UNIVERSITY OF MINNESOTA: TWIN CITIES

CE 4511 HYDRAULIC STRUCTURES

HW 5: Stilling Basins

CHRISTOPHER BULKLEY - LOGSTON

APRIL 13TH, 2015

Contents

0) Preliminaries

0.1) Introduction

The client is a group of spillway managers. Due to extensive scouring, the group is burdened with continuous maintenance of the region immediately-downstream of the structure. Figure 0.1 shows the the spillway's current configuration in which its end acts as a weir, the discharge across which leads to erosion. This firm proposes the addition of a post-spillway stilling basin in order to mitigate these scouring effects.

Considering both the volume of concrete and excavation required for such a project, three separate basin options are considered in order to determine the most cost effective development. The first, as discussed in section 1, is a smooth, sloped face ending in a simple concrete stilling basin. The second second option, as discussed in section 2, swaps the smooth, sloped spillway for one composed of a series of energy-dissipating steps. Finally, the third option in section 3 is analyzed as an abrupt drop structure, also known as a plunge pool.

Finally, as discussed in section 4, the optimal development is decided to be the first option.

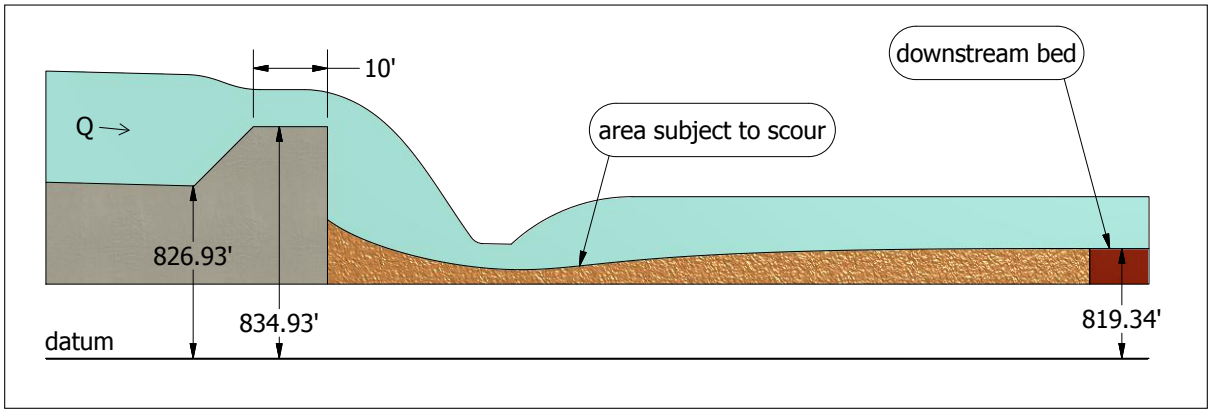

Figure 0.1: Current Scouring

0.2) Given Properties

The client's current system dimensions in addition to that shown in figure 0.1 are as follows:

- upstream bed elevation = 826.93ft
- weir crest elevation: $z_1 = 834.93ft$
- weir breadth: $b = 10ft$
- downstream bed elevation = 819.34ft
- system width, perpendicular to flow: $w = 160ft$

The client has also provided the discharges and tailwater elevations for several flood frequency events:

Flood Frequency (FF) (years)	**Discharge (Q)** (ft^3/s)	**Tailwater Elevation** $(\mathbf{T_w})$ (ft)
5	4,700	826.82
25	7,600	828.77
50	8,900	829.56
100	10,200	830.31
500	13,500	832.14

Table 0.1: Given Flood Frequency Properties

1) Option 1: Simple Smooth Sloped Spillway

A smooth, concrete spillway and stilling basin is the first system to be considered. As shown in figure 1.1, the end sill of the system currently in place will act as a weir after which a sloped surface of (2H:1V) will feed into a stilling basin. This basin will terminate in a riprap feeding into the downstream natural bed. While it is anticipated that the elevation of the basin will differ slightly from that of the downstream bed in order to contain the expected hydraulic jump, this difference is minimized in order to reduce excavation costs associated which this option.

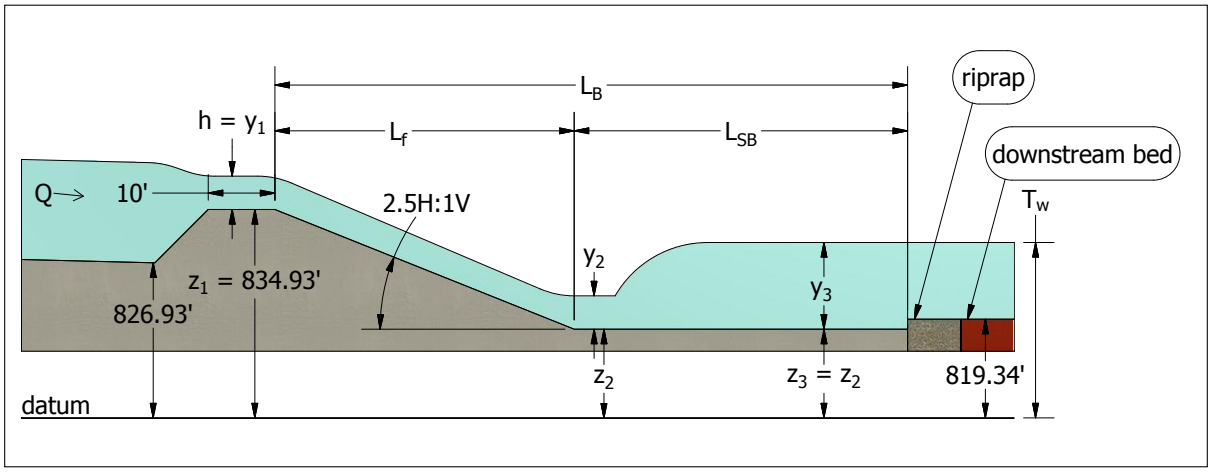

Figure 1.1: Option 1 Basin Detail

1.1) Crest Head Calculations

The formula for discharge over a broad crested weir is:

$$Q = \frac{2}{3}\sqrt{\frac{2}{3}g}\left(C_v C_d L_\perp H^{\frac{3}{2}}\right) \tag{1.1}$$

Where:

- C_v is the coefficient accounting for kinetic effects, assumed at low velocities $C_v \approx 1$;
- C_d is the coefficient of discharge for a broad crested weir;
- $L_\perp$ is the system width perpendicular to flow;
- H is total energy head at the crest;
- g is the acceleration of gravity.

As shown in appendix figure A.4, C_d is a function of not only the crest head to breadth (b) ratio, but also the head to head-dam-height-sum ratio. That is:

$$C_d = f\left(\frac{h}{b}, \frac{h}{h+P}\right)$$

Therefore, although Q is known for each flood event, h cannot simply be isolated from eq. 1.1 and calculated, given the interdependency of C_d. Therefore, the crest head (h) for each flood event is determined by using Matlab® solver to find a value of h for each Q in which the corresponding C_d matches both the $\frac{h}{b}$ as well as the $\frac{h}{h+P}$ parameters.

As shown in figure 1.2, this is achieved by interpolating between neighboring curves of known $\frac{h}{h+P}$ values for any given $\frac{h}{b}$ value. The heads determined through this process are shown in table 0.1.

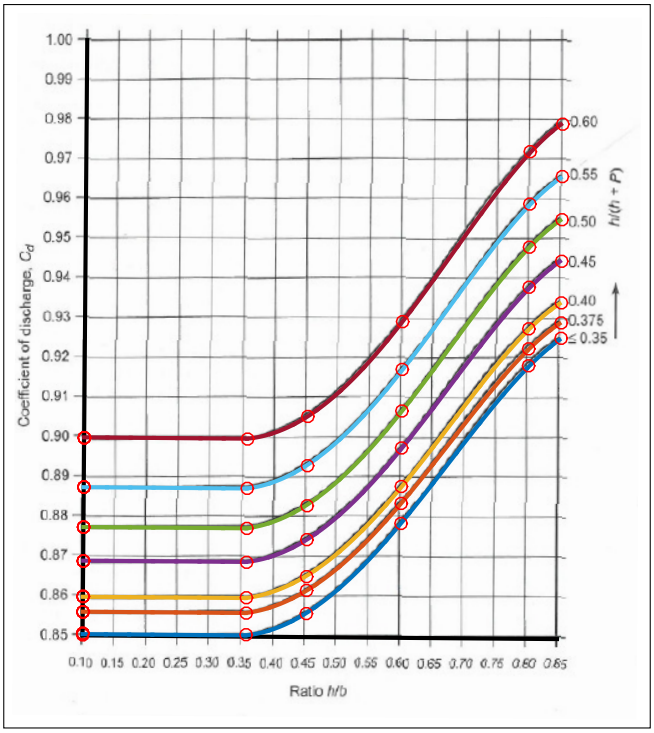

Figure 1.2: Discharge Coefficients Regression

Flood Frequency (years)	**Discharge (Q)** (ft^3/s)	**Tailwater Elevation** $(\mathbf{T_w})$ (ft)	**Crest Head** $(\mathbf{h}, \mathbf{y_1})$ (ft)
5	4,700	826.82	4.935
25	7,600	828.77	6.586
50	8,900	829.56	7.221
100	10,200	830.31	7.818
500	13,500	832.14	9.298

Table 1.1: Given Flood Frequency Properties

1.2) Head Loss Calculation

In order to determine the hydraulic depth at the toe of the slope (y_2), two equations for head loss, each a function of y_2 are used. Head loss is quantified using the theories of energy balance $(h_{l,eb})$ and boundary layers $(h_{l,bl})$.

Starting with energy balance:

$$h_{l,eb} = y_1 + z_1 - y_2 - \frac{{v_2}^2}{2g} - z_2 \tag{1.2}$$

Where:

- g is the acceleration of gravity

And continuing with boundary layers:

$$h_{l,bl} = \frac{\delta_3 U^3}{2gq} \tag{1.3}$$

Where:

- δ_3 is a boundary distance from the sloped face calculated as:

$$\delta_3 = 0.22\delta \tag{1.4}$$

- δ is a further upstream boundary distance from the sloped face calculated as:

$$\delta = L_{hl}(0.02)\left(\frac{L_{hl}}{K_s}\right)^{-0.233} \tag{1.5}$$

- L_{hl} is the length over the weir breadth and sloped face affecting head loss, approximated:

$$L_{hl} \approx b + \frac{z_1 - z_2}{\sin\alpha} \tag{1.6}$$

- α is the against-horizontal-angle of the slope, derived as:

$$\alpha = \tan^{-1}\left(\frac{1}{2.5}\right) \tag{1.7}$$

- z_2 is the stilling basin depth, as shown in figure 1.1.
- K_s is the mean concrete roughness, provided by the client to be $K_s = 0.002ft$.
- q is the total discharge per system width, calculated:

$$q = \frac{Q}{w} \tag{1.8}$$

- U is the velocity at the toe of the sloped surface, approximated:

$$U \approx v_2 = \frac{q}{y_2} \tag{1.9}$$

Given that minimal excavation is preferred, it may seem intuitive to solve for y_2 by satisfying $h_{l,eb} = h_{l,bl}$ with stilling basing bed set equal to natural downstream bed elevation ($z_2 = 819.34ft$). However, in order to keep the post-slope hydraulic jump within the stilling basin, the conjugate depth of y_2 (symbolized y_3) must be checked such that the post-jump surface is no greater than the given tail-water elevation for each discharge. As shown in figure 1.1, this constraint is:

$$y_3 + z_2 < T_w \tag{1.10}$$

Where:

- y_3 is the conjugate depth of y_2, calculated:

$$y_3 = \frac{y_2}{2}\left[1 + 8{Fr_2}^2\right] \tag{1.11}$$

- Fr_2 is the Froude number at the toe of the slope, calculated:

$$Fr_2 = \frac{v_2}{\sqrt{g{y_2}^2}} = \frac{q}{y_2\sqrt{g{y_2}^2}} \tag{1.12}$$

Therefore, the z_2 at which the hydraulic jump is contained within the basin is determined by first starting at some arbitrary elevation for z_2 at which the jump is well-contained within the basin (at the cost of excavation). This initial guess is set at $10ft$ below the natural downstream bed elevation. After determining the corresponding y_2 for this guessed z_2, the conjugate y_3 is calculated. Then, the $y_3 + z_2$ is compared to T_w, and the distance between those values is halved and added to the initially guessed z_2 for a newly elevated, cheaper ${z_2}'$.

$$\Delta z_{2i} = \frac{1}{2}\left[T_w - (y_3 + z_{2i})\right] \tag{1.13}$$

This difference is used to create a new z_2:

$${z_{2_i}}' = z_{2i} + \Delta z_{2i} \tag{1.14}$$

A new y_2 and h_l is calculated:

$$\left[y_{2_i}, h_{l_i}\right] = f\left({z_{2_i}}'\right)$$

The amount by which T_w exceeds $z_2 + y_3$ is tested again, halved and fed back in the same series of steps. This process if carried out over many iterations until the value $T_w - (y_3 + z_2)$ is below some safety threshold, chosen in this case to be $0.2ft$. A sample of this process is shown in table 1.2 for the 500 year flood event.

i	$\mathbf{z_2}$ (ft)	$\mathbf{h_l}$ (ft)	$\mathbf{y_2}$ (ft)	$\mathbf{y_3}$ (ft)	$\mathbf{T_w - (y_3 + z_2)}$ (ft)	$\mathbf{{z_2}'}$ (ft)
0	809.34	0.52023	1.84360	14.59300	8.20750	813.44
1	813.44	0.37877	1.97180	14.02200	4.67460	815.78
2	815.78	0.30878	2.05890	13.66200	2.69720	817.13
3	817.13	0.27183	2.11510	13.44000	1.57030	817.91
4	817.91	0.25147	2.15020	13.30600	0.91952	818.37
5	818.37	0.23993	2.17160	13.22500	0.54041	818.64
6	818.64	0.23328	2.18450	13.17700	0.31831	818.80
7	818.80	0.22942	2.19220	13.14800	0.18775	818.80

Table 1.2: Determination of z_2 required for $Q = 13,500\frac{ft^3}{s}$

Table 1.3 presents the required z_2 each of the five client-provided flooding events. Somewhat surprisingly, it is the 100 year flood, not the 500 year, that demands the lowest, most expensive z_2. Thus, this is set at the excavated z_2 that is used for all flood event calculations for this first option.

FF (years)	**Q** (ft)	$\mathbf{z_2}$ (ft)
5	4,700	819.1171
25	7,600	818.8963
50	8,900	818.8541
100	10,200	818.8318
500	13,500	818.9532

Table 1.3: Choice of safest z_2

With z_2 determined, table 1.4 presents the results of this process is carried out for each of the five client-provided discharge events.

FF (years)	**Q** (ft^3/s)	**q** (ft^2/s)	$\mathbf{y_2}$ (ft)	$\mathbf{z_2}$ (ft)	$\mathbf{h_{l,bl}}$ (ft)	$\mathbf{U, v_2}$ (ft/s)	$\mathbf{h_{l,eb}}$ (ft)	$\mathbf{Fr_2}$ -	$\mathbf{y_3}$ (ft)	$\mathbf{y_3 + z_2}$ (ft)	$\mathbf{T_{w,elev}}$ (ft)
5	4,700	29.375	0.825	818.832	0.521	35.607	0.521	6.909	7.658	826.490	826.820
25	7,600	47.500	1.290	818.832	0.356	36.808	0.356	5.710	9.796	828.627	828.770
50	8,900	55.625	1.495	818.832	0.314	37.219	0.314	5.365	10.617	829.449	829.560
100	10,200	63.750	1.696	818.832	0.283	37.587	0.283	5.086	11.381	830.213	830.310
500	13,500	84.375	2.194	818.832	0.229	38.464	0.229	4.577	13.143	831.975	832.140

Table 1.4: Results of h_l, and y_2 for each flood event event

1.3) Stilling Basin Length Calculation

Appendix chart A.5 presents recommended ratios of total stilling basin length per toe depth (y_2) for various toe Froude numbers (Fr_2). Thus, with known values of Fr_2, these ratios are read from the type III curve as:

$$\frac{L_{SB}}{y_3} = f\,(Fr_2) \tag{1.15}$$

These basin lengths are then calculated as the product this ratio and known y_2:

$$L_{SB} = \frac{L_{SB}}{y_3} y_3 \tag{1.16}$$

Table 1.5 presents the results of this process, added to previous tabulation, for these basin lengths.

FF (years)	**Q** $\left(\frac{ft^3}{s}\right)$	$\mathbf{Fr_2}$ -	$\frac{\mathbf{L_{SB}}}{\mathbf{y_2}}$ -	$\mathbf{L_{SB}}$ (ft)
5	4,700	6.909	2.543	19.471
25	7,600	5.710	2.415	23.657
50	8,900	5.365	2.368	25.146
100	10,200	5.086	2.327	26.484
500	13,500	4.577	2.242	29.466

Table 1.5: Basin length results

Thus, the built total horizontal basin length will be the greatest required for this basin. Reading from table 1.5, this is is a basin length of:

$$\boxed{L_{SB_{opt1}} = 29.466\ ft}$$

Also, the flow conditions $(y_2,\ y_3,\ Fr_2)$, for the flood event that determined this L_B will be used for feature sizing (if any) in section 1.6.

It should be noted that this estimate assumes the hydraulic jump starts immediately at the toe of the sloped face.

1.4) Drop Length Check

In order to make sure the previous head loss calculations are valid, the drop length (L_d) of the system before development must be calculated and compared to the horizontal footprint the sloped face would have (L_f).

Thus, the horizontal footprint of the slope is calculated as follows:

$$L_f = 2.5(z_1 - z_2) = 2.5(834.93\ ft - 818.83\ ft) = 40.25\ ft$$

The drop length is then calculated using:

$$L_d = 4.3(z_1 - z_2){D_r}^{-0.27} \tag{1.17}$$

Where the drop number D_r is calculated as:

$$Dr = \frac{q^2}{g(z_1 - z_1)^3} \tag{1.18}$$

Table 1.6 presents the drop lengths for each flow calculated with the same z_2 using equations 1.17 and 1.18.

FF	Q	Dr	L_d
5	4,700	0.00642	17.715
25	7,600	0.01680	22.964
50	8,900	0.02303	25.008
100	1,0200	0.03025	26.919
500	13,500	0.05300	31.318

Table 1.6: Drop lengths against z_2

Given that each calculated L_d is far less than L_f, it is assured that the post-weir flow will fall upon the sloped face, rendering the related head loss equations valid.

1.5) Basin Type

Table 1.7 presents recommendations of basin types for certain flow conditions, sourced from the USBR 1984 publication "Hydraulic Design of Stilling Basins and Energy Dissipators".

Type	Conditions	Features
I	weak jumps, $Fr_2 < 2.5$	no additional structures needed
IV	$2.5 < Fr_2 < 4.5$	chute blocks, optional solid end sill
III	$Fr_2 > 4.5$, $V_2 < 60\ ^{ft}/_{s}$	chute blocks, baffle blocks, solid end sill
II	$Fr_2 > 4.5$, $V_2 > 60\ ^{ft}/_{s}$	chute blocks, no baffle blocks, dentated end sill

Table 1.7: Recommended basin types for different flow conditions

Thus, regarding table 1.4, where the lowest Fr_2 is above 4.5 and the highest v_2 is below 60 $^{ft}/_{s}$, a type III basin is chosen.

1.6) Basin Feature Sizing

Figure 1.3 form the same USBR 1984 article presents the recommended feature sizing for a type III basin. In this drawing, the dimensions D_1 and D_2 are equivalent to this report's y_2 and y_3, respectively.

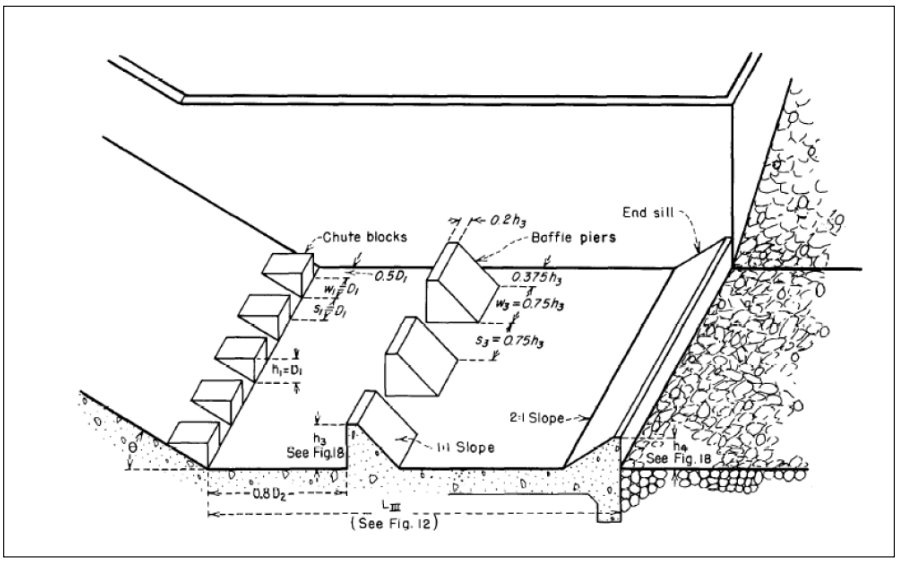

Figure 1.3: Feature detail for type III basin (USBR)

The dimensions for h_3 and h_4 are read from the following USBR chart:

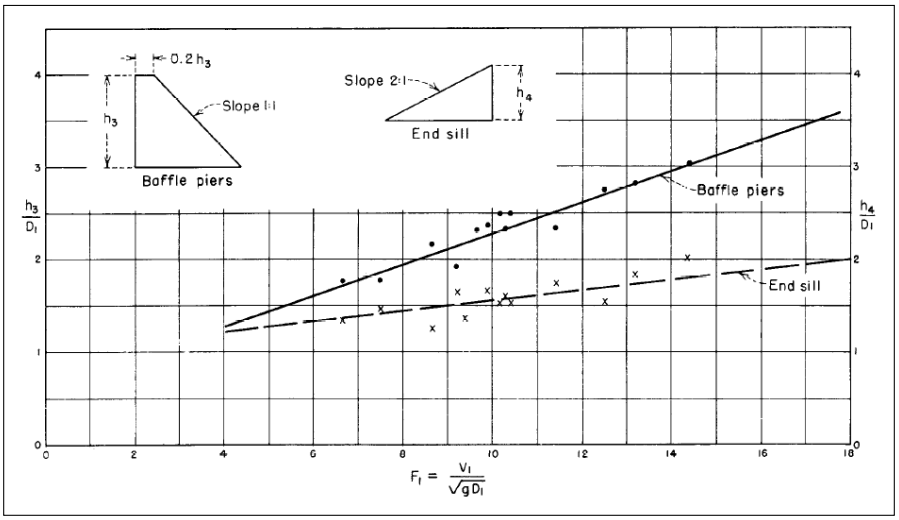

Figure 1.4: Chart for h_3/D_1 and h_4/D_1 as a function of Fr_2

Thus, with $Fr_2 = 4.5767$ for the 500 year event, h_3 and h_4 are read as follows:

- $h_3 = 3.0481\ ft$
- $h_4 = 2.7339\ ft$

Table 1.8 presents the dimensions determined for all featured recommended for a type III basin.

Description	Symbol	Method of Determination	Value (ft)
chute block height	h_1	$= D_1 = y_2$	2.194
chute block spacing	s_1	$s_1 = D_1 = y_2$	2.194
chute block width	w_1	$w_1 = D_1 = y_2$	2.194
chute block – wall gap	-	$= 0.5(D1) = 0.5(y_2)$	1.097
toe - upstream baffle face distance	-	$= 0.8(D2) = 0.8(y_3)$	10.515
baffle pier upstream face height	h_3	read from figure 1.4	3.048
end sill height above downstream elevation	h_4	read from figure 1.4	2.734
baffle pier – wall gap	-	$= 0.375(h_3)$	1.143
baffle pier top breadth	-	$= 0.2(h_3)$	0.610
baffle pier width	w_3	$w_3 = 0.75(h_3)$	2.286
baffle pier spacing	s_3	$s_3 = 0.75(h_3)$	2.286

Table 1.8: Option 1, type III basin feature sizes

1.7) Cost

The costs associated with the construction of this spillway are considered to be mainly from price for concrete volume as well as price of excavation from natural downstream bed elevation as well as the additional depth required to fill in the basin base, assumed to be $t = 4\ ft$ thick.

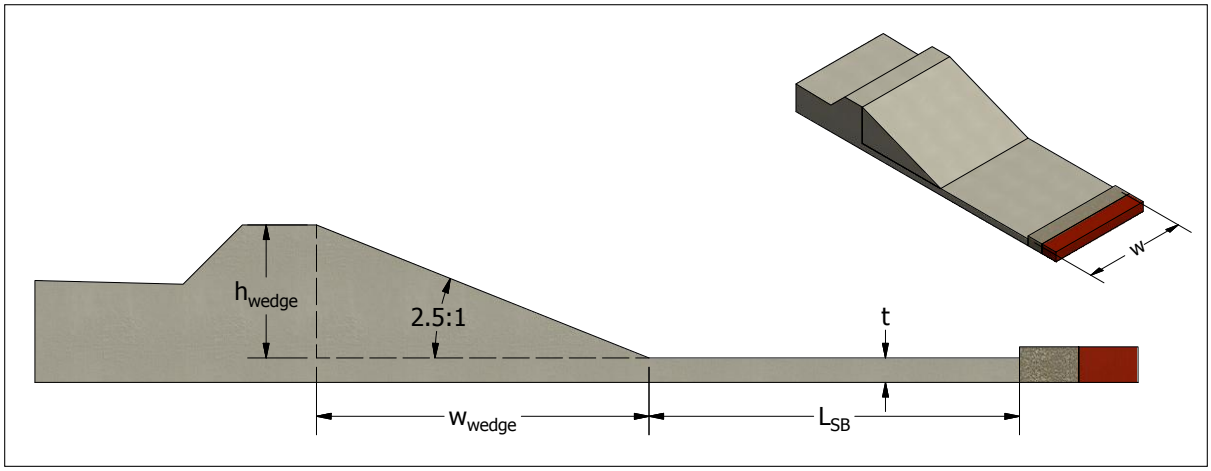

Figure 1.5: Option 1 concrete cost detail

Using the dimensions represented in figure 1.5, the total volume of concrete and excavation is calculated:

$$
\begin{aligned}
V_{\text{concrete}} &= V_{\text{sloped wedge}} + V_{\text{basin}} = \frac{1}{2}\left(b_{\text{wedge}}\right)\left(h_{\text{wedge}}\right) w + t\left(L_{SB}\right) w \\
&= \frac{1}{2}(z_1 - z_2)^2 (2.5) w + t(L_B) w \\
&= 51{,}830\ ft^3 + 18{,}858\ ft^3 = \boxed{70{,}689\ ft^3} \\
V_{\text{excavation}} &= \left[t + (\text{natural downstream } z) - z_2\right](L_B)(w) = \boxed{21{,}254\ ft^3}
\end{aligned}
$$

1.8) Tailwater - Conjugate Depth Plot

As shown in figure 1.6, the post-hydraulic-jump depth of y_3 does not cause the water surface to rise above tailwater elevation.

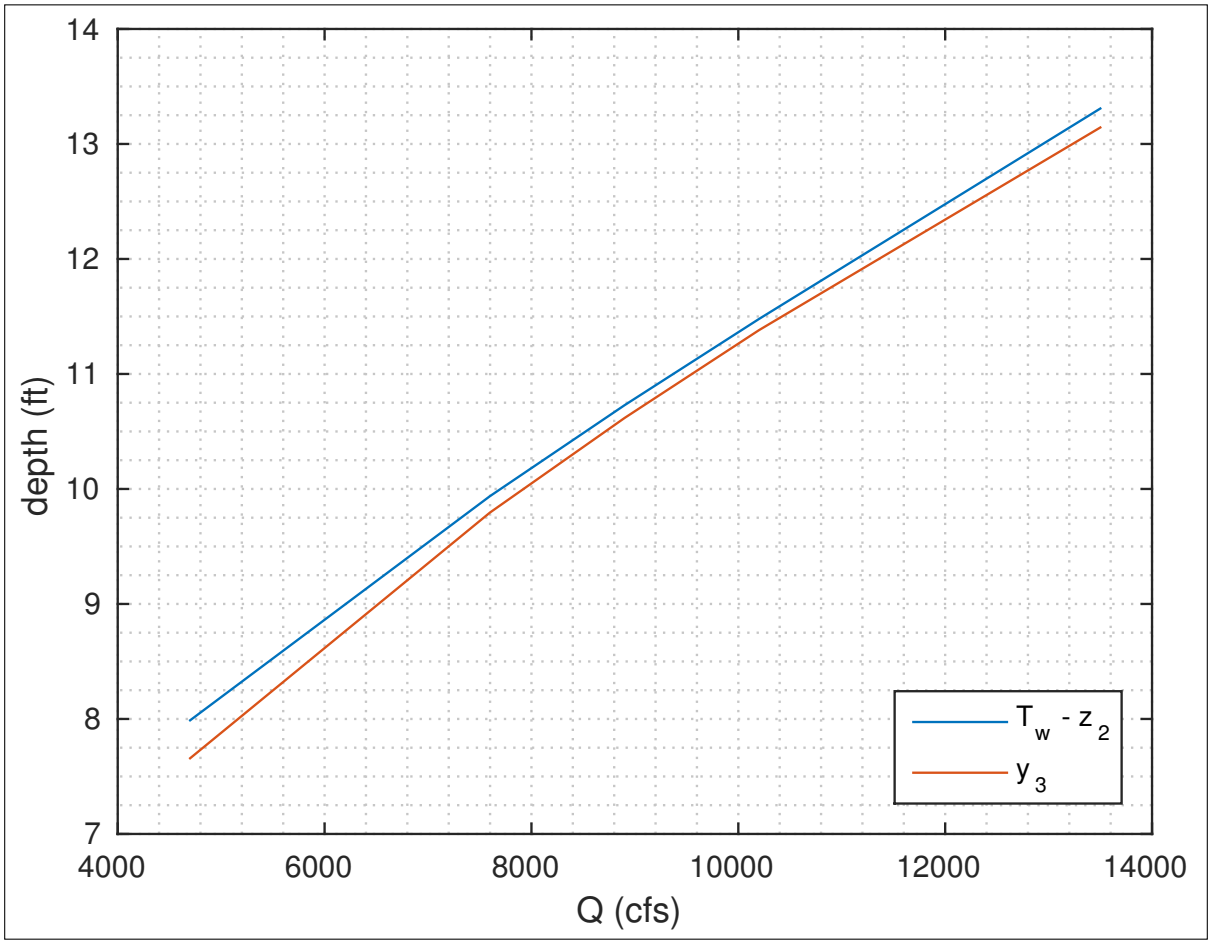

Figure 1.6: Option 1 plot of tailwater and conjugate depths

2) Option 2: Stepped Spillway

The second option under consideration is one involving a stepped spillway.

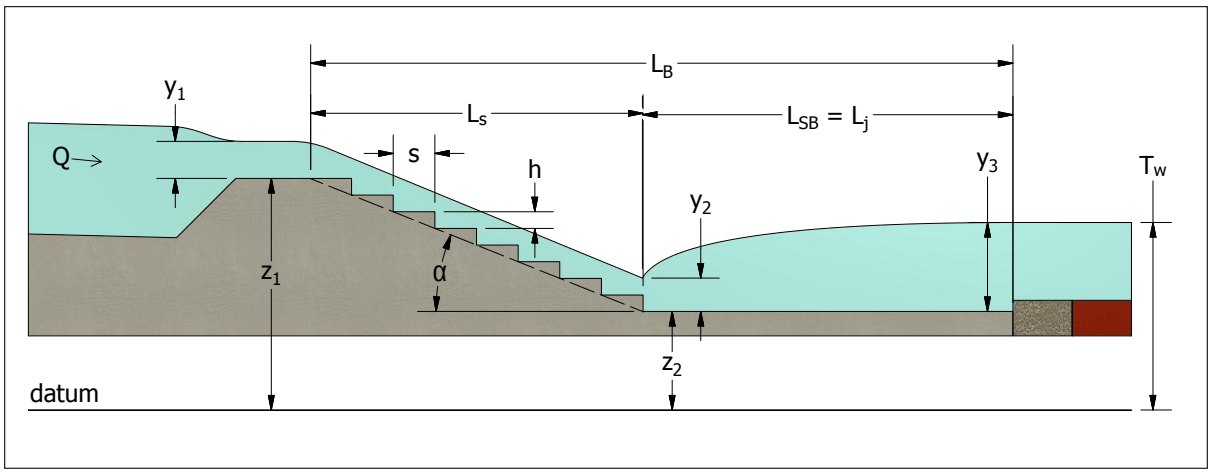

Figure 2.1: Option 2 basin detail

2.1) Testing Trial Parameters Procedure

In determining the most economically efficient construction of such a system, a system can be built that meets the following conditions:

- The hydraulic jump is contained within the basin
- The natural drop distance L_d is less than the horizontal footprint of the stepped spillway L_s, ensuring steps dissipate hydraulic energy via contact with water
- No transitional flow along steps (either entirely skimming or nappe flow)
  - This constraint ensures, no transition between nappe and skimming flow across the range of anticipated discharges.
  - Allowing for transitional flow incurs the risk of significant structural vibration, hence compromising the structural integrity of the entire system.

This is done by developing a system that presents relevant condition tests as a function of stepped spillway parameters s, h and N. This is accomplished as follows:

- A reasonable step width (s) and height (h) are chosen.
- With this known (h), an integer number of steps (N) are chosen such that the total vertical distance covered by the steps is at least the distance between the crest elevation, (z_1) and the natural downstream bed elevation.
- With this known N, the excavated z_2 can be determined easily:

$$z_2 = z_1 - Ns \tag{2.1}$$

- The status of nappe flow vs. skimming is tested as follows:

$$\text{nappe for } \frac{y_c}{h} \leq 0.89 - 0.4\left(\frac{h}{s}\right), \quad \text{skimming for } \frac{y_c}{h} \geq 1.2 - 0.325\left(\frac{h}{s}\right) \tag{2.2}$$

  - where y_c can be solved:

$$y_c = \left(\frac{q^2}{g}\right)^{1/3} \tag{2.3}$$

  - It is important to maintain only either nappe flow of skimming flow for all discharges. That way, detrimental transitional flow can be avoided for all anticipated discharge.

- The depth at the toe (y_2) is then calculated:

$$y_2 = y_c\left[\frac{f_c}{8\sin(\alpha)}\right]^{1/3} \tag{2.4}$$

  - where f_c, the coefficient, of friction is calculated:

$$f_c = \frac{f_d}{2}\left[1 + \tanh\left(2.5\frac{0.5 - c_{\text{mean}}}{c_{\text{mean}}(1 - c_{\text{mean}})}\right)\right] \tag{2.5}$$

  - where f_d, another coefficient, is calculated:

$$f_d = \frac{2}{\sqrt{\pi}}\frac{1}{k} \text{ where } k = 4.5 \tag{2.6}$$

  - where c_{mean}, the mean air concentration, is calculated:

$$c_{\text{mean}} = 1.44\sin(\alpha) - 0.08 \tag{2.7}$$

  - and where α, the pseudo-slope angle, is calculated:

$$\alpha = \tan^{-1}\left(\frac{h}{s}\right) \tag{2.8}$$

- with this y_2 known, the toe velocity (v_2) is calculated:

$$v_2 = \frac{q}{y_2} \tag{2.9}$$

- which allows the toe Froude number (Fr_2) to be calculated:

$$Fr_2 = \frac{v_2}{\sqrt{(g)(y_2)}} \tag{2.10}$$

  - At this point, the known v_2 and Fr_2 can be used to determine basin type using table 1.7. Fr_2 can also be used to determine the conjugate depth (y_3) using equation 1.11.
  - This known y_3 can then be compared to the tailwater depth to ensure the jump is contained within the basin.

- As a final step, it is necessary to check that the drop length L_d is less than the stepped spillway horizontal footprint (L_s). This is accomplished using eqs. 1.17 and 1.18 from the option 1 process.

2.2) Trial Results

Table 2.1 presents the results of this procedure carried out for the parameters $h = 1$, $s = 2.1$, $N = 16$. As shown, the following conditions are met for all discharges:

- Flow is constantly skim throughout given that all y_c/h exceeds $(y_c/h)_{\text{nappe}}$
- Any change in Fr_2 does not indicate transitional flow across changing discharges
- The natural drop length falls below L_s;

However, the values of $\frac{T_w - z_2}{y_3}$ for the 25 though 500 year floods indicate that the jump is in fact *not* held within the basin. Thus, the parameters are adjusted and tried again.

FF (yrs)	Q (ft^3/s)	q (ft^3/s)	yc (ft^2/s)	y_c/h -	$(y_c/h)_{\text{nappe}}$ -	$(y_c/h)_{\text{skim}}$ -	y_2 (ft)	v_2 (ft/s)	Fr_2 -	y_3 (ft)	$T_w - z_2$ (ft)	$\frac{T_w - z_2}{y_3}$ -	$\frac{L_{SB}}{y_3}$ -	L_{SB} (ft)	Dr -	L_d (ft)	$\frac{L_d}{L_s}$ -
5	4,700	29.375	2.993	2.993			0.849	34.614	6.622	7.534	7.890	1.047	2.516	18.959	0.007	17.694	0.527
25	7,600	47.500	4.123	4.123			1.169	40.628	6.622	10.379	9.840	0.948	2.516	26.119	0.017	22.937	0.683
50	8,900	55.625	4.580	4.580	0.700	1.045	1.299	42.824	6.622	11.532	10.630	0.922	2.516	29.018	0.023	24.979	0.743
100	10,200	63.750	5.016	5.016			1.423	44.815	6.622	12.629	11.380	0.901	2.516	31.779	0.031	26.887	0.800
500	13,500	84.375	6.047	6.047			1.715	49.204	6.622	15.224	13.210	0.868	2.516	38.309	0.054	31.281	0.931

Table 2.1: Results for trial with $h = 1$, $s = 2.1$, $N = 16$

For a second round, the step dimensions are kept, but an additional step is added $(N = 17)$, requiring an even lower z_2. However, as shown in table 2.2, this still does not contain the hydraulic jump for the 100 and 500 year discharges. Thus, the step dimension s is modified, and the process is repeated.

FF (yrs)	Q (ft^3/s)	q (ft^3/s)	yc (ft^2/s)	y_c/h -	$(y_c/h)_{\text{nappe}}$ -	$(y_c/h)_{\text{skim}}$ -	y_2 (ft)	v_2 (ft/s)	Fr_2 -	y_3 (ft)	$T_w - z_2$ (ft)	$\frac{T_w - z_2}{y_3}$ -	$\frac{L_{SB}}{y_3}$ -	L_{SB} (ft)	Dr -	L_d (ft)	$\frac{L_d}{L_s}$ -
5	4,700	29.375	2.993	2.993			0.849	34.614	6.622	7.534	8.890	1.180	2.516	18.959	0.005	17.899	0.501
25	7,600	47.500	4.123	4.123			1.169	40.628	6.622	10.379	10.840	1.044	2.516	26.119	0.014	23.203	0.650
50	8,900	55.625	4.580	4.580	0.700	1.045	1.299	42.824	6.622	11.532	11.630	1.009	2.516	29.018	0.020	25.268	0.708
100	10,200	63.750	5.016	5.016			1.423	44.815	6.622	12.629	12.380	0.980	2.516	31.779	0.026	27.199	0.762
500	13,500	84.375	6.047	6.047			1.715	49.204	6.622	15.224	14.210	0.933	2.516	38.309	0.045	31.644	0.886

Table 2.2: Results for trial with $h = 1$, $s = 2.1$, $N = 17$

With a new step horizontal length (s), all conditions are met, as shown in table 2.3. All hydraulic jumps are contained, flow is skimming throghout, no transitional flow is detected, and all drop lengths fall below L_s.

FF (yrs)	Q (ft^3/s)	q (ft^3/s)	yc (ft^2/s)	y_c/h -	$(y_c/h)_{\text{nappe}}$ -	$(y_c/h)_{\text{skim}}$ -	y_2 (ft)	v_2 (ft/s)	Fr_2 -	y_3 (ft)	$T_w - z_2$ (ft)	$\frac{T_w - z_2}{y_3}$ -	$\frac{L_{SB}}{y_3}$ -	L_{SB} (ft)	Dr -	L_d (ft)	$\frac{L_d}{L_s}$ -
5	4,700	29.375	2.993	2.993			1.037	28.331	4.903	6.690	8.890	1.329	2.298	15.373	0.005	17.899	0.458
25	7,600	47.500	4.123	4.123			1.428	33.253	4.903	9.216	10.840	1.176	2.298	21.178	0.014	23.203	0.593
50	8,900	55.625	4.580	4.580	0.716	1.059	1.587	35.050	4.903	10.240	11.630	1.136	2.298	23.529	0.020	25.268	0.646
100	10,200	63.750	5.016	5.016			1.738	36.680	4.903	11.214	12.380	1.104	2.298	25.768	0.026	27.199	0.696
500	13,500	84.375	6.047	6.047			2.095	40.272	4.903	13.518	14.210	1.051	2.298	31.063	0.045	31.644	0.809

Table 2.3: Results for trial with $h = 1$, $s = 2.3$, $N = 17$

Thus the parameters set for further investigation are $h = 1$, $s = 2.3$, $N = 17$. The recommended basin for the corresponding values of Fr_2 and v_2 for all discharges is again type III. Also, given that the 500 year flood requires the largest basin, its conditions will be used for all feature sizing, and this required length will be the design basin length.

$$L_{SB_{opt2}} = 31.063\ ft$$

2.3) Basin Feature Sizing

Table 2.4 presents the dimensions determined for all features recommended for a type III basin in this case.

Description	Symbol	Method of Determination	Value (ft)
chute block height	h_1	$= D_1 = y_2$	2.095
chute block spacing	s_1	$s_1 = D_1 = y_2$	2.095
chute block width	w_1	$w_1 = D_1 = y_2$	2.095
chute block – wall gap	-	$= 0.5(D1) = 0.5(y_2)$	1.048
toe - upstream baffle face distance	-	$= 0.8(D2) = 0.8(y_3)$	10.814
baffle pier upstream face height	h_3	read from figure 1.4	3.027
end sill height above downstream elevation	h_4	read from figure 1.4	2.651
baffle pier – wall gap	-	$= 0.375(h_3)$	1.135
baffle pier top breadth	-	$= 0.2(h_3)$	0.605
baffle pier width	w_3	$w_3 = 0.75(h_3)$	2.270
baffle pier spacing	s_3	$s_3 = 0.75(h_3)$	2.270

Table 2.4: Option 2, type III basin feature sizes

2.4) Cost

Just as with option 1, the primary costs of this stepped spillway system are both cost of concrete as well as excavation, both measured in volume. Stilling basin slab thickness (t) is again set to 4 ft.

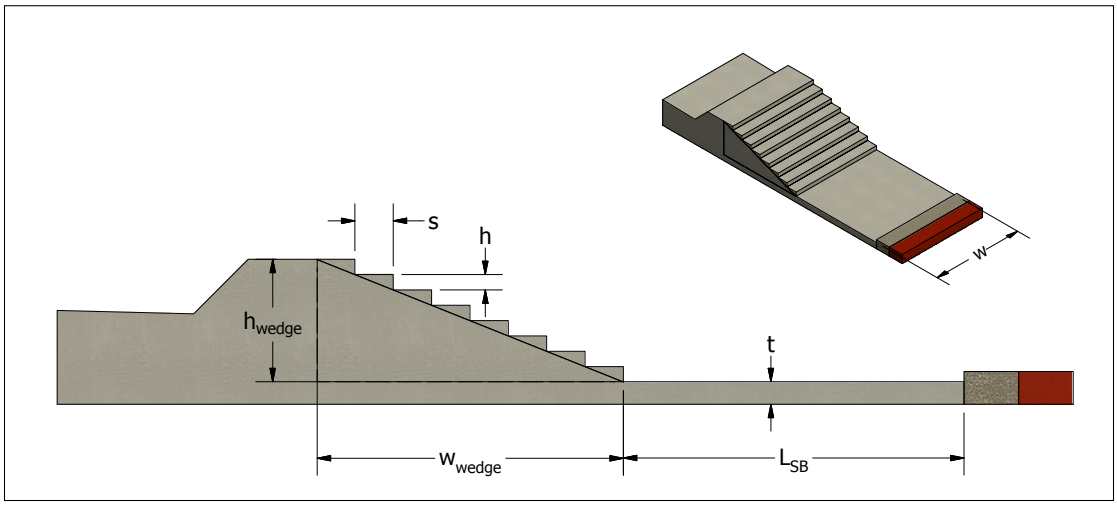

Figure 2.2: Option 2 concrete cost detail

Using the dimensions represented in figure 2.2, the total volume of concrete and excavation is calculated:

$$V_{\text{concrete}} = V_{\text{sloped wedge}} + V_{\text{steps}} + V_{\text{basin}} = \frac{1}{2}\left(w_{\text{wedge}}\right)\left(h_{\text{wedge}}\right)w + \frac{w}{2}(\text{s})(\text{h})w + (t)\left(L_B\right)w$$

$$= \frac{w}{2}(Ns)(Nh) + \frac{w}{2}(sh)(N) + t(w)(L_{SB}) = 53{,}176\ ft^3 + 31{,}280\ ft^3 + 19{,}880\ ft^3 = \boxed{76{,}184\ ft^3}$$

$$V_{\text{excavation}} = [t + (\text{natural downstream } z - z_2)]\left(L_{SB}\right)(w) = \boxed{26{,}888 ft^3}$$

2.5) Tailwater - Conjugate Depth Plot

As shown in figure 2.3, the post-hydraulic-jump depth of y_3 does not cause the water surface to rise above tailwater elevation for this appropriately sized stepped spillway.

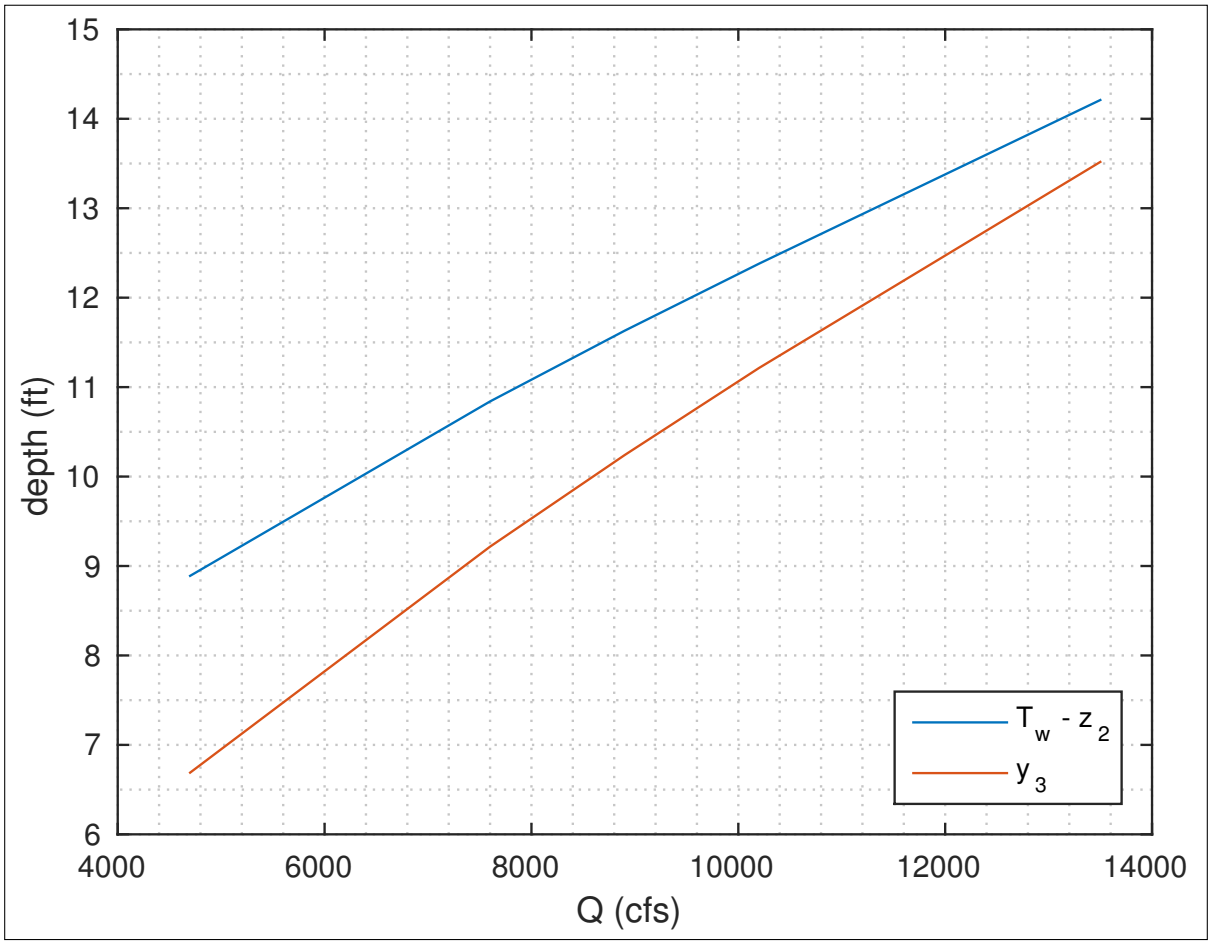

Figure 2.3: Option 2 plot of tailwater and conjugate depths

3) Option 3: Plunge Pool

The final option to be considered is simple drop structure (or plunge pool), over which energy is dissipated via impact with the basin bed as well as the end sill.

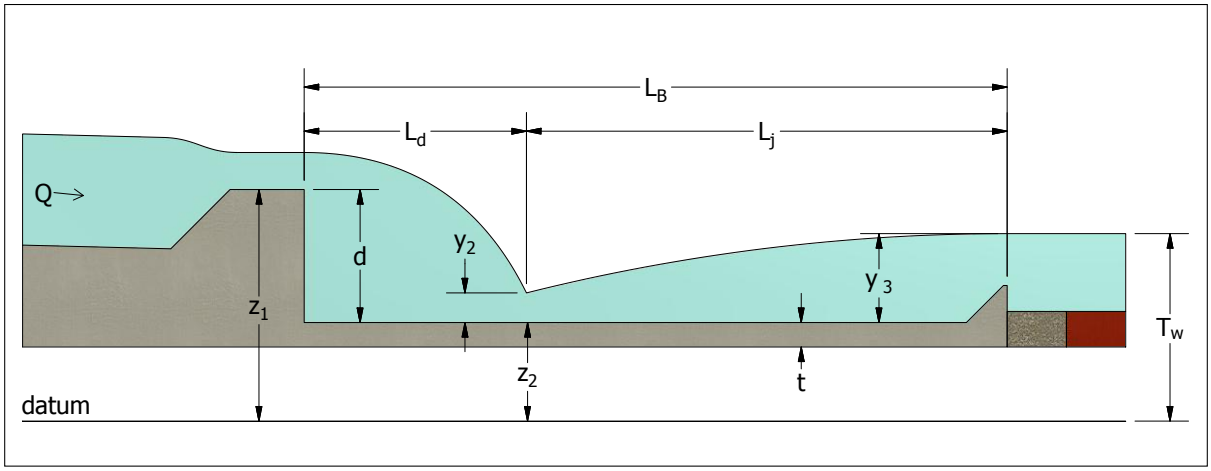

Figure 3.1: Option 3 Basin Detail

As shown in figure 3.1, the system is sized such that the hydraulic jump starts immediately at the point at the drop length (L_d). As with previous options, the relation between y_3 and T_w is primarily a function of chosen z_2.

3.1) Determination of Invert Elevation

USBR recommends differing safety offsets of $T_w - (y_3 + z_2)$ for differing basins. Therefore, an initial z_2 is determined in which this offset is 0. Then, the basin type can be inferred from available conditions and a new z_2 can be set matching the corresponding recommended offset value. This process is carried out as follows:

- z_2 is first set to natural downstream z
- This invert elevation is used to calculate the drop height (d) of the structure:

$$d = (z_1 - z_2) \tag{3.1}$$

- With this d, the drop number (Dr) is calculated using equation 1.18.

$$Dr = \frac{q^2}{g\,(z_1 - z_2)^3} = \frac{q^2}{gd^3}$$

- The drop length hydraulic depth (y_2) is then calculated:

$$y_2 = d(0.54)Dr^{0.425} \tag{3.2}$$

- Critical depth (y_c) is calculated using eq. 2.3.
- The drop number is also used to calculate the jump-end hydraulic depth (y_3):

$$y_3 = d(1.66)Dr^{0.27} \tag{3.3}$$

- This y_3 is used for later comparison with T_w.

- y_2 is then used to calculate drop-point flow conditions:
  - The drop-point Froude number (Fr_2) is calculated using eq. 1.12.
  - The drop-point velocity (v_2) is then calculated:

$$v_2 = \frac{q}{y_2} \tag{3.4}$$

- Dr is used yet again to calculate the drop length:

$$L_d = d(4.3)Dr^{0.27} \tag{3.5}$$

- Fr_2 is then used to calculate the jump length (y_2) using a formula particular to this type of structure:

$$L_j = y_2(220)\tanh\left(\frac{Fr_2 - 1}{22}\right) \tag{3.6}$$

- The total basin length starting at downstream face of the weir (L_B) is then the sum of the drop and jump lengths:

$$L_B = L_d + L_j \tag{3.7}$$

This process is carried out for several trail values of z_2 until all constraints are met, as discussed in the following section.

3.2) Trial Results

Table 3.1 presents this process for the initial $z_2 = 819.34\ ft$. While the hydraulic jump is contained within the system, given $\frac{T_w - z_2}{y_3} > 1$ for all discharges, the Fr_2 values span the ranges of multiple basin types. This can determined regarding table 1.7. Therefore, one of the basin types of concern, type IV, is chosen and used for further calculation.

FF	Q	Dr	y_2	y_c	y_3	$\frac{y_3}{T_w - z_2}$	Fr_2	v_2	L_d	L_j	L_B
5	4,700	0.007	1.026	2.993	6.797	0.909	4.979	28.620	17.607	40.394	58.002
25	7,600	0.018	1.544	4.123	8.811	0.934	4.362	30.760	22.825	51.518	74.343
50	8,900	0.025	1.766	4.580	9.596	0.939	4.177	31.497	24.856	55.716	80.572
100	10,200	0.033	1.983	5.016	10.329	0.942	4.023	32.148	26.755	59.574	86.329
500	13,500	0.058	2.517	6.047	12.017	0.939	3.725	33.528	31.127	68.217	99.345

Table 3.1: Option 3 trial results for z_2 = natural downstream elevation

It is recommended that for type IV basins, the post-jump head be *at least* 10% below the tailwater depth. That is, for all discharges:

$$\frac{y_3}{T_w - z_2} \leq 0.9 \tag{3.8}$$

Regarding these values shown in table 3.1, an invert elevation equal to natural downstream depth does not meet this condition. Therefore, a deeper z_2 is attempted in order to satisfy this condition.

Table 3.2 presents another trial process for a slightly lower $z_2 = 818.84\ ft$. However, it can be seen that the tailwater safety condition violation persists for the 50, 100 and 500 year events.

FF	Q	Dr	y_2	y_c	y_3	$\frac{y_3}{T_w - z_2}$	Fr_2	v_2	L_d	L_j	L_B
5	4,700	0.006	1.018	2.993	6.838	0.857	5.044	28.870	17.713	40.687	58.401
25	7,600	0.017	1.531	4.123	8.864	0.893	4.419	31.028	22.962	51.928	74.890
50	8,900	0.023	1.751	4.580	9.653	0.901	4.232	31.772	25.006	56.173	81.179
100	10,200	0.030	1.966	5.016	10.391	0.906	4.076	32.428	26.916	60.076	86.992
500	13,500	0.053	2.495	6.047	12.089	0.909	3.773	33.820	31.315	68.827	100.140

Table 3.2: Option 3 trial results for $z_2 = 818.84\ ft$

Table 3.3 presents the process for which conditions are met, including the type IV 10% safety tailwater offset. This is achieved with an invert elevation of $z_2 = 818.68\ ft$.

FF	Q	Dr	y_2	y_c	y_3	$\frac{y_3}{T_w - z_2}$	Fr_2	v_2	L_d	L_j	L_B
5	4,700	0.006	1.015	2.993	6.851	0.842	5.064	28.949	17.747	40.779	58.526
25	7,600	0.016	1.527	4.123	8.881	0.880	4.437	31.112	23.005	52.056	75.061
50	8,900	0.022	1.746	4.580	9.672	0.889	4.249	31.858	25.053	56.316	81.369
100	10,200	0.029	1.961	5.016	10.410	0.895	4.093	32.516	26.967	60.233	87.200
500	13,500	0.052	2.488	6.047	12.112	0.900	3.789	33.913	31.374	69.018	100.390

Table 3.3: Option 3 trial results for $z_2 = 818.68\ ft$

The discharge event requiring the largest basin length is used for system sizing as well as flow conditions for feature sizing.

$$L_{B_{opt3}} = 100.390\ ft$$

3.3) Basin Feature Sizing

Figure 3.2 presents the recommended feature sizing for this type IV basin. It should be noted this design will forgo the inclusion of the chute blocks shown. It is assumed that they can be neglected given that their placement would have minimal dissipation effects at the toe of a nearly vertical slope. That is, their location is too far before L_d to function properly. Table 3.4 presents the dimensions of the remaining feature to be sized - the end sill.

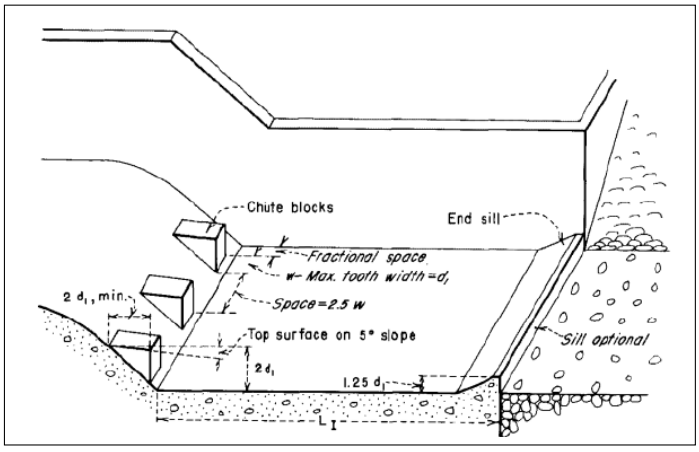

Figure 3.2: Feature detail for type IV basin (USBR)

Description	Symbol	Method of Determination	Value (ft)
end sill height above invert elevation	-	$1.25d_1 = 1.25y_1$	3.11

Table 3.4: Option 3, type IV basin feature sizes

3.4) Cost

With the same slab thickness of $t = 4\ ft$, the concrete volume needed for option 3 is calculated as follows:

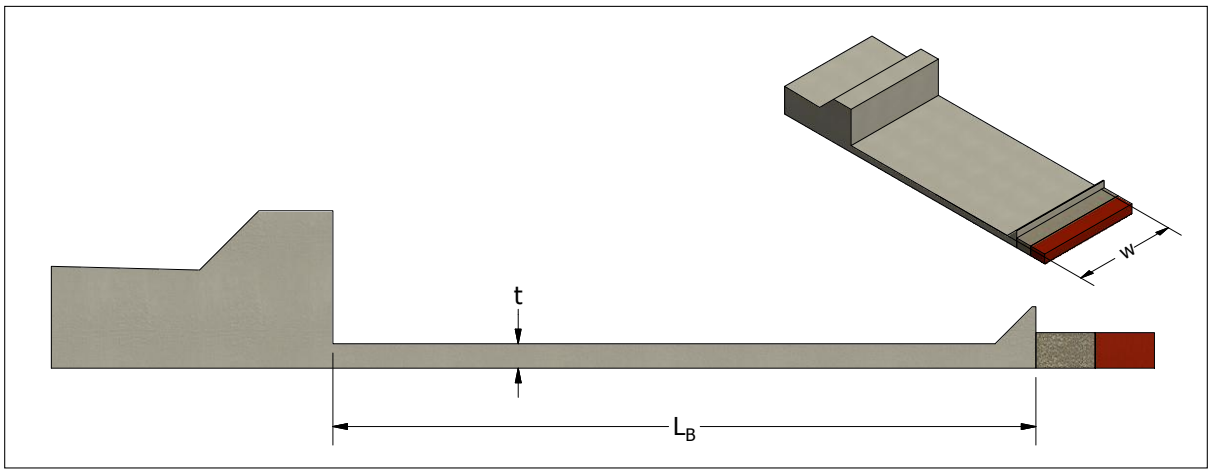

Figure 3.3: Option 3 concrete cost detail

Using the dimensions represented in figure 3.3, the total volume of concrete and excavation is calculated:

$$V_{\text{concrete}} = V_{\text{basin}} = L_B(t)w = \boxed{64,250ft^3}$$

$$V_{\text{excavation}} = [t + (\text{natural downstream } z - z_2)]\,(L_{SB})\,(w) = \boxed{74,852ft^3}$$

3.5) Tailwater - Conjugate Depth Plot

Finally, as shown in figure 3.4, the post-hydraulic-jump depth of y_3 does not cause the water surface to rise above tailwater elevation for drop structure. In fact, there is a sizeable safety distance for all discharges.

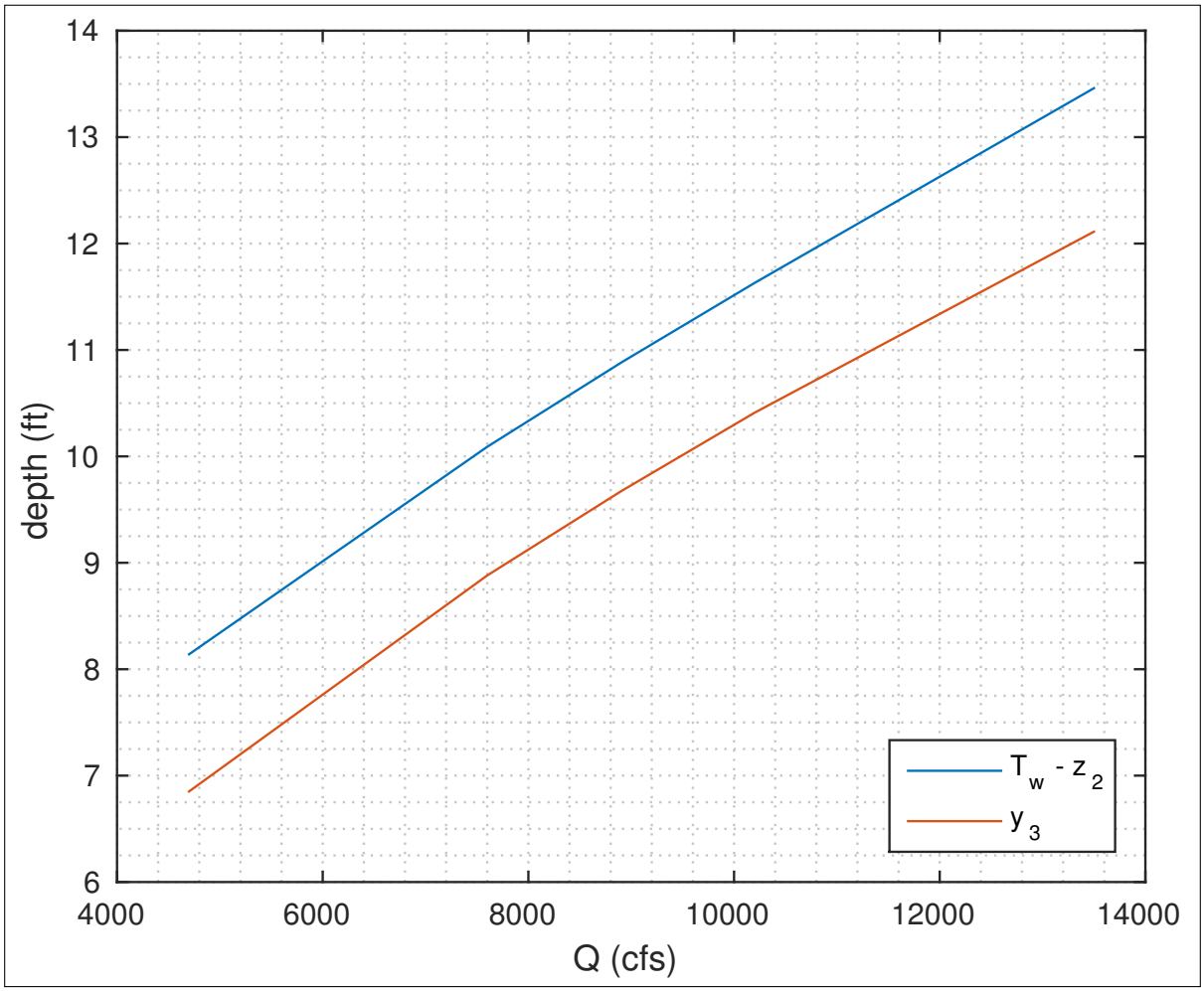

Figure 3.4: Option 3 plot of tailwater and conjugate depths

4) Option Choice

Table 4.1 presents the most salient sizes and properties for each option. As discussed below, it can be seen that each option has their own respective economic trade-offs.

Option	**Description**	$\mathbf{z_2}$ (ft)	$\mathbf{L_{SB}}$ (ft)	$\mathbf{L_B}$ (ft)	$\mathbf{V_{concrete}}$ $\left(ft^3\right)$	$\mathbf{V_{excavation}}$ $\left(ft^3\right)$
1	sloped, basin type III	818.832	29.466	69.711	70,689	21,254
2	stepped, basin type III	817.930	31.063	70.163	76,184	26,888
3	drop, basin type IV	818.680	100.390	100.390	64,250	74,852

Table 4.1: Relevant properties for each option

- Option 1:
  - pro: least total system length;
  - pro: less concrete volume than option 2;
  - pro: least volume excavated;
  - con: greater concrete volume than option 3;
  - con: highest adjustment to invert elevation of all options;
- Option 2:
  - pro: less excavation volume than option 3;
  - pro: occupies less total system length than the drop structure;
  - pro: least adjustment of the invert elevation against natural, downstream elevation;
  - con: more excavation volume than option 1;
  - con: most concrete volume of all options;
  - con: most complicated construction of all options
- Option 3:
  - pro: less adjustment to invert elevation than option 1;
  - pro: least concrete volume of all options;
  - pro: simplest construction of all options;
  - pro: robust containment of hydraulic jump resulting from safety offset between y_3 and $T_w - 2$;
  - con: most excavation of all options;
  - con: greatest system length of all options by far;
  - con: flow may occur for which different basin types are recommended

Table 4.2 presents an extremely rough estimate of each option's construction costs. This table converts each option's concrete and excavation estimates from table 4.1 into costs based on researched average volumetric pricing.

	Concrete Costs	**Excavation Cost**	**Total**
cost per cubic yard	$ 93	$ 90	
Option 1	$243,484	$70,847	$314,331
Option 2	$262,412	$89,627	$352,038
Option 3	$221,306	$249,507	$470,812

Table 4.2: Option Pricing

By working with the lightly researched prices per volume for both concrete and excavation, it is clear that option 1 provides the most economic choice. While it does not require the least amount of concrete, it offers dramatic savings on excavation compared to option 1. It's construction is also quite simple compared to option 2.

Option 1 is chosen

This conclusion was determined with an assumed price per volume. In order to assess option 1's economic superiority across ranging volumetric prices, several extreme cases are tested. Given that option 3's biggest pitfall is its high excavation volume, the case of dramatically reduced excavation is assessed in table 4.3. As shown, it is only at an unrealistically low excavation price that option 3 becomes preferable. Thus, the choice of option 1 is determined to be a confident one.

	Concrete Costs	**Excavation Cost**	**Total**
cost per cubic yard	$ 93	$ 10.8	
Option 1	$243,484	$8,502	$251,986
Option 2	$262,412	$10,755	$273,167
Option 3	$221,306	$29,941	$251,246

Table 4.3: Option Pricing for Extreme Case

It is assumed that when approaching the other extreme, in which the ratio $^{\text{cost}_{\text{excavation}}}/_{\text{cost}_{\text{concrete}}}$ increases, option 1 only becomes more preferable.

A Appendix

A.1 Tailwater - Conjugate Depth Plots for All Options

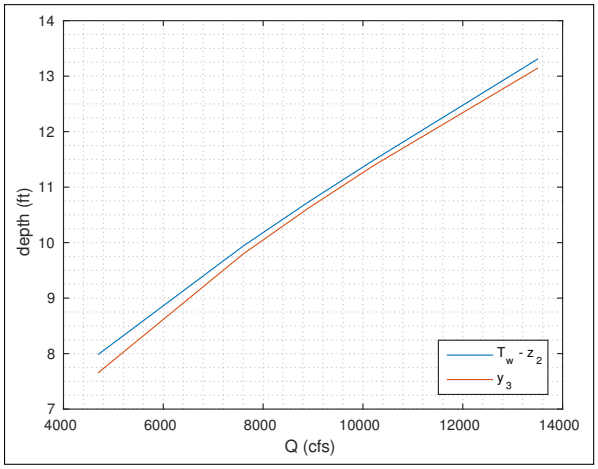

Figure A.1: Option 1

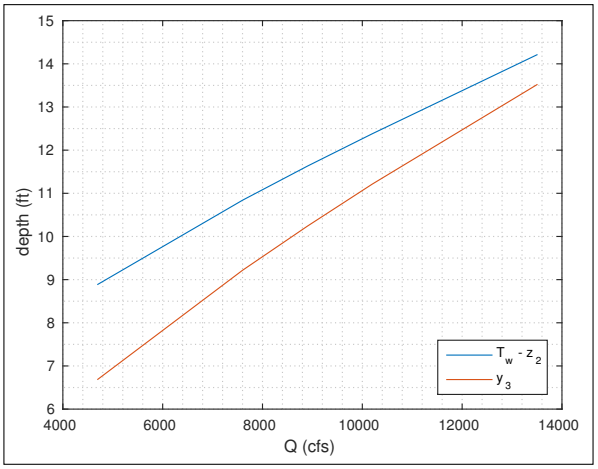

Figure A.2: Option 2

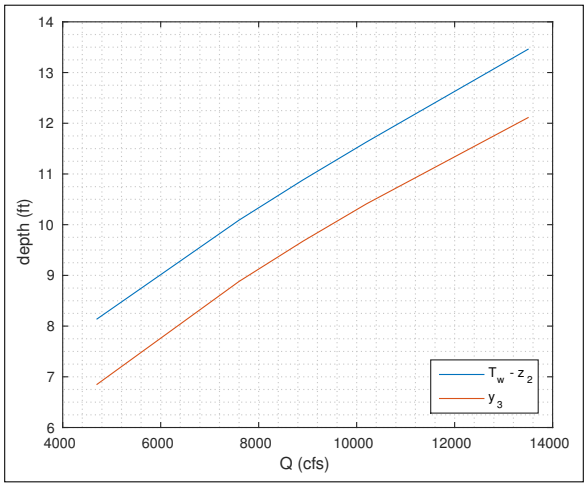

Figure A.3: Option 3

A.2 Charts and Figures Used

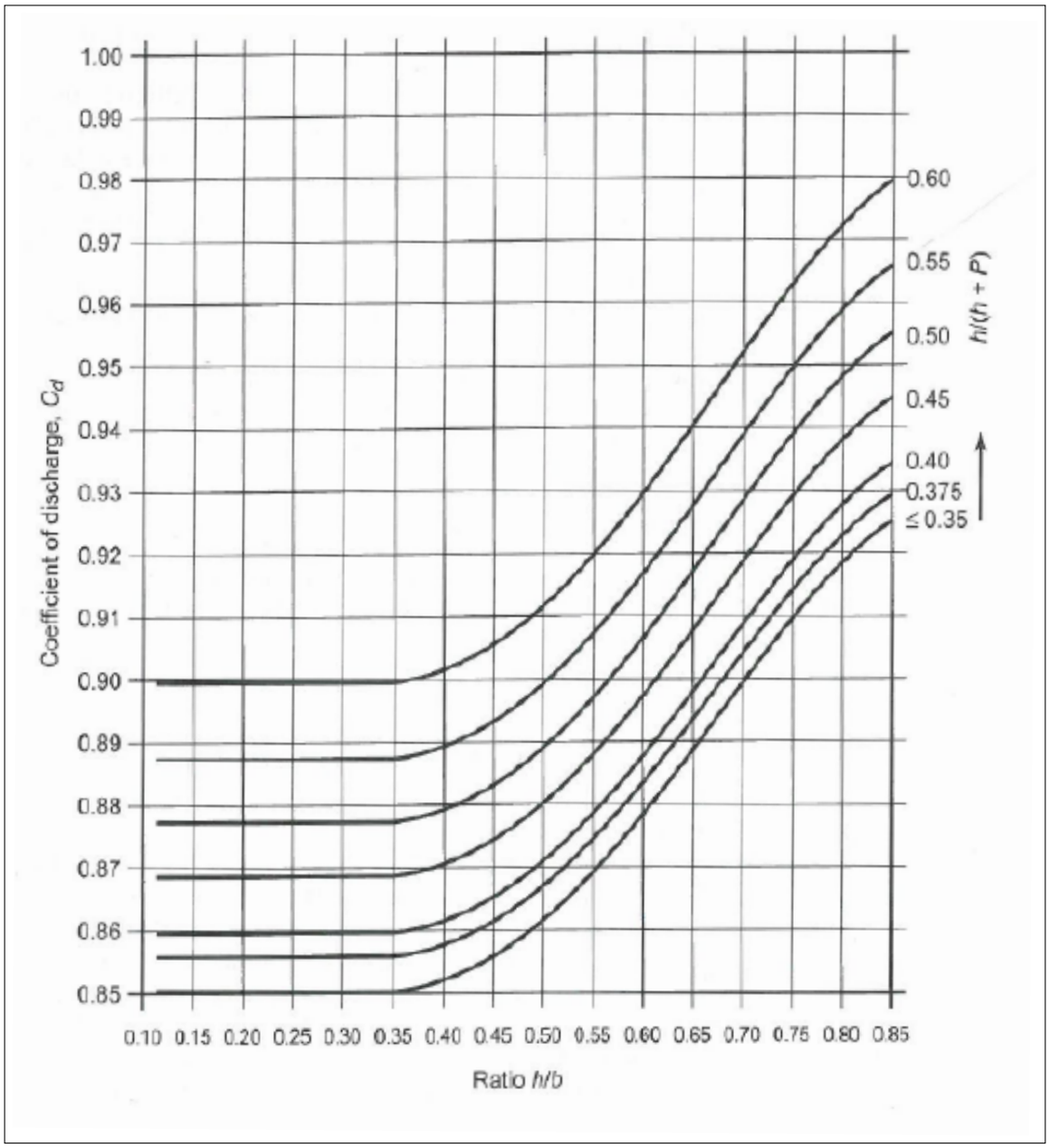

Figure A.4: Discharge Coefficients for broad-crested weirs (British Standards Institution, 1969)

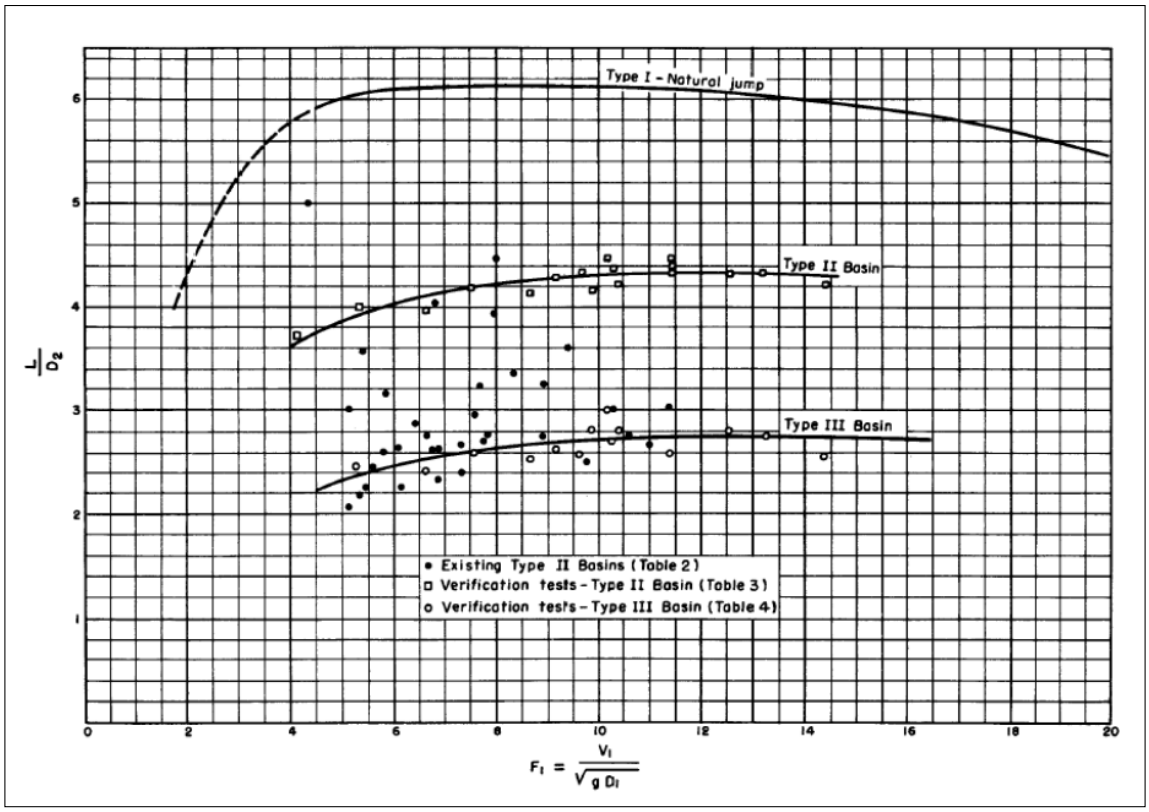

Figure A.5: Stilling basin type III reading (USBR)

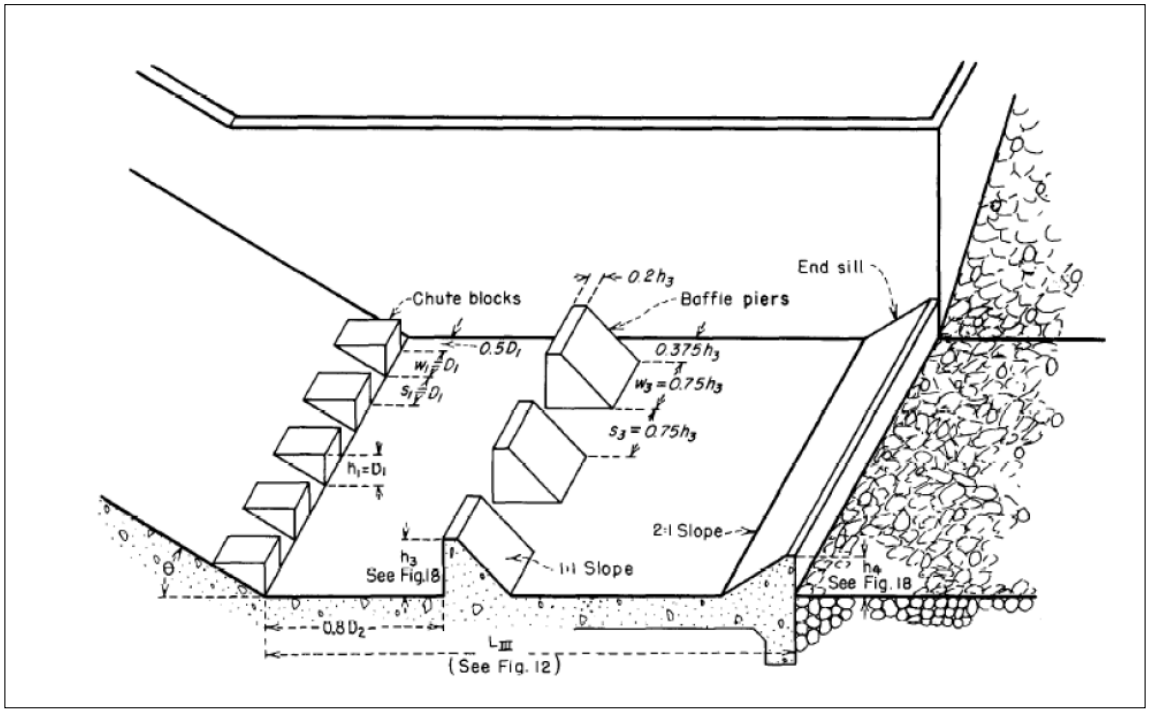

Figure A.6: Feature detail for type III basin (USBR)

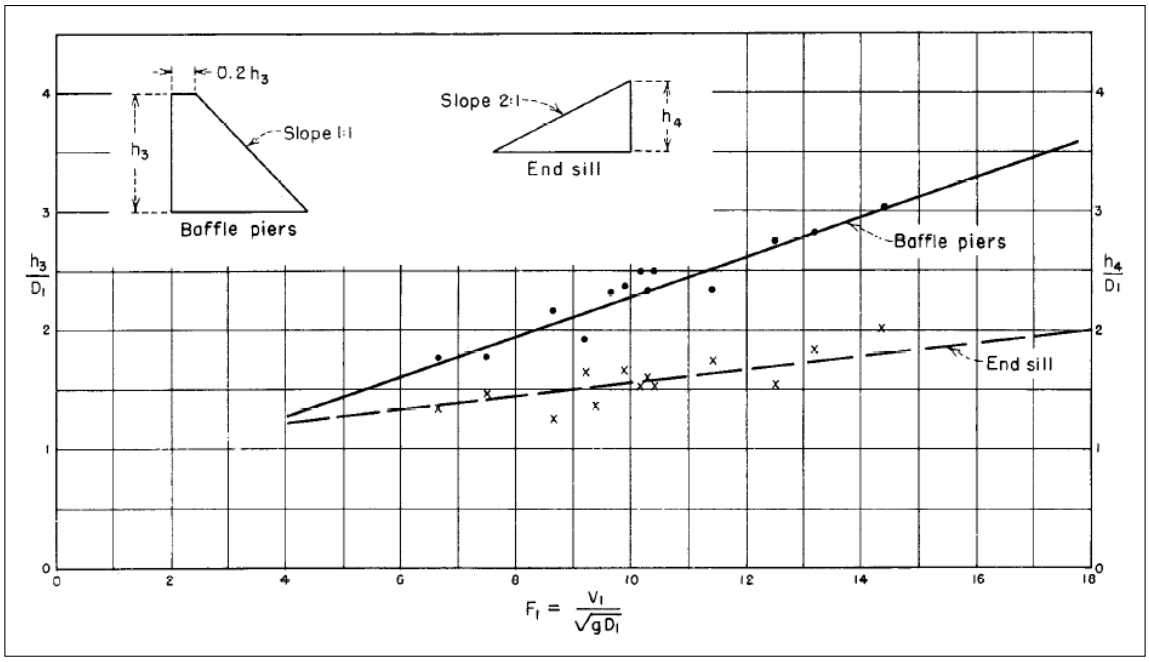

Figure A.7: Chart for h_3/D_1 and h_4/D_1 as a function of Fr_2

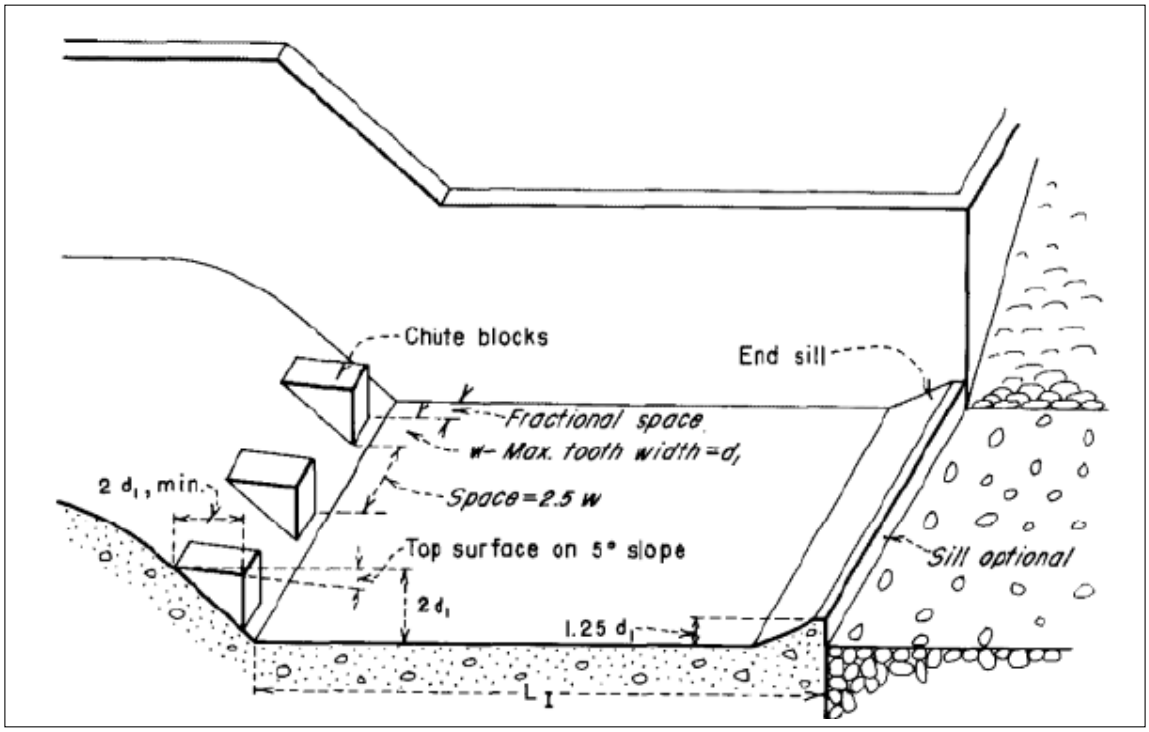

Figure A.8: Feature detail for type IV basin (USBR)

A.3 Matlab® Scripts

A.3.1 Option 1 Master Script

```
clc
clear all
clear all

% --- given properties ------------------------------------------------
elev_crest = 834.93;                                % crest elevation [ft]
elev_bed_u = 826.93;                          % upstream bed elevation [ft]
elev_bed_d = 819.34;
w = 160;                                              % system width [ft]
b = 10;                                              % crest breadth [ft]

% --- given constants -------------------------------------------------
g = 32.2;                                % acceleration of gravity [ft/s^2]
Ks = 0.002;                                      % concrete roughness [ft]

% --- preliminary calcs -----------------------------------------------
P = elev_crest - elev_bed_u;          % crest height above upstream bed [ft]
alpha = atan(1/2.5);                % ds face angle against horizontal (rad)

% --- crest head elevation calc ---------------------------------------

% for ff (years) =    5     25      50     100    500
          Q_array =  [4700  7600    8900   10200  13500];
          Tw_array = [826.82 828.77 829.56 830.31 832.14];

h_array = zeros(1,length(Q_array));
for ii = 1:length(Q_array)
[~, h_array(ii)] = func_opt1_h_from_Q(Q_array(ii),P,b,w,g, 1);
end

print('101','-depsc2','-r300');



% --- head loss calc --------------------------------------------------

   % --- prelininary elevation and array setup ----------------------------

z1 = elev_crest;

      ql_array = zeros(1,length(Q_array));
      y2_array = zeros(1,length(Tw_array));
      %z2_array = zeros(1,length(Q_array));
      hl_headloss_array = zeros(1,length(Q_array));
      hl_energy_array = zeros(1,length(Q_array));
      v2_array = zeros(1,length(Q_array));
      L_loss_array = zeros(1,length(Q_array));
      del_array = zeros(1,length(Q_array));
      del3_array = zeros(1,length(Q_array));
      Fr2_array = zeros(1,length(Q_array));
      y3_array = zeros(1,length(Q_array));
      y3_plus_z2_array = zeros(1,length(Q_array));
      L_over_y2_array = zeros(1,length(Q_array));
      L_array = zeros(1,length(Q_array));
      Dr_array = zeros(1,length(Q_array));
      Ld_array = zeros(1,length(Q_array));

for ii = 1:length(Q_array)
ql_array(ii) = Q_array(ii)/w;
end

   % --- determine z2 compatible with all flows ---------------------------

tol =0.1;             % threshold distance by which Tw exceeds y3 + z2 [ft]

z2=func_z2_solver(b,z1,alpha,ql_array,h_array,Ks,g,...
                                        Tw_array,elev_bed_d,0,tol);


   % --- y2 solve from Tw - store results --------------------------------

L_loss = b + (z1 - z2)/sin(alpha);
del = L_loss*0.02*(L_loss/Ks)^(-0.233);
del3 = 0.22*del;

for ii = 1:length(Q_array)
y2_array(ii)=...
       func_y2_solver_with_known_z2(z1,z2,ql_array(ii),h_array(ii),g,del3);

hl_headloss_array(ii)= h_array(ii) + z1- y2_array(ii) - ...
                                ((ql_array(ii)/y2_array(ii))^2)/(2*g) - z2;
L_loss_array(ii) = b + (z1 - z2)/sin(alpha);
del_array(ii) = L_loss_array(ii)*0.02*(L_loss_array(ii)/Ks)^(-0.233);
del3_array(ii) = 0.22*del_array(ii);
v2_array(ii) = ql_array(ii)/y2_array(ii);
hl_energy_array(ii)= del3_array(ii)*(v2_array(ii)^3)/(2*g*ql_array(ii));

disp({'hl_headloss',num2str(hl_headloss_array(ii))});
disp({'hl_energy_array',num2str(hl_energy_array(ii))});


assert(abs(hl_headloss_array(ii) - hl_energy_array(ii)) <0.001,'hl check');

Fr2_array(ii) = (ql_array(ii)/y2_array(ii))/sqrt(g*y2_array(ii));
y3_array(ii)  = y2_array(ii)*0.5*(sqrt(1 + 8*Fr2_array(ii)^2) - 1);
y3_plus_z2_array(ii) = y3_array(ii) + z2;
disp(y3_plus_z2_array(ii));
disp(Tw_array(ii));

 assert(Tw_array(ii) - y3_plus_z2_array(ii) > 0,'Tw check')

 %  --- calc basin length ------------------------------------------------
 L_over_y2_array(ii) = func_opt1_L_over_y2_from_Fr2(Fr2_array(ii), 0);
 L_array(ii) = L_over_y2_array(ii)*y3_array(ii);
end

    % --- output results -------------------------------------------------
opt1_table1 = zeros(length(Q_array),13);
opt1_table1(:,1) = Q_array;
opt1_table1(:,2) = ql_array;
opt1_table1(:,3) = y2_array;
opt1_table1(:,4) = z2;
opt1_table1(:,5) = hl_headloss_array;
opt1_table1(:,6) = v2_array;
opt1_table1(:,7) = hl_energy_array;
opt1_table1(:,8) =  Fr2_array;
opt1_table1(:,9) = y3_array;
opt1_table1(:,10) = y3_plus_z2_array;
opt1_table1(:,11) = Tw_array;
opt1_table1(:,12) = L_over_y2_array;
opt1_table1(:,13) = L_array;

xlswrite('opt1_table',opt1_table1);

    % --- check Ld -------------------------------------------------------

for ii = 1:length(Dr_array);
Dr_array(ii) = ql_array(ii)^2/(g*(z1 - z2)^3);
Ld_array(ii) = 4.3*(z1-z2)*Dr_array(ii)^0.27;
end

Ld_table = zeros(length(Q_array),4);
Ld_table(:,1) = [5 25 50 100 500];
Ld_table(:,2) = Q_array;
Ld_table(:,3) = Dr_array;
Ld_table(:,4) = Ld_array;
%xlswrite('Ld_table',Ld_table);

   % --- sizing basin features ------------------------------------------
opt1_sizing_table = zeros(11, 1);

D1 = y2_array(5);
D2 = y3_array(5);
Fr2_array =    [ 6.9085    5.7099    5.3652    5.0861    4.5767];

opt1_sizing_table(1,1) = D1;          %h1
opt1_sizing_table(2,1) = D1;          %s1
opt1_sizing_table(3,1) = D1;          %w1
opt1_sizing_table(4,1) = 0.5*D1;      % cb_wall_gap
opt1_sizing_table(5,1) = 0.8*D2;      % toe_bp_spacing
h3 = func_h3_over_D1_from_Fr2(Fr2_array(5),0)*D1;
h4 = func_h4_over_D1_from_Fr2(Fr2_array(5),0)*D1;
opt1_sizing_table(6,1) = h3;              %h3
opt1_sizing_table(7,1) = h4;              %h4
opt1_sizing_table(8,1) = 0.375*h3;        %bp_wall_gap
opt1_sizing_table(9,1) = 0.2*h3;          %bp_top_width
opt1_sizing_table(10,1) = 0.75*h3;        %w3
opt1_sizing_table(11,1) = 0.75*h3;        %s3
clc
%xlswrite('opt1_sizing_table',opt1_sizing_table);


% --- basin cost ------------------------------------------------------

t = 4;
conc_slope_volume = 0.5*(z1 - z2)*(z1-z2)*2.5*w;
con_basin_volume = t*L_array(5)*w;
opt1_total_conc_volume = conc_slope_volume + con_basin_volume;
exc_volume = (elev_bed_d - z2 + t)*L_array(5)*w;

% --- plot y3 and Tw vs Q ---------------------------------------------

figure; axis equal
plot(Q_array, Tw_array - z2); hold on
plot(Q_array, y3_array);
legend('T_w - z_2','y_3','location','southeast');
xlabel('Q (cfs)'); ylabel('depth (ft)');
grid minor;

print('105','-depsc2','-r300');
```

A.3.2 Option 2 Master Script

```
clc
clear all
clear all

% --- given properties ------------------------------------------------
elev_crest = 834.93;                                % crest elevation [ft]
z1= elev_crest;
elev_bed_u = 826.93;                          % upstream bed elevation [ft]
elev_bed_d = 819.34;
w = 160;                                                % system width [ft]
b = 10;                                                 % crest breadth [ft]

% --- given constants -------------------------------------------------
g = 32.2;                                  % acceleration of gravity [ft/s^2]
Ks = 0.002;                                       % concrete roughness [ft]
k = 4.5;                                    % for stepped spillway equations

% --- preliminary calcs -----------------------------------------------
P = elev_crest - elev_bed_u;          % crest height above upstream bed [ft]

% --- crest head elevation calc ---------------------------------------

% for ff (years) =      5     25      50     100    500
          Q_array    =  [4700  7600    8900   10200  13500];
          Tw_array   = [826.82 828.77 829.56 830.31 832.14];
          ql_array   = Q_array/w;
          yc_array   = zeros(1,length(Q_array));
          yc_over_h_array = zeros(1, length(Q_array));
          y2_array   = zeros(1,length(Q_array));
          v2_array   = zeros(1,length(Q_array));
          Fr2_array  = zeros(1,length(Q_array));
          y3_array   = zeros(1,length(Q_array));
          Tw_depth_array = zeros(1,length(Q_array));
          Tw_over_y3_array = zeros(1,length(Q_array));
          L_over_y2_array = zeros(1,length(Q_array));
          LB_array = zeros(1,length(Q_array));
          Dr_array = zeros(1,length(Q_array));
          LD_array = zeros(1,length(Q_array));
          LD_over_Ls_array = zeros(1,length(Q_array));

% --- determination of parameters -------------------------------------
h = 1;
s = 2.3;
N = 17;

z2 = z1 - N*h;
alpha = atan(h/s);  %radians
cmean = 1.44*sin(alpha) - 0.08;
fd = 2/sqrt(pi)/k;
fc = fd*0.5*(1 + tanh(2.5*((0.5 - cmean)/(cmean*(1-cmean)))));
yc_over_h_nappe = 0.89 - 0.4*h/s;
yc_over_h_skim = 1.2 - 0.325*h/s;

for ii = 1:length(Q_array)
yc_array(ii) = (ql_array(ii)^2/g)^(1/3);
yc_over_h_array(ii) = yc_array(ii)/h;
assert(yc_over_h_array(ii) > yc_over_h_skim,'not skimming flow');
y2_array(ii) = yc_array(ii)*(fc/(8*sin(alpha)))^(1/3);
v2_array(ii) = ql_array(ii)/y2_array(ii);
if v2_array(ii) > 60
disp( 'not type III - v2');
end
Fr2_array(ii) = v2_array(ii)/sqrt(y2_array(ii)*g);
if Fr2_array(ii) < 4.5
disp('not type III - Fr');
end
y3_array(ii) = y2_array(ii)*0.5*(sqrt(1+ 8*Fr2_array(ii)^2) - 1);
Tw_depth_array(ii) = Tw_array(ii) - z2;
Tw_over_y3_array(ii) = Tw_depth_array(ii)/(y3_array(ii));
if Tw_over_y3_array(ii) < 1
   disp('HJ uncontained');
end

L_over_y2_array(ii) = func_opt1_L_over_y2_from_Fr2(Fr2_array(ii), 0);
LB_array(ii)= L_over_y2_array(ii)*y3_array(ii);
Dr_array(ii) = ql_array(ii)^2/(g*(z1 - z2)^3);
LD_array(ii) = 4.3*(z1 - z2)*Dr_array(ii)^(0.27);
LD_over_Ls_array(ii) = LD_array(ii)/(N*s);
if LD_over_Ls_array(ii) > 1
    disp('Ld exceeds');
end
end

opt2_table = zeros(length(Q_array),  18);
opt2_table(:,1) =  [ 5     25      50     100    500 ];
opt2_table(:,2) = Q_array;
opt2_table(:,3) = ql_array;
opt2_table(:,4) = yc_array;
opt2_table(:,5) = yc_over_h_array;
opt2_table(:,6) = yc_over_h_nappe*ones(1,length(Q_array));
opt2_table(:,7) = yc_over_h_skim*ones(1,length(Q_array));
opt2_table(:,8) = y2_array;
opt2_table(:,9) = v2_array;
opt2_table(:,10) = Fr2_array;
opt2_table(:,11) = y3_array;
opt2_table(:,12) = Tw_depth_array;
opt2_table(:,13) = Tw_over_y3_array;
opt2_table(:,14) = L_over_y2_array;
opt2_table(:,15) = LB_array;
opt2_table(:,16) = Dr_array;
opt2_table(:,17) = LD_array;
opt2_table(:,18) = LD_over_Ls_array;
%xlswrite('opt2_table03',opt2_table)

% --- sizing basin features -------------------------------------------
opt2_sizing_table = zeros(11, 1);

D1 = y2_array(5);
D2 = y3_array(5);

opt2_sizing_table(1,1) = D1;        %h1
opt2_sizing_table(2,1) = D1;        %s1
opt2_sizing_table(3,1) = D1;        %w1
opt2_sizing_table(4,1) = 0.5*D1;    % cb_wall_gap
opt2_sizing_table(5,1) = 0.8*D2;    % toe_bp_spacing
h3 = func_h3_over_D1_from_Fr2(Fr2_array(5),0)*D1;
h4 = func_h4_over_D1_from_Fr2(Fr2_array(5),0)*D1;
opt2_sizing_table(6,1) = h3;            %h3
opt2_sizing_table(7,1) = h4;            %h4
opt2_sizing_table(8,1) = 0.375*h3;      %bp_wall_gap
opt2_sizing_table(9,1) = 0.2*h3;        %bp_top_width
opt2_sizing_table(10,1) = 0.75*h3;      %w3
opt2_sizing_table(11,1) = 0.75*h3;      %s3
clc
xlswrite('opt2_sizing_table',opt2_sizing_table);

% --- basin cost ------------------------------------------------------

t = 4;
conc_slope_volume = 0.5*(s*N)*(h*N)*w;
conc_steps_volume = 0.5*(s*h)*N*w;
conc_basin_volume = t*(LB_array(5))*w;
cxc_basin_volume = (t + (elev_bed_d - z2))*LB_array(5)*w;

% --- plot y3 and Tw vs Q ---------------------------------------------

figure; axis equal
plot(Q_array, Tw_array - z2); hold on
plot(Q_array, y3_array);

legend('T_w - z_2','y_3','location','southeast');
xlabel('Q (cfs)'); ylabel('depth (ft)');
grid minor;

print('106','-depsc2','-r300');
```

A.3.3 Option 3 Master Script

```
clc
clear all
clear all

% --- given properties ------------------------------------------------
elev_crest = 834.93;                                % crest elevation [ft]
z1= elev_crest;
elev_bed_u = 826.93;                          % upstream bed elevation [ft]
elev_bed_d = 819.34;
w = 160;                                              % system width [ft]
b = 10;                                              % crest breadth [ft]

% --- given constants -------------------------------------------------
g = 32.2;                                  % acceleration of gravity [ft/s^2]
Ks = 0.002;                                      % concrete roughness [ft]
k = 4.5;                                   % for stepped spillway equations

% --- preliminary calcs -----------------------------------------------
P = elev_crest - elev_bed_u;          % crest height above upstream bed [ft]

% --- crest head elevation calc ---------------------------------------

z2 = elev_bed_d-0.66;
d = z1 - z2;

% for ff (years) =      5     25      50     100    500
         Q_array   =  [4700  7600    8900   10200  13500];
         Tw_array  = [826.82 828.77 829.56 830.31 832.14];
         ql_array  = Q_array/w;
         Dr_array = zeros(1,length(Q_array));
         y2_array = zeros(1,length(Q_array));
         yc_array = zeros(1,length(Q_array));
         y3_array = zeros(1,length(Q_array));
         y3_over_Tw_array = zeros(1,length(Q_array));
         Fr2_array = zeros(1,length(Q_array));
         V2_array = zeros(1,length(Q_array));
         LD_array = zeros(1,length(Q_array));
         Lj_array = zeros(1,length(Q_array));
         LB_array = zeros(1,length(Q_array));

for ii = 1:length(Q_array)
    Dr_array(ii) = (ql_array(ii)^2)/(g*(z1 - z2)^3);
    y2_array(ii) = d*0.54*Dr_array(ii)^0.425;
    yc_array(ii) = (ql_array(ii)^2/g)^(1/3);
    y3_array(ii) = d*1.66*Dr_array(ii)^0.27;
    y3_over_Tw_array(ii)= y3_array(ii)/(Tw_array(ii) - z2);
    disp(y3_over_Tw_array(ii));
    if  y3_over_Tw_array(ii) > 1
       disp('HJ not contained');
    end

    V2_array(ii) = ql_array(ii)/y2_array(ii);
    LD_array(ii) = d*4.3*Dr_array(ii)^0.27;
    Fr2_array(ii) = ql_array(ii)/sqrt(g*y2_array(ii)^3);
    Lj_array(ii) = y2_array(ii)*220*tanh(((Fr2_array(ii) - 1)/22));
    LB_array(ii) = Lj_array(ii) + LD_array(ii);
end

opt3_table = zeros(length(Q_array),12);
opt3_table(:,1) = [    5     25      50     100    500    ];
opt3_table(:,2) = Q_array;
opt3_table(:,3) = Dr_array;
opt3_table(:,4) = y2_array;
opt3_table(:,5) = yc_array;
opt3_table(:,6) = y3_array;
opt3_table(:,7) = y3_over_Tw_array;
opt3_table(:,8) = Fr2_array;
opt3_table(:,9) = V2_array;
opt3_table(:,10) = LD_array;
opt3_table(:,11) = Lj_array;
opt3_table(:,12) = LB_array;

xlswrite('opt3_table03',opt3_table);

% --- feature sizing --------------------------------------------------

endsill = 1.25*y2_array(5);

% --- basin cost ------------------------------------------------------

t = 4;
conc_basin_volume = t*(LB_array(5))*w;
cxc_basin_volume = (t + (elev_bed_d - z2))*LB_array(5)*w;

% --- plot y3 and Tw vs Q ---------------------------------------------

figure; axis equal
plot(Q_array, Tw_array - z2); hold on
plot(Q_array, y3_array);

legend('T_w - z_2','y_3','location','southeast');
xlabel('Q (cfs)'); ylabel('depth (ft)');
grid minor;

print('107','-depsc2','-r300');
```

A.3.4 Function: h = f(Q) Solver

```
function [Cd_out, h_out] = func_opt1_h_from_Q(Q,P,b,w,g, plot_fig)

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

ori = [68 547];
x_end = 449;
y_top = 69;

pivot_x = 199;

scale_xx = (x_end - ori(1))/(0.85 - 0.10);
scale_yy = (ori(2)- y_top)/ (1 - 0.85);

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

int_mat_xx =   [pivot_x; 247; 323; 425; 450];

int_mat_yy = ...
  [546   528  516  487  460  428  388;
   528   510  498  469  441  409  370;
   455   439  426  395  365  332  293;
   328   314  298  265  232  198  156;
   306   293  277  243  210  176  133;
  .350  .375 .400 .450 .500 .550 .600];


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

[rows, cols] = size(int_mat_yy);
AA = ones(rows-1,rows-1);

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

CC = zeros(rows-1,cols);

for kk = 1:cols
CC(:,kk) = AA\int_mat_yy(1:rows-1,kk);
end

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

N_plot = 1000;
He_over_b_array = linspace(ori(1),x_end,N_plot);
%He_over_b_array = linspace(ori(1),532,N_plot);
Cd_plot = zeros(cols,N_plot);

for kk = 1:cols
 for ii = 1:N_plot
  if He_over_b_array(ii) < pivot_x
        Cd_plot(kk,ii) = int_mat_yy(1,kk);
  else
    for jj = 1:rows-1
        Cd_plot(kk,ii) = Cd_plot(kk,ii) + ...
                                CC(jj,kk)*He_over_b_array(ii)^(rows-1-jj);
    end
  end
 end
end

% --- interpolate initial output value, with Cd = 0.85 ----------------


trial_Cd = 1.11;
Cd_error = 1.0;
tolerance = 1e-4;
while Cd_error > tolerance
trial_h = (3*Q^(2/3))/(2*(trial_Cd*w)^(2/3)*g^(1/3));
trial_h_over_h_plus_P = trial_h/(trial_h + P);
trial_h_over_b = trial_h/b;
if trial_h_over_h_plus_P < 0.35
    trial_h_over_h_plus_P = 0.350001;
end

trial_h_over_b_plot = (trial_h_over_b-0.1)*scale_xx + ori(1);

for nn = 1:N_plot-1
   if  He_over_b_array(nn) > trial_h_over_b_plot
       xx_index = nn;
       break
   end
end

kk_index = 1;
for kk = 1:cols-1
    if trial_h_over_h_plus_P > int_mat_yy(rows, kk) && ...
       trial_h_over_h_plus_P < int_mat_yy(rows, kk+1)
        kk_index = kk;
        break
    end
end

trial_Cd_out_plot_L = Cd_plot(kk_index,xx_index);
trial_Cd_out_plot_U = Cd_plot(kk_index+1,xx_index);

trial_Cd_out_plot = (trial_Cd_out_plot_U - trial_Cd_out_plot_L)/...
                    (int_mat_yy(rows,kk+1) - int_mat_yy(rows,kk))*...
                    (trial_h_over_h_plus_P - int_mat_yy(rows,kk))+...
                    trial_Cd_out_plot_L;

trial_Cd_out = 1.0 - (trial_Cd_out_plot-y_top)/scale_yy;


Cd_error = abs(trial_Cd_out - trial_Cd);
trial_Cd = trial_Cd - tolerance/10;
end
Cd_out = trial_Cd;
h_out = trial_h;


% --- plot results if requested ---------------------------------------
if plot_fig == 1

I = imread('10.10_read.png');
B = imrotate(I,-0.45);

figure
imshow(B); hold on;
plot(ori(1),ori(2),'or');
plot([ori(1) x_end],[ori(2) ori(2)],'-k','linewidth',2);
plot([ori(1) ori(1)],[y_top ori(2)],'-k','linewidth',2);
for kk = 1:cols
plot(He_over_b_array,Cd_plot(kk,:),'','linewidth',2);
end

 for ii = 1:rows-1
     for jj = 1:length(int_mat_yy)
    plot(int_mat_xx(ii),int_mat_yy(ii,jj),'ro');
    plot(ori(1),int_mat_yy(1,jj),'ro');
     end
 end

 %plot(He_over_b_array(xx_index),trial_Cd_out_plot_L,'bo');
 %plot(He_over_b_array(xx_index),trial_Cd_out_plot_U,'bo');
 %plot(He_over_b_array(xx_index),trial_Cd_out_plot,'co');

% h1 = plot(He_over_P_array_plot,CT_mat_plot(row_aa,:),'-r','linewidth',2);
% h2 = plot(He_over_P_array_plot,CT_mat_plot(row_bb,:),'-b','linewidth',2);
% h3 = plot(He_over_P_array_plot,CT_array_out_plot,':c','linewidth',2);
% legend([h1 h2 h3],{'lower \alpha','upper \alpha','interpolated \alpha'},...
%     'location','northeast');
end
end
```

A.3.5 Function: option 1 z_2 Solver

```
function[z2_out] = func_z2_solver( b, z1, alpha,ql_array,h_array,...
                                   Ks,g,Tw_array,elev_bed_d,gen_table,tol)

out_table = zeros(30,7);
          z2_array = zeros(1,length(ql_array));

for jj = 1:length(ql_array)

disp('---');
disp({'head-',num2str(h_array(jj))});

    diff = 2;
    ql = ql_array(jj);
    h = h_array(jj);
    Tw = Tw_array(jj);

z2 = elev_bed_d - 10;

tt = 1;
while diff > tol;

syms y2
% --- caulculate head loss according to hl equation -----------------------

L_loss = b + (z1 - z2)/sin(alpha);
del = L_loss*0.02*(L_loss/Ks)^(-0.233);
del3 = 0.22*del;

d = y2;
U = ql/d;

% --- calculate head loss according to energy balance ---------------------

v2 = ql/y2;
hl_headloss = (del3*U^3)/(2*g*ql);
hl_energy = h + z1 - y2 - (v2^2)/(2*g) - z2;

% --- account for hydraulic jump ------------------------------------------

y2 = double(vpasolve(hl_energy==hl_headloss,y2));

for ii = 1:length(y2)
   y2(ii) = (real(sqrt(y2(ii))))^2;
   if y2(ii) > h
       y2(ii) = 0;
   end
end

y2 = sum(y2);

Fr2 = (ql/y2)/sqrt(g*y2);
y3 = y2*0.5*(sqrt(1 +8*Fr2^2) - 1);

diff = Tw - (y3 + z2);
disp(diff);

out_table(tt,1) = tt - 1;
out_table(tt,2) = z2;
out_table(tt,3) = h + z1 - y2 - ((ql/y2)^2)/(2*g) - z2;
out_table(tt,4) = y2;
out_table(tt,5) = y3;
out_table(tt,6) = Tw - (y3+ z2);

z2 = z2 + diff*0.5;

out_table(tt,7) = z2;
tt = tt + 1;

end

z2 = z2 - diff*0.5;

out_table(tt-1,7) = z2;

if gen_table == 1 && jj == length(ql_array);
xlswrite('tab02',out_table);
end

z2_array(jj) = z2;

end
z2_out = min(z2_array);

end
```

A.3.6 Function: $y_2 = f(z_2)$ Solver

```
function[y2_out]=func_y2_solver_with_known_z2(z1,z2,ql,h,g,del3)

% --- caulculate head loss according to hl equation -----------------------


syms y2
U = ql/y2;

% --- calculate head loss according to energy balance ---------------------

v2 = ql/y2;
hl_headloss = (del3*U^3)/(2*g*ql);
hl_energy = h + z1 - y2 - (v2^2)/(2*g) - z2;

% --- solve for y2 and output ---------------------------------------------

y2 = double(vpasolve(hl_energy==hl_headloss,y2));

% --- remove negative solutions for y2 that are negative and above y1 -----

for ii = 1:length(y2);
if y2(ii) > h
    y2(ii)= 0;
end
if y2(ii) < 0
    y2(ii)= 0;
end
end

% --- output value --------------------------------------------------------

y2_out = sum(y2);

end
```

A.3.7 Function: $\frac{L_j}{y_2} = f(Fr_2)$ Reader

```
function [L_over_y2_out] = func_opt1_L_over_y2_from_Fr2(Fr2, plot_fig)

%plot_fig = 1;
%Fr2 = Fr2_array(1);

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

ori = [59 578];
x_end = 896;
y_top = 68;

scale_xx = (x_end - ori(1))/20;
scale_yy = (ori(2)- y_top)/6;

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

int = [249 388; 322 367; 436 351; 586 345  ;747 348 ];

% --- 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;

Fr1_array = linspace(int(1,1),max(int(:,1)),N_plot);
L_over_y1_array = zeros(N_plot);

for ii = 1:N_plot
     for jj = 1:rows
         L_over_y1_array(ii) = L_over_y1_array(ii) + ...
                               CC(jj)*Fr1_array(ii)^(rows-jj);
     end
end

% --- interpolate initial output value, with Cd = 0.85 --------------------

Fr1_unscaled = Fr2;                          % accounting for chart notation
Fr1 = Fr1_unscaled*scale_xx + ori(1);

L_over_y1= 0;

 for jj = 1:rows
        L_over_y1 = L_over_y1 + CC(jj)*Fr1^(rows-jj);
 end

L_over_y1_unscaled = 6 - (L_over_y1 - y_top)/scale_yy;
L_over_y2_out = L_over_y1_unscaled;

% --- plot results if requested -------------------------------------------
if plot_fig == 1

    figure;
I = imread('fig6.png');
B = imrotate(I,0.00);

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(Fr1_array, L_over_y1_array,'m','linewidth',2);
%plot(Fr1, L_over_y1,'co','linewidth',2);

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


end
end
```

A.3.8 Function: $\frac{h_3}{D_1} = f(Fr_2)$ Reader

```
function [h3_over_D1_out] = func_h3_over_D1_from_Fr2(Fr2, plot_fig)

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

ori = [36 459];
x_end = 891;
y_top = 67;
y_top_unscaled = 4 ;
x_end_unscaled = 18;

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

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

int = [227 332 ;880 105 ];

% --- 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;

Fr1_array = linspace(int(1,1),max(int(:,1)),N_plot);
h3_over_D1_array = zeros(N_plot);

for ii = 1:N_plot
     for jj = 1:rows
         h3_over_D1_array(ii) = h3_over_D1_array(ii) + ...
                                CC(jj)*Fr1_array(ii)^(rows-jj);
     end
end

% --- interpolate initial output value, with Cd = 0.85 -----------------

Fr1_unscaled = Fr2;                          % accounting for chart notation
Fr1 = Fr1_unscaled*scale_xx + ori(1);

h3_over_D1= 0;

 for jj = 1:rows
         h3_over_D1 = h3_over_D1 + CC(jj)*Fr1^(rows-jj);
 end

h3_over_D1_unscaled = y_top_unscaled - (h3_over_D1 - y_top)/scale_yy;
h3_over_D1_out = h3_over_D1_unscaled;

% --- plot results if requested ----------------------------------------
if plot_fig == 1

    figure;
I = imread('fig18.png');
B = imrotate(I,0.00);

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(Fr1_array, h3_over_D1_array,'m','linewidth',2);
plot(Fr1, h3_over_D1,'co','linewidth',2);

end
end
```

A.3.9 Function: $\frac{h_4}{D_1} = f(Fr_2)$ Reader

```
function [h4_over_D1_out] = func_h4_over_D1_from_Fr2(Fr2, plot_fig)

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

ori = [36 459];
x_end = 891;
y_top = 67;
y_top_unscaled = 4 ;
x_end_unscaled = 18;

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

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

int = [227 340 ;891 261 ];

% --- 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;

Fr1_array = linspace(int(1,1),max(int(:,1)),N_plot);
h4_over_D1_array = zeros(N_plot);

for ii = 1:N_plot
     for jj = 1:rows
         h4_over_D1_array(ii) = h4_over_D1_array(ii) + ...
                                CC(jj)*Fr1_array(ii)^(rows-jj);
     end
end

% --- interpolate initial output value, with Cd = 0.85 -----------------

Fr1_unscaled = Fr2;                          % accounting for chart notation
Fr1 = Fr1_unscaled*scale_xx + ori(1);

h4_over_D1= 0;

 for jj = 1:rows
         h4_over_D1 = h4_over_D1 + CC(jj)*Fr1^(rows-jj);
 end

h4_over_D1_unscaled = y_top_unscaled - (h4_over_D1 - y_top)/scale_yy;
h4_over_D1_out = h4_over_D1_unscaled;

% --- plot results if requested ----------------------------------------
if plot_fig == 1

  figure;
I = imread('fig18.png');
B = imrotate(I,0.00);

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),'or','linewidth',2);
end

plot(Fr1_array, h4_over_D1_array,'r','linewidth',2);
plot(Fr1, h4_over_D1,'co','linewidth',2);

disp(h4_over_D1_out);

end
end
```