```
function [x, y_Richard] = Richardson_Extrpolation_Method(dydx, a, b, n, h, y_ini)
% Richardson Extrapolation Method Function solves 1st order initial value ODE
% with Richardson Extrapolation Method's
% dydx - First Order Differential Equation
% a    - starting point of a range
% b    - ending point of a range
% h    - step size
% n    - number of intervals

% Initialize x and y vectors;
y_Richard = zeros(n, 1);

% Initialize z vector;
z = zeros(n, 1);

% Store all the x values in a vector form.
if a > b
    H = -h;
else
    H = h;
end

%x = a : H : b;

% Size of Sub-step
sub_H = H / n;

% Initial value of x
x(1) = a;

% Initial value of y
y_Richard(1) = y_ini;

% Initial value of z_0 and z_1
z(1) = y_ini;
z(2) = z(1) + sub_H * dydx(x(1), z(1));

% Apply Richardson Extrapolation Method
for counter = 1 : n

    x(counter+1) = x(counter) + H;

    for i = 2 : n

        z(i + 1) = z(i - 1) + 2 * sub_H * dydx(x(counter) + (i-1)*sub_H, z(i));

    end

    y_Richard(counter + 1) = 1 / 2 * (z(end) + z(end-1) + sub_H * dydx(x(counter) + H,↙
z(end)));

end

end
```