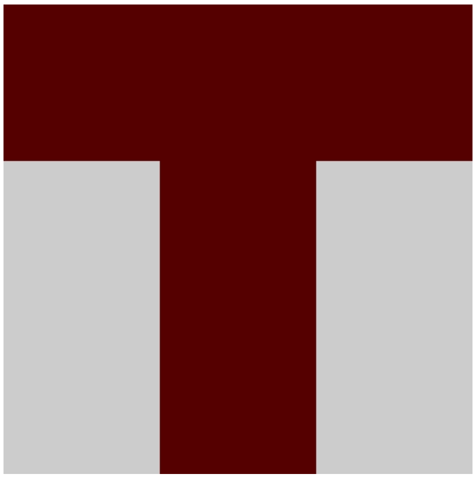
MEHTODOLGY AND ALGORITHM :

Basic idea behind this method is to imagine the 3 dimensional prismatic object as a binary matrix as shown in the given diagram. To get more accuracy in shape, value of n can be increased.



$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

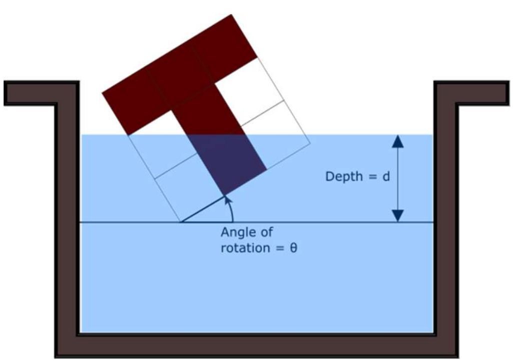
Prismatic object having T-shaped base

In this case, n = 3.

So now, we have to solve for these n^2 variables whose values will either be 1 or 0. For these we plan to get n^2 linearly independent linear equations. Then we will use matrix inversion method to solve for n^2 variables.

MEHTOD FOR OBTAINING A LINEAR EQUATION BY IMMERSING OBJECT IN WATER :

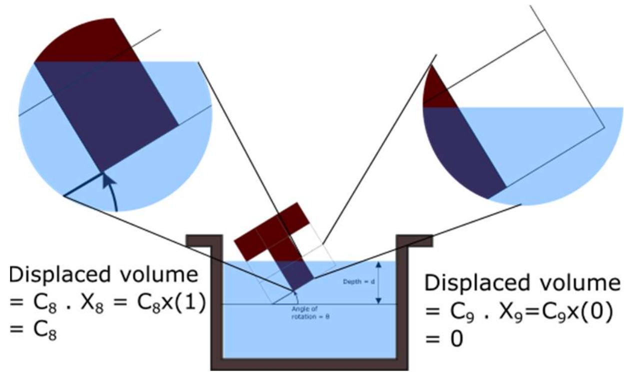
Let's consider an object shown and imagine it as a binary matrix. For simplicity of understanding, let's take n = 3 (i.e. 3x3 matrix).



$$\begin{bmatrix} X_1 & X_2 \cdots\cdots & X_N \\ X_{N+1} & X_{N+2} \cdots\cdots & X_{2N} \\ \vdots & \ddots & \vdots \\ \cdots & \cdots & X_{N^2} \end{bmatrix}$$

Now, rotate the matrix by an angle 'Ɵ' clockwise and dip it in the water by a depth 'd'. As shown in figure below, contribution by each cell in amount of water displaced will be equal to

'$(C_i).(X_i)$'.



So total volume displaced will be

$$V_{displaced} = \sum_{1}^{n^2}(Ci).(Xi)$$

Where,

C_i is the ratio volume submerged in water to total volume of i^{th} cell

$X_i = 1$ if i^{th} cell is filled,

0 if i^{th} cell is empty.

For each value of 'Ɵ' and 'd' we get an equation.

Now, problem arises that many of these equations we get for different values of 'Ɵ' and 'd' are linearly dependent. We need exact n^2 linearly independent equations. This will make sure that we get a unique solution.

Method to get n^2 linearly independent linear equations :

1. Take a square matrix A n^2 x n^2. Make all the elements of the matrix 0. Take different values of 'Ɵ' and 'd'. For each value of 'Ɵ' and 'd', we will get a equation of form $\sum_{1}^{n^2}(Ci).(Xi)$.

2. Replace a row of matrix A having all elements 0 by C_1, C_2, C_3 … $C_{(n^2)}$ respectively. Check whether rank of matrix A has increased after replacing the row.

3. If the rank has not increased, then make all the elements of that row 0 again.

4. If the rank has increased, then replace that row permanently with C_1, C_2, C_3 … $C_{(n^2)}$ respectively and note down the values of ‘Ѳ’ and ‘d’.

5. Continue the same procedure for new values of ‘Ѳ’ and ‘d’ until we get the rank of A equal to n^2. Now this matrix A will be the coefficient matrix which we can use in matrix inversion method.

Basically, if the rank of A increases after replacing a row, it means that the equation we have, is linearly independent with all other equations which previously have replaced a row in matrix A.

So, this is how we can obtain coefficient matrix whose rank is n^2.

VOLUME MATRIX :

VOLUME MATRIX :

For matrix equation of form:

$$[\text{Coefficient matrix}] \times [X] = [\text{Volume matrix}],$$

We have obtained coefficient matrix of size n^2 X n^2 whose rank is n^2. Now we only need to find volume matrix. For getting volume matrix, we use the n^2 values of ‘Ѳ’ and ‘d’ which we had noted down while finding coefficient matrix. Using those values, we transform the object accordingly and dip it. We can then measure the volume of water displaced, and get the Volume matrix.

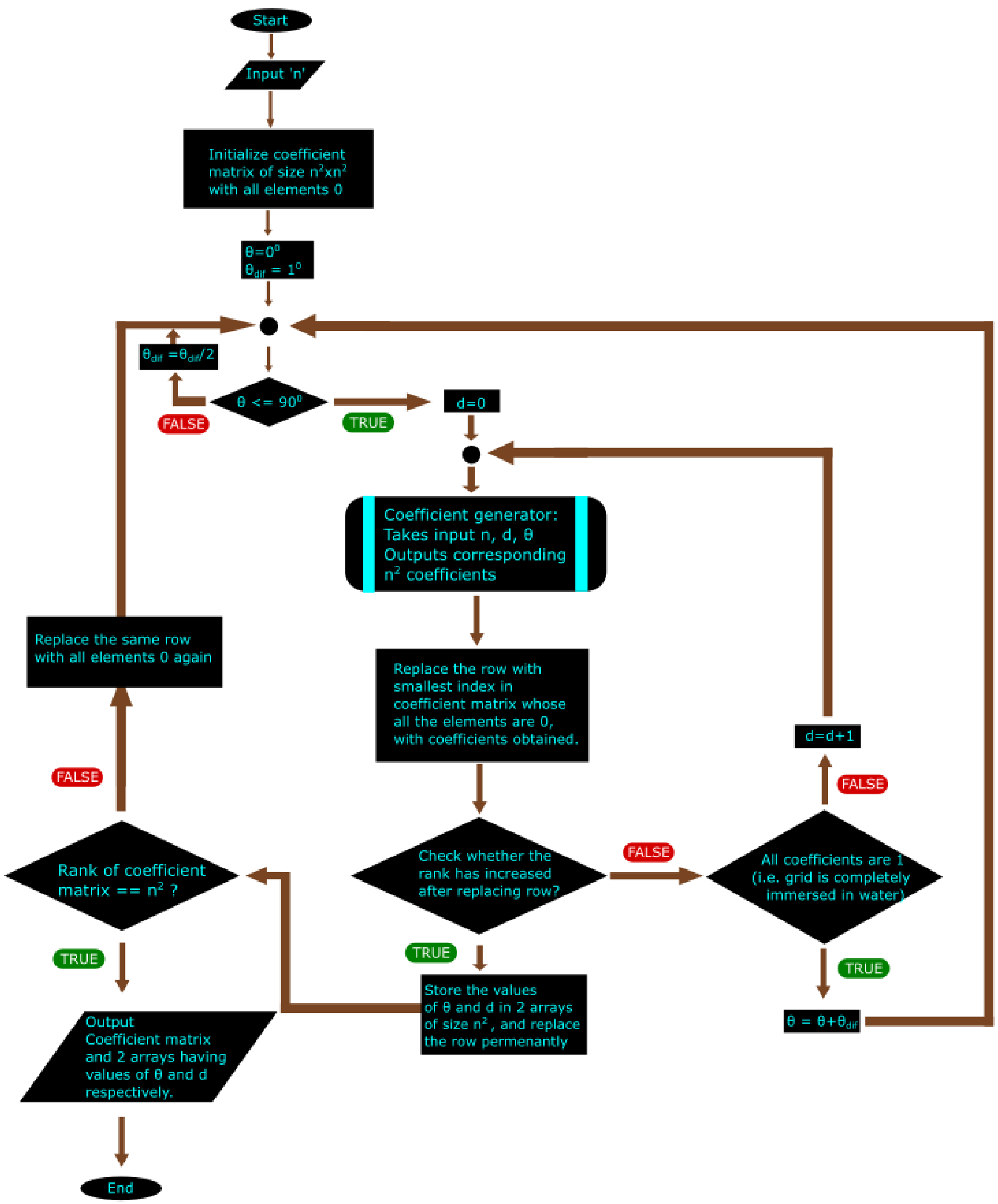
So, now that we have found both, coefficient matrix as well as volume matrix, using simple matrix inversion method, we can get the values of all the variables.

$$[X] = [\text{Coefficient matrix}]^{-1} \times [\text{Volume matrix}]$$

The following algorithm takes ‘n’ as input and outputs the coefficient matrix and corresponding values of ‘Ѳ’ and ‘d’.

A code was implemented which took ‘n’ as input and gave the coefficient matrix and corresponding n^2 values of theta and d. This code is based on following logic.

ALGORITHM :

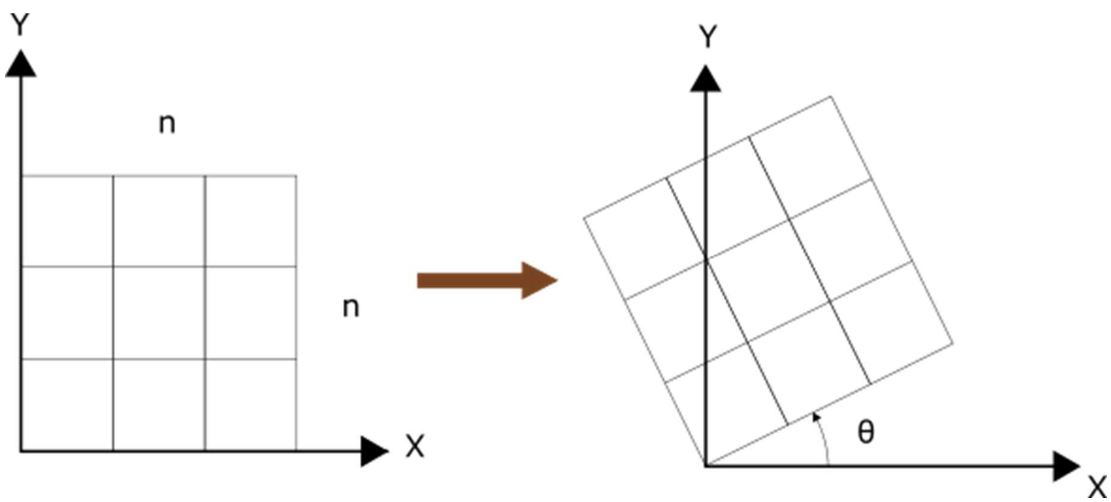

Start
Input 'n'
Initialize coefficient matrix of size n²xn² with all elements 0
θ=0⁰
θdif = 1⁰
θdif =θdif/2
θ <= 90⁰
FALSE
TRUE
d=0
Coefficient generator: Takes input n, d, θ Outputs corresponding n² coefficients
Replace the same row with all elements 0 again
Replace the row with smallest index in coefficient matrix whose all the elements are 0, with coefficients obtained.
d=d+1
FALSE
FALSE
Rank of coefficient matrix == n² ?
Check whether the rank has increased after replacing row?
FALSE
All coefficients are 1 (i.e. grid is completely immersed in water)
TRUE
TRUE
TRUE
Store the values of θ and d in 2 arrays of size n², and replace the row permenantly
θ = θ+θdif
Output Coefficient matrix and 2 arrays having values of θ and d respectively.
End

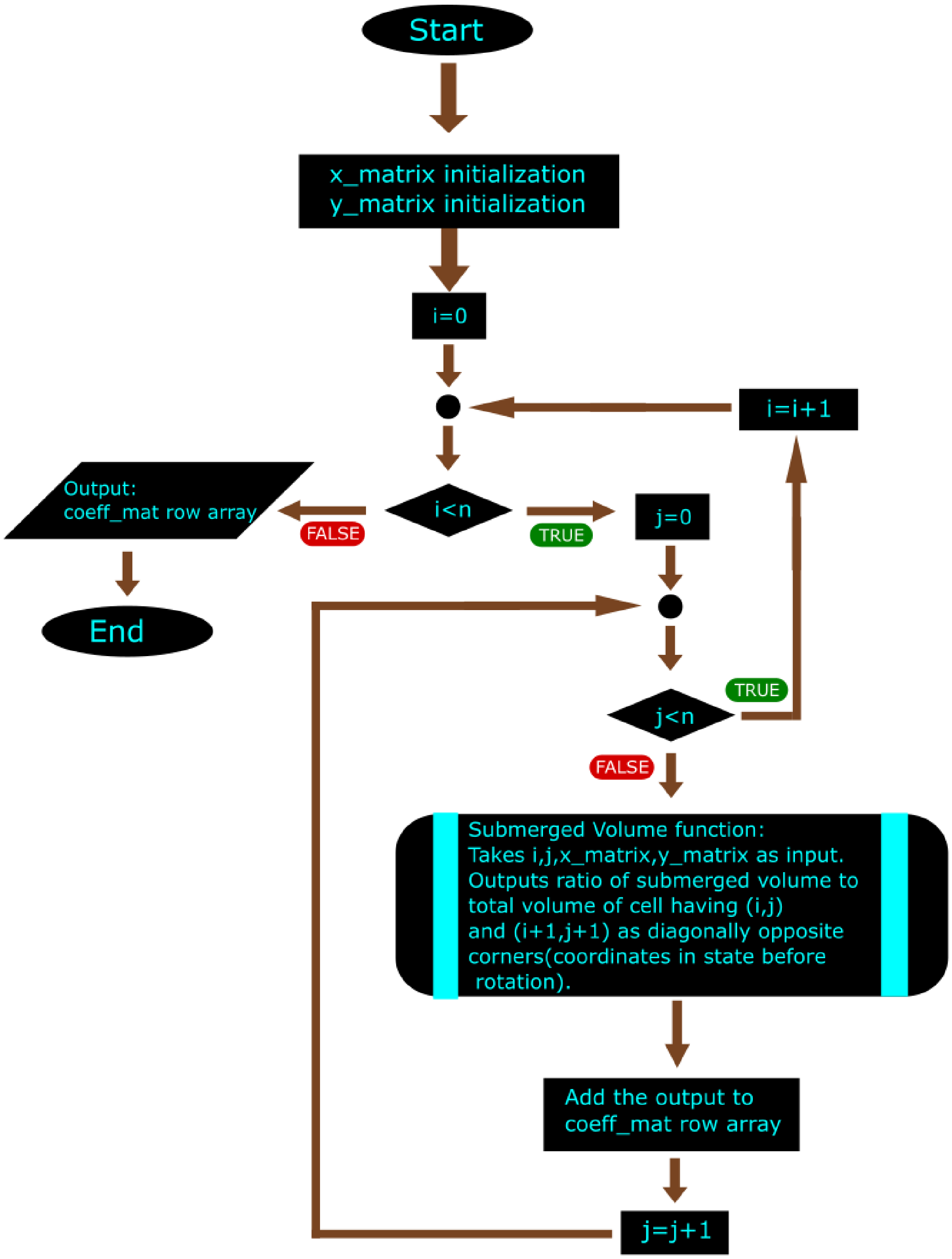
For 0 <= i,j <=n :

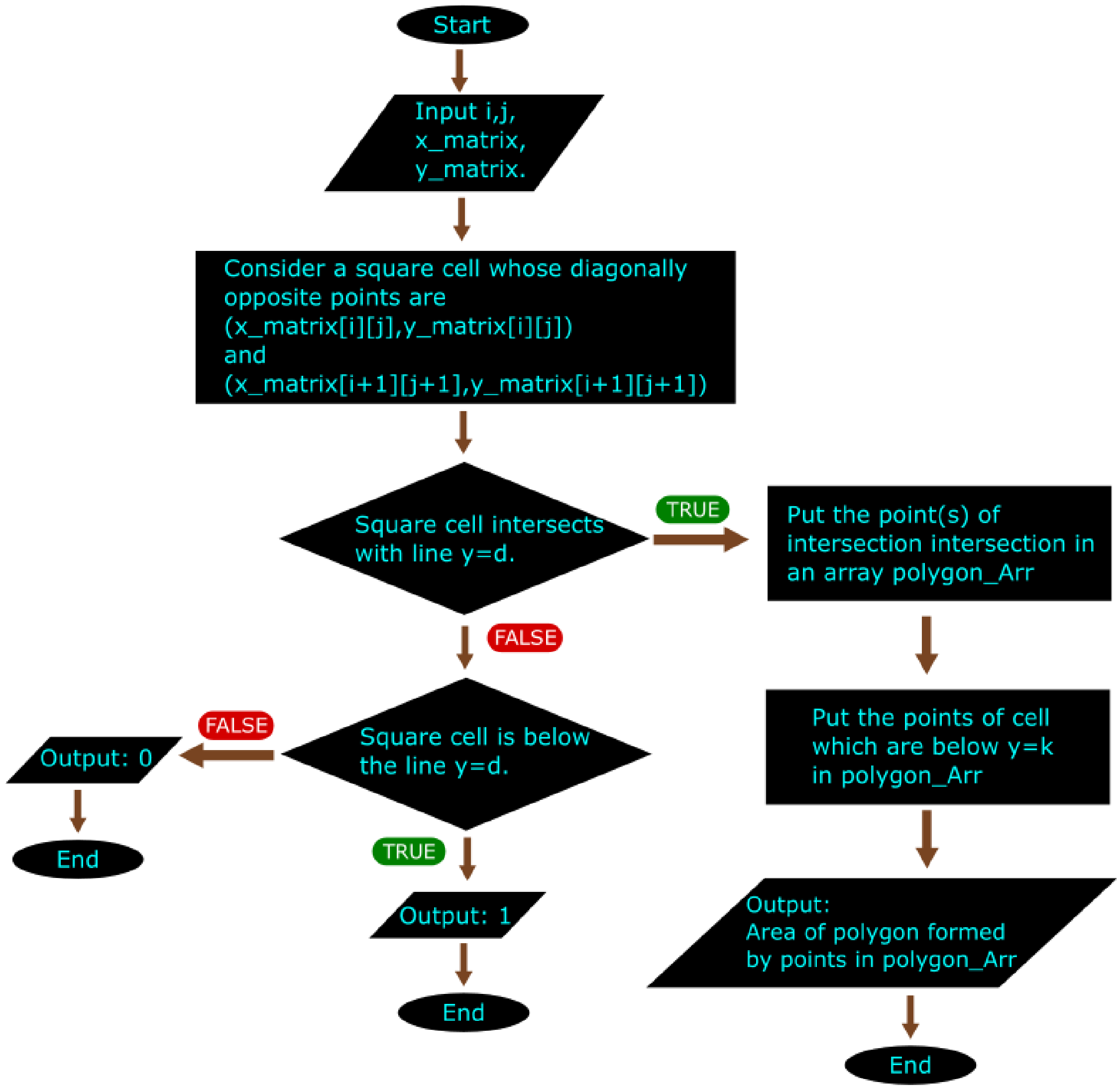
x_matrix and y_matrix are 2D arrays of (n+1)x(n+1) such that,

x_matrix[i][j] = x coordinate of point(i,j) after rotating it by θ in clockwise direction about origin.

y_matrix[i][j] = x coordinate of point(i,j) after rotating it by θ in clockwise direction about origin.




Start
x_matrix initialization
y_matrix initialization
i=0
i=i+1
Output:
coeff_mat row array
FALSE
i<n
TRUE
j=0
End
TRUE
j<n
FALSE
Submerged Volume function:
Takes i,j,x_matrix,y_matrix as input.
Outputs ratio of submerged volume to
total volume of cell having (i,j)
and (i+1,j+1) as diagonally opposite
corners(coordinates in state before
rotation).
Add the output to
coeff_mat row array
j=j+1



Start
Input i,j,
x_matrix,
y_matrix.
Consider a square cell whose diagonally opposite points are (x_matrix[i][j],y_matrix[i][j]) and (x_matrix[i+1][j+1],y_matrix[i+1][j+1])
Square cell intersects with line y=d.
TRUE
Put the point(s) of intersection intersection in an array polygon_Arr
FALSE
Square cell is below the line y=d.
FALSE
Output: 0
End
TRUE
Output: 1
End
Put the points of cell which are below y=k in polygon_Arr
Output:
Area of polygon formed by points in polygon_Arr
End