UNIVERSITY OF MINNESOTA: TWIN CITIES

CE 4511 HYDRAULIC STRUCTURES

HW 6: Tainter Gates

CHRIS [redacted] LOGSTON

APRIL 27TH, 2015

Contents

0 Preliminaries **1**
0.1 Introduction 1

a No Tailwater Effects **2**
a.1 Uncontrolled Discharge 2
a.2 Controlled Discharge 7
a.3 Unsubmerged, Controlled Headwater Elevation vs. Discharge Curves 9

b Considering Tailwater Effects **13**
b.1 Given Scenario 13
b.2 Modified Discharge for Submergence 14
b.3 Submerged, Controlled Headwater Elevation vs. Discharge Curves 15

A Matlab® Scripts **19**
A.1 Master Script 19
A.2 Function Script: $\mathbf{y_n = f(Q)}$ 20
A.3 Function Script: $\mathbf{y_2 = f(y_1, Q)}$ 20
A.4 Reader Script: $\beta, \mathbf{C = f(G_0)}$ 20

0) Preliminaries

0.1) Introduction

This report reviews the process of developing a stage-discharge rating curve for a Tainter gated structure. In part a, the stage-discharge curve is determined assuming the downstream elevation does not cause submergence. This is then checked against the same process, considering submergence in part b.

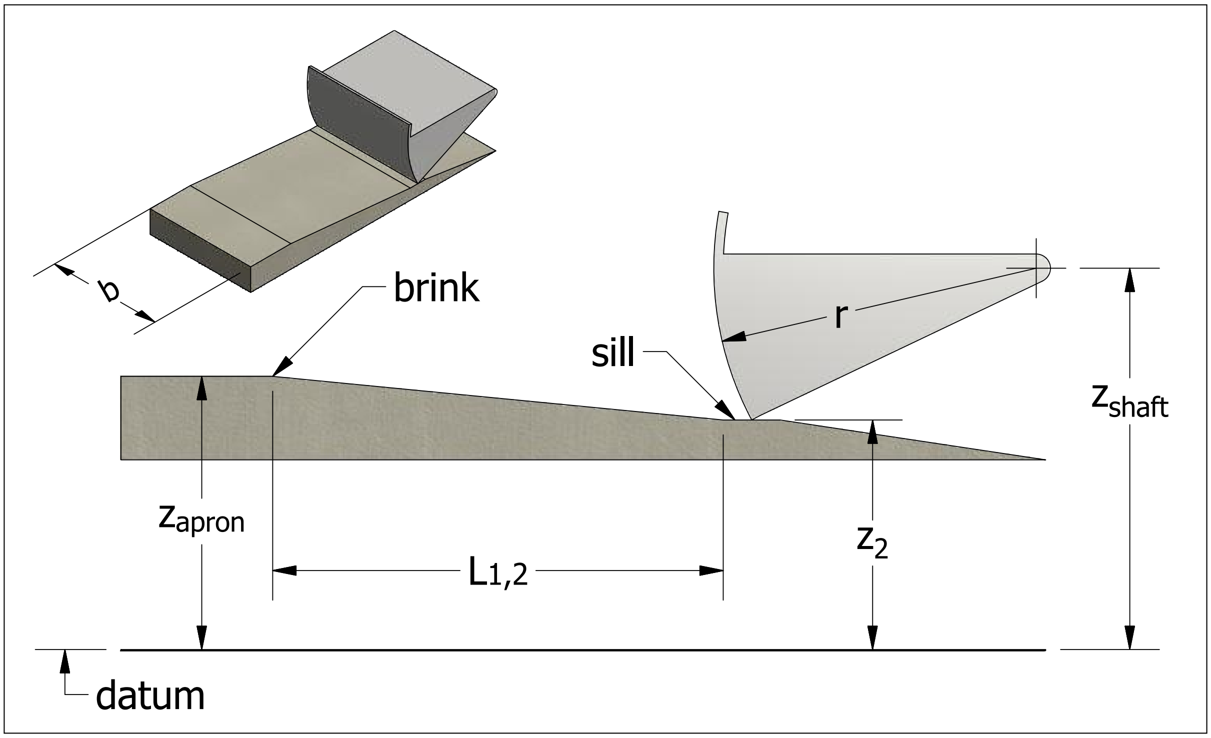

Figure 1: Tainter Gate System Dimensions

As shown in indicated in figure 1, the given system dimensions are as follows:

- Tainter gate system width: $b = 19.5\ ft$
- Tainter gate radius: $r = 17\ ft$
- Tainter gate shaft (pivot point) : $z_{\text{shaft}} = 1179.8\ ft$
- upstream apron elevation: $z_{\text{apron}} = 1174.1\ ft$
- gate sill elevation: $z_2 = 1171.8\ ft$
- horizontal length from brink to sill: $L_{1,2} = 23.75\ ft$

Also, the following properties are inferred:

- acceleration of gravity: $g = 32.2\ {}^{m}\!/_{s^2}$
- Manning's roughness coefficient for concrete : $n = 0.013$

a) No Tailwater Effects

a.1) Uncontrolled Discharge

The maximum water surface elevation in the upstream pool is given to be $H_{w_{max}} = 1185.32\ ft$. The maximum gate opening is also given to be $G_{0_{max}} = 6\ ft$. A hypothetical surface profile for these conditions, with no relevant tailwater height is shown in figure 2. This figure shows flow crossing the brink at an approximated depth of $y_1 = y_c$, incurring head loss to friction along an s_2 profile, and passing under the gate lip, uncontrolled.

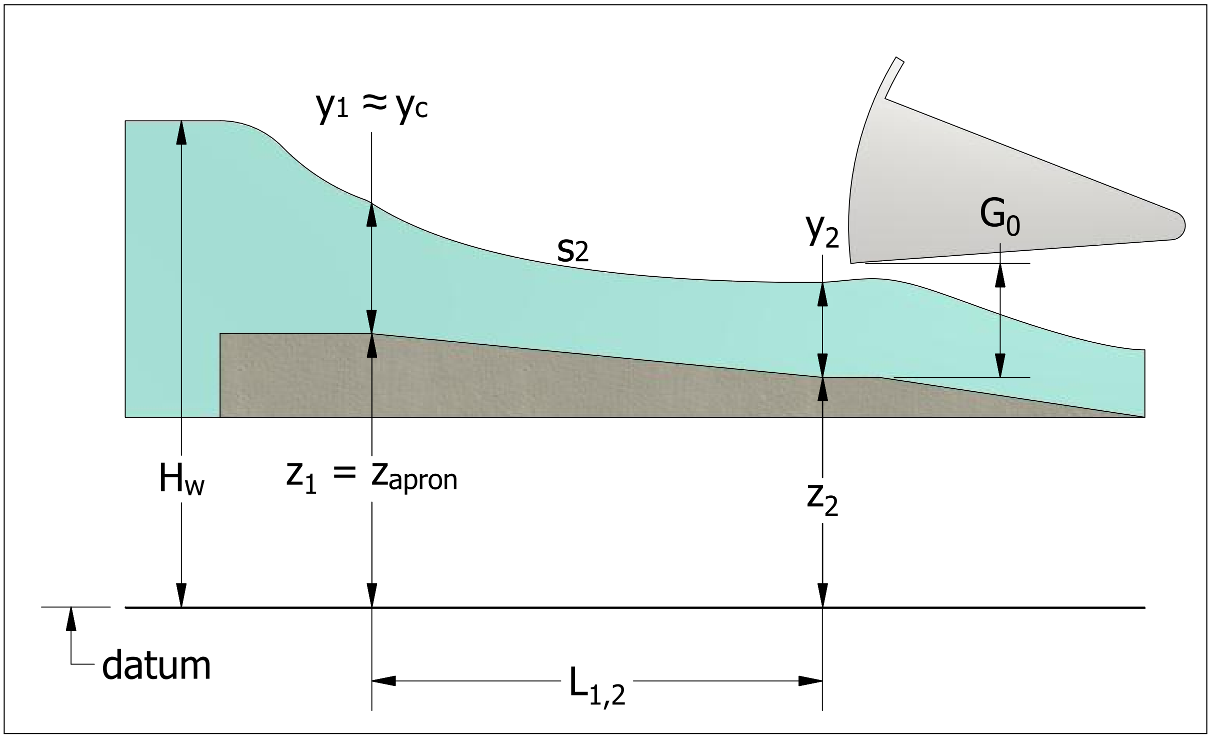

Figure 2: Possible WSP for uncontrolled flow

In order to verify the verify this uncontrolled hypotheses, the following steps are taken:

- The entrance depth at the into the s_2 profile is approximated to occur at the brink, at a depth just below y_c.
- One iteration of the direct step method is applied along the relatively short $L_{1,2}$ to calculate y_2.
- This y_2 is assumed to be the relevant depth to be checked under the gate lip.
  - That is, the ensuing increase in depth along the sill length is negligible if the clearance between G_0 and y_2 is large enough.

a.1.1) Determining Maximum Critical Depth

The critical depth can be solved using the following energy balance between the upstream pool and the brink:

$$\left[h_{\text{bed elevation}} + h_{\text{hydraulic depth}} + h_{\text{velocity}}\right]_{\text{pool}} = \left[h_{\text{bed elevation}} + h_{\text{hydraulic depth}} + h_{\text{velocity}}\right]_1 \tag{1}$$

The following approximations are made:

$$\begin{aligned}
\text{negligible velocity head in pool :} \quad & h_{\text{velocity}_{\text{pool}}} = 0 \\
\text{pool elevation accounts for bed elevation and hydraulic depth :} \quad & H_w = \left[h_{\text{bed elevation}} + h_{\text{hydraulic depth}}\right]_{\text{pool}} \\
\text{square upstream apron corner causes velocity head loss :} \quad & h_{\text{velocity}_1} = (1 + k_e)\frac{V_1{}^2}{2g} \text{ where } k_e = 0.5
\end{aligned}$$

Thus, the energy balance shown in eq. 1 becomes:

$$H_w = y_1 + (1 + k_e)\frac{V_1{}^2}{2g} + z_1 \tag{2}$$

The entrance depth along the s_2 profile of the steep portion is set as just below y_c. That is,

$$y_1 \approx y_c \tag{3}$$

This, in addition to a researched square-corner coefficient of $k_e = 0.5$, turns equation 2 into:

$$H_w = y_c + 1.5\frac{V_c{}^2}{2g} - z_1 \tag{4}$$

Where critical velocity can be solved assuming a rectangular cross section:

$$V_c = \frac{Q}{by_c} \tag{5}$$

That is, eq. 4 can be expressed:

$$H_w = y_c + 1.5\frac{\left(\frac{Q}{by_c}\right)^2}{2g} - z_1 \tag{6}$$

Also, given the critical conditions assumed at the start of the s_2 portion, the following constraint can be used:

$$Fr_1 = 1 = \frac{V_c}{\sqrt{gy_c}} \tag{7}$$

The critical depth can be isolated from eq. 7 as:

$$y_c = \left(\frac{Q}{b\sqrt{g}}\right)^{\frac{2}{3}} \tag{8}$$

Eq. 6 is then specified to the given max pool elevation as follows:

$$H_{w_{max}} = y_{c_{max}} + 1.5\frac{\left(\frac{Q_{max}}{by_{c_{max}}}\right)^2}{2g} - z_1$$

Matlab® solver then calculates both $y_{c_{max}}$ and Q_{max} with known $H_{w_{max}}$ using eqs. 6 and 8 to yield:

$$\boxed{y_{1_{max}} = y_{c_{max}} = 6.411\ ft; \qquad Q_{max} = 1796.4\ \frac{ft^3}{s}}$$

a.1.2) Verification of Slope Steepness

This relevance of calculating $y_{c_{max}}$ is based on the assumption that the slope was is steep. This is confirmed by calculating $y_{n_{max}}$, the normal depth associated with max pool elevation using:

$$Q = \frac{\phi}{n} A_n R_n^{\frac{2}{3}} \sqrt{s_0} \tag{9}$$

Where:

$$\text{for English units}: \quad \phi = 1.486$$

$$\text{normal cross sectional area is calculated}: \quad A_n = by_n \tag{10}$$

$$\text{normal hydraulic radius is calculated}: \quad R_n = \frac{A_n}{P_n} = \frac{by_n}{b + 2y_n} \tag{11}$$

$$\text{bed Slope from brink to sill is calculated}: \quad s_0 = \frac{z_1 - z_2}{L_{1,2}} \tag{12}$$

Eq. 9 is then specified to the calculated max discharge as follows:

$$Q_{max} = \frac{\phi}{n} A_{n_{max}} R_{n_{max}}^{\frac{2}{3}} \sqrt{s_0}$$

Matlab® solver then calculates $y_{n_{max}}$ with known Q_{max} to yield:

$$\boxed{y_{n_{max}} = 1.901\ ft}$$

Therefore, the normal depth is less than the critical depth for the maximum discharge:

$$\boxed{y_c > y_n}$$

This superiority is assumed for all discharges less than maximum as well. Therefore, the steepness of the slope is confirmed:

$$\boxed{\text{slope from brink to sill confirmed steep}} \tag{13}$$

a.1.3) Determination of Maximum Sill Depth

With $y_{1_{max}}$ known, assumed to be equal to $y_{c_{max}}$, the maximum hydraulic depth at the sill, $y_{2_{max}}$, can be calculated. This is done using the Direct Step Method, which uses anticipated head loss over the slope to determine a relation between pre and post slope energies. Given the relatively short horizontal slope length of $L_{1,2}$, only one iteration of the direct step method is used:

$$dl = L_{1,2} = \frac{E_2 - E_1}{s_0 - \overline{s_f}} \tag{14}$$

Where:

$$\text{the pre-slope, brink energy is calculated}: \quad E_1 = y_{1_{max}} + \frac{{V_{1_{max}}}^2}{2g} \tag{15}$$

$$\text{the post-slope, sill energy is calculated}: \quad E_2 = y_{2_{max}} + \frac{{V_{2_{max}}}^2}{2g} \tag{16}$$

$$\text{bed slope is calculated using 12}: \quad S_0 = \frac{z_1 - z_2}{L_{1,2}}$$

$$\text{average friction slope is calculated}: \quad \overline{s_f} = \frac{sf_1 + sf_2}{2} \tag{17}$$

$$\text{where friction slope at brink is calculated}: \quad sf_1 = \left[\frac{nQ_{max}}{(A_{1_{max}})(R_{1_{max}})^{\frac{2}{3}}}\right]^2 \tag{18}$$

$$\text{and friction slope at sill is calculated}: \quad sf_2 = \left[\frac{nQ_{max}}{(A_{2_{max}})(R_{2_{max}})^{\frac{2}{3}}}\right]^2 \tag{19}$$

Matlab® solver then calculates $y_{2_{max}}$ with known $y_{1_{max}}$ and Q_{max} to yield:

$$\boxed{y_{\text{sill}_{max}} = y_{2_{max}} = 4.158\ ft}$$

Given that $G_{0_{max}} = 6\ ft$, it is determined that even under maximum flow conditions, the hydraulic depth at the sill depth does not surpass the height of the bottom gate lip. That is, when the gate is fully opened, the system has uncontrolled flow.

a.1.4) Uncontrolled Headwater Elevation vs. Discharge Curve

With this uncontrolled condition established, the rating curve for stage (H_w) vs. uncontrolled discharge for a range of pool elevations can be developed using the following process:

- An array of H_w is created between $H_w = z_1$ and $H_{w_{max}}$.
- The process detailed in section a.1.1 is repeated for each H_w value to find each's corresponding y_c.
- This y_c is then set as the brink y_1.
- Eq. 5 is rearranged to calculate discharge from each of these y_1 values:

$$Q = b\,(y_c)^{\frac{3}{2}}\,(g)^{\frac{1}{2}} = b\,(y_1)^{\frac{3}{2}}\,\sqrt{g} \tag{20}$$

These steps are carried produce the values shown in table 1, which are plotted as the uncontrolled rating curve shown in figure 3.

$\mathbf{H_w}$ (ft)	$\mathbf{H_w - z_{apron}}$ (ft)	**Q** $\left(\frac{ft^3}{s}\right)$
1174.1	0.00	0.000
1174.5	0.39	11.503
1174.9	0.77	32.534
1175.3	1.16	59.770
1175.6	1.55	92.021
1176.0	1.93	128.600
1176.4	2.32	169.050
1176.8	2.71	213.030
1177.2	3.10	260.280
1177.6	3.48	310.570
1178.0	3.87	363.750
1178.4	4.26	419.650
1178.7	4.64	478.160
1179.1	5.03	539.150
1179.5	5.42	602.550
1179.9	5.80	668.240
1180.3	6.19	736.170
1180.7	6.58	806.250
1181.1	6.96	878.430
1181.5	7.35	952.640
1181.8	7.74	1028.800
1182.2	8.12	1106.900
1182.6	8.51	1186.900
1183.0	8.90	1268.800
1183.4	9.29	1352.400
1183.8	9.67	1437.800
1184.2	10.06	1525.000
1184.5	10.45	1613.800
1184.9	10.83	1704.300
1185.3	11.22	1796.400

Table 1: Uncontrolled rating values

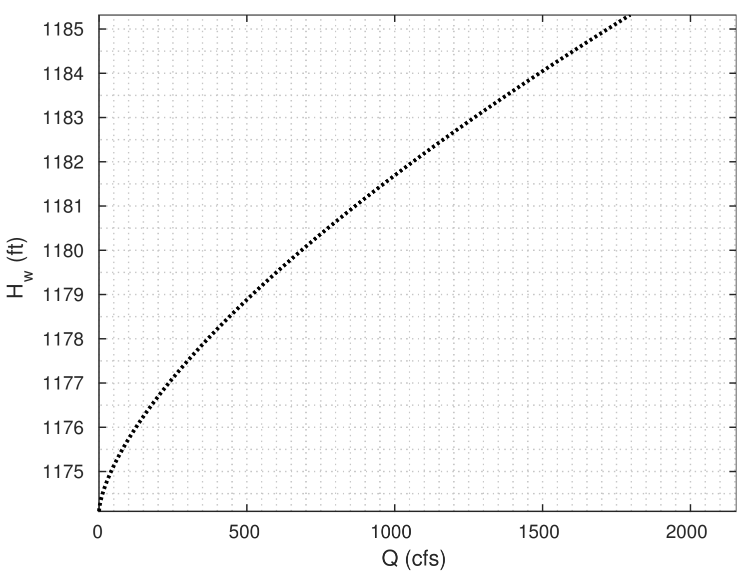

Figure 3: Uncontrolled, unsubmerged rating curve

a.2) Controlled Discharge

While controlled flow is not relevant when the gate lip is 6 ft above the sill, it does become worth considering for all openings for which the gate lip is below maximum hydraulic depth at sill. That is:

$$\text{controlled flow possible for :} \quad G_0 < y_{2_{max}}$$

Therefore, stage-discharge curves for controlled flow need to be developed for several gate openings below this max sill depth. As a cursory inspection, the rating curves for the following gate openings are to be inspected:

$$G_0 = 0.5\ f; \quad 1.0\ ft; \quad 2.0\ ft; \quad 4.0\ ft;$$

a.2.1) Reading Discharge Coefficient

The US Army Corps of Engineers provides the following figure presenting the formula and discharge coefficient for a range of gate opening angles:

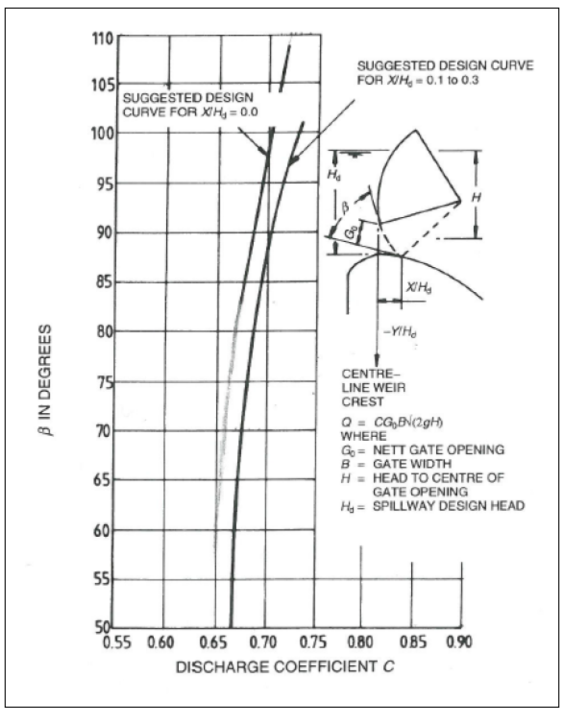

Figure 4: Discharge Coefficient for Tainter Gates (USACE)

Figure 4 is interpreted for the given scenario as follows:

- H is calculated as the difference between the pool elevation and the vertical center of the gate clearance:

$$H = \left(H_w - z_2 - \frac{1}{2}G_0\right) \tag{21}$$

- With this, the equation shown for controlled, (unsubmerged) discharge becomes:

$$Q_{cus} = CG_0b\sqrt{2g\left(H_w - z_2 - \frac{1}{2}G_0\right)} \tag{22}$$

- Discharge coefficient C is read from the figure, where β is calculated as follows:
  - As shown in figure 5, the shaft elevation (z_{shaft}), gate opening (G_0), gate radius (r) and sill elevation (z_2) are all used to develop a function of β in terms of G_0.

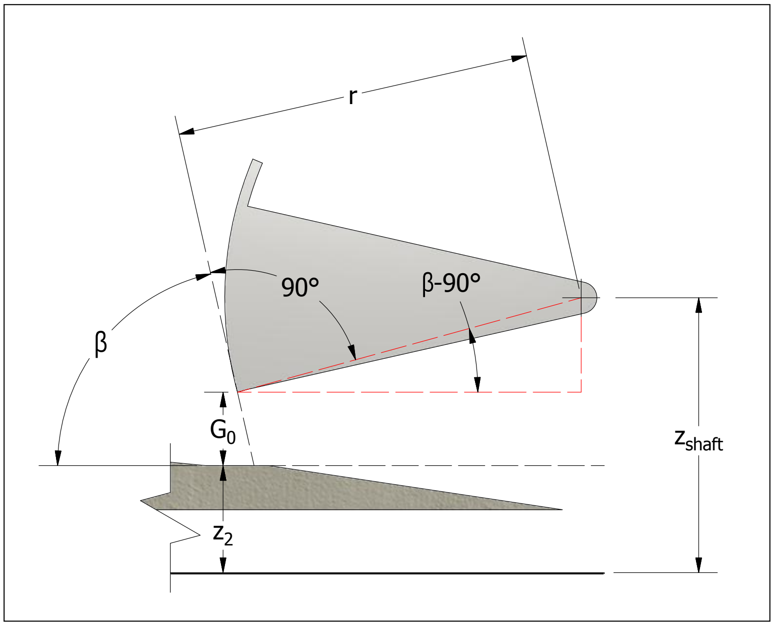

Figure 5: β calculation detail

  - A right triangle is drawn, as shown in red, for which the hypotenuse is r and both the acute angle and shorter leg length can be equated:

$$\sin(\beta - 90^\circ) = \frac{z_{\text{shaft}} - G_0 - z_2}{r} \tag{23}$$

  - This can be simplified, knowing $\sin(\theta - 90^\circ) = \cos(\theta)$

$$\cos(\beta) = \frac{z_{\text{shaft}} - G_0 - z_2}{r} \tag{24}$$

  - β can then be isolated:

$$\beta = \cos^{-1}\left(\frac{z_{\text{shaft}} - G_0 - z_2}{r}\right) \tag{25}$$

This process yields β values for each gate opening which are then used to read C values from figure 4 along the $^{X}/_{H_d} = 0$ curve. This produces the values shown in table 2.

$\mathbf{G_0}$	β	**C**
0.5	63.821	0.657
1	65.684	0.658
2	69.333	0.661
4	76.391	0.668

Table 2: β and discharge coefficient results for each gate opening

a.3) Unsubmerged, Controlled Headwater Elevation vs. Discharge Curves

a.3.1) Raw Values

Using eq. 22 along with the discharge coefficient procedure detailed in section a.2.1, the discharge is calculated for each pool elevation of the previously used H_w array for each gate opening. This process produces the values of controlled, unsubmerged discharge (Q_{cus}) shown in table 3, which also includes the uncontrolled discharge (Q_{uc}) for comparison. These rating values are then used to plot the curves shown in figure 6.

	G_0	0.5 ft	1.0 ft	2.0 ft	4.0 ft	$> y_{2_{max}}$
	β	63.821°	65.684°	69.333°	76.391°	-
	C	0.657	0.658	0.661	0.668	-
H_w	$H_w - z_2$	$Q_{cus,\ 0.5\ ft}$	$Q_{cus,\ 1.0\ ft}$	$Q_{cus,\ 2.0\ ft}$	$Q_{cus,\ 4.0\ ft}$	Q_{uc}
1174.10	2.30	73.573	138.170	235.870	229.010	0.000
1174.50	2.69	80.216	152.290	268.690	346.530	11.503
1174.90	3.07	86.350	165.210	297.910	433.270	32.534
1175.30	3.46	92.075	177.200	324.510	505.330	59.770
1175.60	3.85	97.465	188.420	349.090	568.330	92.021
1176.00	4.23	102.570	199.010	372.050	625.010	128.600
1176.40	4.62	107.440	209.070	393.680	676.960	169.050
1176.80	5.01	112.090	218.660	414.170	725.200	213.030
1177.20	5.40	116.560	227.850	433.700	770.430	260.280
1177.60	5.78	120.860	236.680	452.390	813.140	310.570
1178.00	6.17	125.020	245.200	470.330	853.720	363.750
1178.40	6.56	129.040	253.430	487.620	892.450	419.650
1178.70	6.94	132.940	261.400	504.310	929.580	478.160
1179.10	7.33	136.730	269.130	520.470	965.270	539.150
1179.50	7.72	140.410	276.650	536.140	999.690	602.550
1179.90	8.10	144.000	283.970	551.360	1033.000	668.240
1180.30	8.49	147.510	291.100	566.180	1065.200	736.170
1180.70	8.88	150.930	298.070	580.620	1096.500	806.250
1181.10	9.26	154.280	304.870	594.700	1126.900	878.430
1181.50	9.65	157.550	311.530	608.470	1156.500	952.640
1181.80	10.04	160.760	318.050	621.920	1185.400	1028.800
1182.20	10.43	163.910	324.430	635.100	1213.600	1106.900
1182.60	10.81	167.000	330.700	648.000	1241.200	1186.900
1183.00	11.20	170.030	336.840	660.650	1268.100	1268.800
1183.40	11.59	173.010	342.880	673.070	1294.500	1352.400
1183.80	11.97	175.940	348.810	685.260	1320.400	1437.800
1184.20	12.36	178.810	354.640	697.230	1345.800	1525.000
1184.50	12.75	181.650	360.380	709.010	1370.700	1613.800
1184.90	13.13	184.440	366.030	720.590	1395.100	1704.300
1185.32	13.52	187.190	371.590	731.990	1419.100	1796.400

Table 3: Lip-controlled rating values, also showing uncontrolled

a.3.2) Raw Curves

The values developed in table 3 are shown in figure 6.

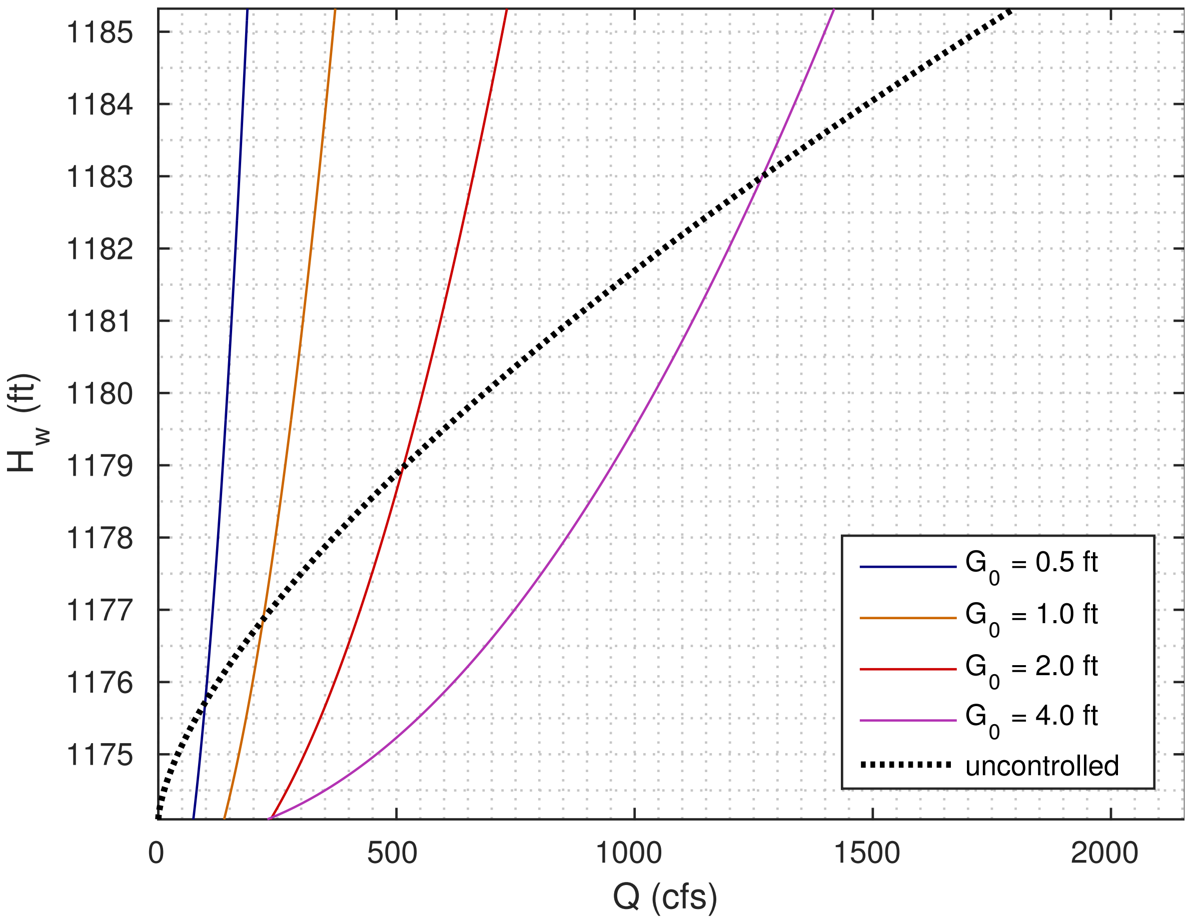

Figure 6: Lip-controlled rating curves, also showing uncontrolled

a.3.3) Functional Rating Values

At each gate opening, the functional, unsubmerged discharge $\left(Q_{fus}\right)$ for any pool elevation is the lesser of that of the controlled and uncontrolled. That is:

$$Q_{fus}\left(H_w\right) = \min\left[Q_{cus}\left(H_w\right), \quad Q_{uc}\left(H_w\right)\right] \tag{26}$$

The functional rating curves shown in table 4 are developed for each gate opening, and plotted in figure 7.

	$\mathbf{G_0}$	**0.5 ft**	**1.0 ft**	**2.0 ft**	**4.0 ft**
	β	63.821°	65.684°	69.333°	76.391°
	C	0.657	0.658	0.661	0.668
$\mathbf{H_w}$	$\mathbf{H_w - z_2}$	$\mathbf{Q_{fus,\,0.5\,ft}}$	$\mathbf{Q_{fus,\,1.0\,ft}}$	$\mathbf{Q_{fus,\,2.0\,ft}}$	$\mathbf{Q_{fus,\,4.0\,ft}}$
1174.10	2.30	0.000	0.000	0.000	0.000
1174.50	2.69	11.503	11.503	11.503	11.503
1174.90	3.07	32.534	32.534	32.534	32.534
1175.30	3.46	59.770	59.770	59.770	59.770
1175.60	3.85	92.021	92.021	92.021	92.021
1176.00	4.23	102.570	128.600	128.600	128.600
1176.40	4.62	107.440	169.050	169.050	169.050
1176.80	5.01	112.090	213.030	213.030	213.030
1177.20	5.40	116.560	227.850	260.280	260.280
1177.60	5.78	120.860	236.680	310.570	310.570
1178.00	6.17	125.020	245.200	363.750	363.750
1178.40	6.56	129.040	253.430	419.650	419.650
1178.70	6.94	132.940	261.400	478.160	478.160
1179.10	7.33	136.730	269.130	520.470	539.150
1179.50	7.72	140.410	276.650	536.140	602.550
1179.90	8.10	144.000	283.970	551.360	668.240
1180.30	8.49	147.510	291.100	566.180	736.170
1180.70	8.88	150.930	298.070	580.620	806.250
1181.10	9.26	154.280	304.870	594.700	878.430
1181.50	9.65	157.550	311.530	608.470	952.640
1181.80	10.04	160.760	318.050	621.920	1028.800
1182.20	10.43	163.910	324.430	635.100	1106.900
1182.60	10.81	167.000	330.700	648.000	1186.900
1183.00	11.20	170.030	336.840	660.650	1268.100
1183.40	11.59	173.010	342.880	673.070	1294.500
1183.80	11.97	175.940	348.810	685.260	1320.400
1184.20	12.36	178.810	354.640	697.230	1345.800
1184.50	12.75	181.650	360.380	709.010	1370.700
1184.90	13.13	184.440	366.030	720.590	1395.100
1185.32	13.52	187.190	371.590	731.990	1419.100

Table 4: Fuctional rating values, unsubmerged

a.3.4) Functional Rating Curves

The values developed in table 4 are shown in figure 7.

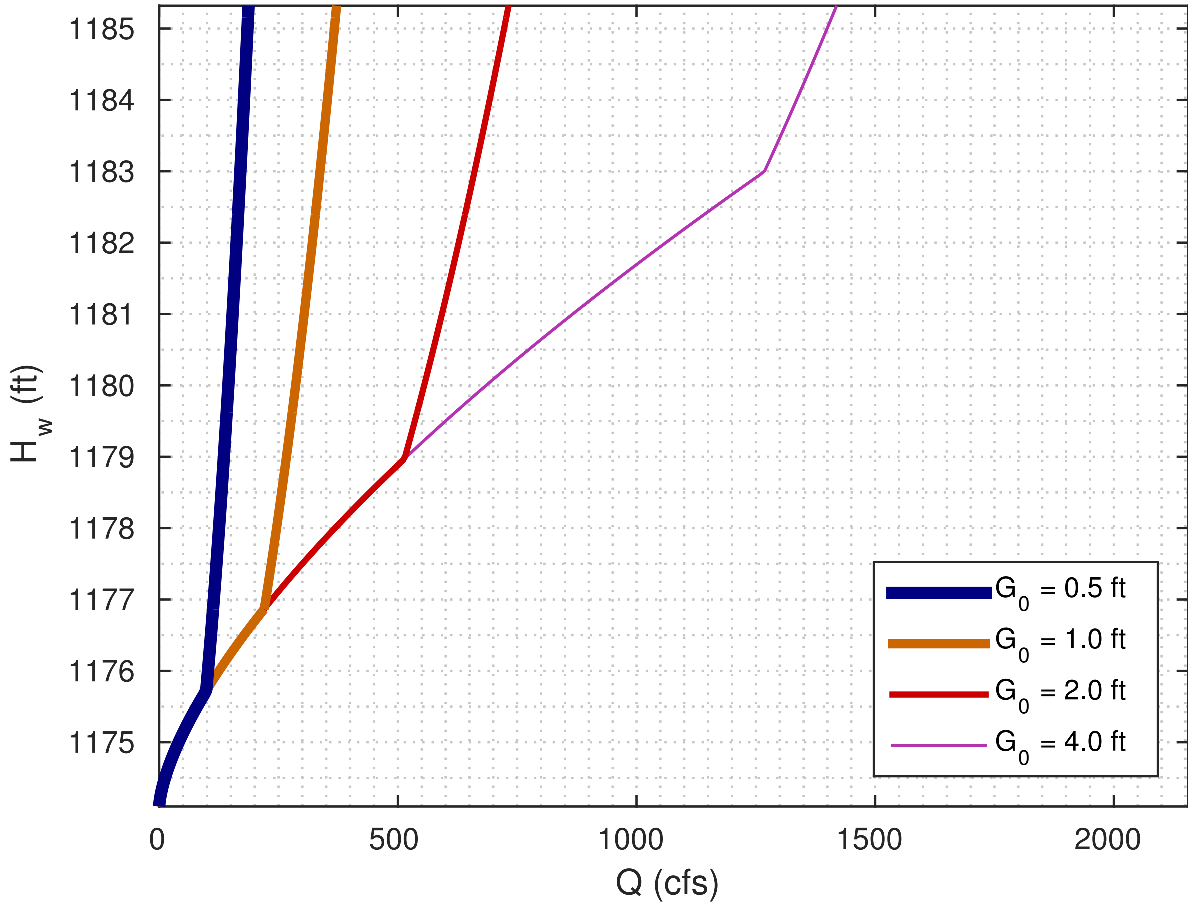

Figure 7: Functional rating curves, unsubmerged

b) Considering Tailwater Effects

b.1) Given Scenario

If there is a tailwater elevation that causes the downstream portion of the gate to become submerged, as shown in figure 8, a slightly different approach is required to determine the controlled discharge for each gate opening.

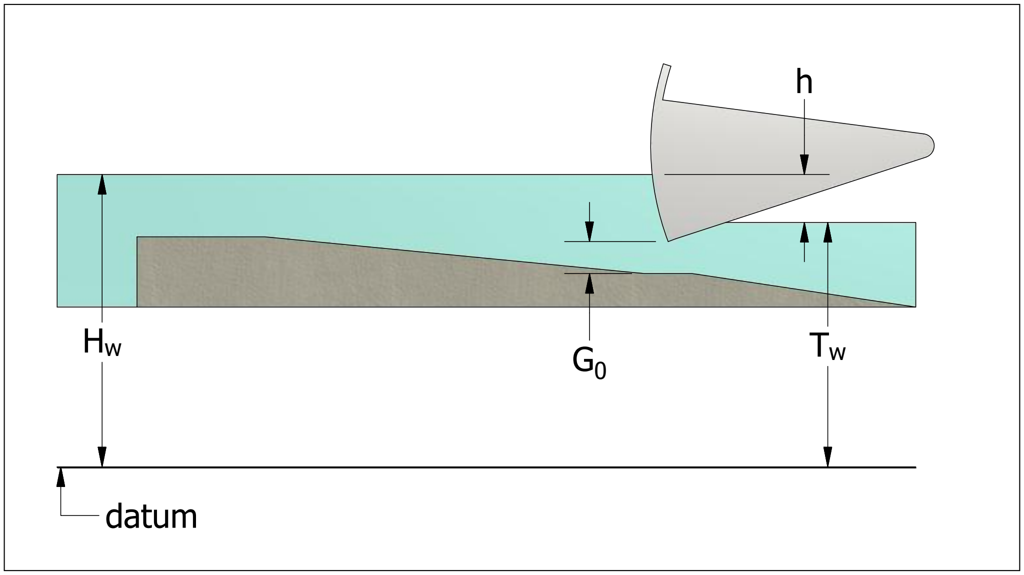

Figure 8: Submergence scenario

This figure presents the approximations that are used in subsequent calculations:

- Upstream velocity head is negligible.
- Under submerged conditions, the water surface remains constant from the upstream pool, all the way up to the upstream gate face.
- There is no contraction of flow through the gate opening. That is, the water surface touching the downstream gate face is approximated to be equal to the tailwater elevation.

The client provides a tailwater elevation to be analysed of $T_w = 1175\ ft$. In order to modify the rating curves considering this effect, a submerged, controlled discharge Q_{cs} is to be calculated.

With this, the submerged, functional discharge $\left(Q_{fs}\right)$ at each value of the H_w array is calculated again as the minimum among controlled submerged, controlled unsubmerged and uncontrolled.

$$Q_{fs}\left(H_w\right) = \min\left[Q_{cs}\left(H_w\right), \quad Q_{cus}\left(H_w\right), \quad Q_{uc}\left(H_w\right)\right] \tag{27}$$

b.2) Modified Discharge for Submergence

The same USACE publication for gates provides the following chart for calculating such discharge:

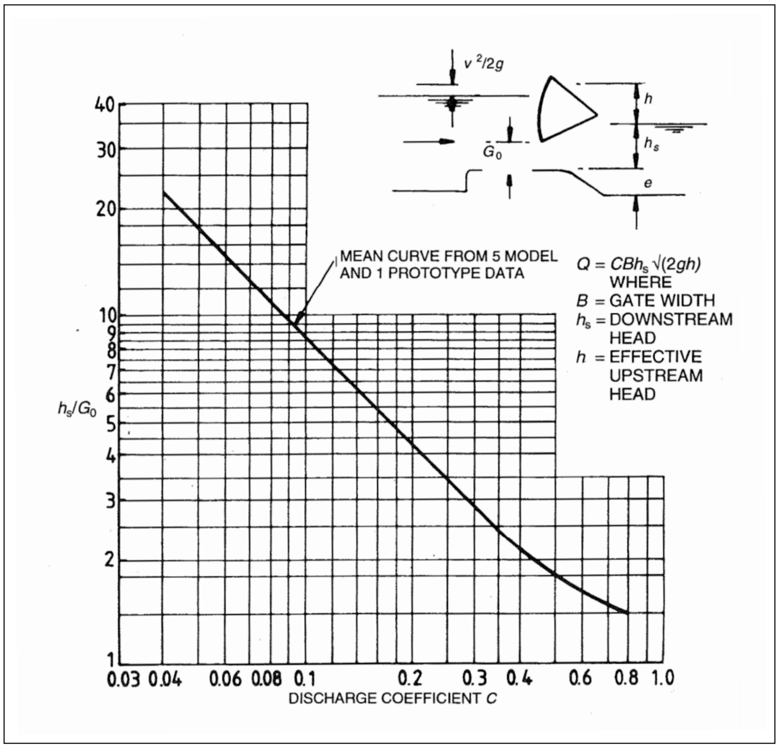

Figure 9: Coefficient of discharge under submerged flow conditions

This figure provides the following equation:

$$Q = CBh_s\sqrt{2gh} \tag{28}$$

However, it can be seen that the given curve for $\frac{h_s}{G_0} = f(C)$ is approximately linear for a a significant range:

$$C \approx \frac{0.85}{h_s/G_0} \quad \text{for} \quad 2.5 < \frac{h_S}{G_0} < 25 \tag{29}$$

Therefore, eq. 28 can be simplified using eq. 29 as follows:

$$Q = 0.85G_0B\sqrt{2gh} \quad \text{for} \quad 2.5 < \frac{h_S}{G_0} < 25 \tag{30}$$

This equation for controlled, submerged discharge (Q_{cs}) is adopted into local notation as:

$$Q_{cs} = 0.85G_0b\sqrt{2g\left(H_w - T_w\right)} \quad \text{for} \quad 2.5 < \frac{T_w - z_2}{G_0} < 25 \tag{31}$$

b.3) Submerged, Controlled Headwater Elevation vs. Discharge Curves

b.3.1) Raw Values

Eq. 31 is used to calculate a Q_{cs} for each value of the previously used H_w array, as shown in table 5. It should be noted that for all values of H_w below T_w, (for which $H_w - T_w$ are negative), the actual value of $H_w - T_w$ is set to zero, rendering the corresponding controlled discharge value zero. This process is assumed valid given that in reality, a T_w greater than H_w would cause flow upstream. This is a scenario in which the investigators take no interest.

Previously calculated values of controlled, unsubmerged discharge are also shown immediately after their submerged counterparts for comparison.

H_w	$H_w - T_w$	Q_{cs}, 0.5 ft	Q_{cus}, 0.5 ft	Q_{cs}, 1.0 ft	Q_{cus}, 1.0 ft	Q_{cs}, 2.0 ft	Q_{cus}, 2.0 ft	Q_{cs}, 4.0 ft	Q_{cus}, 4.0 ft	Q_{uc}
1174.10	0.00	0.000	73.573	0.000	138.170	0.000	235.870	0.000	229.010	0.000
1174.50	0.00	0.000	80.216	0.000	152.290	0.000	268.690	0.000	346.530	11.503
1174.90	0.00	0.000	86.350	0.000	165.210	0.000	297.910	0.000	433.270	32.534
1175.30	0.26	33.957	92.075	67.914	177.200	135.830	324.510	271.660	505.330	59.770
1175.60	0.65	53.520	97.465	107.040	188.420	214.080	349.090	428.160	568.330	92.021
1176.00	1.03	67.644	102.570	135.290	199.010	270.580	372.050	541.150	625.010	128.600
1176.40	1.42	79.291	107.440	158.580	209.070	317.160	393.680	634.320	676.960	169.050
1176.80	1.81	89.433	112.090	178.870	218.660	357.730	414.170	715.470	725.200	213.030
1177.20	2.20	98.537	116.560	197.070	227.850	394.150	433.700	788.300	770.430	260.280
1177.60	2.58	106.870	120.860	213.740	236.680	427.470	452.390	854.950	813.140	310.570
1178.00	2.97	114.600	125.020	229.190	245.200	458.380	470.330	916.770	853.720	363.750
1178.40	3.36	121.830	129.040	243.670	253.430	487.340	487.620	974.670	892.450	419.650
1178.70	3.74	128.670	132.940	257.330	261.400	514.660	504.310	1029.300	929.580	478.160
1179.10	4.13	135.150	136.730	270.300	269.130	540.610	520.470	1081.200	965.270	539.150
1179.50	4.52	141.340	140.410	282.680	276.650	565.370	536.140	1130.700	999.690	602.550
1179.90	4.90	147.270	144.000	294.540	283.970	589.080	551.360	1178.200	1033.000	668.240
1180.30	5.29	152.970	147.510	305.940	291.100	611.880	566.180	1223.800	1065.200	736.170
1180.70	5.68	158.470	150.930	316.930	298.070	633.860	580.620	1267.700	1096.500	806.250
1181.10	6.06	163.780	154.280	327.550	304.870	655.110	594.700	1310.200	1126.900	878.430
1181.50	6.45	168.920	157.550	337.840	311.530	675.680	608.470	1351.400	1156.500	952.640
1181.80	6.84	173.910	160.760	347.820	318.050	695.650	621.920	1391.300	1185.400	1028.800
1182.20	7.22	178.760	163.910	357.530	324.430	715.060	635.100	1430.100	1213.600	1106.900
1182.60	7.61	183.490	167.000	366.980	330.700	733.950	648.000	1467.900	1241.200	1186.900
1183.00	8.00	188.090	170.030	376.190	336.840	752.370	660.650	1504.700	1268.100	1268.800
1183.40	8.39	192.590	173.010	385.180	342.880	770.360	673.070	1540.700	1294.500	1352.400
1183.80	8.77	196.980	175.940	393.960	348.810	787.930	685.260	1575.900	1320.400	1437.800
1184.20	9.16	201.280	178.810	402.560	354.640	805.110	697.230	1610.200	1345.800	1525.000
1184.50	9.55	205.490	181.650	410.970	360.380	821.940	709.010	1643.900	1370.700	1613.800
1184.90	9.93	209.610	184.440	419.220	366.030	838.430	720.590	1676.900	1395.100	1704.300
1185.30	10.32	213.650	187.190	427.300	371.590	854.610	731.990	1709.200	1419.100	1796.400

Table 5: Submergence-controlled rating values, also showing controlled submerged and uncontrolled

b.3.2) Raw Curves

The values developed in table 5 are shown in figure 10.

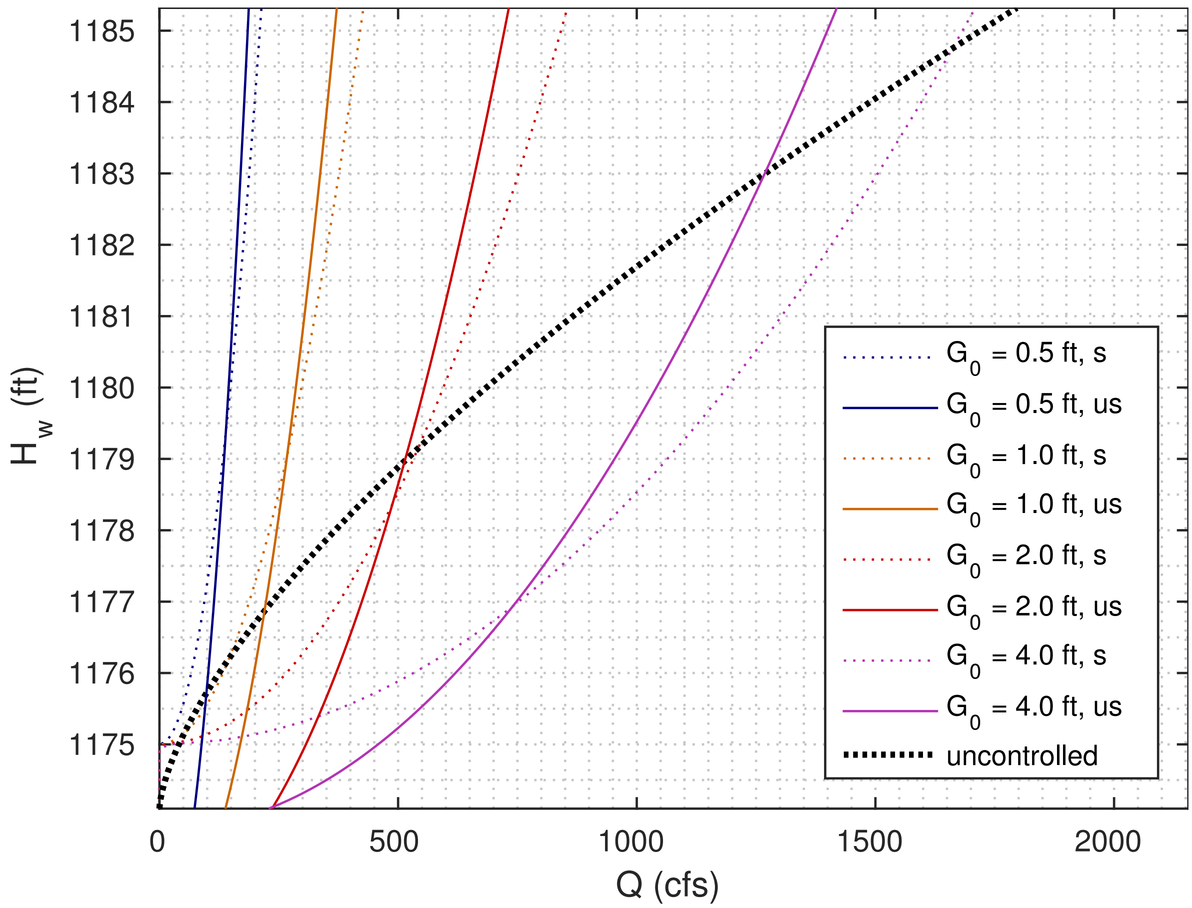

Figure 10: Submergence-controlled rating curves, including unsubmerged and uncontrolled

b.3.3) Functional Rating Values

The functional rating curves considering submergence effects are developed using eq. 27. That is, for each H_w value, the least-valued discharge is taken from either Q_{cs}, Q_{cus} or Q_{uc} from table 5.

This produces the functional rating values, considering submergence, shown in table 6 and plotted in figure 11.

	$\mathbf{G_0}$	**0.5 ft**	**1.0 ft**	**2.0 ft**	**4.0 ft**
$\mathbf{H_w}$	$\mathbf{H_w - z_2}$	$\mathbf{Q_{fs,\ 0.5\ ft}}$	$\mathbf{Q_{fs,\ 1.0\ ft}}$	$\mathbf{Q_{fs,\ 2.0\ ft}}$	$\mathbf{Q_{fs,\ 4.0\ ft}}$
1174.10	0.00	0.000	0.000	0.000	0.000
1174.50	0.00	0.000	0.000	0.000	0.000
1174.90	0.00	0.000	0.000	0.000	0.000
1175.30	0.26	33.957	59.770	59.770	59.770
1175.60	0.65	53.520	92.021	92.021	92.021
1176.00	1.03	67.644	128.600	128.600	128.600
1176.40	1.42	79.291	158.580	169.050	169.050
1176.80	1.81	89.433	178.870	213.030	213.030
1177.20	2.20	98.537	197.070	260.280	260.280
1177.60	2.58	106.870	213.740	310.570	310.570
1178.00	2.97	114.600	229.190	363.750	363.750
1178.40	3.36	121.830	243.670	419.650	419.650
1178.70	3.74	128.670	257.330	478.160	478.160
1179.10	4.13	135.150	269.130	520.470	539.150
1179.50	4.52	140.410	276.650	536.140	602.550
1179.90	4.90	144.000	283.970	551.360	668.240
1180.30	5.29	147.510	291.100	566.180	736.170
1180.70	5.68	150.930	298.070	580.620	806.250
1181.10	6.06	154.280	304.870	594.700	878.430
1181.50	6.45	157.550	311.530	608.470	952.640
1181.80	6.84	160.760	318.050	621.920	1028.800
1182.20	7.22	163.910	324.430	635.100	1106.900
1182.60	7.61	167.000	330.700	648.000	1186.900
1183.00	8.00	170.030	336.840	660.650	1268.100
1183.40	8.39	173.010	342.880	673.070	1294.500
1183.80	8.77	175.940	348.810	685.260	1320.400
1184.20	9.16	178.810	354.640	697.230	1345.800
1184.50	9.55	181.650	360.380	709.010	1370.700
1184.90	9.93	184.440	366.030	720.590	1395.100
1185.30	10.32	187.190	371.590	731.990	1419.100

Table 6: Functional rating values, submerged

b.3.4) Functional Rating Curves

The values developed in table 6 are shown in figure 11.

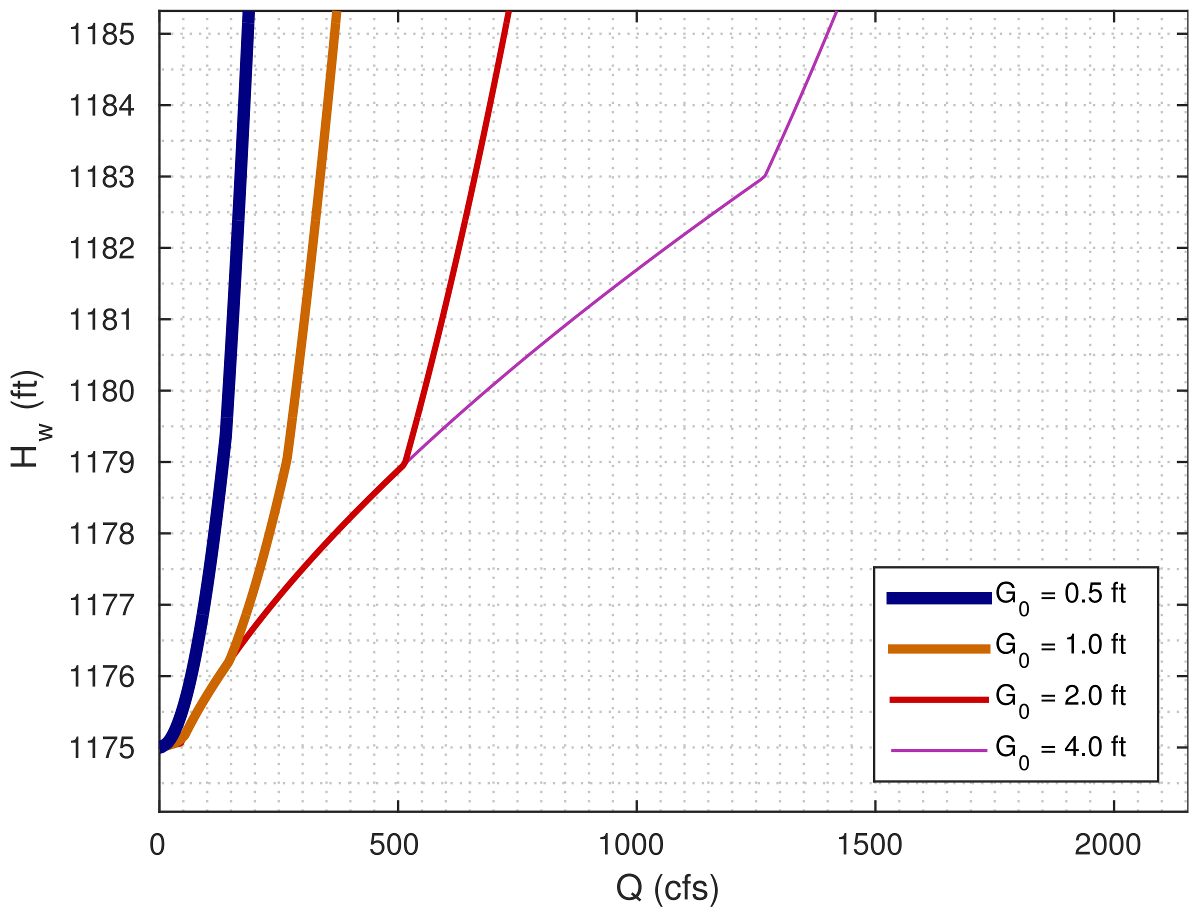

Figure 11: Functional rating curves, considering submergence

A Matlab® Scripts

A.1 Master Script

```matlab
clc; close all; clear all
% --- given properties ---------------------------------------------------------
b = 19.5;                                                  %Tainter gate width [ft]
r = 17;                                                   %Tainter gate radius [ft]
uAE = 1174.1;                                        %upstream apron elevation [ft]
maxWSE = 1185.32;                              %max water surface elevation [ft]
minGLE = 1171.8;                                      %min gate lip elevation [ft]
maxGLE = 1177.8;                                      %max gate lip elevation [ft]
Lbrink_to_lip = 9 + 14 + 9/12;                   %length from brink to gate [ft]
SHE = 1179.8;                                                %shaft elevation [ft]
n = 0.013;                                                           %Manning's n
g = 32.2;                                        %acceleration of gravity [ft/s^2]
ke = 0.5;                                               %apron corner coefficient

% --- preliminary calculations -------------------------------------------------
maxGO = maxGLE - minGLE;                                   %max gate opening [ft]
s0  = (uAE-minGLE)/(Lbrink_to_lip);                       %upstream bed slope [%]

% --- uncontrolled test --------------------------------------------------------
maxHw = maxWSE - uAE;                                         %max headwater [ft]
syms QQ
yc = (QQ/(b*sqrt(g)))^(2/3);
maxQ = double(vpasolve(yc + (1+ke)*((QQ/(yc*b))^2)/(2*g) == maxHw,QQ));
maxyc = (maxQ/(b*sqrt(g)))^(2/3);
maxyn = yn_of_Q(maxQ,n,b,s0);
assert(maxyc > maxyn,'slope not steep');
maxy2 = y2_of_y1_and_Q_ds(maxQ,maxyc,n,b,s0,g,Lbrink_to_lip);

% --- Hw / uncontrolled discharge curves ---------------------------------------
NHw = 30;
Hw_array= linspace(0, maxHw, NHw);
Q_uc_array = zeros(1,length(Hw_array));
for ii = 1:length(Hw_array)
syms QQ
yc = (QQ/(b*sqrt(g)))^(2/3);
Q_uc_array(ii)=...
      double(vpasolve(yc+(1+ke)*((QQ/(yc*b))^2)/(2*g)==Hw_array(ii),QQ));
disp(ii);
end
curve_tab_00 = zeros(length(Hw_array), 3);
curve_tab_00(:,1) = Hw_array + uAE;
curve_tab_00(:,2) = Hw_array;
curve_tab_00(:,3) = Q_uc_array;

figure;
plot( Q_uc_array, Hw_array + uAE,':k','linewidth',2); grid minor
ylabel('H_w (ft)'); xlabel('Q (cfs)');
axis([0 1.2*max(Q_uc_array) min(Hw_array + uAE) max(Hw_array + uAE)]);

% --- controlled discharge for each gate opening -------------------------------
G0_array = [0.5 1 2 4];
C_array = zeros(1,length(G0_array));
beta_array = zeros(1,length(G0_array));
curve_tab_01 = zeros(NHw,length(G0_array)+3);
for jj = 1:length(G0_array)
 [beta_array(jj), C_array(jj)] = func_beta_and_C_from_G0(...
                   G0_array(jj), r, minGLE, SHE, 0); %store C, beta values
end
for ii = 1:NHw
curve_tab_01(ii,1) = Hw_array(ii) + uAE;                     %pool elevation [ft]
curve_tab_01(ii,2) = curve_tab_01(ii,1)-minGLE;                 %head at sill [ft]
curve_tab_01(ii,length(G0_array)+3) = Q_uc_array(ii);
for jj = 1:length(G0_array)
curve_tab_01(ii,jj+2) = C_array(jj)*G0_array(jj)...
*b*sqrt(2*g*(curve_tab_01(ii,2)-0.5*G0_array(jj))); %store discharge values
end
end

figure;
plot(curve_tab_01(:,3),curve_tab_01(:,1),'color',[0   0   0.5]);
hold on, grid minor
plot(curve_tab_01(:,4),curve_tab_01(:,1),'color',[0.8 0.4   0]);
plot(curve_tab_01(:,5),curve_tab_01(:,1),'color',[0.8   0   0]);
plot(curve_tab_01(:,6),curve_tab_01(:,1),'color',[0.7 0.2 0.7]);
plot(Q_uc_array,curve_tab_01(:,1),':k','linewidth',2);
ylabel('H_w (ft)'); xlabel('Q (cfs)');
legend('G_0 = 0.5 ft','G_0 = 1.0 ft','G_0 = 2.0 ft',...
     'G_0 = 4.0 ft','uncontrolled','location','southeast');
axis([0 1.2*max(Q_uc_array)...
                           min(curve_tab_01(:,1)) max(curve_tab_01(:,1))]);

% --- functional rating curves -------------------------------------------------
curve_tab_02 = zeros(NHw,length(G0_array)+3);
for ii = 1:NHw
curve_tab_02(ii,1) = Hw_array(ii) + uAE;                     %pool elevation [ft]
curve_tab_02(ii,2) = curve_tab_02(ii,1)-minGLE;                 %head at sill [ft]
curve_tab_02(ii,length(G0_array)+3) = Q_uc_array(ii);
for jj = 1:length(G0_array)
curve_tab_02(ii,jj+2) = min([curve_tab_01(ii,jj+2) Q_uc_array(ii)]);
end
end

figure;
h4 = plot(curve_tab_02(:,6),curve_tab_02(:,1),...
                                     'color',[0.7 0.2 0.7],'linewidth',1);
hold on, grid minor
h3 = plot(curve_tab_02(:,5),curve_tab_02(:,1),...
                                     'color',[0.8   0   0],'linewidth',2);
h2 = plot(curve_tab_02(:,4),curve_tab_02(:,1),...
                                     'color',[0.8 0.4   0],'linewidth',3);
h1 = plot(curve_tab_02(:,3),curve_tab_02(:,1),...
                                     'color',[0   0   0.5],'linewidth',4);
ylabel('H_w (ft)'); xlabel('Q (cfs)');
legend([h1 h2 h3 h4],...
{'G_0 = 0.5 ft','G_0 = 1.0 ft','G_0 = 2.0 ft','G_0 = 4.0 ft'},...
                                               'location','southeast');
axis([0 1.2*max(Q_uc_array) ...
                           min(curve_tab_01(:,1)) max(curve_tab_01(:,1))]);

% --- considering submergence --------------------------------------------------
Tw = 1175;                                                 %tailwater level [ft]
curve_tab_03 = zeros(NHw, 11);
for ii = 1:NHw
    curve_tab_03(ii,1) = Hw_array(ii)+uAE;                              %Hw [ft]
    curve_tab_03(ii,11) = Q_uc_array(ii);
    if curve_tab_03(ii,1) > Tw
    curve_tab_03(ii,2) = Hw_array(ii)+uAE-Tw;                           %hs [ft]
    else
    curve_tab_03(ii,2) = 0;
    end
    for jj = 1:length(G0_array)
    curve_tab_03(ii,2*jj+1) = ...
    0.85*b*G0_array(jj)*sqrt(2*g*curve_tab_03(ii,2));    %Q_cont, submerged
    curve_tab_03(ii,2*(jj+1)) = curve_tab_01(ii,jj+2);      %Q_cont, unsub
    end
end

figure;
aa1 =plot(curve_tab_03(:,3),  curve_tab_03(:,1),':','color',[0   0   0.5]);
hold on, grid minor
aa2 =plot(curve_tab_03(:,5),  curve_tab_03(:,1),':','color',[0.8 0.4   0]);
aa3 =plot(curve_tab_03(:,7),  curve_tab_03(:,1),':','color',[0.8   0   0]);
aa4 =plot(curve_tab_03(:,9),  curve_tab_03(:,1),':','color',[0.7 0.2 0.7]);
aa5 =plot(Q_uc_array,curve_tab_01(:,1),':k','linewidth',2);
ab1 =plot(curve_tab_03(:,4),  curve_tab_03(:,1),'color',[0   0   0.5]);
ab2 =plot(curve_tab_03(:,6),  curve_tab_03(:,1),'color',[0.8 0.4   0]);
ab3 =plot(curve_tab_03(:,8),  curve_tab_03(:,1),'color',[0.8   0   0]);
ab4 =plot(curve_tab_03(:,10), curve_tab_03(:,1),'color',[0.7 0.2 0.7]);
ylabel('H_w (ft)'); xlabel('Q (cfs)');
legend([aa1 ab1 aa2 ab2 aa3  ab3 aa4 ab4 aa5],...
      {'G_0 = 0.5 ft, s','G_0 = 0.5 ft, us',...
       'G_0 = 1.0 ft, s','G_0 = 1.0 ft, us',...
       'G_0 = 2.0 ft, s','G_0 = 2.0 ft, us',...
       'G_0 = 4.0 ft, s','G_0 = 4.0 ft, us',...
                         'uncontrolled'},'location','southeast');
axis([0 1.2*max(Q_uc_array) min(curve_tab_01(:,1)) ...
                                                  max(curve_tab_01(:,1))]);

% --- functional curves considering tailwater ----------------------------------
curve_tab_04 = zeros(NHw, 6);
for ii = 1:NHw
    curve_tab_04(ii,1) = curve_tab_03(ii,1);
    curve_tab_04(ii,2) = curve_tab_03(ii,2);
    for jj = 1:length(G0_array)
    curve_tab_04(ii,jj+2) = ...
    min([curve_tab_03(ii, 2*jj+1)...
         curve_tab_03(ii, 2*jj+2) ...
         curve_tab_03(ii,   11)]);
    end
end
curve_tab_05 = zeros(NHw, 6);
for ii = 1:NHw
    if curve_tab_03(ii,1) <= Tw
    curve_tab_05(ii,1) = Tw;
    curve_tab_05(ii,2) = 0;
    else
    curve_tab_05(ii,1) = curve_tab_03(ii,1);
    curve_tab_05(ii,2) = curve_tab_03(ii,2);
    end
    for jj = 1:length(G0_array)
    curve_tab_05(ii,jj+2) = ...
    min([curve_tab_03(ii, 2*jj+1)...
         curve_tab_03(ii, 2*jj+2) ...
         curve_tab_03(ii,   11)]);
    end
end

figure;
h6d = plot(curve_tab_05(:,6),  curve_tab_05(:,1),...
                                     'color',[0.7 0.2 0.7],'linewidth',1);
hold on, grid minor
h6c = plot(curve_tab_05(:,5),  curve_tab_05(:,1),...
                                     'color',[0.8   0   0],'linewidth',2);
h6b = plot(curve_tab_05(:,4),  curve_tab_05(:,1),...
                                     'color',[0.8 0.4   0],'linewidth',3);
h6a = plot(curve_tab_05(:,3),  curve_tab_05(:,1),...
                                     'color',[0   0   0.5],'linewidth',4);
ylabel('H_w (ft)'); xlabel('Q (cfs)');
legend([h6a h6b h6c h6d],...
{'G_0 = 0.5 ft','G_0 = 1.0 ft','G_0 = 2.0 ft','G_0 = 4.0 ft'},...
                                               'location','southeast');
axis([0 1.2*max(Q_uc_array) uAE max(Hw_array + uAE)]);
```

A.2 Function Script: $\mathbf{y_n = f(Q)}$

```
function [ yn_out ] = yn_of_Q( Q,n,b,s0 )
syms yn
An = yn*b;
Pn = 2*yn + b;
Rn = An/Pn;
yn_out = double(vpasolve((1.486/n)*An*Rn^(2/3)*sqrt(s0)==Q,yn));
end
```

A.3 Function Script: $\mathbf{y_2 = f(y_1, Q)}$

```
function [ y2_out ] = y2_of_y1_and_Q_ds(Q,y1,n,b,s0,g,L)

ql = Q/b;
E1 = y1 + ((ql/y1)^2)/(2*g);
A1 = y1*b;
P1 = 2*y1 + b;
R1 = A1/P1;
sf1 = (n*Q/(1.49*A1*R1^(2/3)))^2;

y2 = y1*0.95; tol  = 0.00000001; err = tol+1;
while err > tol
E2 = y2 + ((ql/y2)^2)/(2*g);
A2 = y2*b;
P2 = 2*y2 + b;
R2 = A2/P2;
sf2 = (n*Q/(1.49*A2*R2^(2/3)))^2;
sfbar = 0.5*(sf1 + sf2);
LL = (E2 - E1)/(s0 - sfbar);
err = L - LL;
y2 = y2 - 0.001*err;
end
y2_out = y2;
end
```

A.4 Reader Script: $\beta, \mathbf{C = f(G_0)}$

```
function [beta_out, C_out] = func_beta_and_C_from_G0(G0, r, minGLE, SHE, plot_fig )

beta = (180/pi)*acos((SHE - minGLE - G0)/r);

% --- set axis scaling ---------------------------------------------------

ori = [110 640];
ori_unscaled =[0.55 50];
x_end = 473;
x_end_unscaled = .9;
y_top = 25;
y_top_unscaled = 110;

scale_xx = (x_end - ori(1))/(x_end_unscaled(1) - ori_unscaled(1));
scale_yy = (ori(2)- y_top)/(y_top_unscaled - ori_unscaled(2));

% --- record points for interpolation -------------------------------------

int = [218 539; 261 198; 290 35; 283 75; 242 300];

% --- create matrix of interpolated values --------------------------------

 [rows, ~] = size(int);
 AA = ones(rows,rows);

 for ii = 1:rows
      for jj = 1:rows
          AA(ii,jj) = int(ii,1)^(rows-jj);
      end
 end

 CC = AA\int(:,2);

% -- create matrix of Cd values, from which the output Cd is interpolated -

N_plot = 1000;

xx_array = linspace(int(1,1),max(int(:,1)),N_plot);
yy_array = zeros(N_plot);

 for ii = 1:N_plot
      for jj = 1:rows
          yy_array(ii) = yy_array(ii) + ...
                                   CC(jj)*xx_array(ii)^(rows-jj);
        end
 end

% --- back-solve interpolated output value --------------------------------

C = 0.65;            %initially-guessed C
tol = 0.0001;
err = tol+1;
while abs(err) > tol;
xx_unscaled = (C - ori_unscaled(1));          % accounting for chart notation
xx = xx_unscaled*scale_xx + ori(1);

yy = 0;
for jj = 1:rows
          yy = yy + CC(jj)*xx^(rows-jj);
end

beta_out = y_top_unscaled - (yy - y_top)/scale_yy;
err = beta - beta_out;
C = C + err*0.001;
end
C_out = C;
% --- plot results if requested -------------------------------------------
if plot_fig == 1

    figure;
I = imread('102_b.png');
B = imrotate(I,0.5);

imshow(B); hold on;
plot(ori(1),ori(2),'or');
plot([ori(1) x_end],[ori(2) ori(2)],'-b','linewidth',2);
plot([ori(1) ori(1)],[y_top ori(2)],'-r','linewidth',2);

 for ii = 1:length(int)
    plot(int(ii,1),int(ii,2),'om','linewidth',2);
 end

plot(xx_array, yy_array,'m','linewidth',2);
plot(xx, yy,'co','linewidth',2);

%print('103','-dpng','-r600');
end
end
```