UNIVERSITY OF MINNESOTA: TWIN CITIES

CE 4511 HYDRAULIC STRUCTURES

HW 4: Labyrinth Weir

CHRIS [redacted] LOGSTON

MARCH 22ND, 2015

Contents

0 Preliminaries

0.1 Introduction

This assignment explores the benefits of a Labyrinth Spillway's discharge capacity over its linear counterparts. As shown in sections a and b, a labyrinth weir can support a significantly greater discharge at the same approach surface elevation. This improved performance does come in exchange of greater design complexity, footprint, and construction costs.

The discharge capacity of a labyrinth weir can be improved by adjusting various geometric parameters, as discussed in section c. Given the sophisticated, interdependent effects of these basic parameters on the capacity, many adjustments can be made for specific expected flows. In this report, improvements are only speculated, and more in depth analysis is reserved for future discussion.

Finally, as shown in section d, the water surface profile (WSP) along the down stream channel indicates a transition from sub- to super-critical flow, after which the hydraulic depth essentially plateaus and dips slightly. The sophistication of this WSP is due in part to the non-constant discharge along the channel length, given that each horizontal position receives a different amount of flow from different stretches of the inlet crest above.

It should be noted that while accuracy is prioritized, the methods used in the report are prone to error, the sources of which include, but are not limited to:

- the ignoring of tail-water effects;
- the ignoring of discharge over the flow-perpendicular portions of the weir;
- the approximation of the plan-trapezoidal system as a triangular one, therefore producing a conservative estimate of total discharge capacity;
- the difference in acceleration of gravity present at differing elevations;
- the approximation of C_T for a non-researched α through interpolation.

0.2 Given Properties

Description	Symbol	Value
upstream apron elevation	UAE	1971 ft
crest elevation	CE	1975 ft
maximum water surface elevation	$MWSE$	1978 ft
channel bed slope	s_0	5 %
Manning's n of weir	n	0.014

Table 0.1: Given Weir Property Values

- the total labyrinth weir system consists of 2 cycles ($N = 2$);
- as shown in given channel plan view (appendix figure A.1):
  - each channel top width $\left(w_{\text{top}}\right)$ increases linearly downstream from $w_{\text{top}_{\text{min}}} = 24\ ft$ to $w_{\text{top}_{\text{max}}} = 60\ ft$;
  - each channel bottom width (w_{bot}) increases linearly downstream from $w_{\text{bot}_{\text{min}}} = 20\ ft$ to $w_{\text{bot}_{\text{max}}} = 50\ ft$;
  - each channel length is $S = 120\ ft$
- as shown in the given channel cross sectional view (appendix figure A.2):
  - each weir's crest shape is "quarter round";
  - each channel's cross section's trapezoidal side slope is $z = 0.5$;

0.2.1 Preliminary Depth Calculations

Using the values given in table 0.1 and read appendix figures A.1 and A.2, the following depths are determined:

- Dam height: $(P) = CE - UAE = 1975\ ft - 1971\ ft = 4\ ft$
- Maximum hydraulic depth at crest: $(H_{\text{max}}) = MWSE - CE = 1978\ ft - 1975\ ft = 3\ ft$

0.3 Geometric Analysis

Later calculations require several geometric properties for the weir that are not given. As shown in figure 0.1, these values include the channel plan-angle α as well as the total developed weir length L_{lbyr}.

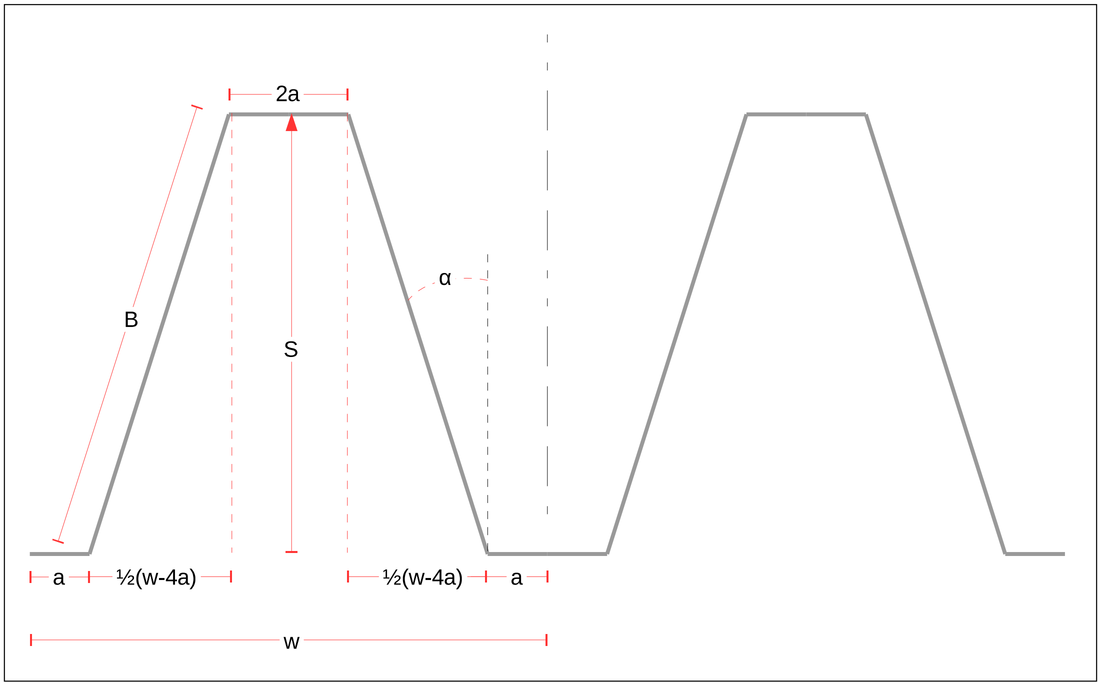

Figure 0.1: Top geometric properties for one of both weir cycles

- the perpendicular crest portion can be determined from given top channel width as follows:

$$2a = w_{\text{top}_{\text{min}}} = 24\ ft \to a = \frac{24\ ft}{2} = 12\ ft$$

- using a and the outlet top width, the total width for a single cycle is calculated as follows:

$$w = w_{\text{top}_{\text{max}}} + 2a = 60\ ft + 2(12\ ft) = 84\ ft$$

- with this, the diagonal crest length (B) can be calculated from w, a and S as follows:

$$B = \sqrt{(w-4a)^2 + S^2} = \sqrt{\left(\frac{1}{2}(84\ ft - 4(12\ ft))\right)^2 + (120\ ft)^2} = 121.3425\ ft$$

- with B known, α can be calculated as follows:

$$\alpha = \tan^{-1}\left(\frac{\frac{1}{2}(w-4a)}{S}\right) = \tan^{-1}\left(\frac{\frac{1}{2}(84\ ft - 4(12\ ft))}{120\ ft}\right) = 8.5308^\circ$$

- total labyrinth crest length per cycle can then be calculated as the sum of these lengths:

$$\frac{L_{\text{lbyr}}}{N} = a + B + 2a + B + a = 4a + 2B = 4(12\ ft) + 2(121.3425\ ft) = 290.6850\ ft$$

- and in turn, the total crest length of the weir over both cycles can be calculated as follows:

$$L_{\text{lbyr}} = N\left(\frac{L_{\text{lbyr}}}{N}\right) = 2(290.6850\ ft) = 581.3700\ ft$$

a) Discharge at Maximum Elevation

It is first requested to calculate the discharge over the weir when the maximum surface elevation occurs. This requires the use of the following equation:

$$Q = C_T L_{\text{lbyr}} \frac{2}{3} \sqrt{2g} H_e^{\frac{3}{2}} \tag{a.1}$$

Where:

- C_T is a discharge coefficient for quarter-round crest shapes, taken from appendix figure A.3;
- L_{lbyr} is the total labyrinth length;
- g is the acceleration of gravity $\left(\text{known to be } 32.2\ \frac{ft}{s^2}\right)$;
- H_e is the total head (energy and surface height) over the crest

Thus, with a known α, the discharge corresponding to a given hydraulic depth (H) can be solved using solver as follows:

1. discharge Q is guessed
2. approach velocity is calculated with guessed discharge:

$$V_0 = \frac{Q}{L_{\text{upstream width}}(P+H)} = \frac{Q}{2w(P+H)}$$

3. velocity head is calculated with guessed approach velocity:

$$h_a = \frac{V_0{}^2}{2g}$$

4. total energy head is calculated from guessed velocity head:

$$H_e = H + h_a$$

5. the discharge coefficient C_T is determined reading from the interpolated curve of $C_T = f\left(\alpha, \frac{H_e}{P}\right)$ shown in appendix figure A.4 for $\alpha_{int} = 8.5308°$
6. eq. a.1 is used to calculate a guessed discharge
7. this process is repeated/solved with variable Q such that the equivalence of eq. a.1 is held.

Using this process, Matlab® solver is used to determine the maximum head discharge with known values of α, H, P, L_{lbyr} and g in the following table:

Q $\left(\frac{ft^3}{s}\right)$	$\mathbf{V_0}$ $\left(\frac{ft}{s}\right)$	$\mathbf{h_a}$ (ft)	$\mathbf{H_e}$ (ft)	$\mathbf{C_T}$ -	**Q** $\left(\frac{ft^3}{s}\right)$
5797.7	4.9300	0.3774	3.3774	0.3003	5797.7

$$\boxed{Q_{\text{labyrinth}}(H_{\max} = 3\ ft) = 5797.7 \frac{ft^3}{s}}$$

It should be noted that this solving process assumes **no significant tailwater effects**.

b) Comparison to Linear Weir

In order to explore the efficacy of this labyrinth weir as compared to a perfectly linear weir in its place, the discharge of the same head is to be calculated for the latter structure. This is done as follows:

- the total, undeveloped length across both cycles L_0 is calculated from the given plan view (appendix figure A.1) as follows:

$$L_0 = Nw = N\left(w_{\text{top}_{\text{max}}} + w_{\text{top}_{\text{min}}}\right) = 2(60\ ft + 24\ ft) = 168\ ft$$

- the solving steps in section a are carried out again, with the following changes:
  - the length is set to $L = L_0 = 168\ ft$;
  - α is set to 90°, and C_T is read from the top curve of appendix figure A.3;
  - given that $\frac{H}{P}$ itself is above 0.5, and it is assumed that $\frac{H_e}{P}$ will only be larger, this C_T is set to equal 0.76.

With this, Matlab® solver is again used to determine maximum head discharge over a linear weir. These results are shown in the following table:

Q $\left(\frac{ft^3}{s}\right)$	$\mathbf{V_0}$ $\left(\frac{ft}{s}\right)$	$\mathbf{h_a}$ (ft)	$\mathbf{H_e}$ (ft)	$\mathbf{C_T}$ -	**Q** $\left(\frac{ft^3}{s}\right)$
3848.6	3.2726	0.1663	3.1663	0.76	3848.6

$$\boxed{Q_{\text{linear}}\left(H_{\text{max}} = 3\ ft\right) = 3848.6\frac{ft^3}{s}}$$

The improvement of the labyrinth weirs discharge capacity over that of its linear counterparts is determined:

$$\%\ \text{Improvement} = \frac{Q_{\text{labyrinth}} - Q_{\text{linear}}}{Q_{\text{linear}}} = \frac{5797.7\frac{ft^3}{s} - 3848.6\frac{ft^3}{\equiv}s}{3848.6\frac{ft^3}{s}} = 50.6447\%$$

Thus, as expected the labyrinth weir provided a significantly improved discharge capacity over its linear counterpart. In fact, at the maximum expected upstream surface elevation, this improvement is almost 51% .

c) Further Improvement

The purpose any labyrinth weir is to extend the crest length over which water spills such that the total discharge capacity is improved. Given the many geomtric properties at play, the discharge capacity of any weir can be improved in the following ways:

- increase number of cycles
  - while this decreases the size of a for each cycle, it adds an additional $2B$ to the total length.
  - this can be increased until a contributes nothing to the total length, producing a sawtooth appearance.
  - this will improve Q only until the effects of stream interference become prohibitive.
  - thus Q is maximized by adjusting cycle quantity somewhere along $0 < N < \infty$.
- adjust α
  - assuming a is kept constant, a lower α increases B.
  - this enlarged B increases total weir length.
  - given that $C_T = f\left(\alpha, \frac{H_e}{P}\right)$, there exists some maximum discharge between $\alpha = 0°$ and $\alpha = 90°$ where increasing α yields increasing C_T, but decreasing α yields a higher B.
  - thus, Q is maximized by adjusting α somewhere along $0° < \alpha < 90°$.
- adjust dam height
  - given that $C_T = f\left(\alpha, \frac{H_e}{P}\right)$, the P can be adjusted such that the C_T curve for the given α is at its maximum for an anticipated H_e.
  - while too high a dam height would completely restrict flow and cause upstream flooding, too low a dam height would yield a higher $\frac{H_e}{P}$, thus a lower C_T (after the C_T-curve's maximum), thus a lower Q.
  - thus, Q is maximized by adjusting P somewhere along $0 < P < \infty$.
- build the dam on a different planet
  - given that Q is proportional to the square root of g as shown in equation a.1, a greater g, such as that found on Jupiter, Neptune or Saturn would yield an improved Q.
  - such an approach would in itself incur several restrictions including:
    - cost of extraterrestrial construction
    - hostility of surrounding environment to human personnel
    - lack of flowing water to manage

In this case, given the general inability to:

- increase N without needing increase w or decrease a or B;
- build a dam elevation higher than the MWSE of 1978 ft;
- feasibly build in a field of larger g,

It is recommended to increase α slightly such that C_T is increased in exchange for a small decrease in L, thus increasing the total Q .

d) Water Surface Profile for 1 ft Hydraulic Depth

d.1) Finding Critical Depth and Position

The water surface profile along the length of the channel with an approach head of $H = 1\ ft$ is requested.

- using methods discussed in section a, the total discharge at this head calculated using equation a.1 and setting total head as $H_e = 1\ ft$

$$Q_{1ft} = 1645.6\frac{ft^3}{s}$$

- with the known developed length, the discharge per crest length q_l is calculated:

$$q_{l,1ft} = \frac{Q_{1ft}}{L_{\text{lbyr}}} = 6.8566\frac{ft^3}{s}$$

This process requires that critical depth y_c be determined along with its location along the channel x_c. This is done using a solver as follows:

1. x_c is guessed
2. the channel base width is determined for this guessed x_c
$$b = \frac{x_c}{S}\left(w_{\text{bot}_{max}} - w_{\text{bot}_{min}}\right) + w_{\text{bot}_{min}}$$
3. the discharge having spilled over the crest before the guessed x_c is determined:
$$Q = f(x) = q_{l,1ft}x_c$$
4. the associated critical depth is determined by finding whichever y_c satisfies the following equivalence for critical depth in a trapezoidal channel:
$$\frac{Q^2}{g} = \frac{\left(by_c + zy_c{}^2\right)^3}{b + 2zy_c}$$
5. this solved y_c for the guessed x_c is then used to determine wetted perimeter:
$$P_w = b + 2y_c\sqrt{1+z^2}$$
6. as well as cross sectional area of flow:
$$A_w = y_c\left(b + zy_c\right)$$
7. both wetted area and perimeter are then used to calculate the hydraulic radius:
$$R_w = \frac{A_w}{P_w}$$
8. and stream velocity (cross sectional mean):
$$V = \frac{Q}{A_w}$$
9. the guessed/determined velocity and hydraulic radius are then used with given values to determine friction slope:
$$s_f = \left(\frac{nV}{1.49R_w{}^{\frac{2}{3}}}\right)^2$$
10. finally, x_c is solved such that it satisfies the following energy slope balance equation:
$$s_f = s_0 - 2q_l\frac{V}{gA_w}$$

Using this process, Matlab® solver is used to determine y_c at x_c. These steps are shown in the following table:

$\mathbf{x_c}$	$\mathbf{Q}$	$\mathbf{b}$	$\mathbf{y_c}$	$\mathbf{P_w}$	$\mathbf{A_w}$	$\mathbf{R_w}$	$\mathbf{V}$	$\mathbf{s_f}$	$\mathbf{s_0 - 2q_l\frac{V}{gA_w}}$
(ft)	$\left(\frac{ft^3}{s}\right)$	(ft)	(ft)	(ft)	(ft^2)	(ft)	$\left(\frac{ft}{s}\right)$	-	-
70.63	484.2829	37.6575	1.7122	41.4861	65.9429	1.5895	7.3440	0.0026	0.0026

$$x_{c_{H=1ft}} = 70.6300\ ft$$

$$y_{c_{H=1ft}} = 1.7122\ ft$$

d.2) Water Surface Profile

With a known x_c and y_c, the sub- and super-critical portions of the water surface profile can be calculated separately and plotted along the same length.

d.2.1) Sub Critical Portion

For the sub critical portion, a starting x is chosen just before x_c. The same steps from section d.1 are then used to determine a Q, b, y, A_w, P_w, R_w, V_w and s_f from the guessed x. With those values, a froude number is calculated using eq. d.1.

$$Fr = \frac{V}{\sqrt{gy}} \tag{d.1}$$

After this, the non-linear exchange between hydraulic energy and friction losses is determined along the length of the channel via discretization by solving for the change in depth associated with a change in energy. This is accomplished by starting at the just-before-x_c value of $x_{i=1}$ and continuing in steps upstream $x_{i=2}$, $x_{i=3}$, executing the following calculations for each step:

1. an average velocity between steps is determined:

$$\overline{V_i} = \frac{V_i + V_{i-1}}{2} \text{ where } \overline{V_1} = 0$$

2. an average cross sectional flow area between steps is determined:

$$\overline{A_{w_i}} = \frac{A_{w_i} + V_{w_{i-1}}}{2} \text{ where } \overline{A_{w_1}} = 0$$

3. an average friction slope between steps is determined:

$$\overline{s_{f_i}} = \frac{s_{f_i} + s_{f_{i-1}}}{2} \text{ where } \overline{s_{f_1}} = 0$$

4. an average Froude number between steps is determined:

$$\overline{Fr_i} = \frac{Fr_i + Fr_{i-1}}{2} \text{ where } \overline{Fr_1} = 0$$

5. a change in distance along the channel is calculated:

$$\Delta x_i = x_i - x_{i-1} \text{ where } \Delta x_1 = 0$$

6. the associated change in hydraulic depth is calculated:

$$\Delta y_i = \Delta x_i \left(\frac{s_0 - \overline{s_{f_i}} - \frac{2q_l \overline{V_i}}{g \overline{A_i}}}{1 - \overline{Fr_i}^2} \right) \text{ where } \Delta y_i < 0 \text{ moving upstream}$$

7. the hydraulic depth calculated from the energy exchange is checked against the guessed/determined hydraulic depth that would be expected from the sequence:

$$y_{\text{next},i} = y_{i-1} - \Delta y_i$$

For each step, moving from x_c upstream, a set of guessed $x_{i \to end}$ is synthesised, used and modified such that the following constraints are maintained throughout:

- $Fr < 1$
- $y_{\text{next},i} = y_i$

This process requires the first few values of guessed x to be solved for ad-hoc. The perseverance of these constraints is also dependent on the step between each x. They are more easily upheld with smaller steps at the cost of additional calculation. A portion of this process is shown in table d.1. As shown in red, the Froude number for all steps of along this sub-critical region are less than 1, as expected.

x (ft)	Q $\left(\frac{ft^3}{s}\right)$	y (ft)	A_w $\left(ft^2\right)$	P_w (ft)	R_w (ft)	V $\left(\frac{ft}{s}\right)$	$\overline{V}$ $\left(\frac{ft}{s}\right)$	$\overline{A_w}$ $\left(ft^2\right)$	s_f -	$\overline{s_f}$ -	Fr -	$\overline{Fr}$ -	Δx (ft)	Δy (ft)	y_{nxt} (ft)	$y - y_{nxt}$ (ft)
70.600	484.280	1.712	65.943	41.486	1.590	7.344	0.000	0.000	0.003	0.000	0.9891	0.0000	0.000	0.000	0.000	0.000
70.200	481.330	1.712	65.748	41.378	1.589	7.321	7.332	65.846	0.003	0.003	0.9860	0.9876	0.400	0.000	1.712	0.000
69.700	477.910	1.711	65.512	41.252	1.588	7.295	7.308	65.630	0.003	0.003	0.9827	0.9844	0.500	0.001	1.711	0.000
69.000	473.110	1.711	65.178	41.075	1.587	7.259	7.277	65.345	0.003	0.003	0.9781	0.9804	0.700	0.001	1.711	0.000
68.303	468.330	1.710	64.843	40.898	1.586	7.223	7.241	65.010	0.002	0.003	0.9735	0.9758	0.697	0.001	1.710	0.000
67.606	463.550	1.709	64.507	40.722	1.584	7.186	7.204	64.675	0.002	0.002	0.9688	0.9712	0.697	0.001	1.709	0.000
66.909	458.770	1.708	64.170	40.545	1.583	7.149	7.168	64.338	0.002	0.002	0.9642	0.9665	0.697	0.001	1.708	0.000
66.212	453.990	1.706	63.832	40.369	1.581	7.112	7.131	64.001	0.002	0.002	0.9595	0.9618	0.697	0.001	1.706	0.000
65.515	449.210	1.705	63.492	40.192	1.580	7.075	7.094	63.662	0.002	0.002	0.9548	0.9571	0.697	0.001	1.705	0.000
64.818	444.430	1.704	63.152	40.015	1.578	7.038	7.056	63.322	0.002	0.002	0.9500	0.9524	0.697	0.001	1.704	0.000
64.121	439.650	1.703	62.811	39.838	1.577	7.000	7.019	62.981	0.002	0.002	0.9452	0.9476	0.697	0.001	1.703	0.000
63.424	434.880	1.702	62.468	39.661	1.575	6.962	6.981	62.639	0.002	0.002	0.9404	0.9428	0.697	0.001	1.702	0.000
62.727	430.100	1.701	62.125	39.484	1.573	6.923	6.942	62.297	0.002	0.002	0.9356	0.9380	0.697	0.001	1.701	0.000
62.030	425.320	1.699	61.780	39.307	1.572	6.884	6.904	61.952	0.002	0.002	0.9307	0.9331	0.697	0.001	1.699	0.000
61.333	420.540	1.698	61.434	39.130	1.570	6.845	6.865	61.607	0.002	0.002	0.9258	0.9282	0.697	0.001	1.698	0.000
60.636	415.760	1.697	61.087	38.953	1.568	6.806	6.826	61.261	0.002	0.002	0.9208	0.9233	0.697	0.001	1.697	0.000
59.939	410.980	1.695	60.739	38.775	1.566	6.766	6.786	60.913	0.002	0.002	0.9159	0.9184	0.697	0.001	1.695	0.000
59.242	406.200	1.694	60.390	38.598	1.565	6.726	6.746	60.565	0.002	0.002	0.9108	0.9134	0.697	0.001	1.694	0.000
58.545	401.420	1.692	60.040	38.420	1.563	6.686	6.706	60.215	0.002	0.002	0.9058	0.9083	0.697	0.002	1.692	0.000
57.848	396.640	1.691	59.688	38.242	1.561	6.645	6.666	59.864	0.002	0.002	0.9007	0.9032	0.697	0.002	1.691	0.000
57.152	391.870	1.689	59.335	38.064	1.559	6.604	6.625	59.512	0.002	0.002	0.8956	0.8981	0.697	0.002	1.689	0.000
56.455	387.090	1.687	58.981	37.886	1.557	6.563	6.584	59.158	0.002	0.002	0.8904	0.8930	0.697	0.002	1.687	0.000
55.758	382.310	1.686	58.626	37.708	1.555	6.521	6.542	58.803	0.002	0.002	0.8852	0.8878	0.697	0.002	1.686	0.000
55.061	377.530	1.684	58.269	37.530	1.553	6.479	6.500	58.447	0.002	0.002	0.8799	0.8826	0.697	0.002	1.684	0.000
54.364	372.750	1.682	57.911	37.352	1.550	6.437	6.458	58.090	0.002	0.002	0.8746	0.8773	0.697	0.002	1.682	0.000
53.667	367.970	1.680	57.552	37.173	1.548	6.394	6.415	57.731	0.002	0.002	0.8693	0.8720	0.697	0.002	1.680	0.000
52.970	363.190	1.678	57.191	36.995	1.546	6.351	6.372	57.371	0.002	0.002	0.8639	0.8666	0.697	0.002	1.678	0.000
52.273	358.410	1.676	56.829	36.816	1.544	6.307	6.329	57.010	0.002	0.002	0.8585	0.8612	0.697	0.002	1.676	0.000
51.576	353.640	1.674	56.465	36.637	1.541	6.263	6.285	56.647	0.002	0.002	0.8530	0.8558	0.697	0.002	1.674	0.000
50.879	348.860	1.672	56.101	36.458	1.539	6.218	6.241	56.283	0.002	0.002	0.8475	0.8503	0.697	0.002	1.672	0.000
50.182	344.080	1.670	55.734	36.279	1.536	6.174	6.196	55.917	0.002	0.002	0.8420	0.8447	0.697	0.002	1.670	0.000
49.485	339.300	1.667	55.367	36.100	1.534	6.128	6.151	55.550	0.002	0.002	0.8363	0.8392	0.697	0.002	1.667	0.000
48.788	334.520	1.665	54.997	35.920	1.531	6.083	6.105	55.182	0.002	0.002	0.8307	0.8335	0.697	0.002	1.665	0.000
48.091	329.740	1.663	54.627	35.741	1.528	6.036	6.059	54.812	0.002	0.002	0.8250	0.8278	0.697	0.002	1.663	0.000
47.394	324.960	1.660	54.254	35.561	1.526	5.990	6.013	54.440	0.002	0.002	0.8192	0.8221	0.697	0.002	1.660	0.000
46.697	320.180	1.658	53.880	35.381	1.523	5.943	5.966	54.067	0.002	0.002	0.8134	0.8163	0.697	0.003	1.658	0.000
46.000	315.400	1.655	53.505	35.201	1.520	5.895	5.919	53.693	0.002	0.002	0.8075	0.8104	0.697	0.003	1.655	0.000
45.303	310.630	1.652	53.128	35.021	1.517	5.847	5.871	53.316	0.002	0.002	0.8016	0.8045	0.697	0.003	1.652	0.000
44.606	305.850	1.650	52.749	34.840	1.514	5.798	5.823	52.938	0.002	0.002	0.7956	0.7986	0.697	0.003	1.650	0.000
43.909	301.070	1.647	52.368	34.660	1.511	5.749	5.774	52.559	0.002	0.002	0.7895	0.7925	0.697	0.003	1.647	0.000
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
27.182	186.380	1.546	42.615	30.252	1.409	4.374	4.407	42.833	0.001	0.001	0.6199	0.6240	0.697	0.006	1.546	0.000
26.485	181.600	1.540	42.177	30.064	1.403	4.306	4.340	42.396	0.001	0.001	0.6115	0.6157	0.697	0.006	1.540	0.000
25.788	176.820	1.534	41.735	29.876	1.397	4.237	4.271	41.956	0.001	0.001	0.6029	0.6072	0.697	0.006	1.534	0.000
25.091	172.040	1.527	41.290	29.688	1.391	4.167	4.202	41.512	0.001	0.001	0.5942	0.5985	0.697	0.006	1.527	0.000
24.394	167.260	1.521	40.841	29.499	1.385	4.095	4.131	41.065	0.001	0.001	0.5853	0.5897	0.697	0.007	1.521	0.000
23.697	162.480	1.514	40.389	29.309	1.378	4.023	4.059	40.615	0.001	0.001	0.5762	0.5808	0.697	0.007	1.514	0.000
23.000	157.700	1.507	39.933	29.119	1.371	3.949	3.986	40.161	0.001	0.001	0.5670	0.5716	0.697	0.007	1.507	0.000
22.303	152.920	1.499	39.473	28.929	1.365	3.874	3.912	39.703	0.001	0.001	0.5576	0.5623	0.697	0.007	1.499	0.000
21.606	148.140	1.492	39.009	28.737	1.357	3.798	3.836	39.241	0.001	0.001	0.5479	0.5527	0.697	0.008	1.492	0.000
20.909	143.370	1.484	38.541	28.546	1.350	3.720	3.759	38.775	0.001	0.001	0.5381	0.5430	0.697	0.008	1.484	0.000
20.212	138.590	1.476	38.068	28.354	1.343	3.641	3.680	38.304	0.001	0.001	0.5281	0.5331	0.697	0.008	1.476	0.000
19.515	133.810	1.468	37.591	28.161	1.335	3.560	3.600	37.830	0.001	0.001	0.5178	0.5229	0.697	0.008	1.468	0.000
18.818	129.030	1.459	37.109	27.967	1.327	3.477	3.518	37.350	0.001	0.001	0.5073	0.5125	0.697	0.009	1.459	0.000
18.121	124.250	1.450	36.623	27.773	1.319	3.393	3.435	36.866	0.001	0.001	0.4965	0.5019	0.697	0.009	1.450	0.000
17.424	119.470	1.441	36.131	27.578	1.310	3.307	3.350	36.377	0.001	0.001	0.4855	0.4910	0.697	0.009	1.441	0.000
16.727	114.690	1.431	35.634	27.382	1.301	3.219	3.263	35.882	0.001	0.001	0.4741	0.4798	0.697	0.010	1.431	0.000
16.030	109.910	1.421	35.131	27.186	1.292	3.129	3.174	35.382	0.001	0.001	0.4625	0.4683	0.697	0.010	1.421	0.000
15.333	105.130	1.411	34.621	26.988	1.283	3.037	3.083	34.876	0.001	0.001	0.4505	0.4565	0.697	0.010	1.411	0.000
14.636	100.360	1.400	34.106	26.790	1.273	2.943	2.990	34.364	0.001	0.001	0.4382	0.4444	0.697	0.011	1.400	0.000
13.939	95.577	1.389	33.584	26.591	1.263	2.846	2.894	33.845	0.001	0.001	0.4256	0.4319	0.697	0.011	1.389	0.000
13.242	90.798	1.377	33.055	26.390	1.253	2.747	2.796	33.320	0.000	0.001	0.4125	0.4190	0.697	0.012	1.377	0.000
12.545	86.019	1.365	32.519	26.189	1.242	2.645	2.696	32.787	0.000	0.000	0.3990	0.4057	0.697	0.012	1.365	0.000
11.848	81.241	1.353	31.974	25.987	1.230	2.541	2.593	32.246	0.000	0.000	0.3850	0.3920	0.697	0.013	1.353	0.000
11.152	76.462	1.340	31.421	25.783	1.219	2.433	2.487	31.698	0.000	0.000	0.3705	0.3778	0.697	0.013	1.340	0.000
10.455	71.683	1.326	30.859	25.578	1.207	2.323	2.378	31.140	0.000	0.000	0.3555	0.3630	0.697	0.014	1.326	0.000
9.758	66.904	1.311	30.288	25.372	1.194	2.209	2.266	30.574	0.000	0.000	0.3399	0.3477	0.697	0.014	1.311	0.000
9.061	62.125	1.296	29.706	25.164	1.181	2.091	2.150	29.997	0.000	0.000	0.3237	0.3318	0.697	0.015	1.296	0.000
8.364	57.346	1.281	29.113	24.955	1.167	1.970	2.031	29.409	0.000	0.000	0.3067	0.3152	0.697	0.016	1.281	0.000
7.667	52.567	1.264	28.507	24.744	1.152	1.844	1.907	28.810	0.000	0.000	0.2890	0.2979	0.697	0.016	1.264	0.000
6.970	47.789	1.247	27.889	24.531	1.137	1.714	1.779	28.198	0.000	0.000	0.2704	0.2797	0.697	0.017	1.247	0.000
6.273	43.010	1.229	27.256	24.316	1.121	1.578	1.646	27.572	0.000	0.000	0.2509	0.2607	0.697	0.018	1.229	0.000
5.576	38.231	1.210	26.607	24.098	1.104	1.437	1.507	26.931	0.000	0.000	0.2303	0.2406	0.697	0.019	1.210	0.000
4.879	33.452	1.189	25.941	23.879	1.086	1.290	1.363	26.274	0.000	0.000	0.2084	0.2193	0.697	0.020	1.189	0.000
4.182	28.673	1.168	25.255	23.656	1.068	1.135	1.213	25.598	0.000	0.000	0.1852	0.1968	0.697	0.022	1.168	0.000
3.485	23.894	1.145	24.547	23.431	1.048	0.973	1.054	24.901	0.000	0.000	0.1603	0.1728	0.697	0.023	1.145	0.000
2.788	19.115	1.120	23.814	23.202	1.026	0.803	0.888	24.180	0.000	0.000	0.1337	0.1470	0.697	0.024	1.120	0.000
2.091	14.337	1.094	23.053	22.969	1.004	0.622	0.712	23.433	0.000	0.000	0.1048	0.1192	0.697	0.026	1.094	0.000
1.394	9.558	1.066	22.259	22.732	0.979	0.429	0.526	22.656	0.000	0.000	0.0733	0.0890	0.697	0.028	1.066	0.000
0.697	4.779	1.035	21.425	22.490	0.953	0.223	0.326	21.842	0.000	0.000	0.0386	0.0560	0.697	0.031	1.035	0.000
0.000	0.000	1.002	20.545	22.241	0.924	0.000	0.112	20.985	0.000	0.000	0.0000	0.0193	0.697	0.033	1.002	0.000

Table d.1: Sub Critical WSP Table

d.2.2) Super Critical Portion

To determine the super critical portion of the water surface profile, the same steps as in section d.2.1 are taken with the following exceptions:

- the guessed x is set to start soon *after* x_c, increasing *downstream*
- the Froude numbers are constrained to be above 1 $(Fr > 1)$ for the super - critical flow

A portion of this process is shown in table

x (ft)	Q $\left(\frac{ft^3}{s}\right)$	y (ft)	A_w (ft^2)	P_w (ft)	R_w (ft)	V $\left(\frac{ft}{s}\right)$	$\overline{V}$ $\left(\frac{ft}{s}\right)$	$\overline{A_w}$ (ft^2)	s_f -	$\overline{s_f}$ -	Fr -	$\overline{Fr}$ -	Δx (ft)	Δy (ft)	y_{nxt} (ft)	$y - y_{nxt}$ (ft)
72.600	497.100	1.714	66.743	41.954	1.591	7.448	0.000	0.000	0.003	0.000	1.0031	0.0000	0.000	0.000	0.000	0.000
72.601	497.800	1.714	66.868	41.983	1.593	7.445	7.446	66.806	0.003	0.003	1.0020	1.0025	-0.001	0.000	1.714	0.000
72.609	497.850	1.714	66.875	41.986	1.593	7.445	7.445	66.871	0.003	0.003	1.0020	1.0020	-0.008	0.000	1.714	0.000
72.617	497.910	1.714	66.881	41.988	1.593	7.445	7.445	66.878	0.003	0.003	1.0020	1.0020	-0.008	0.000	1.714	0.000
72.625	497.960	1.715	66.887	41.990	1.593	7.445	7.445	66.884	0.003	0.003	1.0020	1.0020	-0.008	0.000	1.715	0.000
72.633	498.010	1.715	66.893	41.992	1.593	7.445	7.445	66.890	0.003	0.003	1.0020	1.0020	-0.008	0.000	1.715	0.000
72.641	498.070	1.715	66.898	41.994	1.593	7.445	7.445	66.896	0.003	0.003	1.0020	1.0020	-0.008	0.000	1.715	0.000
72.648	498.120	1.715	66.904	41.996	1.593	7.445	7.445	66.901	0.003	0.003	1.0020	1.0020	-0.008	0.000	1.715	0.000
72.656	498.180	1.715	66.909	41.998	1.593	7.446	7.446	66.906	0.003	0.003	1.0020	1.0020	-0.008	0.000	1.715	0.000
72.664	498.230	1.715	66.914	42.000	1.593	7.446	7.446	66.911	0.003	0.003	1.0021	1.0020	-0.008	0.000	1.715	0.000
72.672	498.280	1.715	66.919	42.002	1.593	7.446	7.446	66.916	0.003	0.003	1.0021	1.0021	-0.008	0.000	1.715	0.000
72.680	498.340	1.715	66.923	42.004	1.593	7.446	7.446	66.921	0.003	0.003	1.0021	1.0021	-0.008	0.000	1.715	0.000
72.688	498.390	1.715	66.928	42.006	1.593	7.447	7.447	66.926	0.003	0.003	1.0021	1.0021	-0.008	0.000	1.715	0.000
72.696	498.450	1.715	66.933	42.008	1.593	7.447	7.447	66.930	0.003	0.003	1.0022	1.0022	-0.008	0.000	1.715	0.000
72.704	498.500	1.715	66.937	42.010	1.593	7.447	7.447	66.935	0.003	0.003	1.0022	1.0022	-0.008	0.000	1.715	0.000
72.712	498.560	1.715	66.941	42.013	1.593	7.448	7.448	66.939	0.003	0.003	1.0022	1.0022	-0.008	0.000	1.715	0.000
72.720	498.610	1.715	66.946	42.015	1.593	7.448	7.448	66.943	0.003	0.003	1.0023	1.0023	-0.008	0.000	1.715	0.000
72.727	498.660	1.715	66.950	42.017	1.593	7.448	7.448	66.948	0.003	0.003	1.0023	1.0023	-0.008	0.000	1.715	0.000
72.735	498.720	1.715	66.954	42.019	1.593	7.449	7.448	66.952	0.003	0.003	1.0024	1.0023	-0.008	0.000	1.715	0.000
72.743	498.770	1.715	66.958	42.021	1.594	7.449	7.449	66.956	0.003	0.003	1.0024	1.0024	-0.008	0.000	1.715	0.000
72.751	498.830	1.715	66.962	42.023	1.594	7.449	7.449	66.960	0.003	0.003	1.0025	1.0024	-0.008	0.000	1.715	0.000
72.759	498.880	1.715	66.966	42.025	1.594	7.450	7.450	66.964	0.003	0.003	1.0025	1.0025	-0.008	0.000	1.715	0.000
72.767	498.940	1.715	66.970	42.027	1.594	7.450	7.450	66.968	0.003	0.003	1.0025	1.0025	-0.008	0.000	1.715	0.000
72.775	498.990	1.715	66.974	42.029	1.594	7.451	7.450	66.972	0.003	0.003	1.0026	1.0026	-0.008	0.000	1.715	0.000
72.783	499.040	1.715	66.978	42.031	1.594	7.451	7.451	66.976	0.003	0.003	1.0026	1.0026	-0.008	0.000	1.715	0.000
72.791	499.100	1.715	66.982	42.033	1.594	7.451	7.451	66.980	0.003	0.003	1.0027	1.0027	-0.008	0.000	1.715	0.000
72.799	499.150	1.715	66.986	42.035	1.594	7.452	7.451	66.984	0.003	0.003	1.0027	1.0027	-0.008	0.000	1.715	0.000
72.806	499.210	1.715	66.990	42.037	1.594	7.452	7.452	66.988	0.003	0.003	1.0028	1.0027	-0.008	0.000	1.715	0.000
72.814	499.260	1.715	66.994	42.039	1.594	7.452	7.452	66.992	0.003	0.003	1.0028	1.0028	-0.008	0.000	1.715	0.000
72.822	499.310	1.715	66.997	42.041	1.594	7.453	7.453	66.996	0.003	0.003	1.0029	1.0028	-0.008	0.000	1.715	0.000
72.830	499.370	1.715	67.001	42.043	1.594	7.453	7.453	66.999	0.003	0.003	1.0029	1.0029	-0.008	0.000	1.715	0.000
72.838	499.420	1.715	67.005	42.045	1.594	7.454	7.453	67.003	0.003	0.003	1.0030	1.0029	-0.008	0.000	1.715	0.000
72.846	499.480	1.715	67.009	42.047	1.594	7.454	7.454	67.007	0.003	0.003	1.0030	1.0030	-0.008	0.000	1.715	0.000
72.854	499.530	1.715	67.013	42.049	1.594	7.454	7.454	67.011	0.003	0.003	1.0031	1.0030	-0.008	0.000	1.715	0.000
72.862	499.590	1.715	67.017	42.051	1.594	7.455	7.455	67.015	0.003	0.003	1.0031	1.0031	-0.008	0.000	1.715	0.000
72.870	499.640	1.715	67.020	42.053	1.594	7.455	7.455	67.018	0.003	0.003	1.0032	1.0031	-0.008	0.000	1.715	0.000
72.878	499.690	1.715	67.024	42.055	1.594	7.455	7.455	67.022	0.003	0.003	1.0032	1.0032	-0.008	0.000	1.715	0.000
72.885	499.750	1.715	67.028	42.057	1.594	7.456	7.456	67.026	0.003	0.003	1.0033	1.0032	-0.008	0.000	1.715	0.000
72.893	499.800	1.715	67.032	42.059	1.594	7.456	7.456	67.030	0.003	0.003	1.0033	1.0033	-0.008	0.000	1.715	0.000
72.901	499.860	1.715	67.036	42.061	1.594	7.457	7.456	67.034	0.003	0.003	1.0034	1.0033	-0.008	0.000	1.715	0.000
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
119.690	820.680	1.722	87.473	53.774	1.627	9.382	9.382	87.471	0.004	0.004	1.2598	1.2598	-0.008	0.000	1.722	0.000
119.700	820.740	1.722	87.476	53.776	1.627	9.382	9.382	87.475	0.004	0.004	1.2598	1.2598	-0.008	0.000	1.722	0.000
119.710	820.790	1.722	87.479	53.778	1.627	9.383	9.383	87.478	0.004	0.004	1.2599	1.2599	-0.008	0.000	1.722	0.000
119.720	820.840	1.722	87.483	53.780	1.627	9.383	9.383	87.481	0.004	0.004	1.2599	1.2599	-0.008	0.000	1.722	0.000
119.720	820.900	1.722	87.486	53.782	1.627	9.383	9.383	87.484	0.004	0.004	1.2599	1.2599	-0.008	0.000	1.722	0.000
119.730	820.950	1.722	87.489	53.784	1.627	9.384	9.383	87.487	0.004	0.004	1.2600	1.2600	-0.008	0.000	1.722	0.000
119.740	821.010	1.722	87.492	53.786	1.627	9.384	9.384	87.491	0.004	0.004	1.2600	1.2600	-0.008	0.000	1.722	0.000
119.750	821.060	1.722	87.496	53.788	1.627	9.384	9.384	87.494	0.004	0.004	1.2601	1.2600	-0.008	0.000	1.722	0.000
119.760	821.110	1.722	87.499	53.790	1.627	9.384	9.384	87.497	0.004	0.004	1.2601	1.2601	-0.008	0.000	1.722	0.000
119.760	821.170	1.722	87.502	53.792	1.627	9.385	9.384	87.500	0.004	0.004	1.2601	1.2601	-0.008	0.000	1.722	0.000
119.770	821.220	1.722	87.505	53.794	1.627	9.385	9.385	87.504	0.004	0.004	1.2602	1.2602	-0.008	0.000	1.722	0.000
119.780	821.280	1.722	87.508	53.796	1.627	9.385	9.385	87.507	0.004	0.004	1.2602	1.2602	-0.008	0.000	1.722	0.000
119.790	821.330	1.722	87.512	53.798	1.627	9.385	9.385	87.510	0.004	0.004	1.2603	1.2602	-0.008	0.000	1.722	0.000
119.790	821.390	1.722	87.515	53.800	1.627	9.386	9.386	87.513	0.004	0.004	1.2603	1.2603	-0.008	0.000	1.722	0.000
119.800	821.440	1.722	87.518	53.802	1.627	9.386	9.386	87.516	0.004	0.004	1.2603	1.2603	-0.008	0.000	1.722	0.000
119.810	821.490	1.722	87.521	53.804	1.627	9.386	9.386	87.520	0.004	0.004	1.2604	1.2603	-0.008	0.000	1.722	0.000
119.820	821.550	1.722	87.525	53.806	1.627	9.387	9.386	87.523	0.004	0.004	1.2604	1.2604	-0.008	0.000	1.722	0.000
119.830	821.600	1.722	87.528	53.808	1.627	9.387	9.387	87.526	0.004	0.004	1.2604	1.2604	-0.008	0.000	1.722	0.000
119.830	821.660	1.722	87.531	53.810	1.627	9.387	9.387	87.529	0.004	0.004	1.2605	1.2605	-0.008	0.000	1.722	0.000
119.840	821.710	1.722	87.534	53.812	1.627	9.387	9.387	87.533	0.004	0.004	1.2605	1.2605	-0.008	0.000	1.722	0.000
119.850	821.760	1.722	87.537	53.814	1.627	9.388	9.387	87.536	0.004	0.004	1.2606	1.2605	-0.008	0.000	1.722	0.000
119.860	821.820	1.722	87.541	53.816	1.627	9.388	9.388	87.539	0.004	0.004	1.2606	1.2606	-0.008	0.000	1.722	0.000
119.870	821.870	1.722	87.544	53.818	1.627	9.388	9.388	87.542	0.004	0.004	1.2606	1.2606	-0.008	0.000	1.722	0.000
119.870	821.930	1.722	87.547	53.820	1.627	9.388	9.388	87.545	0.004	0.004	1.2607	1.2607	-0.008	0.000	1.722	0.000
119.880	821.980	1.722	87.550	53.822	1.627	9.389	9.389	87.549	0.004	0.004	1.2607	1.2607	-0.008	0.000	1.722	0.000
119.890	822.040	1.722	87.554	53.824	1.627	9.389	9.389	87.552	0.004	0.004	1.2607	1.2607	-0.008	0.000	1.722	0.000
119.900	822.090	1.722	87.557	53.826	1.627	9.389	9.389	87.555	0.004	0.004	1.2608	1.2608	-0.008	0.000	1.722	0.000
119.910	822.140	1.722	87.560	53.828	1.627	9.390	9.389	87.558	0.004	0.004	1.2608	1.2608	-0.008	0.000	1.722	0.000
119.910	822.200	1.722	87.563	53.830	1.627	9.390	9.390	87.562	0.004	0.004	1.2609	1.2608	-0.008	0.000	1.722	0.000
119.920	822.250	1.722	87.566	53.832	1.627	9.390	9.390	87.565	0.004	0.004	1.2609	1.2609	-0.008	0.000	1.722	0.000
119.930	822.310	1.722	87.570	53.833	1.627	9.390	9.390	87.568	0.004	0.004	1.2609	1.2609	-0.008	0.000	1.722	0.000
119.940	822.360	1.722	87.573	53.835	1.627	9.391	9.391	87.571	0.004	0.004	1.2610	1.2610	-0.008	0.000	1.722	0.000
119.940	822.410	1.722	87.576	53.837	1.627	9.391	9.391	87.574	0.004	0.004	1.2610	1.2610	-0.008	0.000	1.722	0.000
119.950	822.470	1.722	87.579	53.839	1.627	9.391	9.391	87.578	0.004	0.004	1.2610	1.2610	-0.008	0.000	1.722	0.000
119.960	822.520	1.722	87.583	53.841	1.627	9.391	9.391	87.581	0.004	0.004	1.2611	1.2611	-0.008	0.000	1.722	0.000
119.970	822.580	1.722	87.586	53.843	1.627	9.392	9.392	87.584	0.004	0.004	1.2611	1.2611	-0.008	0.000	1.722	0.000
119.980	822.630	1.722	87.589	53.845	1.627	9.392	9.392	87.587	0.004	0.004	1.2612	1.2611	-0.008	0.000	1.722	0.000
119.980	822.690	1.722	87.592	53.847	1.627	9.392	9.392	87.591	0.004	0.004	1.2612	1.2612	-0.008	0.000	1.722	0.000
119.990	822.740	1.722	87.595	53.849	1.627	9.393	9.392	87.594	0.004	0.004	1.2612	1.2612	-0.008	0.000	1.722	0.000
120.000	822.790	1.722	87.599	53.851	1.627	9.393	9.393	87.597	0.004	0.004	1.2613	1.2613	-0.008	0.000	1.722	0.000

Table d.2: Super Critical WSP Table

d.2.3) Water Surface Profile for Full Channel Length

The y-values in the 3rd columns are plotted against the x-values in the 1st columns of tables d.1 and d.2 in figure d.1. As a check, the Froude values of column 12 for both sets are also plotted. As expected, all sub-critical values are below 1, while all super critical are above.

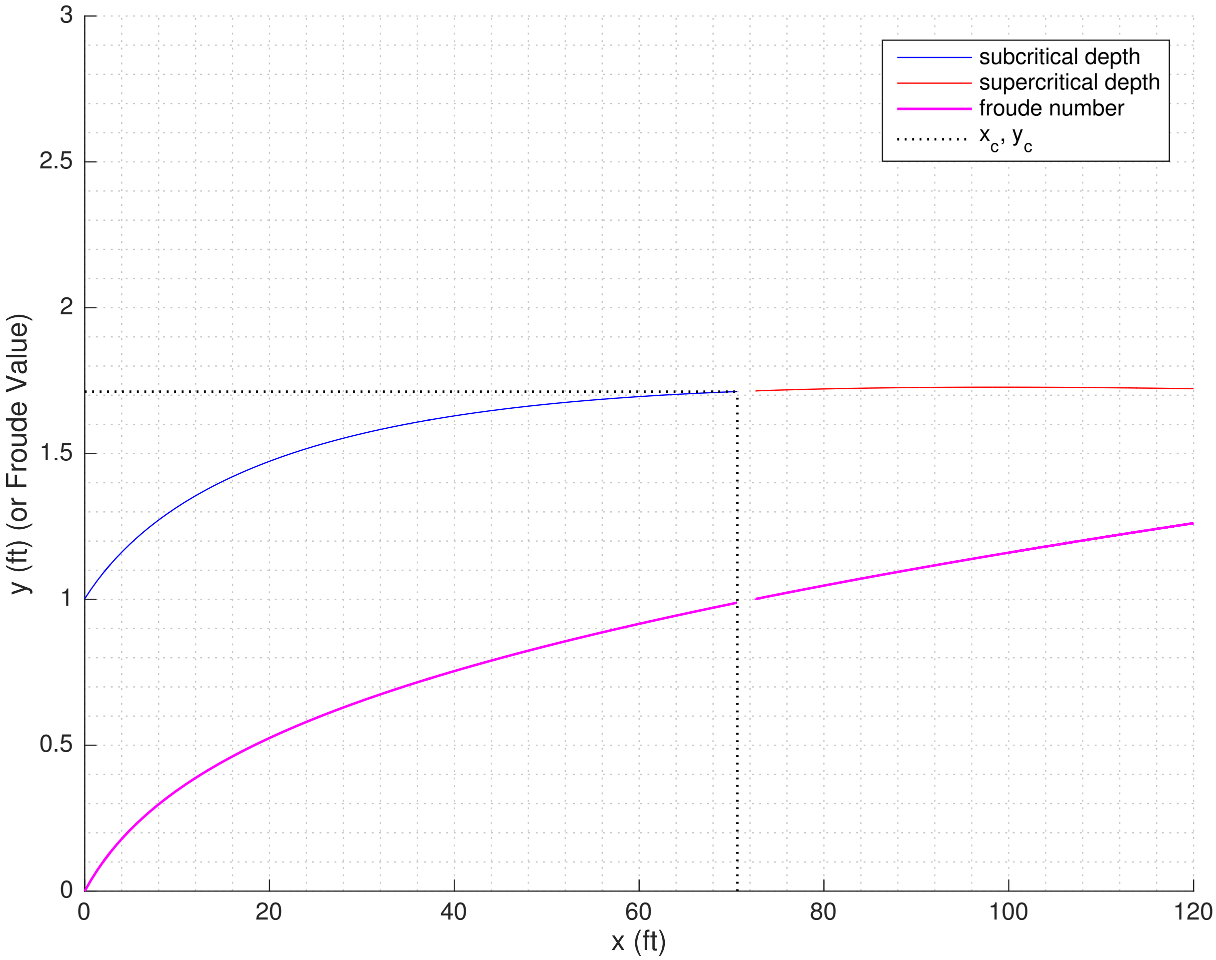

Figure d.1: Water surface profile along outlet channel

The gap for both y and Fr can be seen near the critical point. This reflects the region were the constraints for either the sub or super critical portions could not be met.

A Appendix

A.1 Given Weir Drawings

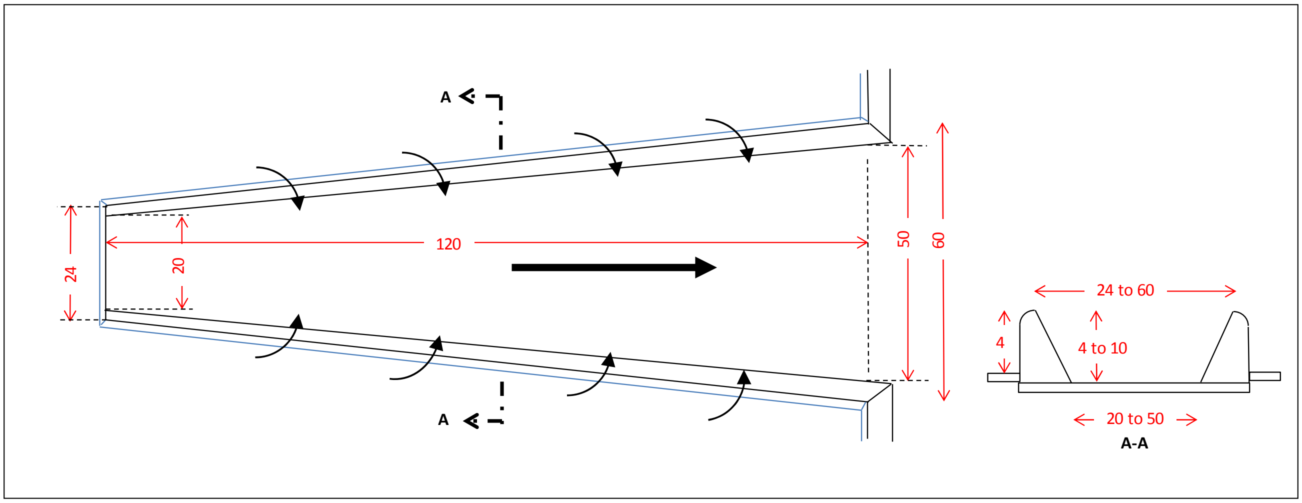

Figure A.1: 1 cycle of labyrinth weir (plan view)

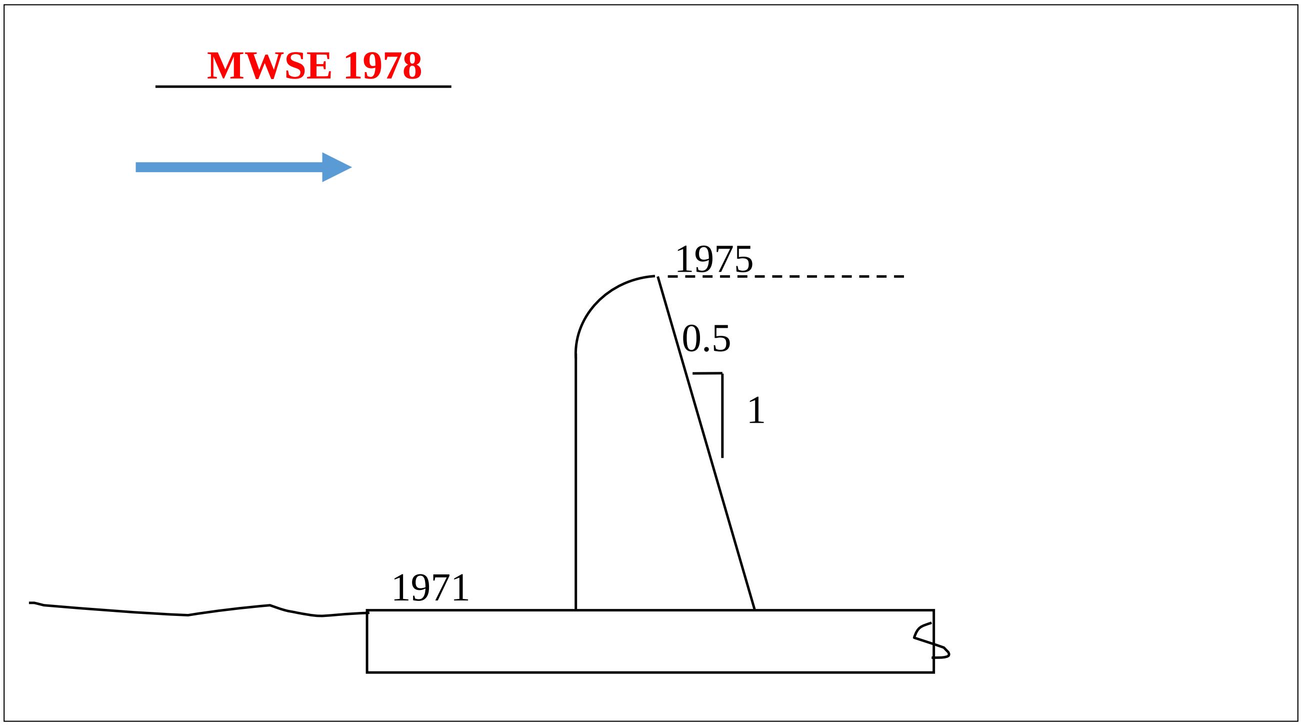

Figure A.2: Weir cross sectional view

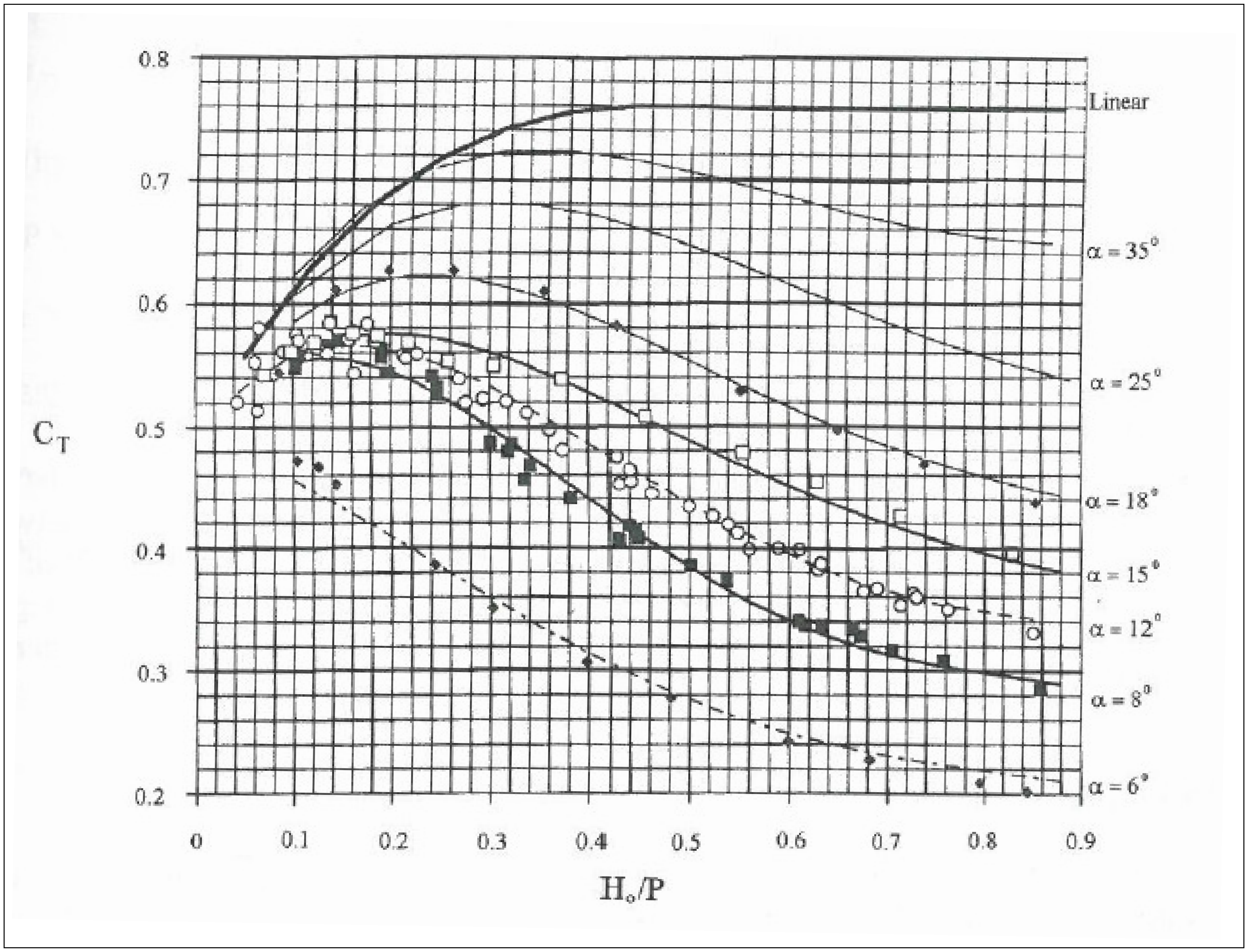

Figure A.3: Discharge coefficient chart for quarter-round crest and triangular weirs, Tullis, 1995

A.2 Reading C_T from Figure A.3

Design of Labyrinth Spillways by ASCE member J. Paul Tullis, 1995 provides an expression for discharge coefficient C_T as a function of the ratio of the total crest head and the dam height $\left(\frac{H_e}{P}\right)$as follows:

$$C_T = C_0 + C_1 \frac{H_e}{P} + C_2 \left(\frac{H_e}{P}\right)^2 + C_3 \left(\frac{H_e}{P}\right)^3 + +C_4 \left(\frac{H_e}{P}\right)^4 \tag{A.1}$$

The article also provides a set of coefficients for each ordered term for set of different weir geometry α's, as shown in table A.1. These coefficients yield the given chart shown in appendix figure A.3. It should be noted that in this figure, total head is symbolized as H_0.

α	$\mathbf{C_0}$	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$
6	0.49	-0.24	-1.20	2.17	-1.03
8	0.49	1.08	-5.27	6.79	-2.83
12	0.49	1.06	-4.43	5.18	-1.97
15	0.49	1.00	-3.57	3.82	-1.38
18	0.49	1.32	-4.13	4.24	-1.50
25	0.49	1.51	-3.83	3.40	-1.05
35	0.49	1.69	-4.05	3.62	-1.10
90	0.49	1.46	-2.56	1.44	0.00

Table A.1: Coefficients for $C_T = f\left(\frac{H_e}{P}\right)$

When a value of C_T is needed for an α whose coefficients are not available, they are interpolated as follows:

- the upper and lower known α's are determined as α_U and α_L
- for each n-powered term of eq. A.1, a coefficient is interpolated from each of the known upper and lower coefficients for the interpolated α_int, as follows:

$$C_{\mathrm{int}_n} = \frac{\alpha_{int} - \alpha_\mathrm{L}}{\alpha_\mathrm{U} - \alpha_\mathrm{L}} \left(C_{U_n} - C_{L_n}\right) + C_{L_n} \tag{A.2}$$

- with that, an interpolated C_{T_int} is calculated as follows:

$$C_{T_\mathrm{int}} = \sum_{n=0}^{4} C_{\mathrm{int}_n} \left(\frac{H_e}{P}\right)^n \tag{A.3}$$

Having calculated α in section 0.3, the coordinating curve of $C_T = f\left(\frac{H_e}{P}\right)$ is interpolated between an upper and lower known α as shown in figure A.4.

A.3 Interpolated Discharge Coefficient Curve

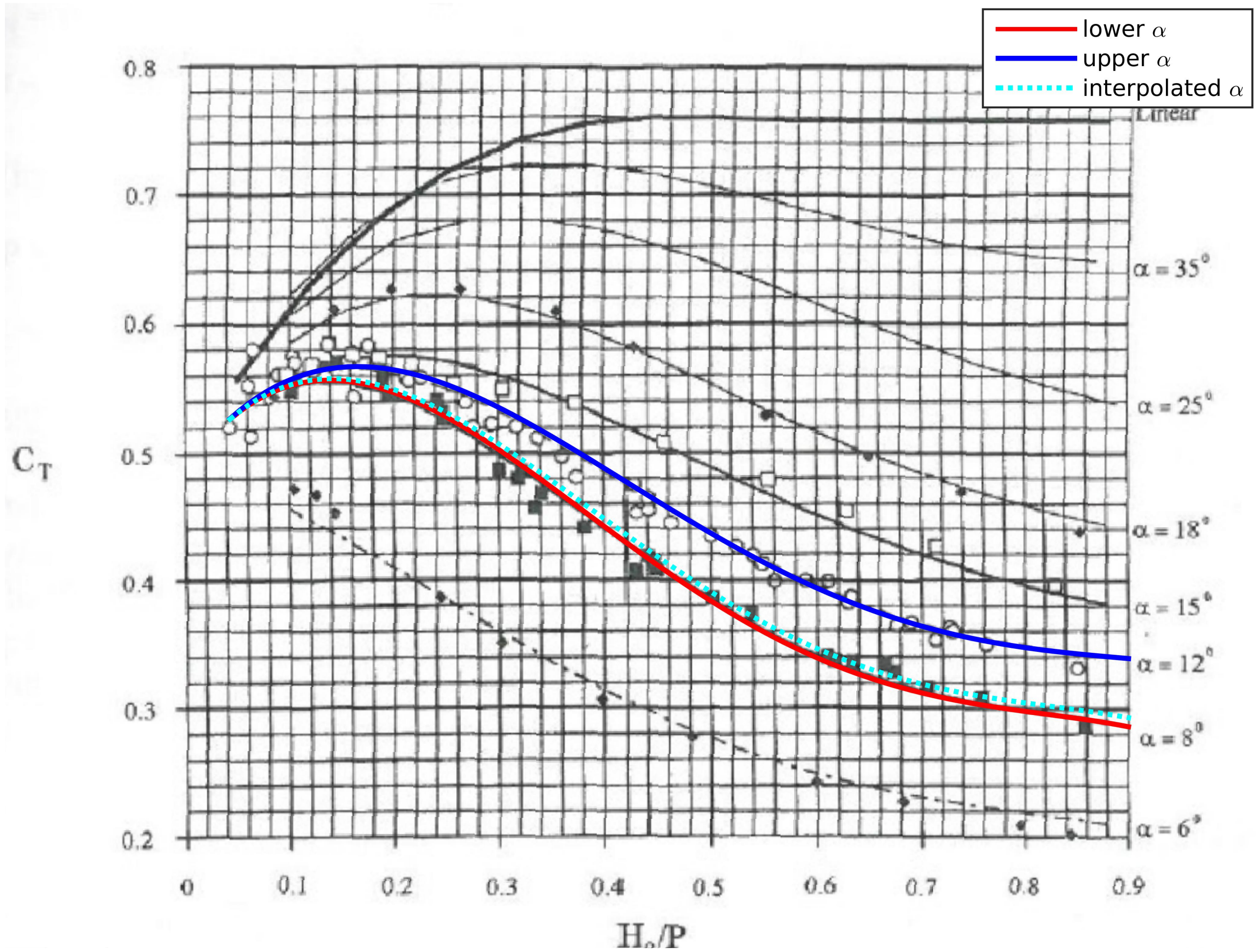

Figure A.4: Interpolation for $\alpha_{\text{int}} = 8.53°$

A.4 Matlab® Scripts

A.4.1 Master Script

```
clc
close all
clear all

% ------ given/known constants -------------------------------------------

g = 32.2;                                    %acceleration of gravity [ft/s^2]
n = 0.014;                                                         %Manning's n
kn = 1.49;

% ------ input given geometric spillway properties -----------------------

L_chan_t = 120;                                      %channel top length [ft]
w_b = [20 50];                        %bottom width [upstream downstream] [ft]
w_t = [24 60];                           %top width [upstream downstream] [ft]
z = 0.5;                                                  %channel side slope
s0 = 5/100;                                               %channel slope [%]
N_cyc = 2;                                                   %cycle quantity

% ------ input given head conditions -------------------------------------

H_max_elev = 1978;                           %max upstream pool elevation [ft]
P_elev = 1975;                                          %dam elevation [ft]
H_b_elev = 1971;                               %upstream apron elevation [ft]

% ------ preliminary geometric calculations ------------------------------
w = sum(w_t);                                    %total cycle width [ft]
a = w_t(1)/2;
B = sqrt((0.5*(w_t(2) - w_t(1)))^2 + L_chan_t^2);
L = N_cyc*(2*B + 4*a);
L0 = 2*w;
alpha = asin((w - 4*a)/(L/N_cyc - 4*a))*180/pi;

P = P_elev-H_b_elev;                                        %dam height [ft]
H_max = H_max_elev - P_elev;                         %max recorded head [ft]
H_1ft = 1;

% ------ max discharge - developed weir ----------------------------------

[Q_max, He_max, CT_max] = func_Q(L,P,H_max,g,alpha,w);

% ------ max discharge - linear weir -------------------------------------

[Q_max_lin, He_max_lin, CT_max_lin] = func_Q(L0,P,H_max,g,90,w);

% ------ labyrinth - linear efficacy improvment --------------------------

IMP = 100*(Q_max - Q_max_lin)/Q_max_lin;

% ------ H0 = 1ft discharge calculations ---------------------------------

Q_1ft = func_Q_1ft(L,P,H_1ft,g,alpha);
ql_1ft = (Q_1ft/L_chan_t)/N_cyc;

% ----- solve for xc and yc ----------------------------------------------

N_test = 50;
xc_eqn_test_array = linspace(1,L_chan_t,N_test);
sf_cond_test_output  = zeros(1,N_test);

for kk = 1:N_test
sf_cond_test_output(kk) = func_xc_solver(xc_eqn_test_array(kk),...
                               ql_1ft,g,z,w_b(1),w_b(2),L_chan_t,n,kn,s0);
end


plot(xc_eqn_test_array , sf_cond_test_output); grid minor; hold on;
plot([0 L_chan_t],[0 0],'k','linewidth',2);

xc_test  = 70.63;
eqn_xc = func_xc_solver(xc_test,ql_1ft,g,z,w_b(1),w_b(2),L_chan_t,n,kn,s0);
disp(eqn_xc);

xc = xc_test;
yc = func_yc(ql_1ft,xc,g,z,w_b(1),w_b(2),L_chan_t);

% --- subcritical WSP, leading up to xc ----------------------------------
sub_N = 100;
WSP_sub_table = func_WSP_sub_table_generator(...
                          sub_N, ql_1ft, xc,yc, w_b, L_chan_t,s0, kn,n, z,g);

% --- supcritical WSP, downstream of xc ----------------------------------

sup_N = 6000;
WSP_sup_table = func_WSP_sup_table_generator(...
                         sup_N, ql_1ft, xc,yc, w_b, L_chan_t,s0, kn, n, z,g);

% --- plot full WSP ------------------------------------------------------

figure; grid minor; hold on;
h1 = plot(WSP_sup_table(:,1),WSP_sup_table(:,3),'r');
h2 = plot(WSP_sub_table(:,1),WSP_sub_table(:,3),'b');

h3 = plot(WSP_sup_table(:,1),WSP_sup_table(:,12),'m','linewidth',1);
h4 = plot(WSP_sub_table(:,1),WSP_sub_table(:,12),'m','linewidth',1);
h5 = plot([xc xc],[0 yc],':k','linewidth',1);
h6 = plot([0 xc],[yc yc],':k','linewidth',1);
legend([h2 h1 h4 h5],'subcritical depth','supercritical depth',...
                          'froude number','x_c, y_c','location','northeast');
xlabel('x (ft)'); ylabel('y (ft) (or Froude Value)')
axis([0 L_chan_t 0 3]);
print('WSP_01','-depsc2','-r300');
```

A.4.2 Function: $C_T = f(H_e, \alpha)$

```
function [CT_out] = func_CT(alpha, He_over_P, plot_fig)

N = 100;                                            %interpolation plot points

% --- given interpolation coefficients (J. Hydraul. Eng, 1995) ------------

int_mat = [0.49 -0.24 -1.20  2.17 -1.03   6;
           0.49  1.08 -5.27  6.79 -2.83   8;
           0.49  1.06 -4.43  5.18 -1.97  12;
           0.49  1.00 -3.57  3.82 -1.38  15;
           0.49  1.32 -4.13  4.24 -1.50  18;
           0.49  1.51 -3.83  3.40 -1.05  25;
           0.49  1.69 -4.05  3.62 -1.10  35;
           0.49  1.46 -2.56  1.44  0.00  90];

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

[rows, cols] = size(int_mat);

He_over_P_array = linspace(0.04,0.9,N);

CT_mat = zeros(rows,N);
for nn = 1:N
    for ii = 1:rows;
for jj = 1:cols-1
CT_mat(ii,nn)=CT_mat(ii,nn)+He_over_P_array(nn)^(jj-1)*int_mat(ii,jj);
end
    end
end

% --- re-interpolate values for known alpha ------------------------------
row_aa = 0;
bow_bb = 0;
row_xx = 1;
while row_aa == 0
   if  int_mat(row_xx,cols) > alpha
       row_aa = row_xx - 1;
       row_bb = row_xx;
   end
   row_xx = row_xx + 1;
end

int_out = int_mat(row_aa,:) + (int_mat(row_bb,:) - int_mat(row_aa,:))*...
(alpha - int_mat(row_aa,cols))/(int_mat(row_bb,cols)-int_mat(row_aa,cols));

CT_array_out = zeros(1,N);
for nn = 1:N
for jj = 1:cols-1
CT_array_out(nn)=CT_array_out(nn)+...
                                  He_over_P_array(nn)^(jj-1)*int_out(jj);
end
end

% --- output CT for known He_over_P --------------------------------------

CT_out = 0;
for jj = 1:cols-1
CT_out=CT_out+He_over_P^(jj-1)*int_out(jj);
end

% --- plot results if requested ------------------------------------------
if plot_fig == 1
ori = [242 977];
x_end = 1343;
y_top = 120;


scale_xx = abs(x_end - ori(1))/0.9;
scale_yy = abs(y_top - ori(2))/0.6;

CT_mat_plot = zeros(rows,N);
CT_array_out_plot = zeros(1,N);
He_over_P_array_plot = zeros(1,N);

for nn  = 1:N
          He_over_P_array_plot(nn) = He_over_P_array(nn)*scale_xx + ori(1);
         CT_array_out_plot(nn) = ori(2) - (CT_array_out(nn)-0.2)*scale_yy;
for ii = 1:rows
          CT_mat_plot(ii,nn) = ori(2) - (CT_mat(ii,nn)-0.2)*scale_yy;
end
end

I = imread('CT_chart.png');
B = imrotate(I,-0.8);

figure
imshow(B); hold on;
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
```

A.4.3 Function: $\mathbf{Q = f(H)}$

```
function [Q_out, He_out, CT_out] = func_Q(L,P,H,g,alpha)

syms QQ
V0 = QQ/(L*(P+H));
ha = (V0^2)/(2*g);
He = H + ha;
He_over_P = He/P;
if alpha == 90
CT = 0.76;
else
CT = func_CT(alpha, He_over_P, 0);
end
eqn_QQ = CT*L*(2/3)*sqrt(2*g)*He^1.5;

% output
Q_out = double(vpasolve(eqn_QQ == QQ));
V0_out = Q_out/(L*(P+H));
ha_out = (V0_out^2)/(2*g);
He_out = H + ha_out;
if alpha == 90
    CT_out = CT;
else
CT_out = func_CT(alpha, He_out/P, 0);
end

if alpha == 90 && He_out/P < 0.45
   disp('CT determined improperly - He/P assumed to be greater than 0.45');
end
end
```

A.4.4 Function: $\mathbf{Q = f(H)}$ with $H_e = 1\ ft$

```
function [Q_out] = func_Q_1ft(L,P,He,g,alpha)

He_over_P = He/P;
if alpha == 90
CT = 0.76;
else
CT = func_CT(alpha, He_over_P, 0);
end
Q_out = CT*L*(2/3)*sqrt(2*g)*He^1.5;

if alpha == 90
disp('CT maybe dtrmnd improperly-He/P assumed to be greater than 0.45');
end

end
```

A.4.5 Function: $\mathbf{x_c}$ Solver, with $\mathbf{y_c}$ Solver

```
function [eqn_xc] = func_xc_solver(xc, ql,g,z,w_min,w_max,L_chan_t,n,kn,s0)

Q = ql*xc;
b = func_b(xc,w_min,w_max,L_chan_t);

yc = func_yc(ql,xc,g,z,w_min,w_max,L_chan_t);

Pw = b + 2*yc*sqrt(1 + z^2);
Aw = yc*(b + z*yc);
Rw = Aw/Pw;
V = Q/Aw;
sf = ((n*V)/(kn*Rw^(2/3)))^2;
eqn_xc = s0 - sf - 2*ql*V/(g*Aw);
end
```

A.4.6 Function: $\mathbf{b = f(x)}$

```
function [ b ] = func_b(x,w_min,w_max,L)
b = (x/L)*(w_max-w_min) + w_min;
end
```

A.4.7 Function: $\mathbf{y_c}$ Solver

```
function [yc_out] = func_yc(ql,x,g,z,w_min,w_max,L_chan_t)

Q = ql*x;
b = func_b(x,w_min,w_max,L_chan_t);

syms yc
left_term = ((b*yc + z*yc^2)^3)/(b + 2*z*yc);
right_term = (Q^2)/g;
eqn = left_term == right_term  ;
yc_out = sum(real(sqrt(real(double(vpasolve(eqn,yc))))))^2;

end
```

A.4.8 Function: $\mathbf{yy = f(xx)}$

```
function [yy_out] = func_yy_solver(xx, ql_max,yy_prev,VV_prev,...
                   AA_prev,sf_prev,Fr_prev,xx_prev,w_b,L_chan_t,z,n,kn,g,s0)

syms yy
QQ =  ql_max*xx;
bb = func_b(xx,w_b(1),w_b(2),L_chan_t);
AA = yy*(bb + z*yy);
PP = bb + 2*yy*sqrt(1 + z^2);
RR = AA/PP;
VV = QQ/AA;
sf = ((n*VV)/(kn*RR^(2/3)))^2;
Fr = VV/sqrt(g*yy);

VV_bar = 0.5*(VV + VV_prev);
AA_bar = 0.5*(AA + AA_prev);
sf_bar = 0.5*(sf + sf_prev);
Fr_bar = 0.5*(Fr + Fr_prev);

del_x =  xx_prev - xx;
del_y = del_x*(s0 - sf_bar - (2*ql_max*VV_bar)/(g*AA_bar))/(1 - Fr_bar^2);

y_2 = yy_prev - del_y;
eqn_yy = y_2 - yy == 0 ;

yy_out = sum(real(sqrt(real(double(vpasolve(eqn_yy,yy))))))^2;

end
```

A.4.9 Function: Subcritical WSP Plotter

```
function [ WSP_sub_table ] = func_WSP_sub_table_generator(...
                          sub_N, ql_1ft, xc,yc, w_b, L_chan_t,s0, kn, n, z,g)

x_select_sub_start = [70.6 70.2 69.7];
x_select_sub_end = floor(xc)-1 - linspace(0,floor(xc)-1,sub_N);

x_select_sub= zeros(1,length(x_select_sub_start)+length(x_select_sub_end));
x_select_sub(1:length(x_select_sub_start)) = x_select_sub_start;
x_select_sub(length(x_select_sub_start)+1:end) = x_select_sub_end;

WSP_sub_table = zeros(sub_N + length(x_select_sub_start),17);
[WSP_sub_rows, ~] = size(WSP_sub_table);

xx_prev = xc;

QQ_prev =  ql_1ft*xx_prev;
bb_prev = func_b(xx_prev,w_b(1),w_b(2),L_chan_t);
yy_prev = yc;

AA_prev = yy_prev*(bb_prev + z*yy_prev);
PP_prev = bb_prev + 2*yy_prev*sqrt(1 + z^2);
RR_prev = AA_prev/PP_prev;
VV_prev = QQ_prev/AA_prev;
sf_prev = ((n*VV_prev)/(kn*RR_prev^(2/3)))^2;
Fr_prev = VV_prev/sqrt(g*yy_prev);

xx = x_select_sub(1);
yy_out = func_yy_solver(xx, ql_1ft,yy_prev,VV_prev,AA_prev,...
                         sf_prev,Fr_prev,xx_prev,w_b,L_chan_t,z,n,kn,g,s0);

WSP_sub_table(1,1) = xx;
WSP_sub_table(1,2) = QQ_prev;
WSP_sub_table(1,3) = yy_out;
WSP_sub_table(1,4) = AA_prev;
WSP_sub_table(1,5) = PP_prev;
WSP_sub_table(1,6) = RR_prev;
WSP_sub_table(1,7) = VV_prev;
WSP_sub_table(1,10) = sf_prev;
WSP_sub_table(1,12) = Fr_prev;

for ii = 2:WSP_sub_rows
WSP_sub_table(ii,1) = x_select_sub(ii);                                 % x
WSP_sub_table(ii,2) = ql_1ft*WSP_sub_table(ii,1);                       % Q
WSP_sub_table(ii,3) = func_yy_solver(WSP_sub_table(ii,1),ql_1ft,...
                                     WSP_sub_table(ii-1,3),...
                                     WSP_sub_table(ii-1,7),...
                                     WSP_sub_table(ii-1,4),...
                                     WSP_sub_table(ii-1,10),...
                                     WSP_sub_table(ii-1,12),...
                                     WSP_sub_table(ii-1,1),...
                                     w_b,L_chan_t,z,n,kn,g,s0);         % y
bb = func_b(WSP_sub_table(ii,1),w_b(1),w_b(2),L_chan_t);
WSP_sub_table(ii,4) = WSP_sub_table(ii,3)*(bb + z*WSP_sub_table(ii,3)); % A
WSP_sub_table(ii,5) = bb + 2*WSP_sub_table(ii,3)*sqrt(1 + z^2);         % P
WSP_sub_table(ii,6) = WSP_sub_table(ii,4)/WSP_sub_table(ii,5);          % R
WSP_sub_table(ii,7) = WSP_sub_table(ii,2)/WSP_sub_table(ii,4);          % V
WSP_sub_table(ii,8) = 0.5*(WSP_sub_table(ii,7)...
                                          +WSP_sub_table(ii-1,7)); % V_bar
WSP_sub_table(ii,9) = 0.5*(WSP_sub_table(ii,4)...
                                         + WSP_sub_table(ii-1,4)); % A_bar
WSP_sub_table(ii,10) = ((n*WSP_sub_table(ii,7))/...
                                    (kn*WSP_sub_table(ii,6)^(2/3)))^2; % sf
WSP_sub_table(ii,11) = 0.5*(WSP_sub_table(ii,10) +...
                                          WSP_sub_table(ii-1,10)); % sf_bar
WSP_sub_table(ii,12) = WSP_sub_table(ii,7)/sqrt(g*WSP_sub_table(ii,3));% Fr
WSP_sub_table(ii,13) = 0.5*(WSP_sub_table(ii,12) +...
                                           WSP_sub_table(ii-1,12)); %Fr_bar
WSP_sub_table(ii,14) = WSP_sub_table(ii-1,1) - WSP_sub_table(ii,1); % del_x
WSP_sub_table(ii,15) = WSP_sub_table(ii,14)*(s0 - WSP_sub_table(ii,11)- ...
                          (2*ql_1ft*WSP_sub_table(ii,8))/...
               (g*WSP_sub_table(ii,9)))/(1 - WSP_sub_table(ii,13)^2); %del_y
WSP_sub_table(ii,16) = WSP_sub_table(ii-1,3)- WSP_sub_table(ii,15);   % y_2
WSP_sub_table(ii,17) = WSP_sub_table(ii,16) - WSP_sub_table(ii,3);  % check
end

end
```

A.4.10 Function: Supercritical WSP Plotter

```
function [ WSP_sup_table ] = func_WSP_sup_table_generator(...
                          sup_N, ql_1ft, xc,yc, w_b, L_chan_t,s0, kn, n, z,g)

xx_prev = xc + 1.8700;
x_select_sup_start = [xx_prev + 0.1];
x_select_sup_end = linspace(max(x_select_sup_start) +0.001,L_chan_t,sup_N);

x_select_sup= zeros(1,length(x_select_sup_start)+length(x_select_sup_end));
x_select_sup(1:length(x_select_sup_start)) = x_select_sup_start;
x_select_sup(length(x_select_sup_start)+1:end) = x_select_sup_end;

WSP_sup_table = zeros(sup_N + length(x_select_sup_start),17);
[WSP_sup_rows, ~] = size(WSP_sup_table);

QQ_prev =  ql_1ft*xx_prev;
bb_prev = func_b(xx_prev,w_b(1),w_b(2),L_chan_t);
yy_prev = yc;

AA_prev = yy_prev*(bb_prev + z*yy_prev);
PP_prev = bb_prev + 2*yy_prev*sqrt(1 + z^2);
RR_prev = AA_prev/PP_prev;
VV_prev = QQ_prev/AA_prev;
sf_prev = ((n*VV_prev)/(kn*RR_prev^(2/3)))^2;
Fr_prev = VV_prev/sqrt(g*yy_prev);

xx = x_select_sup(1);
yy_out = func_yy_solver(xx, ql_1ft,yy_prev,VV_prev,AA_prev,...
                         sf_prev,Fr_prev,xx_prev,w_b,L_chan_t,z,n,kn,g,s0);

WSP_sup_table(1,1) = xx;
WSP_sup_table(1,2) = QQ_prev;
WSP_sup_table(1,3) = yy_out;
WSP_sup_table(1,4) = AA_prev;
WSP_sup_table(1,5) = PP_prev;
WSP_sup_table(1,6) = RR_prev;
WSP_sup_table(1,7) = VV_prev;
WSP_sup_table(1,10) = sf_prev;
WSP_sup_table(1,12) = Fr_prev;

for ii = 2:WSP_sup_rows
WSP_sup_table(ii,1) = x_select_sup(ii);                                 % x
WSP_sup_table(ii,2) = ql_1ft*WSP_sup_table(ii,1);                       % Q
WSP_sup_table(ii,3) = func_yy_solver(WSP_sup_table(ii,1),ql_1ft,...
                                     WSP_sup_table(ii-1,3),...
                                     WSP_sup_table(ii-1,7),...
                                     WSP_sup_table(ii-1,4),...
                                     WSP_sup_table(ii-1,10),...
                                     WSP_sup_table(ii-1,12),...
                                     WSP_sup_table(ii-1,1),...
                                     w_b,L_chan_t,z,n,kn,g,s0);         % y
bb = func_b(WSP_sup_table(ii,1),w_b(1),w_b(2),L_chan_t);
WSP_sup_table(ii,4) = WSP_sup_table(ii,3)*(bb + z*WSP_sup_table(ii,3)); % A
WSP_sup_table(ii,5) = bb + 2*WSP_sup_table(ii,3)*sqrt(1 + z^2);         % P
WSP_sup_table(ii,6) = WSP_sup_table(ii,4)/WSP_sup_table(ii,5);          % R
WSP_sup_table(ii,7) = WSP_sup_table(ii,2)/WSP_sup_table(ii,4);          % V
WSP_sup_table(ii,8) = 0.5*(WSP_sup_table(ii,7)...
                                          +WSP_sup_table(ii-1,7)); % V_bar
WSP_sup_table(ii,9) = 0.5*(WSP_sup_table(ii,4)...
                                         + WSP_sup_table(ii-1,4)); % A_bar
WSP_sup_table(ii,10) = ((n*WSP_sup_table(ii,7))/...
                                    (kn*WSP_sup_table(ii,6)^(2/3)))^2; % sf
WSP_sup_table(ii,11) = 0.5*(WSP_sup_table(ii,10) +...
                                          WSP_sup_table(ii-1,10)); % sf_bar
WSP_sup_table(ii,12) = WSP_sup_table(ii,7)/sqrt(g*WSP_sup_table(ii,3));% Fr
WSP_sup_table(ii,13) = 0.5*(WSP_sup_table(ii,12) +...
                                           WSP_sup_table(ii-1,12)); %Fr_bar
WSP_sup_table(ii,14) = WSP_sup_table(ii-1,1) - WSP_sup_table(ii,1); % del_x
WSP_sup_table(ii,15) = WSP_sup_table(ii,14)*(s0 - WSP_sup_table(ii,11)- ...
                          (2*ql_1ft*WSP_sup_table(ii,8))/...
               (g*WSP_sup_table(ii,9)))/(1 - WSP_sup_table(ii,13)^2); %del_y
WSP_sup_table(ii,16) = WSP_sup_table(ii-1,3)- WSP_sup_table(ii,15);   % y_2
WSP_sup_table(ii,17) = WSP_sup_table(ii,16) - WSP_sup_table(ii,3);  % check
end

disp(WSP_sup_table(1:100,:));

end
```