Error Checking Code Team

Guneet Sandhu and Kafayat Salawu

05/16/2014

1 Problem Definition

1.1 Introduction

AguaClara is an open source technology that allows multiple user interface. Due to the flexibility of the design tool, which allows users to modify certain parameters for special design requirements, it is necessary to ensure that the Expert Inputs, which are an integral part of the design remain consistent for standard AguaClara designs.

A feature needed to test the integrity of the designs produced by the AguaClara design tool is an independent error checking code. The objective of having such a code is primarily to identify potential sources of error that could lead to serious design issues or failures. The independent code ensures that all constraints critical to the overall design of the plant are checked for accuracy and consistency. This independent error checking code will be used for all final plant designs provided by the design tool.

"Expert Inputs" consist of a list of constants that are used in the design code to calculate various parameters that are central to the overall design. These values reflect the latest developments in AguaClara technology based on ongoing research and field requirements. The scope of the error checking code will consist of, but will not be limited to, comparing the values of expert inputs used for each design to the values approved by the experts as per an official signed document.

Another major function of the error checking code will be to perform an overall hydraulic gradient analysis of the plant to ensure there were no errors in the head loss estimation for each design which could lead to flooding or other potential failures by virtue of incorrect head loss estimation.

1.2 Design Details

Having an independent error checking code for hydraulic head estimation and expert inputs acts as a validation mechanism to bolster the reliability of each final plant design through various check points within the error checking code.

1.2.1 Expert Inputs Comparison Code

An integral part of the Expert Inputs Comparison Code is an independent approved list of Expert Inputs, solely a part of this code, with which the values used by the design tool were compared. In case of discrepancies, the code displays an error statement, hence alerting the user to trace the error. Since the code is meant to be an independent document, any approved changes to the Expert Inputs will need to be updated in the Expert Inputs Comparison Code independently.

1.2.2 Error Checking Code

The Error Checking Code primarily consists of accounting for head losses across each unit and associated inlet and outlet systems. The hydraulic gradient was mapped from the exit of the plant all the way to the entrance. This ensures that the overall head loss is in agreement with the head loss per component estimated by the design tool. Due consideration was given to relevant dimension of the component that is affected by the head loss e.g. depth of the tanks. Thorough analysis of the head loss incurred across each unit was performed. The head loss functions pertaining to each unit was utilized to compute the head loss step wise across the following components:

- The stacked rapid sand filter including the exit and entrance channels
- The exit channel and weir of sedimentation tank
- The sedimentation tank launders
- The sedimentation tank plate settlers
- The sedimentation tank diffusers
- The sedimentation tank inlet manifold
- The sedimentation tank inlet channel
- The flocculator
- The rapid mix
- The linear flow orifice meter
- The entrance tank

Analysis of any elevation constraints associated with each unit of the plant was included to ensure that the water level in the unit satisfies that constraint. The code utilized fluid functions specific to calculating head loss for each component described above. The aim was make the code as independent as possible in order to keep infiltration of error due to previously defined functions to a minimum i.e. no files (except the Approved Expert Inputs) from the design tool are being referenced in the code.

1.3 Solution Approach

1.3.1 Expert Inputs Comparison Code

The Expert Inputs Comparison Code is a simple comparison of values of each of the parameters in the expert inputs file used in the design tool to the values of parameters in the approved expert inputs file. If the values are the same, there is no error reported otherwise an error message is displayed. An integral part of the code was to obtain an approved Expert Inputs file with which the current values used by the design tool was compared. The variable names of the expert inputs comparison code had to be modified in order to compare the value of the same parameters without the risk of redefinition.

1.3.2 Error Checking Code

The first step was to copy the fluid functions to the error checking code as a separate section at the beginning of the code. From the math functions, pipe functions, and materials database, only functions and constants needed for the headloss calculations were transferred to the code and this was done as we continued along with the transfer of headloss functions and pertinent dimensional constraints for each component. Next, we thoroughly combed through the design code for each component of the plant and extracted equations pertinent for the headloss calculations. The headloss was then computed for each of the components according to the design tool sequence for the entire plant. The values computed were then used to calculate the hydraulic grade line for the entire plant starting from the exit of the plant all the way to the entrance. Finally, we ensured that the corresponding tank depths took all relevant head losses and constraints into account and compared all values with those determined by the design tool.

2 Documented Progress

2.1 Expert Inputs Comparison Code

The objective of creating the Expert Inputs Comparison Code was to compare each parameter in the approved expert inputs to the expert inputs used in the design tool in order to ensure that the expert inputs used in the design tool are consistent with the approved list.. To accomplish this objective and in order to limit the size of the Error Checking Code, an independent file was created. The Approved Expert Inputs file obtained is a separate file inaccessible to the design tool, compiled and approved by Prof. Weber-Shirk. Given that this file is independent of the Design Tool Expert Inputs, any changes made to the Design Tool Expert Inputs have to be added to the Approved Expert Inputs File and the Expert Inputs Comparison Code.

Since the same parameter is being compared in the Expert Inputs Comparison Code, the variable name in the approved list had to be modified by adding a suffix of "EC" (Error Checking) to avoid the risk of redefinition and to allow for comparison. Once the variable name was changed, a simple "if-otherwise" function was used to compare the value of corresponding parameters in both the approved expert inputs and design tool expert inputs. In order to make tracing of error easy for the user, it was decided that each of the parameters will be checked independently. This had to be repeated for all 189 parameters currently defined in the expert inputs file, i.e. $if\ (Parameter_{EC} = Parameter(design\ tool))$, "OK" message is displayed otherwise an error message is displayed. The output or result of the Expert Inputs Comparison Code is a message for each parameter displaying whether the value of each of the parameters match or if there is a discrepancy between the values. At the end of the code, the results for each of the parameters are displayed in a section, which is a comprehensive list that displays the name of the parameter, for which the values are not matching.

2.2 Error Checking Code

The objective of the Error Checking Code is to independently calculate the head loss of the different components of the plant in order to obtain the hydraulic head of the component. In order to ensure that the head estimated in the design tool is accurate, the hydraulic head will be mapped from the exit of the plant and the depth of water in the component will be estimated. If the depth of water is less than the calculated depth of the component, the head loss estimation is accurate. The hydraulic grade line is essentially determining the piezometric or hydraulic head for each of the point of analysis. The hydraulic head for each unit (He_{Unit}) was calculated as the sum of the pressure head due to the water in the unit (HW_{Unit}) and the elevation head due to the elevation difference between the bottom of the unit and the datum selected for the analysis of the particular unit (Z_{Unit}) i.e. for any general unit, $He_{Unit} = HW_{Unit} + Z_{Unit}$. In order to ensure that the head loss calculations are accurate and there is no flooding in the unit, it was also ensured that the mapped hydraulic gradient is continuous between different units of the plant such that the head at the exit of one unit is equal to the head at the entrance of next unit.

In order to make the Error Checking Code as independent as possible, the file referenced in the code is approved expert inputs file and not the expert inputs file used by the design tool. The first task involved was to calculate head loss across each of the units using functions that were consistent with the code and ensuring that the code was consistent with the CEE 4540 design philosophy. Since most of the head loss calculations required different predefined functions, all relevant functions were imported from the main code with revision number and date duly noted. To avoid redundant functions, only functions relevant to head loss calculations were imported except for fluid functions. The fluid functions defined were also checked for consistency with CEE 4540. Before incorporation, the head loss calculation methodology used in the design code was cross checked with CEE 4540 design philosophy for each of the individual components and discrepancies were noted for clarification. Head loss calculations for some elements, as per the latest design, were included in the error checking code, which were originally not accounted for in the design code. The main concern was to ensure that the minor head losses did not amplify when considered together with other components. The results section was added to the Error Checking Code for the purposes of compilation of the outputs of various checks in each of the sections.

2.2.1 Stacked Rapid Sand Filter

To begin the analysis for hydraulic head estimation, the first step is to select the datum from which each individual elevation will be measured as per the methodology outlined in the previous section. The selection of the datum was tricky for the stacked rapid sand filter primarily because origin of the filter and origin of the plant are different since the filter unit is deeper than the other units. To compute the head of the units in the SRSF section, the elevations considered were corrected to account for the bottom of the filter which is the filter origin so that the head calculation is relatively straight forward and is not negative for units below the plant origin. The code for Autocad scripts for SRSF was used to find the filter bottom thus enabling the code to integrate both the design and Autocad script values to determine the head. This leads to another consistency check between the parameters calculated in different codes. Upon final calculation of hydraulic head in the SRSF unit, the head calculated was corrected back to the plant origin which will be the consistent datum selected for the hydraulic head analysis. The SRSF analysis included in the error checking code is for filtration mode of the SRSF since the dimensions of the main units were constrained in the filtration mode. For the backwash mode and siphon operation, the constraints outlined or mentioned in the main SRSF design code were incorporated in the error checking code as it is. The starting point of the analysis for the SRSF unit is the point where the distribution pipe joins the filter exit tank. The piezometric head is then traced in various units like the filter exit piping, filter box, filter inlet piping, filter entrance tank by incorporating head loss in each of the units mentioned. As anticipated, the head increases across each unit as it is traced back into other preceding units and the difference is equal to the head loss calculated across each unit. As a method for checking errors due to incorrect head estimation leading to flooding, the depth of water in each unit was compared to the depth/height of each unit. If the former is lesser than the latter no error is reported in the code otherwise an error message is displayed i.e. ideally $HW_{Unit} < H_{Unit}$. After the filter entrance tank head estimation, head of other units are traced in a similar manner by simply adding the calculated head loss. Another check for consistency that was added to ensure if the hydraulic grade line is correctly estimated is by checking if the difference in head between two points of analysis is equal to the head loss calculated between the same points. During the compilation of the code for the stacked rapid sand filter, there was a check in the design code, which states that $Z_{FiSiphonCouplingMax} - Z_{FiSiphonCouplingMin} \geq 0$, but for flowrates above 20 L/s this value is negative so some corrective action was needed to mitigate this issue. The issue was brought to the notice of the design team and it is currently being resolved. The LFSRSF team had run into an error in the design code for the design of the slotted pipes used in the filter. For low flowrates the calculated length of the backwash manifold slots was larger than the circumference of the slotted pipe. The team had incorporated a check to mitigate the problem by adjusting the headloss through the backwash manifold slots. Although this issue was prevalent in the LFSRSF, this check was modified for the SRSF and was incorporated in the main SRSF code to make it more comprehensive. Subsequently, the check was also added to the main Error Checking Code based on the team's feedback.

2.2.2 Sedimentation Tank

The head estimation for all units starting with the sedimentation tank exit or filter inlet was relatively straight forward since the bottom of each unit coincides with the plant origin or is elevated. As long as the plant origin is the bottom most point in the elevation analysis, the procedure is direct and there is no need for elevation correction. For the sedimentation tank, discrepancy between the design philosophies used in the code and CEE 4540 was found pertaining to the hydraulics discussed in CEE 4540, wherein $V_{SedUpBod} \times W_{SedBay} = V_{SedDiffuser} \times W_{SedDiffuserOutlet}$ but this equality is violated as per the values computed in the code. This issue is currently being resolved by the design team. Head loss pertaining to the flow expansions in the sedimentation tank inlet channel, inlet manifold, diffusers and plate settlers was also incorporated in the error checking code. The hydraulic head calculated had only one point of discrepancy upon comparison with the values computed by the main design code. The discrepancy is due to the fact the the error checking code has accounted for head loss across each component in the sedimentation tank including the inlet channel, inlet pipe manifold, diffusers ,and sedimentation tank plate settlers that the original design code has not accounted for, since they were minor. Although the difference is a few millimeters and is within the plant freeboard design margin, it should not be a cause of concern for lower flowrates but for higher flowrates it can be an issue.

The total head loss by virtue of flow expansion and frictional loss in the inlet channel of the sedimentation tank is given by:

$$HL_{SedInlet} = h_{eRect} + h_{fRect}$$

$$h_{eRect} = \frac{K_{Exp}.Q_{Plant}^2}{2.g.(W_{SedInletChannel}.H_{SedInletChannel})^2}$$

$$h_{fRect} = \frac{f_{rect} L_{SedChannelInlet} Q_{Plant}^2}{4.R_h.2g(W_{SedInletChannel}.H_{SedInletChannel})^2}$$

$$R_{hOpen} = \frac{W._{SedInletChannel}.H_{SedInletChannel}}{W._{SedInletChannel}+2.H._{SedInletChannel}.}$$

The head loss due to flow expansion from the inlet manifold into the diffuser ports is given by:

$$HL_{SedManifoldPort} = h_e$$

$$h_e = \frac{K_{PipeEnt}.8.Q^2_{SedManifoldPort}}{g.\pi^2 D^4_{SedManifoldPort}}$$

The head loss due to the flow expansion of the jet through the diffuser and jet reverser is given by:

$$HL_{SedDiffuser} = h_{eGen}$$

$$h_{eGen} = \frac{K_{Exp} V^2_{SedDiffuser}}{2.g}$$

The head loss due to the flow through the plate settlers is given by:

$$HL_{SedPlate} = \frac{2.\nu}{g}.(\frac{V_{SedUp}}{V_{SedCapture}} - 1)(\frac{6.V_{SedUp}}{S_{SedPlate}.(sin^2\alpha.cos\alpha)})$$

The head loss incurred in the inlet manifold by virtue of 90º elbow and flow expansion is given by:

$$HL_{InletManifoldEnt} = h_{eGen1}$$

$$HL_{InletManifoldElbow} = h_{eGen2}$$

$$h_{eGen1} = \frac{K_{PipeEnt} V^2_{SedInletManifold}}{2.g}$$

$$h_{eGen2} = \frac{K_{El90} V^2_{SedInletManifold}}{2.g}$$

$$V_{SedInletManifold} = \frac{ED_{SedInlet}.D^{1/3}_{PipeED}.Pi^{7/6}_{VenaContractaOrifice}}{Pi_{JetRound}}$$

2.2.3 Flocculator

In the flocculator, three different sets of equations were transferred - each for the vertical, horizontal, and low flow flocculators. Each type of flocculator has different dimensions and hence, different variables are applicable for use when calculating the head loss. The head loss through the horizontal and vertical flocculators are both calculated based on the minor head loss through a baffle multiplied by the number of baffle spaces. The head loss through the ports between channels was added to the head loss calculations for both the horizontal and vertical flocculator cases. The equations for the expansion losses through the ports are different for those between channels and for those at exit locations due to their different dimensions. Consistent with the conventions used in the design tool, either a V, H, or LF subscript is added to the equations to differentiate between the different flocculators. The minor loss expression for the vertical and horizontal flocculator channel ports are given below:

$$HL_{FlocChannelPortV} = (N_{FlocChannels} - 1)\frac{k_{PipeExit}}{2g}\left(\frac{Q_{Plant}}{H_{FlocPort_0}W_{FlocPort_0}}\right)^2$$

The expansion loss expression for the exit port is given by:

$$HL_{FlocExitPortV} = \frac{k_{PipeExit}}{2g}\left(\frac{Q_{Plant}}{H_{FlocPortV}W_{FlocPort_i}}\right)^2$$

The expressions for the horizontal flocculator are:

$$HL_{FlocChannelPortH} = (N_{FlocChannels} - 1)\frac{k_{FlocBaffle}}{2g}\left(\frac{Q_{Plant}}{H_{FlocPort_0}W_{FlocPort_0}}\right)^2$$

The expansion loss expression for the exit port is given by:

$$HL_{FlocExitPortV} = \frac{k_{FlocBaffle}}{2g}\left(\frac{Q_{Plant}}{H_{FlocPort_i}W_{FlocPort_i}}\right)^2$$

2.2.4 Rapid Mix, Linear Flow Orifice Meter, and Entrance Tank

The design methodology for entrance tank is similar to other units described above. For the LFOM, there are two scenarios - one is for normal flowrates and the other is for high flowrates i.e 80 L/s and above. For normal flowrates the LFOM is a pipe and for high flowrates there is an LFOM channel along with a rapid mix channel. The equations for both scenarios were transferred. To compute the hydraulic head in the LFOM for normal flowrates, the head in the vertical section of the entire LFOM/ Rapid Mix pipe had to be considered. Autocad files were referred to obtain the length of the pipe and the headloss incurred in the LFOM was included. For the check on the overall plant hydraulic gradient, the value calculated by the Error Checking Code is compared with the value calculated by the design code to ensure that there is consistency between the two codes.

3 Future Work

The error checking code as of now is consistent and fully inclusive however, in order to maintain the integrity of the code an ongoing task will be to update the Error Checking Code and Expert Inputs Comparison Code for any changes that take place in the design tool as the individual codes are independent and do not reference any file. The importance of this task was highlighted during the course of the semester when changes in the design code were not carried over into the error checking code leading to discrepancies in the computation and comparison of key parameters. Any open comments that were highlighted in the discrepancies list, if resolved in the future need to be implemented in the respective codes also. In order to make the error checking code more comprehensive, it will be beneficial to incorporate changes made to the design code based on feedback received from other teams.