MIE1210

Project 3 Report

Kyu Mok Kim
998745381

1. Describe the discretization for Eq. (2). For the discretization of the diffusion term, you can simply refer to your previous report. You should include a description for the central difference scheme and the 1st order upwind scheme.

In order to discretize the equation $\nabla \cdot \vec{u}\phi = \nabla \cdot \Gamma \nabla \phi$, the equation is integrated over the control volume: $(\rho u A \phi)_e - (\rho u A \phi)_w = \left(\Gamma A \frac{\partial \phi}{\partial x}\right)_e - \left(\Gamma A \frac{\partial \phi}{\partial x}\right)_w$. To simplify the integrated equations, the following variables are defined: F=ρu and D=Γ/(δx). Then, the integrated equation can now be written as: $F_e\phi_e - F_w\phi_w = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W)$, along with continuity equation of $\frac{d(\rho u)}{dx} = 0$ or $F_e - F_w = 0$.

For central differencing: $a_P\phi_P = a_W\phi_W + a_E\phi_E$ where

a_W	a_E	a_P
$D_w + \frac{F_w}{2}$	$D_e - \frac{F_e}{2}$	$a_W + a_E + (F_e - F_w)$

.

For 1st order upwind: $a_P\phi_P = a_W\phi_W + a_E\phi_E$ with $a_P = a_W + a_E + (F_e - F_w)$ where

a_W	a_E
$D_w + \max(F_w,\ 0)$	$D_e + \max(0,\ -F_e)$

.

These are for 1-D problems. However, for this project, the problem is 2-D. Therefore, the following discretization is derived and used.

For central differencing: $a_p\phi_p = a_W\phi_W + a_E\phi_E + a_S\phi_S + a_N\phi_N + S_u$ with central coefficient $a_p = a_W + a_E + (F_e - F_w) + a_N + a_s + (F_N - F_S) - S_p$.

Where $a_W = D_w + F_w/2$, $a_S = D_S + F_s/2$, $a_E = D_E + F_E/2$, $a_N = D_N + F_N/2$

Where $S_u = (2D+F)\phi_i$ or $(2D-F)\phi_f$ and where $S_p = -(2D+F)$ or $-(2D-F)$ for initial and end boundaries, respectively.

For 1st order upwind: $a_p\phi_p = a_W\phi_W + a_E\phi_E + a_S\phi_S + a_N\phi_N + S_u$ with central coefficient $a_p = a_W + a_E + (F_e - F_w) + a_N + a_s + (F_N - F_S) - S_p$.

Where $a_W = D_w + \max(F_W, 0)$, $a_S = D_S + \max(F_S, 0)$, $a_E = D_E + \max(0, -F_E)$, $a_N = D_N + \max(0, -F_N)$

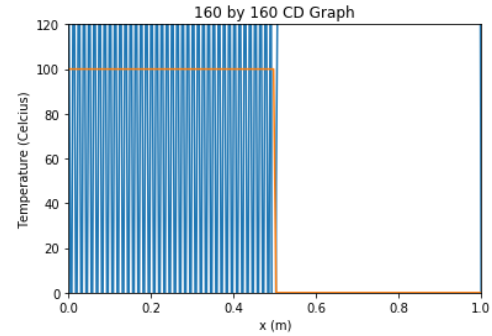
Where $S_u = (2D+F)\phi_i$ or $2D\phi_f$ and where $S_p = -(2D+F)$ or $-2D$ for initial and end boundaries, respectively.

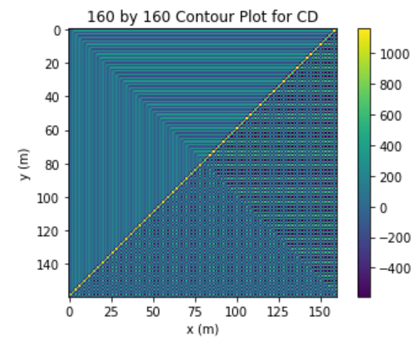
2. With Γ = 0, test the central difference and upwind schemes for ux = 2 and uy = 2. Do comparisons similar to Fig. 5.15 in your text book. Are there any stability issues? If so, explain why.

To analyze both CD and upwind for Γ = 0, 80x80, 160x160, and 320x320 mesh are analyzed (using "Advection-Diffusion23.py". The solution is then compared with the exact solution. (Temperature from diagonal line from northwest to southeast)

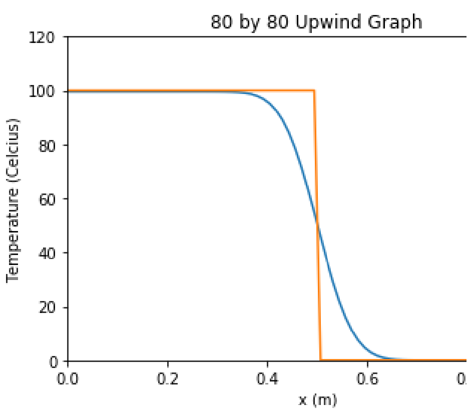
For CD, as gamma reaches zero, the Peclet number (F/D) reaches infinity. CD does not possess Transportiveness at high Pe (convection > diffusion) number since ϕ at a point is average of neighbouring nodes for all Pe. Therefore, there are stability issues (shown on the right).

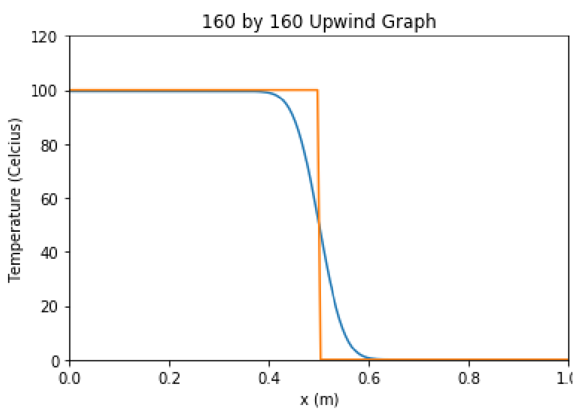
For Upwind, it is fine for high Pe number, so there are no stability issues. The contour plots and comparison with the exact solution is shown below for different mesh. False diffusion can be seen and this false diffusion reduces as higher mesh leads to closer solution to the exact solution.

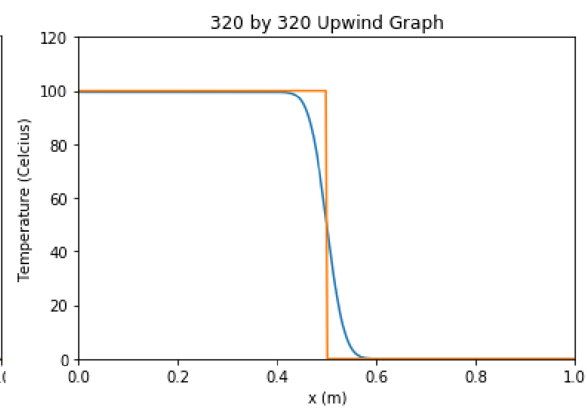


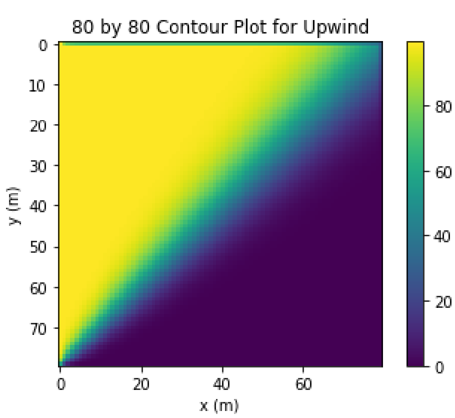


Execution time = 0.6131801605224609

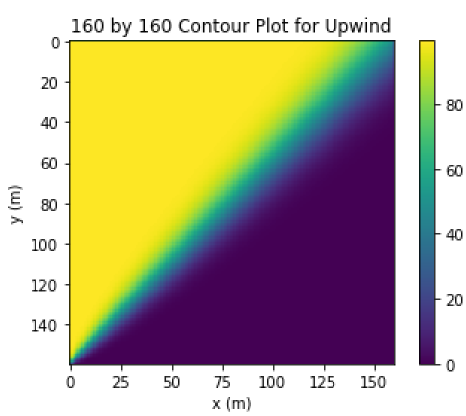




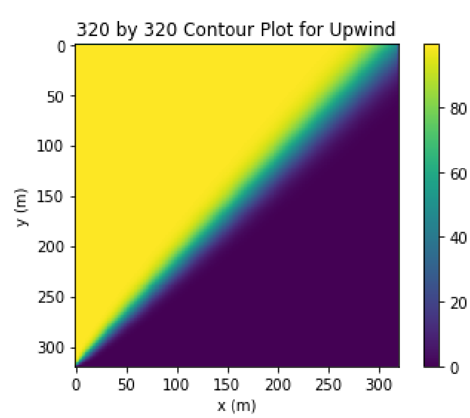




Execution time = 0.3964879512786865



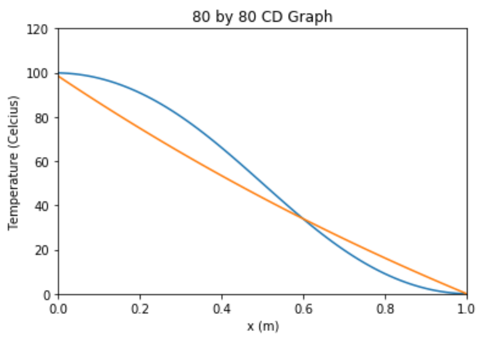
Execution time = 0.5968823432922363

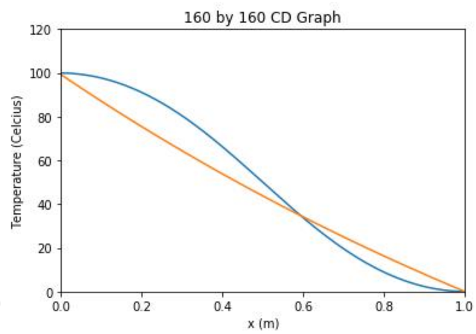


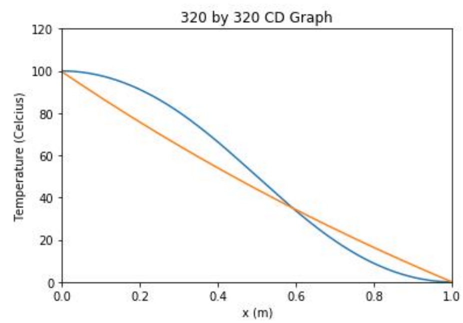
Execution time = 1.467745065689087

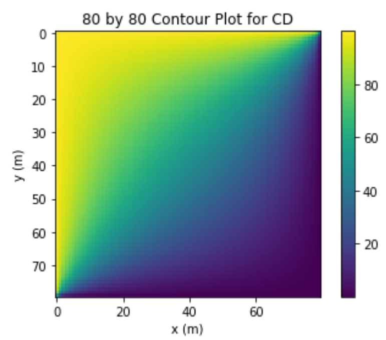
3. Repeat the previous step with Γ = 5. If there were stability issues it the previous step, did they persist? Explain.

Using the equation , $\frac{\phi - \phi_0}{\phi_L - \phi_0} = \frac{\exp(\rho u x/\Gamma) - 1}{\exp(\rho u L/\Gamma) - 1}$ the exact solution for Γ = 5 would be: ϕ=100*(exp(1.2*2*(1-x/nx)/g)-1)/(exp(1.2*2/g)-1), which can be seen with the orange line. The solution for CD and Upwind is compared with this exact solution. The stability issues in the previous step did not persist as the Peclet number is no longer infinite.

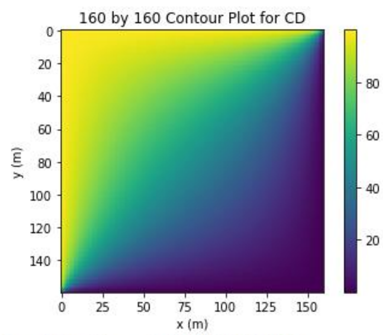




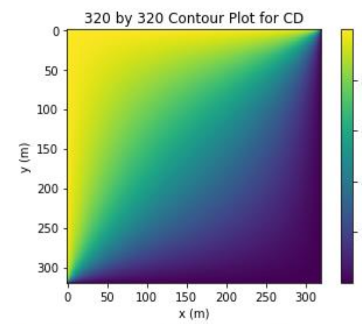




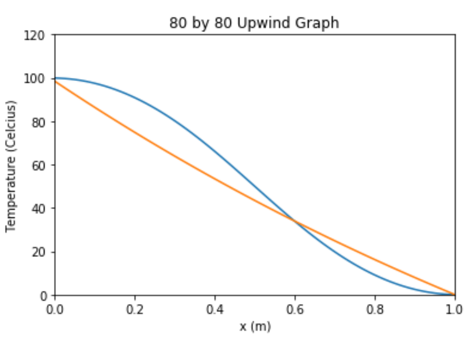
Execution time = 0.3166513442993164

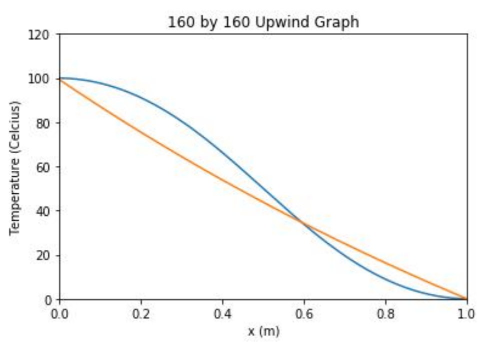


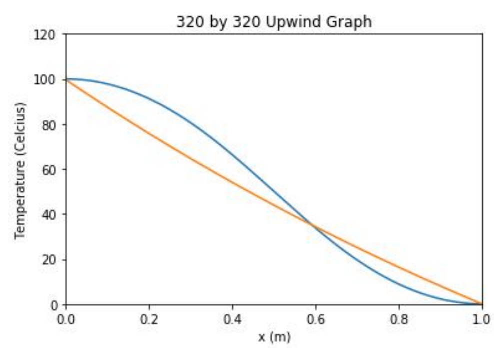
Execution time = 0.5806474685668945

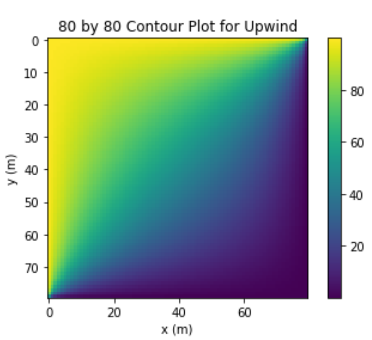


Execution time = 1.461259126663208

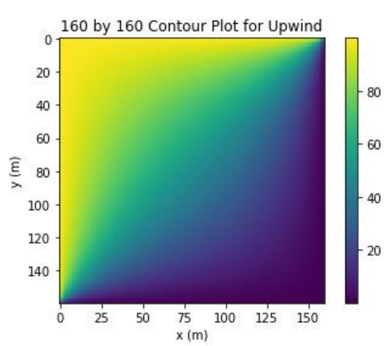




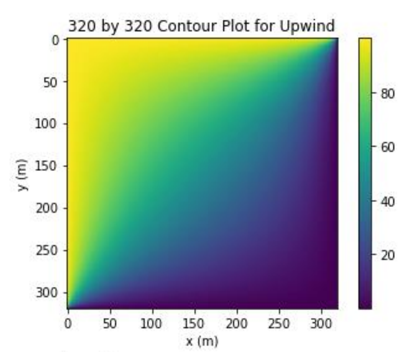




Execution time = 0.3954493999481201



Execution time = 0.5514960289001465



Execution time = 1.4584815502166748

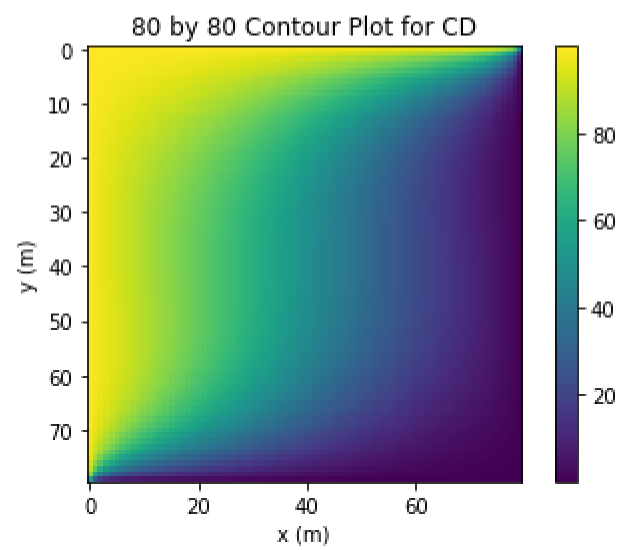
4. Create contour plots for a rotational velocity field with $\tilde{u}_x = -r \sin\theta$ and $\tilde{u}_y = r \cos\theta$. Repeat the order of convergence test from assignment 2 (only for a uniform mesh), and compare the results for central difference and first order upwind. Comment on your findings. Use grids with cell resolutions of 80 × 80, 160 × 160 and 320 × 320 (use finer if you wish).

To add rotational velocity field (circular wind), array of wind velcity for x and y direction is used as shown below for an example for 6 by 6 grid (code shown on the right).

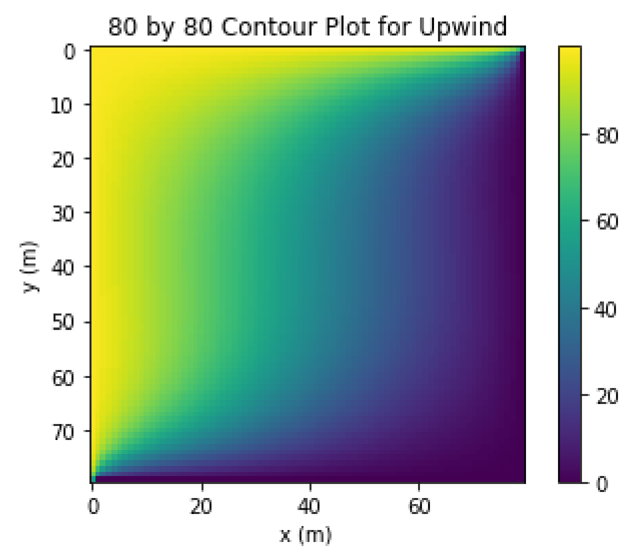
```
[[-2.5 -2.5 -2.5 -2.5 -2.5 -2.5]
 [-1.5 -1.5 -1.5 -1.5 -1.5 -1.5]
 [-0.5 -0.5 -0.5 -0.5 -0.5 -0.5]
 [ 0.5  0.5  0.5  0.5  0.5  0.5]
 [ 1.5  1.5  1.5  1.5  1.5  1.5]
 [ 2.5  2.5  2.5  2.5  2.5  2.5]]
[[-2.5 -1.5 -0.5  0.5  1.5  2.5]
 [-2.5 -1.5 -0.5  0.5  1.5  2.5]
 [-2.5 -1.5 -0.5  0.5  1.5  2.5]
 [-2.5 -1.5 -0.5  0.5  1.5  2.5]
 [-2.5 -1.5 -0.5  0.5  1.5  2.5]
 [-2.5 -1.5 -0.5  0.5  1.5  2.5]]
Execution time = 0.0
```

```
#Parameters
g=5     #gamma
ux=np.zeros(shape=(nx,ny))
uy=np.zeros(shape=(nx,ny))
for i in range (1,nx+1):
    for j in range (1,ny+1):
        y=-((i-1)-(nx-1)/2)
        x=(j-1)-(ny-1)/2
        r=np.sqrt(np.power(y,2)+np.power(x,2))
        if ((j-1)-(ny-1)/2)>0:
            theta=np.arctan(y/x)
        else:
            theta=np.pi+np.arctan(y/x)
        ux[i-1,j-1]=-r*np.sin(theta)
        uy[i-1,j-1]=r*np.cos(theta)
rho=1.2 #density of air (kg/m^3)
De=Dw=g/dx
Dn=Ds=g/dy
Fe=Fw=rho*ux
Fn=Fs=rho*uy
```

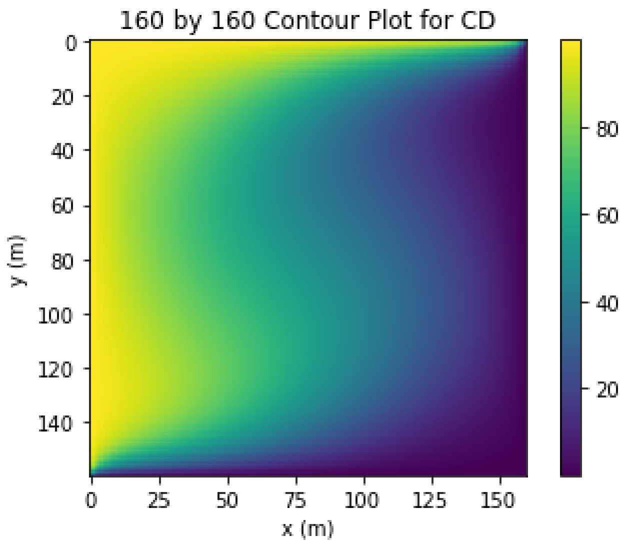
The above array shows that wind is blowing counter clockwise direction. The order of convergence is calculated for both methods using the same method from 2nd assignment: 80x80, 160x160, 320x320 meshes are used to calculate errors for coarse, fine, finest mesh. Order of convergence are shown below the contour plots. (Refer to "Advection-Diffusion4.py")



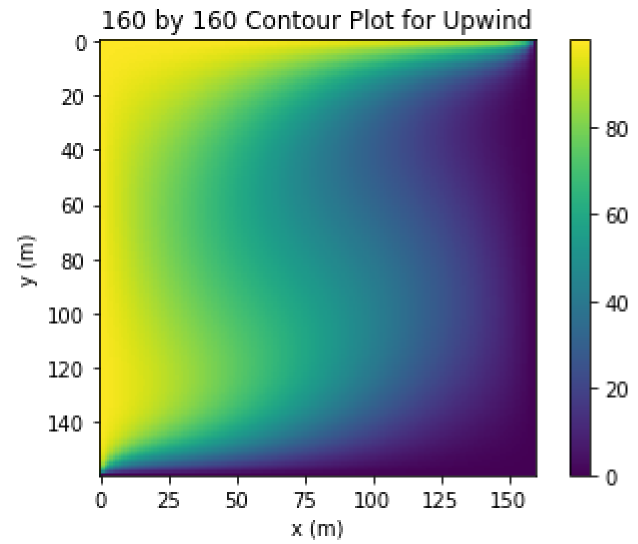

```
Order of Convergence for CD is: 0.850354707761
Order of Convergence for Upwind is: 0.84842448374
Execution time = 7.435666084289551
```



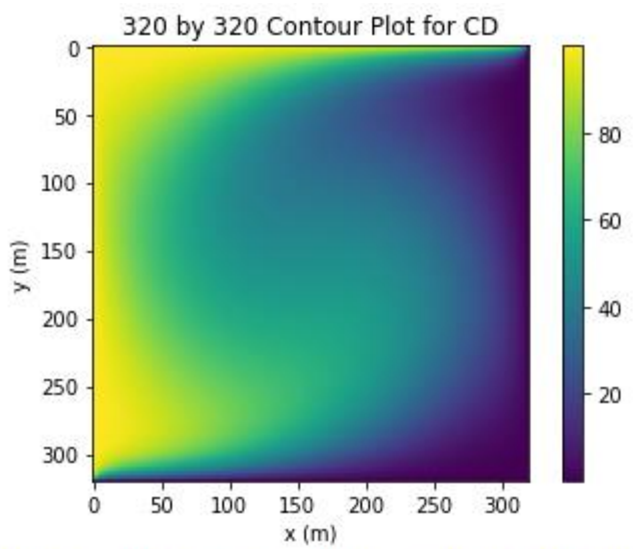

```
Order of Convergence for CD is: 0.850354707761
Order of Convergence for Upwind is: 0.848424483744
Execution time = 7.560286521911621
```



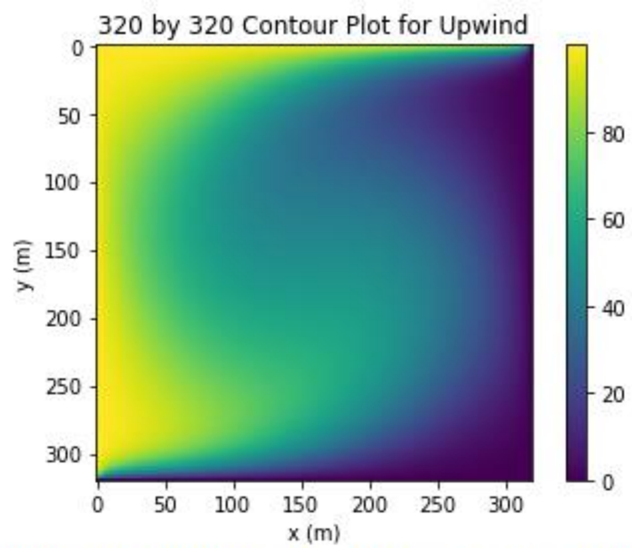

```
Order of Convergence for CD is: 0.850354707761
Order of Convergence for Upwind is: 0.848424483744
Execution time = 7.928232431411743
```




```
Order of Convergence for CD is: 0.850354707761
Order of Convergence for Upwind is: 0.848424483744
Execution time = 8.084009170532227
```




```
Order of Convergence for CD is: 0.850354707761
Order of Convergence for Upwind is: 0.848424483744
Execution time = 9.984901905059814
```




```
Order of Convergence for CD is: 0.850354707761
Order of Convergence for Upwind is: 0.848424483744
Execution time = 10.174554109573364
```

As seen, the higher mesh gives more details on the temperature and can observe the effects of rotating velocity fields clearer. The central difference and upwind contour plots look almost identical. The order of convergence is slightly higher for CD. However, as seen from #2, CD is not recommended when gamma is small (close to zero).