes made to the modular code this semester, the code for drawing the LFOM model in the files EntranceTankLowAC and EntranceTankCompactAC need to be changed to the be the same exact code produced in the modular design. When doing this, some of the variables specific to drawing the template need to be included in the files since the modular code for drawing the model is dependent on these variables. These variables are the ones located in calculating the rotation distance, Xorigin, and the corresponding conditions that dictate the specific rotation calculation.

Another challenge that was not completed this semester, and should be completed as soon as possible is the section cuts challenge. As of December 2016, the design team has not received any code from Autodesk to test for producing sections cuts. Next semester's team working on that challenge will have to reestablish connection with the Autodesk team currently working on the challenge. The team hopes that viable code will be received early next semester to test and implement by the end of the Spring 2017 semester.

Lastly, the Fusion 360 implementation project will be continuing in future semesters as the design team hopes to learn enough about the APIs to start testing code by Spring 2017, and have early developments of translators. This will require continued learning about APIs, the coding language to be used, and how the programs will interact with the infrastructure already in place. The team will also look for support on this project from Autodesk and the Autodesk community to help create code that meets our demands.

7 Task Map

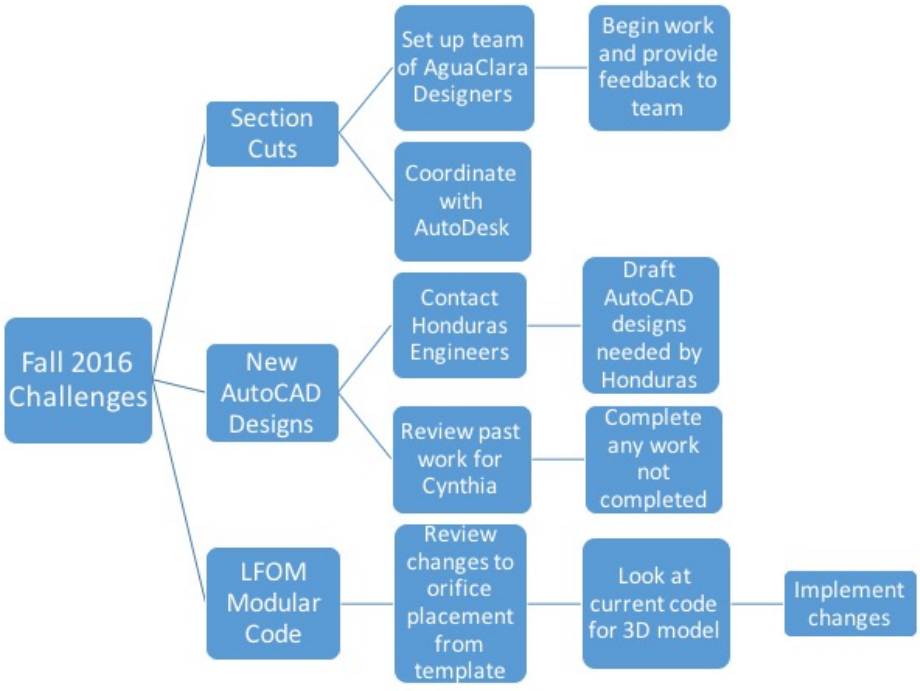

Figure 26: Fall 2016 Task Map

7.a Task Map Details

- Section Cuts
  - Set up meeting with Kean from Autodesk to talk about project goals and timeline
  - Communicate feedback from AguaClara designers about the code to Autodesk for improvements and tweaks
  - Implement finalized section cuts code
- New AutoCAD Designs
  - Collaborate with Skyler and Ethan in Honduras to figure out whether there are any plant pieces that need to be drafted
  - Obtain sketches and dimensions of pieces to be drafted in AutoCAD
  - Review and complete any work not done by Cynthia from Spring 2016
- LFOM Modular Code
  - Look over orifice origin code for the template to review what changes were made to orifice placement
  - Investigate Mathcad code for the LFOM model design to check what changes the cylinder placement and subtraction that creates orifice cut outs
  - Implement the orifice changes and update the server to reflect new design