Lab 1 Task 9 - Formula Clarification

Summarised by Ali Arab and Anne Lavergne

The view from the side of the pyramid involves drawing two lines from the base of the pyramid to the top, as shown in figure 1.

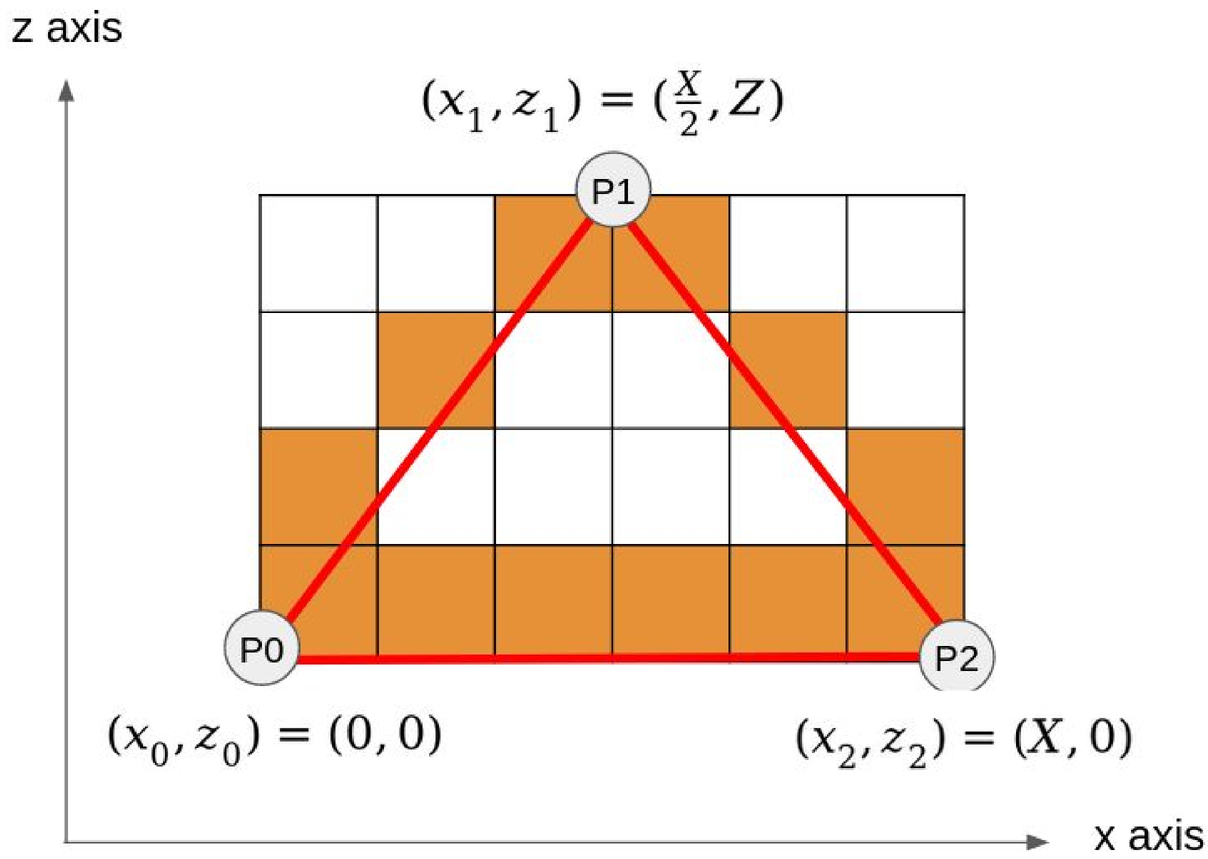

Figure 1

The left line starts from $P_0 = (x_0, z_0) = (0, 0)$ and ends at $P_1 = (x_1, z_1) = (\frac{X}{2}, Z)$. The equation for a line where we can feed in the z values and compute the x values is $x = mz + b$ where m is the slope and b is the intercept. If we know the end points P_0 and P_1 (which we do) we can compute the slope $m = \frac{d_x}{d_z}$ where $d_z = z_1 - z_0$ and $d_x = x_1 - x_0$. So m will be computed as following:

$$m = \frac{d_x}{d_z} = \frac{x_1 - x_0}{z_1 - z_0} = \frac{\frac{x}{2} - 0}{Z} = \frac{X}{2Z}$$

The intercept will be computed as $b = x_0 - mz_0$.

$$b = x_0 - mz_0 = 0 - (\frac{X}{2Z}) * 0 = 0$$

So the formula to find the value of x at z, i.e., point (x,z) on the left line (left slope of the pyramid) would be simplified as:
$x = mz + b = \frac{X}{2Z} z$ => $x = \frac{X}{2Z} z$

Using P_0 = (x_0, z_0) = $(0, 0)$ and P_1 = (x_1, z_1) = $(\frac{X}{2}, Z)$, where X and Z are user input (dimensions of our pyramid), the pseudocode given in our Lab 1 Task 9 becomes:

```
dx = x1 - x0 = X/2 - 0 = X/2
dz = z1 - z0 = Z - 0 = Z
for z from 0 to Z {          /* from z0 to z1 */
  x = x1 + dx*(z-z1)/dz = X/2 + X/2*(z-Z)/Z = (X/(2*Z))*z
  plot(x, z)
}
```

Similarly, for the right line, we need to find the parameters for the line $x = mz + b$ crossing P_1 and P_2.

$$m = \frac{d_x}{d_z} = \frac{x_2 - x_1}{z_2 - z_1} = \frac{X - \frac{X}{2}}{0 - Z} = \frac{-X}{2Z}$$

The intercept will be computed as $b = x_2 - mz_2$.

$$b = x_2 - mz_2 = X - (\frac{-X}{2Z}) * 0 = X$$

So the formula to find the value of x at z, i.e., point (x,z) on the right line (right slope of the pyramid) would be simplified as:
$x = mz + b = -\frac{X}{2Z} z + X$ => $x = X - \frac{X}{2Z} z$

Using P_1 = (x_1, z_1) = ($\frac{X}{2}$, Z) and P2 = (x_2, z_2) = (X , 0), where X and Z are user input (dimensions of our pyramid), the pseudocode given in our Lab 1 Task 9 becomes:

```
dx = x2 - x1 = X - X/2 = X/2
dz = z2 - z1 = 0 - Z = -Z
for z from Z to 0 {        /* for z from z1 to z2 */
  x = x1 + dx*(z-z1)/dz = X/2 + X/2*(z-Z)/-Z = X - (X/(2*Z))*z
  plot(x, z)
}
```

Both pseudocodes above would then plot all pixels (x,z) making up the slopes of the pyramid.

In Lab 1 Task 9, we draw the outline of our pyramid not by plotting (x,z), but by printing “#” characters in a text-based interface Terminal window, once the user has entered the dimensions X (base 1), Y (base 2) and Z (height) of the pyramid.

A text-based interface is like a grid of cells, where one character can be printed in each cell. To print the outline of a X by Y by Z pyramid, from the Y perspective, as we are asked to do in this task, we only need to consider the X by Z grid of cells, i.e., the grid of cells made of Z rows and X columns as illustrated in Figure 1 above.

The coordinates of the cells of the first row (base of pyramid) of this grid of cells, where z = 0, are (0,0), (1,0), (2,0), …, (X-1,0). The coordinates of the cells of the second row of this grid of cells, where z = 1, are (0,1), (1,1), (2,1), …, (X-1,1). The coordinates of the cells of the remaining rows of this grid of cells would follow a similar pattern until the last row (top of pyramid) of this grid of cells where the cells would have the coordinates (0,Z-1), (1,Z-1), (2,Z-1), …, (X-1,Z-1).

If the dimensions of the pyramid are X = 6, Y= 6 and Z = 4, then the coordinates of the cells, in our X by Z grid of cells (of 6 columns by 4 rows), would be:

(0,0), (1,0), (2,0), (3,0), (4,0), (5,0) <- bottom row of pyramid
(0,1), (1,1), (2,1), (3,1), (4,1), (5,1)
(0,2), (1,2), (2,2), (3,2), (4,2), (5,2)
(0,3), (1,3), (2,3), (3,3), (4,3), (5,3) <- toprow of pyramid

Since we want the base of the pyramid to be the bottom row or our grid of cells, we would then flip our grid of cells upside down and get the following coordinates:

(0,3), (1,3), (2,3), (3,3), (4,3), (5,3) <- top row of pyramid
(0,2), (1,2), (2,2), (3,2), (4,2), (5,2)
(0,1), (1,1), (2,1), (3,1), (4,1), (5,1)
(0,0), (1,0), (2,0), (3,0), (4,0), (5,0) <- bottom row of pyramid

Notice that these coordinates are the coordinates of the cells in Figure 1 above.

In order to print the outline of our pyramid upon our X by Z grid of cells, we shall construct a loop in our Task 9 program. Each iteration of this loop will print all the needed characters for each row of our pyramid. We shall use the iterating variable **j** for this loop. We can first set the starting value of **j** to be 0 (the initialization part), where 0 represent the first row of our pyramid, and then, at each iteration of our loop, we increase the value of **j** by 1 for each of the other rows, all the way to the value Z - 1 (i.e., 3, when Z is 4). Yes, our loop would print the pyramid upside down, but once we do this successfully, we can easily rewrite our *for* loop - its initialization and stopping condition parts - such that our pyramid is printed the correct way up.

Let's now have a look at what our program needs to do in each iteration of our loop in order to print all the necessary characters for each row of our pyramid.

The first thing we need to do in each iteration of our loop is calculate

- the x value of the coordinate (x,z) at which we print the # character representing the left slope of our pyramid, and
- the x value of the coordinate (x,z) at which we print the # character representing the right slope of our pyramid.

These two equations have already been created for us. The first one is:

```
left = floor( j * (x/(2.0*z)) );
```

which comes from the first pseudocode, above: `x = (X/(2*Z)) * z` where `x` has been replaced by `left`, `z` by **j**, X and Z by lower case `x` and `z` but they still represent the user input X, Y and X. The function *floor* is needed because the grid coordinates are integral values.

The second one is:

```
right = ceil( (x-1) + -j * (x/(2.0*z)) );
```

which comes from the second pseudocode above: `x = X-(X/(2*Z))*z` where `x` has been replaced by `right`, `z` by **j**, X and Z (in the *slope* part of the equation) by lower case `x` and `z` but they still represent the user input X, Y and X, and X by `(x-1)` because, in our grid, `(x-1)` represents the x value of the last cell. The function *ceil* is needed because the grid coordinates are integral values.

What comes after the calculation of `left` and `right` in each iteration of our loop is left to you to discover!

Good luck!