

Computation PDEs: Project 3

An Annular Model of Planetary Convection

Yadu Bhageria

In this final project the Convection Equations for annular flow are considered in terms of polar coordinates (r, θ) . This is a simplified model of thermally driven convection between two spheres.

$$T_t + \mathbf{u} \cdot \nabla T = \nabla^2 T \quad (1)$$

$$P_r^1(\mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) = \nabla p R_a T \hat{g} + \nabla^2 \mathbf{u} \quad (2)$$

$$\nabla \mathbf{u} = 0 \quad (3)$$

The domain is restricted to $1 < r < b$ and is split into a grid of size $(M + 1) \times N$. Therefore $\delta r = \frac{b-1}{M}$ and $\delta\theta = \frac{2\pi}{N}$. Results in Project 2 suggested that a more efficient way to get accurate results is to have $\frac{M}{N} < 1$ rather than have $M = N$. Thus I have maintained $\frac{M}{N} = \frac{1}{4}$ throughout Project 3 as well unless testing for the impact of varying M or N on the error.

The temperature boundary conditions are $T = 1$ on $r = 1$ and $T = 0$ on $r = b$. The boundaries $r = 1$ and $r = b$ are mentioned together as $\partial\Omega$ in the rest of the report.

`plotAnnulusScalarField.m` and `plotAnnulusVectorField.m` have been used throughout the project to plot the graphs.

The figures are at the end of the written assignment rather than incorporated through it for easier flow of reading. They have been referenced by Figure name where applicable.

1 Advection

First we try to advect a quantity $Q(r, \theta, t)$ by $Q_t + \mathbf{u} \cdot \nabla Q = 0$ such that it is second order in space and time. In polar co-ordinates we have that

$$\nabla Q = \frac{\partial Q}{\partial r} \hat{r} + \frac{1}{r} \frac{\partial Q}{\partial \theta} \hat{\theta} \quad (4)$$

$Q(r, \theta)$ is a scalar field and ∇ is the del operator. Joining this with a velocity vector field $\mathbf{u}(r, \theta) = u_{(r)} \hat{r} + u_{(\theta)} \hat{\theta}$ we get that

$$\mathbf{u} \cdot \nabla Q = u_{(r)} \frac{\partial Q}{\partial r} + u_{(\theta)} \frac{1}{r} \frac{\partial Q}{\partial \theta} \quad (5)$$

Now Q can be advected by solving the given equation $Q_t = -\mathbf{u} \cdot \nabla Q$ using Richtmyer's second-order Lax-Wendroff routine. This is done in `annulusAdvect.m`. This function assumes that $Q = \Omega_1(\theta)$ and $Q = \Omega_b(\theta)$ as well as that $\mathbf{u} = 0$ on the boundaries, $\partial\Omega$.

A suitable velocity field is found by deriving it from a stream function, ψ , using the relations

$$\begin{aligned} u_{(r)} &= \frac{\psi_\theta}{r} \\ u_{(\theta)} &= \psi_r \end{aligned}$$

I have chosen a non uniform stream function (Equation 6) that obeys $\psi = \psi_r = 0$ on the boundaries $\partial\Omega$. This can be visualized in the Figure 1.

$$\psi(r, \theta) = e^{-(r-\frac{b}{2})^2} e^{-(\theta-\pi)^2} (r-1)^2 (r-b)^2 \quad (6)$$

The initial non uniform temperature chosen is as in Equation 7. This is shown in first image of Figure 2 or the first image of Figure 3.

$$Q_0(r, \theta) = .01 \sin(\pi r) \sin(\theta) \quad (7)$$

Advecting this field forward in time is shown in Figure 2. I chose values of $M = 2^6$, $N = 2^8$, $\delta t = 0.01 \min(\delta r, \delta \theta)$ and number of time steps = 10000. It can be seen, by considering the velocity field given in Figure 1, that the quantity Q follows the vectors of the velocity field, \mathbf{u} . Furthermore, advecting the quantity Q back in time returns it to its initial state that is visually indistinguishable (Figure 3) but has some small error. The errors in this specific case are

```
max_error =
1.6651e-06

average_error =
1.0875e-09
```

where max error = $\max_{r,\theta}(|Q_0 - \hat{Q}|)$ and average error = $\frac{|Q_0 - \hat{Q}|}{(M-1)N}$. Note that $Q(1, :)$ and $Q(M+1, :)$ are not changed by the advection function.

Thus the routine seems to be functioning as intended. These results can be reproduced using `q1.m`.

1.1 Check that the Routine is 2nd Order in Space and Time

Now we need to check whether this routine is 2nd order in space and time. That is the errors should be $\mathcal{O}(\delta r^2, \delta \theta^2, \delta t^2)$. This can be done by keeping two of the three values out of $\delta r, \delta \theta, \delta t$ constant and varying the other to find various values for the average error. Then the inbuilt MATLAB function can be used to fit this data to a polynomial of order 2 to make sure that the average error results are indeed 2nd order. I have implemented this in `q1ErrorPlots.m` that also calls `q1Errors.m`. The results can be seen in Figure 4 that demonstrate the 2nd order behaviour of the implemented routine.

2 Diffusion

Now we wish to diffuse a quantity by $Q_t = \nabla^2 Q$. This can be done using the MultiGrid routine formulated in project 2 with revised Dirichlet boundary conditions and revised Gauss-Seidel and residual update steps. Using the Crank-Nicolson routine, we can write

$$Q^{(i+1)} - \frac{\delta t}{2} \nabla^2 Q^{(i+1)} = Q^{(i)} + \frac{\delta t}{2} \nabla^2 Q \quad (8)$$

(9)

This can be written in the simplified form

$$Q^{(i+1)} - \frac{\delta t}{2} \nabla^2 Q^{(i+1)} = f$$

where f is easily computable (done in `T_delsqr.m`). `T_MultiGrid.m`, `T_GaussSeidel.m`, and `T_residual.m` are used to then compute $Q^{(i+1)}$ by solving the system $AQ^{(i+1)} = f$ where $A = I - \frac{\delta t}{2} \nabla^2$.

To make sure the routine is working properly, we use a similar quantity Q_0 defined above (in Equation 7) but multiplied by 100 and with added boundary condition of $Q_0(1,:) = 1$. We then advect this quantity using the function `annulusAdvect.m` until the maximum error between each iteration drops below a specified tolerance. This is done in `q2DiffuseTest.m` with the same values of $M = 2^6$, $N = 2^8$, and $\delta t = 0.01 \min(\delta r, \delta \theta)$ as in `q1.m` along with a tolerance level of 10^{-5} . The results can be seen in Figure 5 which agree with the expected intuition of the annulus tending to a linear gradient from $r = 1$ to $r = b$ normalizing any initial perturbations due to the Dirichlet boundary conditions of $Q_0(1,:) = 1$ and $Q_0(b,:) = 0$. That is the quantity moves from areas of high concentration to areas of low concentration.

2.1 Combining Advection and Diffusion

Now that we are happy with the functioning of the diffusion routine, it is combined with the advection routine to solve for the Temperature field in the annulus. This is done by trying solving the combined problem

$$T_t + \mathbf{u} \cdot \nabla T = \kappa \nabla^2 T \quad (10)$$

where κ controls the rate of diffusion compared to that of advection. This equation is solved by first solving for $T_t = -\mathbf{u} \cdot \nabla T$ and then $T_t = \kappa \nabla^2 T$.

In order to test the impact of *kappa* term, we have chosen the non-negative temperature field

$$T_0(r, \theta) = \sin(\pi r)^2 \sin(\theta)^2 + \frac{r-b}{1-b} \quad (11)$$

which obeys $T_0(1,:) = 1$ and $T_0(b,:) = 0$. The chosen stream function, ψ , is

$$\psi = (r - b)^2(r - 1)^2 e^{\sin(2\theta)} \quad (12)$$

which obeys $\psi = \psi_r = 0$ on $\partial\Omega$ and so we have a suitable velocity field.

Figure 8 shows the results of the varying κ on advection-diffusion routine. These results are derived by varying the value of κ in q2.m. $\kappa = 0$ gives a purely advected solution. This created large gradients as the Temperature is advected around the swirls without diffusing at all. $\kappa = 0.001$ mimics the pure advection solution closely but with the large gradients in the Temperature smoothed out. As κ is increased, the solution starts to become more uniform. For $\kappa = 1$ the solution is an almost uniform (i.e. the pure diffusion solution) but with slight bulges due to the velocity field. For $\kappa = 2$ the solution looks almost identical to the pure diffusion linear gradient solution. In terms of the number of iterations to converge, as $\kappa \rightarrow 0$ the number of iterations $\rightarrow \infty$. For $\kappa = 0$, the solution never converges (for the given definition) as it keeps on advecting. This is reasonable as for larger κ there is greater diffusion and the field becomes more uniform more quickly.

3 Vorticity

The vorticity equation is

$$\omega_t + \mathbf{u} \cdot \nabla \omega = \nabla^2 \omega + F \quad (13)$$

where F is some forcing. The Dirichlet conditions on ψ are $\psi = 0$ on $\partial\Omega$.

The Dirichlet condition for ω is slightly more complicated. We know that $\psi_r = 0$ on $\partial\Omega$. Starting from

$$\begin{aligned} -\omega &= \nabla^2 \psi \\ \omega &= -\psi_{rr} - \frac{1}{3}\psi_r - \frac{1}{r^2}\psi_{\theta\theta} \end{aligned}$$

Since $\psi = 0$ on $\partial\Omega$ then so does ψ_θ and $\psi_{\theta\theta}$. Thus we can further simplify to

$$\omega = -\psi_{rr}$$

This can now be solved using the same method as in Lecture 20 by taking a Taylor expansion ψ around the boundaries, eliminating known terms, estimating the unknown term and finally getting individual Dirichlet conditions on each boundary. This has been implemented in omega_BC.m.

So now to solve the vorticity equation, we do exactly as was done in Question 2 where we first advect the field and then diffuse it. Since advection does not change the values on $\partial\Omega$, it can be used without any alterations for this too. For the diffusion routine, a simple alteration of enforcing the

non-zero Dirichlet boundary condition is added. This code can be found in `omega_MultiGridV.m`, `omega_GaussSeidel.m` and `omega_residual.m`.

In order to test the complete routine, we use the same function (Equation 11) as for testing the advection-diffusion routine in the previous section. Further we have chosen a forcing function, F , same as Equation 7.

This has been implemented in `q3.m` that utilizes `psi_Eqn.m`, `psi_MultiGridV.m`, `psi_GaussSeidel.m` and `psi_residual.m`. The solution is considered converged once the maximum difference between errors is less than 10^{-10} . The results can be seen in Figure 9 along with Figure 2 for the initial ω . The routine appears to be working correctly as the computed final stream function, ψ , resembles the forcing function, F , closely. Changing the initial ω to within reason results in this final same solution and can be done by changing the code in `q3.m`.

4 The Full Problem

Now that we can confident that the Advection-Diffusion routines are working properly for the Temperature as well as routines for solving the vorticity equation and finding ψ from ω , we can tackle the full problem. This has be done in `q4.m`. The main difference in the codes in the inclusion of 2 parameters: Rayleigh number and Prandtl number. The Rayleigh number, R_a , is a measure of the thermal driving force. The Prandtl number, P_r , is the ratio of the fluid kinematic viscosity to the thermal diffusivity which basically serves the same function as κ for the Temperature field. Noting that the solution must be studied for various values of b and various values of κ as well this means that the parameter space we are interested in exploring is 4 dimension. This is obviously quite large and thus will not be exhaustively studied. Furthermore, we will stick to assessing situations arising from our specific perturbations.

We use a perturbation of

$$T_0(r, \theta) = 0.01 \left[\sin(\pi r)^2 \sin(\theta)^2 + \frac{r - b}{1 - b} \right] \quad (14)$$

$$T_0(1, \theta) = 1 \quad (15)$$

throughout this section.

Firstly, we go about computing the mentioned R_c value in the assignment.

4.1 Approximate Value of R_c

In this section it is assumed $\kappa = P_r = 1$ and $b = 2$ to begin with.

This gives an approximation for $R_a \approx 3000$. This can be seen in Figure 10. Furthermore it seems that for larger values of R_a , the rolls appear to be closer to the $r = b$ boundary. I.e. they are in a sense stronger which makes sense with the given definition of the Rayleigh number being the thermal driving force as a stronger thermal driving force will result in stronger advection currents due to the perturbations in the temperature. For $b = 2$, it seems that the varying the value of R_a does not impact the number of rolls that appear as there are still 4 rolls for $R_a = 10000$.

Now we consider the impact on the value of R_c when changing the value of b .

For $b = 3$, and $b = 5$ we get that $R_c \approx 300$ (Figure 11) and $R_c \approx 25$ (Figure 12) respectively as well. So it seems that increasing the value of b quickly decreases the value of R_c .

4.2 Varying P_r

Choosing a small value of P_r reduces the impact of thermal diffusivity which in turn results in a smaller impact from the thermal driving force. So going back to $b = 2$, it can be seen in Figure 14 that even for $R_a = 10000$ when $P_r = 0.01$, the solution converges to no rolls. Even for $b = 3$, with $R_a = 500$ and with $P_r = 0.01$, no rolls appear.

Based on these results I believe the ratio between P_r and R_a matters as well as than the individual values themselves.

4.3 Conclusion

The number of rolls are impacted by the chosen values of R_a , P_r , and b .

Finally it is worth noting that a lot of the derived results are specific to the initial conditions and vary massively from situation to the situation. But nevertheless, it seems that the entire routine is function sufficiently and thus can be used, given enough time, to solve any specified problem of this type.

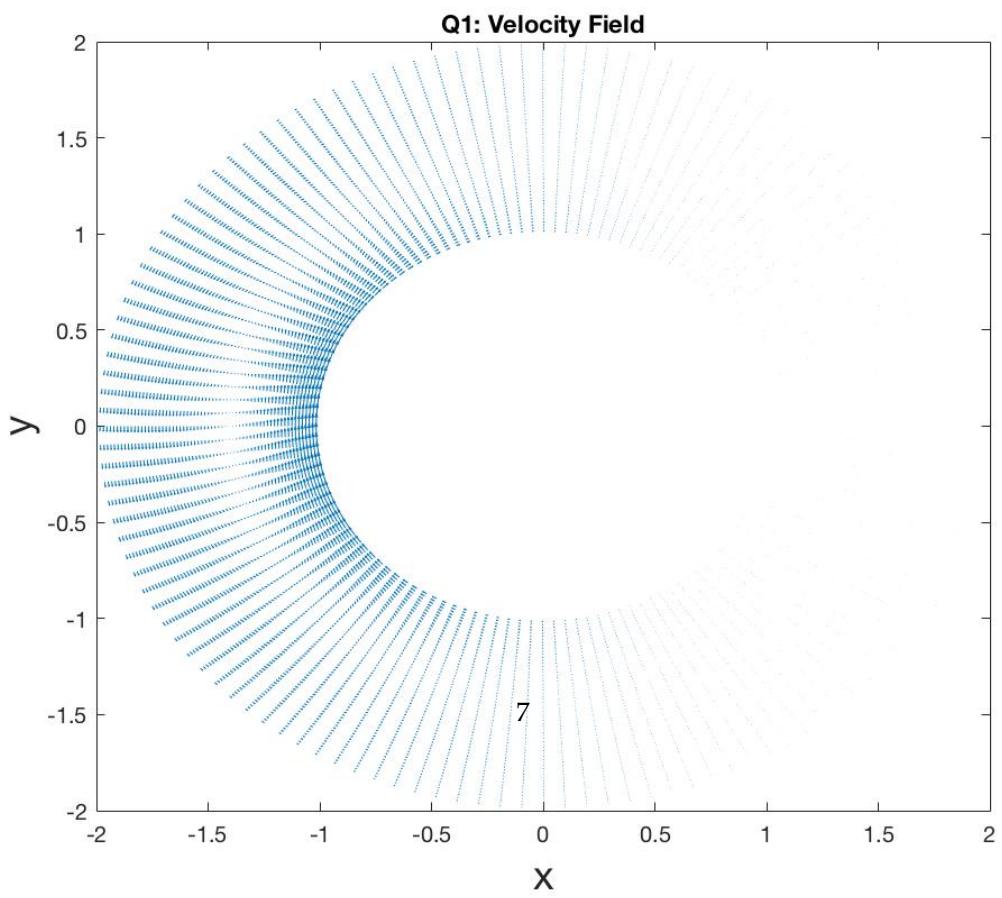
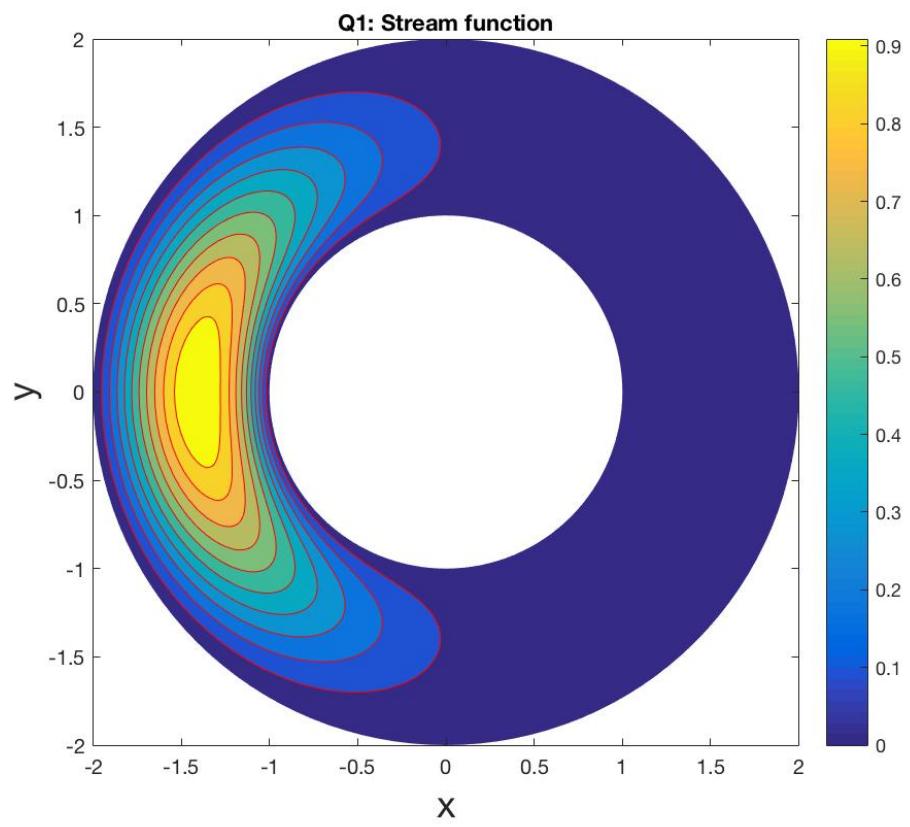


Figure 1: The stream function, ψ , and the corresponding velocity field, \mathbf{u} , chosen for Q1 with $b = 2$

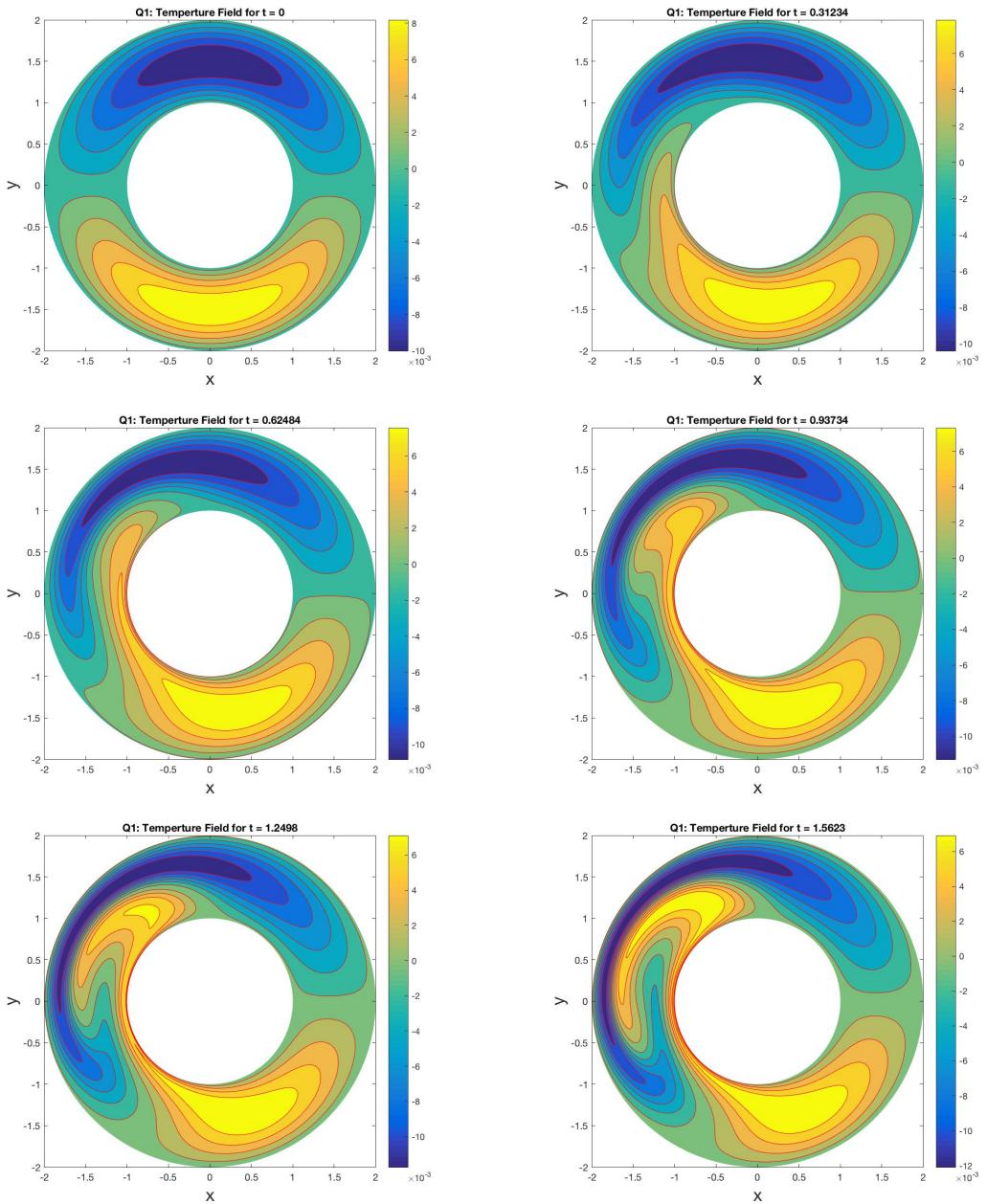


Figure 2: Advecting the specified field Q_0 for $\delta t = 1.5625e - 04$ and $tstep = 10000$ with $M = 2^6$ and $N = 2^8$

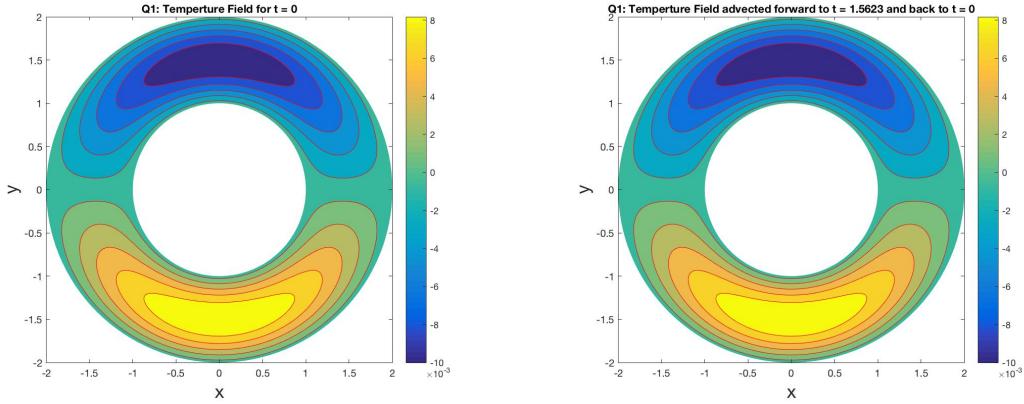


Figure 3: Comparison of the initial quantity Q_0 and the quantity advected forwards to $t = \delta t \cdot \text{tstep}$ and then back again to $t = 0$. Parameters are $\delta t = 1.5625e - 04$ and $\text{tstep} = 10000$ with $M = 2^6$ and $N = 2^8$.

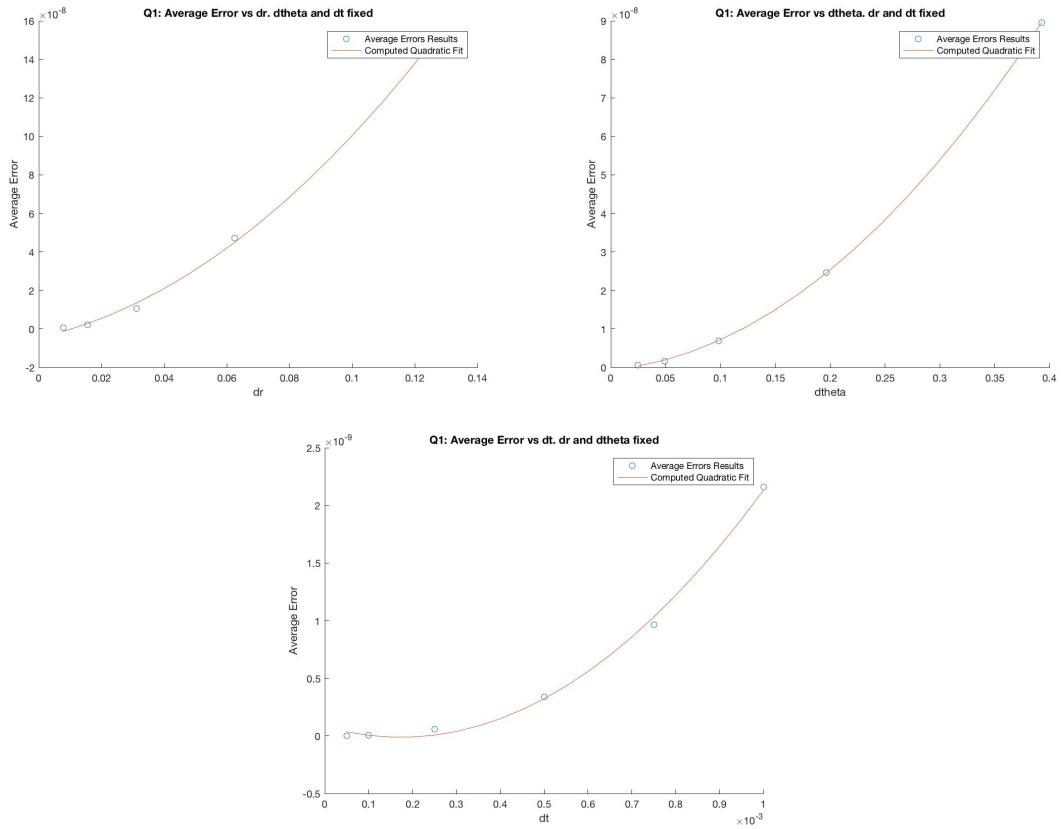


Figure 4: Average Error plots for varying $\delta r, \delta\theta, \delta t$ one at a time.

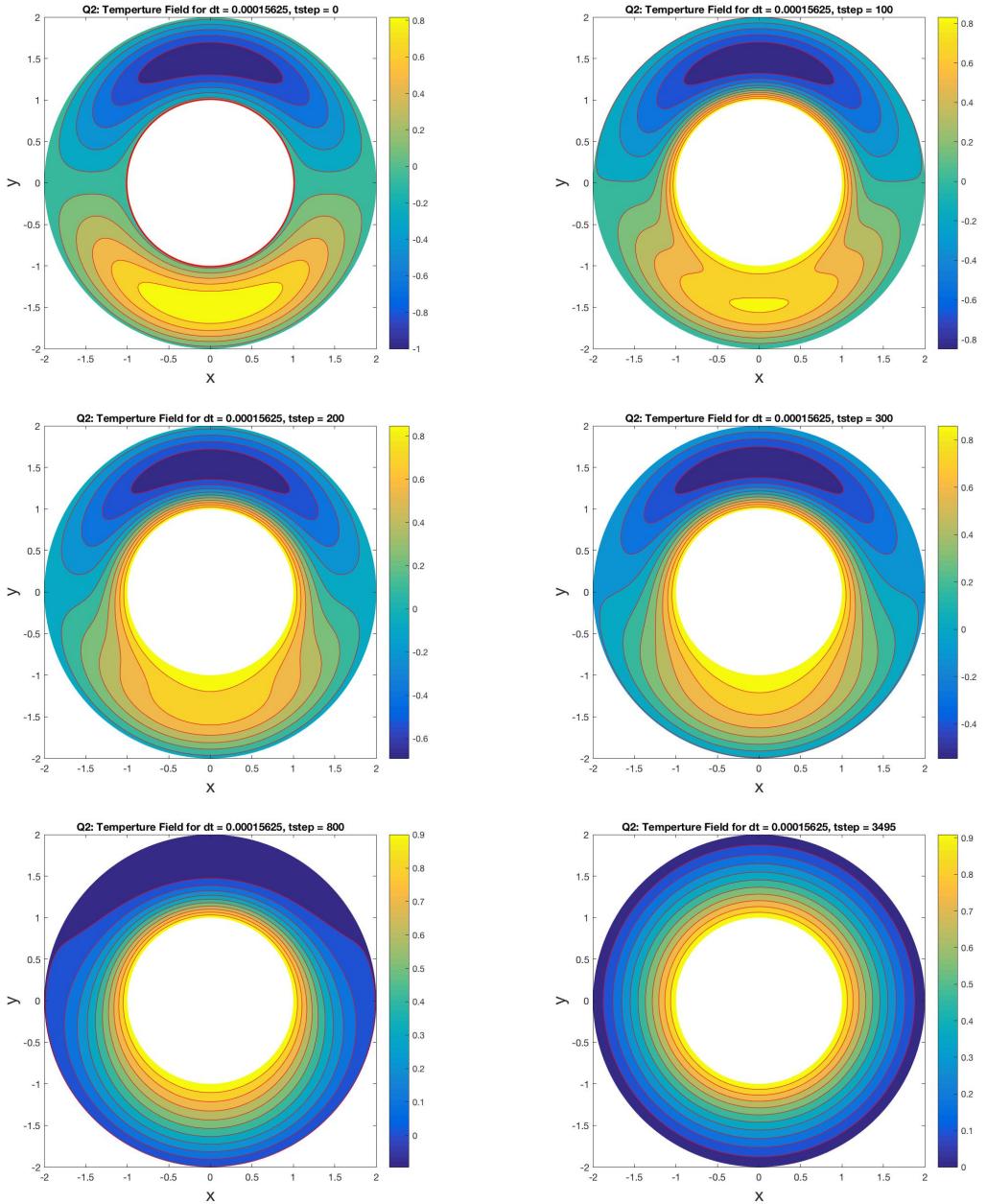


Figure 5: Q2: Diffusing the specified field Q_0 for $\delta t = 1.5625e - 04$ with $M = 2^6$ and $N = 2^8$. The solution converges to a tolerance of 10^{-5} for the difference between the maximum error between iterations after 3495 iterations.

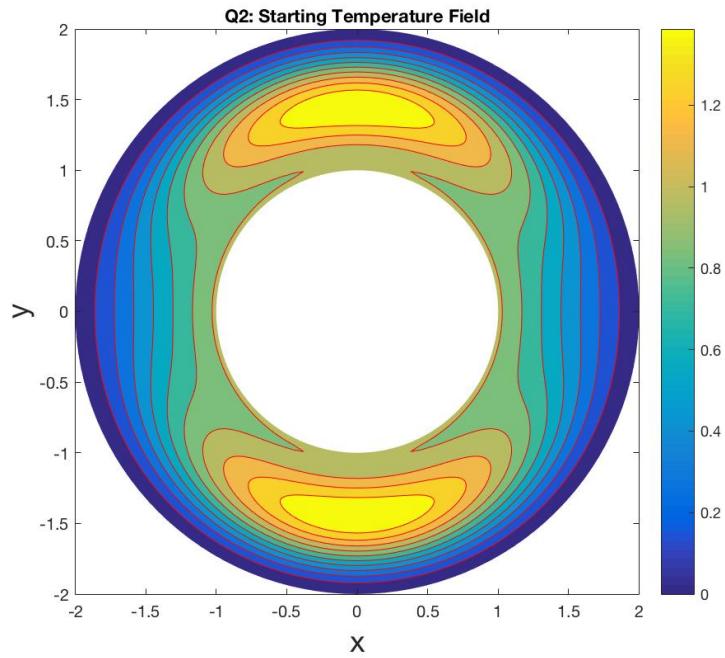


Figure 6: Q2: Starting Temperature Field defined in Equation 11 to test the impact of varying κ with $M = 2^6$ and $N = 2^8$.

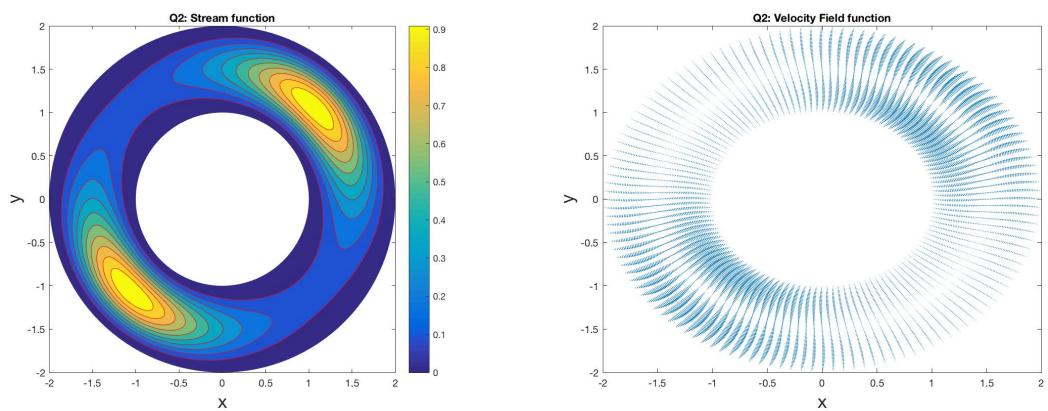


Figure 7: Q2: Stream Function Field defined in Equation 12 to test the impact of varying κ with $M = 2^6$ and $N = 2^8$ along with the derived velocity field.

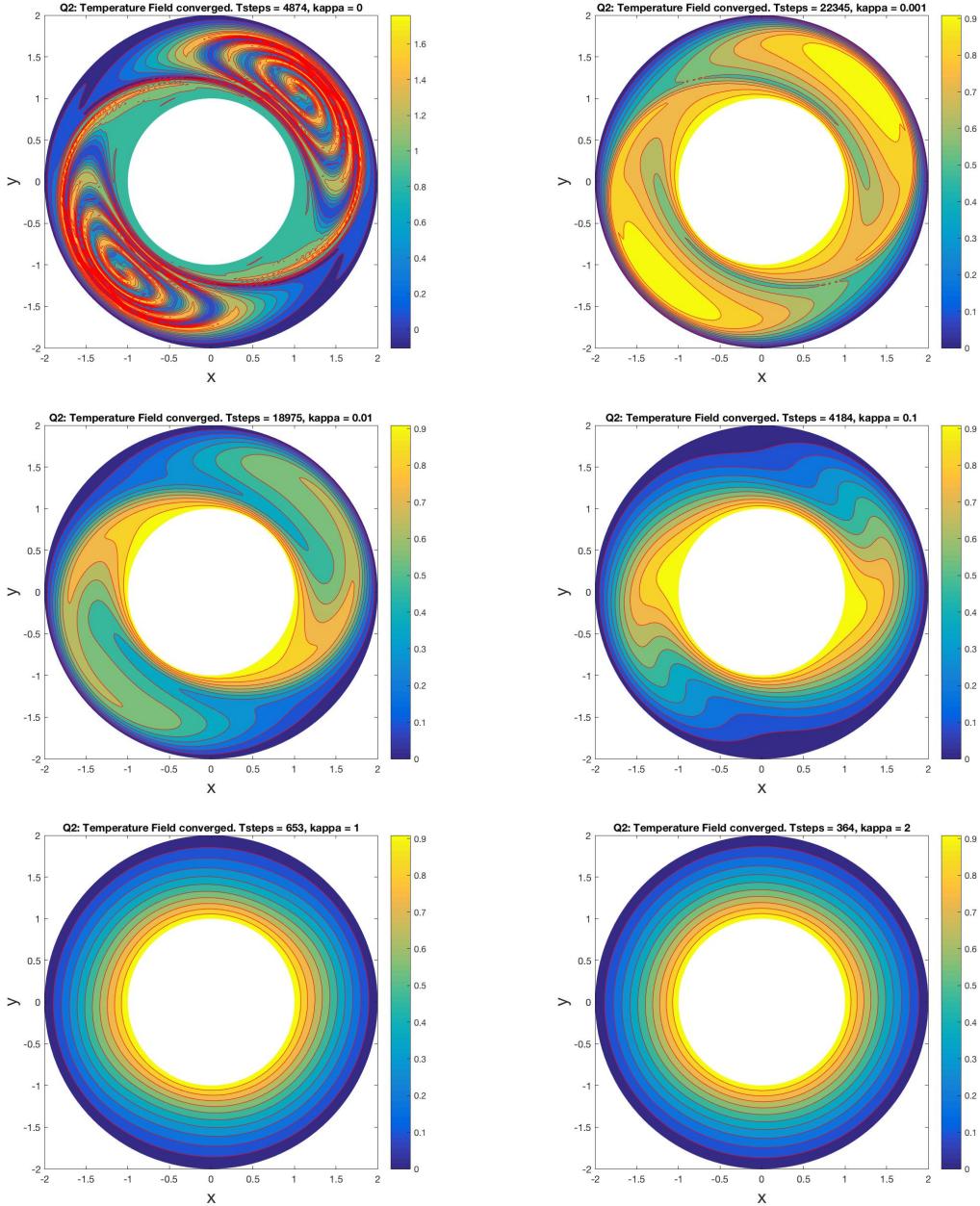


Figure 8: Q2: Advection and Diffusing the specified field T_0 for $\delta t = 1e - 03$ with $M = 2^6$ and $N = 2^8$. Iterations are stopped when the maximum difference between subsequent T 's converges to a tolerance of 10^{-5} .

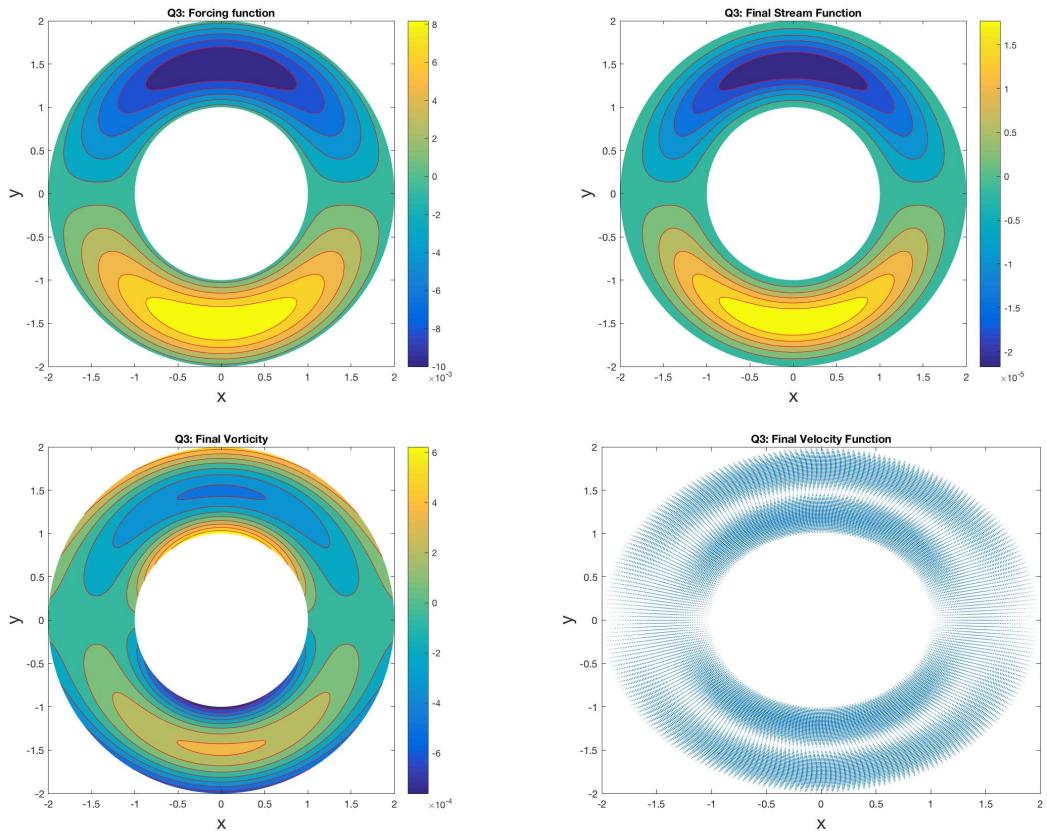


Figure 9: Q3: Advection and Diffusing the vorticity with $M = 2^6$ and $N = 2^8$. Iterations are stopped when the maximum difference between subsequent T 's converges to a tolerance of 10^{-10} .

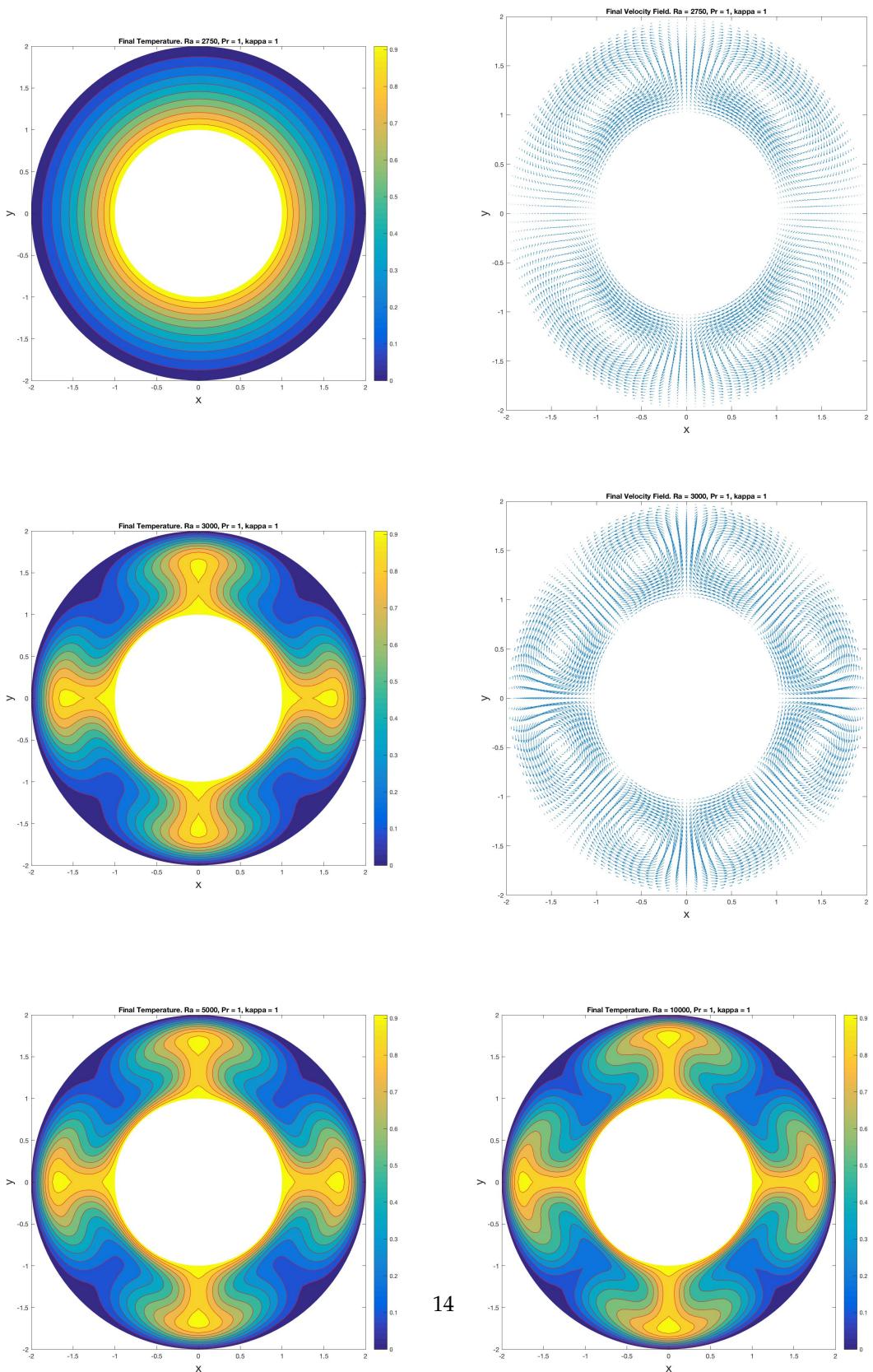


Figure 10: Q4: Final Temperature Field after solving the full problem for $P_r = \kappa = 1$ and $b = 2$ for various values of R_a .

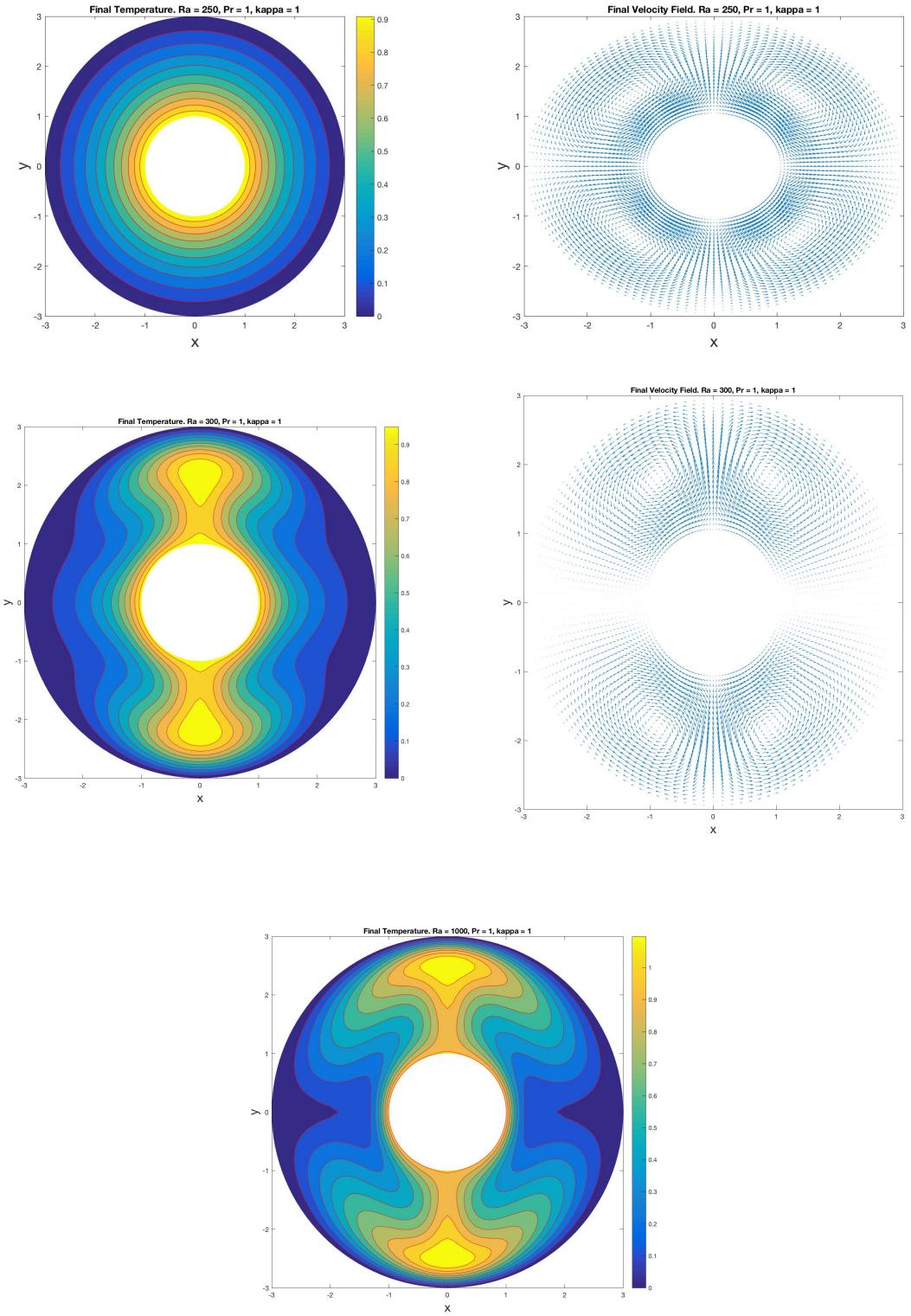


Figure 11: Q4: Final Temperature Field after solving the full problem for $P_r = \kappa = 1$ and $b = 3$ for various values of R_a .

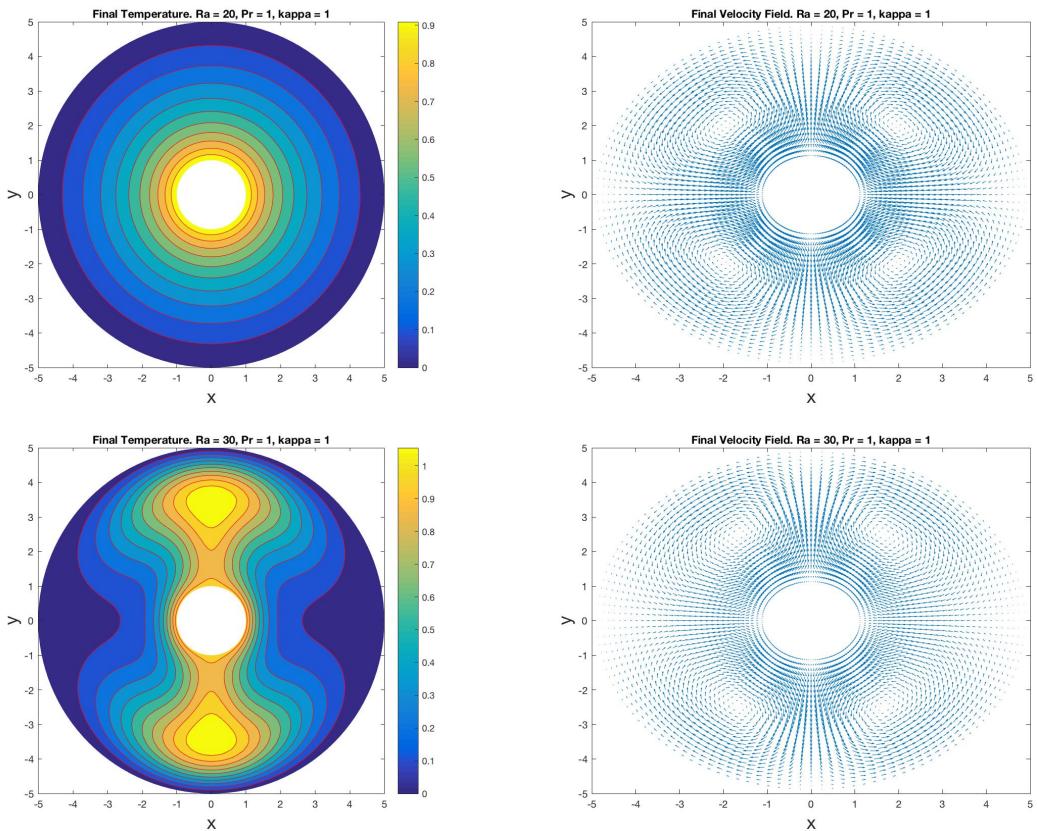


Figure 12: Q4: Final Temperature Field after solving the full problem for $P_r = \kappa = 1$ and $b = 5$ for various values of R_a .

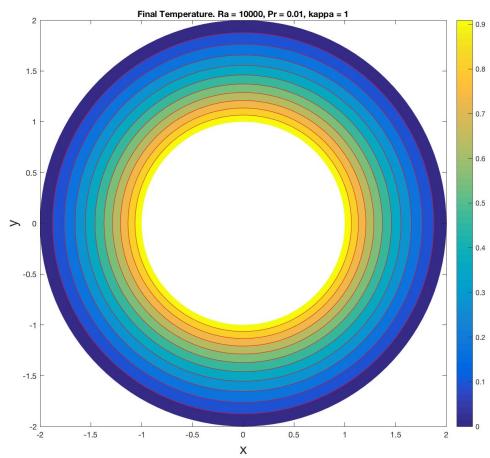


Figure 13: Q4: Final Temperature Field after solving the full problem for $P_r = 0.01$, $R_a = 10000$ and $b = 2$.

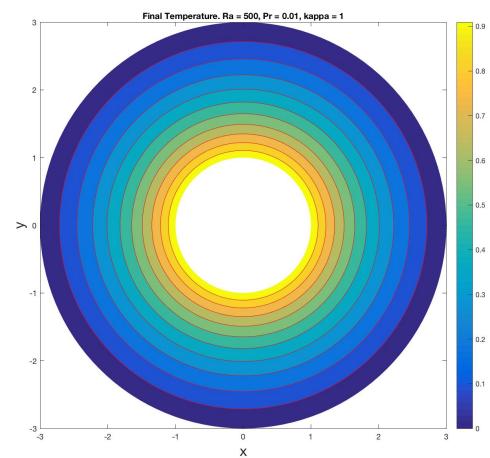


Figure 14: Q4: Final Temperature Field after solving the full problem for $P_r = 0.01$, $R_a = 500$ and $b = 3$.