Flow Across a Flat Plate: A Numerical Analysis

University of Colorado at Boulder

APPM 4650

Intermediate Numerical Analysis 1

Authors:	*Affiliations:*
Nathan Biesterfeld	Applied Mathematics
Cooper Simpson	Applied Mathematics
Jaden Wang	Applied Mathematics

July 16, 2020

Contents

1 Introduction:

We consider laminar fluid flow across a horizontal plate in a two dimensional plane. Our study of this situation will be carried out numerically, and we will specifically investigate the temperature and velocity of the fluid as it passes over the plate. We will use a set of coupled partial differential equations (PDE's) known as the "boundary layer equations" to examine the thickness of the so-called momentum and thermal boundary layer.

We begin by detailing the mathematical model and setup for this problem, although we note that this paper is more concerned with the numerical analysis and less so with the derivations and analytic analysis. Next, we show how we can transform our coupled set of PDE's into a coupled system of ordinary differential equations (ODE's). Finally, we solve this coupled system numerically and investigate our solutions to inform our boundary layer thickness discussions.

2 Mathematical Model:

We consider a fluid with temperature T_∞ and horizontal velocity U_∞ that flows across a horizontal plate in the x-y plane. The plate is assumed to maintain a constant temperature T_w and is stationary. We consider the upper half plane (above the plate) and will denote the temperature distribution of the fluid as $T(x, y)$, the horizontal velocity as $u(x, y)$, and the vertical velocity as $v(x, y)$. The figure below provides a visualization of a portion of the problem:

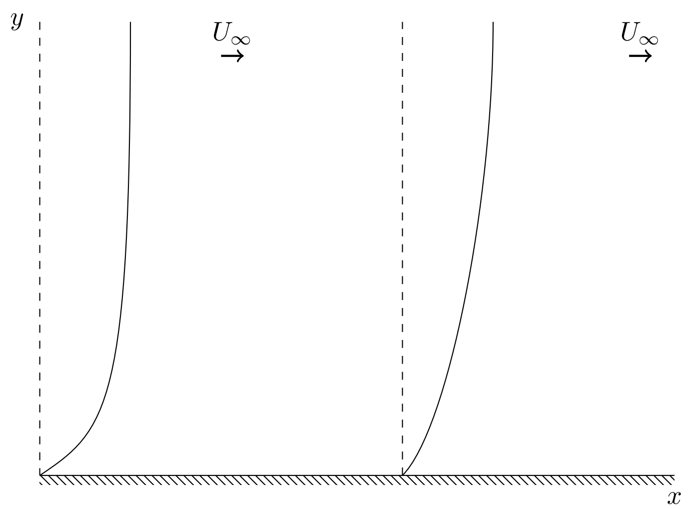

Figure 1: A schematic illustration of the momentum boundary layer problem. The liquid moves horizontally to the right with uniform speed U_∞, makes contact with the lamina at $(x, y) = (0, 0)$, and is subsequently slowed down by the lamina near the region where they contact. The lag in speed gradually propagates toward the far field as shown by the right half of the figure. The illustration of thermal boundary layer problem is similar and therefore omitted.

Given our definitions for u, v, and T, boundary layer equations describing conservation of momentum and energy are given below:

$$
\begin{aligned}
&u_x + v_y = 0 \\
&uu_x + vu_y = \nu u_{yy} \\
&uT_x + vT_y = \alpha T_{yy}
\end{aligned} \tag{1}
$$

The constants ν and α denote the kinematic viscosity and the thermal diffusivity, respectively, of the fluid. Given that the fluid is at free stream velocity and temperature before encountering the plate, we have the following x-dimension boundary conditions.

$$
T(0,y) = T_\infty, \qquad u(0,y) = U_\infty, \qquad v(0,y) = 0 \tag{2}
$$

In the y-dimension, we have the following boundary conditions. The first set require the fluid near the plate to stick to the plate and maintain the same temperature and the second set require the fluid be at free stream velocity and temperature far from the plate.

$$
\begin{aligned}
&T(x,0) = T_w, \qquad u(x,0) = 0, \qquad v(x,0) = 0 \\
&T(x,\infty) = T_\infty, \qquad u(x,\infty) = U_\infty
\end{aligned} \tag{3}
$$

In order to solve this problem we transform our coupled system. To do this, we introduce a similarity transform defined below.

$$
\text{Similarity variable: } \eta = \frac{y}{\sqrt{\frac{\nu x}{U_\infty}}} \tag{4}
$$

$$
\begin{aligned}
&u = U_\infty F'(\eta) \\
&v = \frac{1}{2}\sqrt{\frac{\nu U_\infty}{x}}[\eta F'(\eta) - F(\eta)]
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
&\text{Prandtl Number: } Pr = \frac{\nu}{\alpha} \\
&G(\eta, Pr) = \frac{T - T_\infty}{T_w - T_\infty}
\end{aligned} \tag{6}
$$

Note that the Prandtl number Pr describes the trade-off of viscosity and thermal diffusivity in the fluid. Applying the transform in (4)–(6) to (1), the following coupled set of ODE's can be obtained.

$$
F''' + \frac{1}{2}FF'' = 0 \tag{7}
$$

$$
G'' + \frac{Pr}{2}FG' = 0 \tag{8}
$$

We note that G is dependent on F, while F is independent of G.[1] This tells us that the fluid velocity mechanics (defined only in terms of F) are independent of the temperature mechanics

[1] For the rest of this paper, F is the shorthand for $F(\eta)$, and G is the shorthand for $G(\eta, Pr)$. The prime notation denotes the derivative of the function with respective to η.

(defined in terms of G). On the other hand, the temperature mechanics are dependent on the velocity of the fluid. Intuitively this makes sense as temperature really is just a macroscopic manifestation of kinematics. This one-sided independence will be useful in solving the problem.

To further reduce the complexity of the problem, we convert the two higher-order ODEs to a system of first-order ODEs. Let $y_1 = F$, $y_2 = F'$, $y_3 = F''$, $y_3' = F''' = -\frac{1}{2}FF''$, $y_4 = G$, $y_5 = G'$, and $y_5' = G'' = -\frac{Pr}{2}FG'$. This gives us

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ y_3' &= -\frac{1}{2}y_1y_3 \\ y_4' &= y_5 \\ y_5' &= -\frac{Pr}{2}y_1y_5 \end{aligned} \tag{9}$$

The last piece of the transformation are the constraints, with (4)–(6) transforming (3) to three initial conditions and two boundary conditions.

$$\begin{aligned} y_1(0) &= F(0) = 0 \\ y_2(0) &= F'(0) = 0 \\ y_3(0) &= F'(\infty) = 1 \\ y_4(\infty) &= G(0) = 1 \\ y_5(\infty) &= G(\infty) = 0 \end{aligned} \tag{10}$$

We have obtained our fully defined boundary-value problem in terms of a partially coupled system of ODEs subject to the initial and boundary conditions described by (10). We will now move on to solving this system and investigating what it reveals about the physical properties of the fluid flow.

3 Results:

3.1 Numerical Solution:

We desire to solve the system defined by (9) and (10) using the iterative Runga-Kutta fourth-order (RK4) method, but (10) only specifies initial conditions for y_1, y_2, and y_4, specifying boundary conditions in place of initial conditions for y_3 and y_5. To get around this, we employ the so-called "garden hose method". In this method, instead of applying $F'(\infty) = 1$, we assume that $F''(0)$ is equal to some initial guess and adjust the value of the guess until it is the case that the condition $F'(\infty) = 1$ is met. Since all functions involved represent macroscopic physical processes and hence are considered continuous and infinitely differentiable, we know that the boundary-value problems have unique solutions. This fact guarantees that the garden hose method will work. In practice, this method was implemented by writing a function in MATLAB that does RK4 on the first three equations in (9) with $\Delta\eta = 0.1$ on the domain [0,10]. Initial bounds on the true value of $F''(0)$ were set at 0 and 1 – these bounds were refined by iteratively running the RK4 function to converge on a single value. These iterations were continued until the bounds were less than 10^{-12} apart. Note that F asymptotes in its end behavior, so $\eta = 10$ was sufficient for approximating the boundary conditions. This yielded (to six significant figures):

$$F''(0) = 0.332057$$

Likewise, for solving the full five-equation coupled system including G we applied a very similar procedure as for finding $F''(0)$, this time using RK4 with the five coupled equations and initially bounding G'(0) between -1 and 0 (because we know temperature decreases as we move away from the plate). Given we don't specify a particular fluid, we ran the garden hose method for a representative set of $Pr = 0.1$, 0.2, 0.5, 1, 2, 5, and 10, yielding the following results:

Pr	0.1	0.2	0.5	1	2	5	10
$G'(0)$	-0.147182	-0.185210	-0.259298	-0.332057	-0.422308	-0.576689	-0.728141

Table 1: $G'(0)$ given different Prandtl numbers.

Note that in order to apply the RK4 method on the system defined by (9) to solve for $G'(0)$, we needed to use the previously found value for $F''(0)$. This underlines the importance of F being independent of G. Had F had G dependence, we'd have needed to loop through many $(F''(0), G'(0))$ pairs to find values that sufficiently meet the boundary conditions. This would have been significantly more time-intensive given each pair requires a 100-step RK4 on five coupled equations.

Next, we plugged our values for $F''(0)$ and $G'(0)$ for Pr=5 into the RK4 five-equation method with $\Delta\eta = 0.1$ to get approximations for F, F', F'', G, and G' over [0,10]. Numerical results are only shown for the first and last few steps. Note that each Prandtl number in Table 1 gives us different values of G and G' at each step, but these other tables have been omitted for simplicity.

η	F	F'	F''	G	G'
0.0	0.000000	0.000000	0.332057	1.000000	-0.576689
0.1	0.001660	0.033206	0.332048	0.942333	-0.576609
0.2	0.006641	0.066408	0.331984	0.884694	-0.576051
0.3	0.014941	0.099599	0.331809	0.827155	-0.574539
0.4	0.026560	0.132764	0.331470	0.769834	-0.571605
0.5	0.041493	0.165885	0.330911	0.712896	-0.566803
...	...	...	...	...	...
9.5	7.779212	1.0	6.292236e-08	1.022145e-15	-2.834154e-30
9.6	7.879212	1.0	4.254435e-08	1.022145e-15	-9.068577e-31
9.7	7.979212	1.0	2.862270e-08	1.022145e-15	-2.970544e-31
9.8	8.079212	1.0	1.916070e-08	1.022145e-15	-9.974141e-32
9.9	8.179212	1.0	1.276275e-08	1.022145e-15	-3.437105e-32
10.0	8.279212	1.0	8.458809e-09	1.022145e-15	-1.217012e-32

Table 2: Numerical values for transformed coupled equations with $Pr = 5$.

In the table above we see the numerical solution for our coupled system of ODE's (7). We note that the table clearly shows that $F'(\infty) \to 1$ and $G(\infty) \to 0$ as expected.

3.2 Boundary Locations:

With the solution to (7), we must define and identify the thickness of the momentum and thermal boundary layers. These values are defined as the vertical distance required for the fluid to return to its free stream (far field) values. Here we assume that the edges of the momentum and thermal boundary layers are represented by $F'(\eta_m) = \frac{U}{U_\infty} = 0.95$ and $G(\eta_t, Pr) = \frac{T-T_\infty}{T_w-T_\infty} = 0.01$, respectively. Since the step size we use is relatively crude, we do not obtain the values of η_m or η_t that make $F'(\eta_m)$ exactly 0.95 or $G(\eta_t, Pr)$ exactly 0.01 from Table 2 alone. In order to obtain a more accurate value for the thickness values η_m and η_t, we identify the η values closest to our target value and apply Neville's interpolation method using 4 points before and after the this edge value. Specifically, we use the function output of F' or G as the independent variable in Neville's method and the corresponding η value as the dependent variable. We are allowed to do this because both F' and G are monotonic functions, guaranteeing that their inverses exist. The interpolation gives us the following momentum boundary layer thickness (to three decimal places):

$$\eta_m = 3.918$$

Since G and G' are functions of the Prandtl number, η_t is also a function of the Prandtl number Pr. The Neville's interpolated values for η_t are presented in the table below (again to three decimals):

Pr	0.1	0.2	0.5	1	2	5	10
η_t	9.638	8.886	6.437	4.910	3.799	2.750	2.169

Table 3: Refined η_t given different Pr.

In the plot below, we see that η_t decreases roughly exponentially as the Prandtl number increases. This means that the thermal boundary layer thickness decreases with increasing Prandtl number. We note that for a given value of x the similarity variable η and y (our physical coordinate) are proportional. Thus increasing η implies increasing y and vice-versa, so while we are not yet looking at y specifically, the same relationships hold.

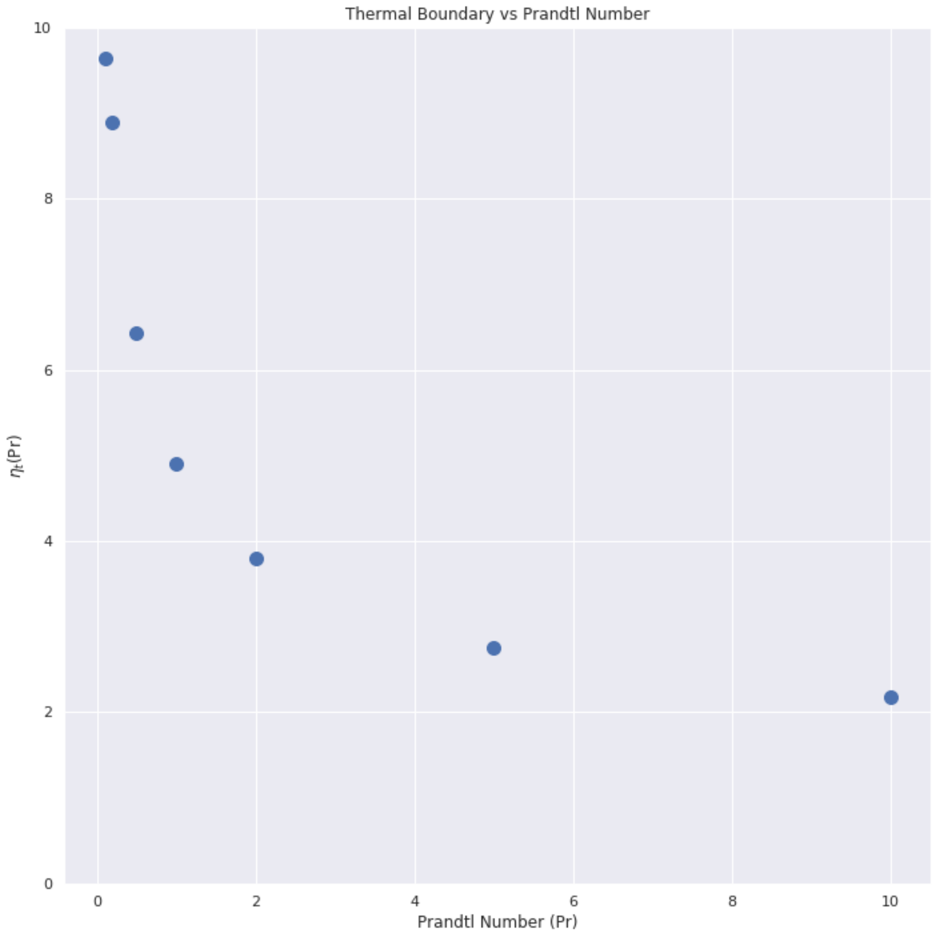

Figure 2: Values of η_t given different Prandtl numbers.

3.3 Velocity Profile:

Having solved our coupled system of equations numerically and refined our boundary locations, we next investigate the evolution of the horizontal and vertical velocities of our fluid. We note that initially the fluid does not exhibit any vertical velocity component, but that one is induced by the interactions between the plate and the fluid. We desire to nondimensionalize our velocities, so we consider the following expressions noting that $Re = \frac{U_\infty L}{\nu}$ is the Reynolds number.

$$\frac{v}{U_\infty}\sqrt{\frac{x}{L}}\sqrt{Re} = \frac{1}{2}[\eta F'(\eta) - F(\eta)] \tag{11}$$

$$\frac{u}{U_\infty} = F'(\eta) \tag{12}$$

It is clear to see that the second of the two equations is dimensionless and represents the ratio of the actual horizontal velocity to its far field value. Plugging Reynolds number into the first equation we see the following:

$$\frac{v}{U_\infty}\sqrt{\frac{x}{L}}\sqrt{\frac{U_\infty L}{\nu}} = v\sqrt{\frac{x}{U_\infty \nu}}$$

Noting that the kinematic viscosity ν has units of $\frac{m^2}{s}$ the above equation results in the following:

$$\frac{m}{s}\sqrt{m\frac{s}{m}\frac{s}{m^2}} = 1$$

Thus the second equation is clearly dimensionless. Below we see a plot of these two dimensionless groupings on the interval of η we have integrated over.

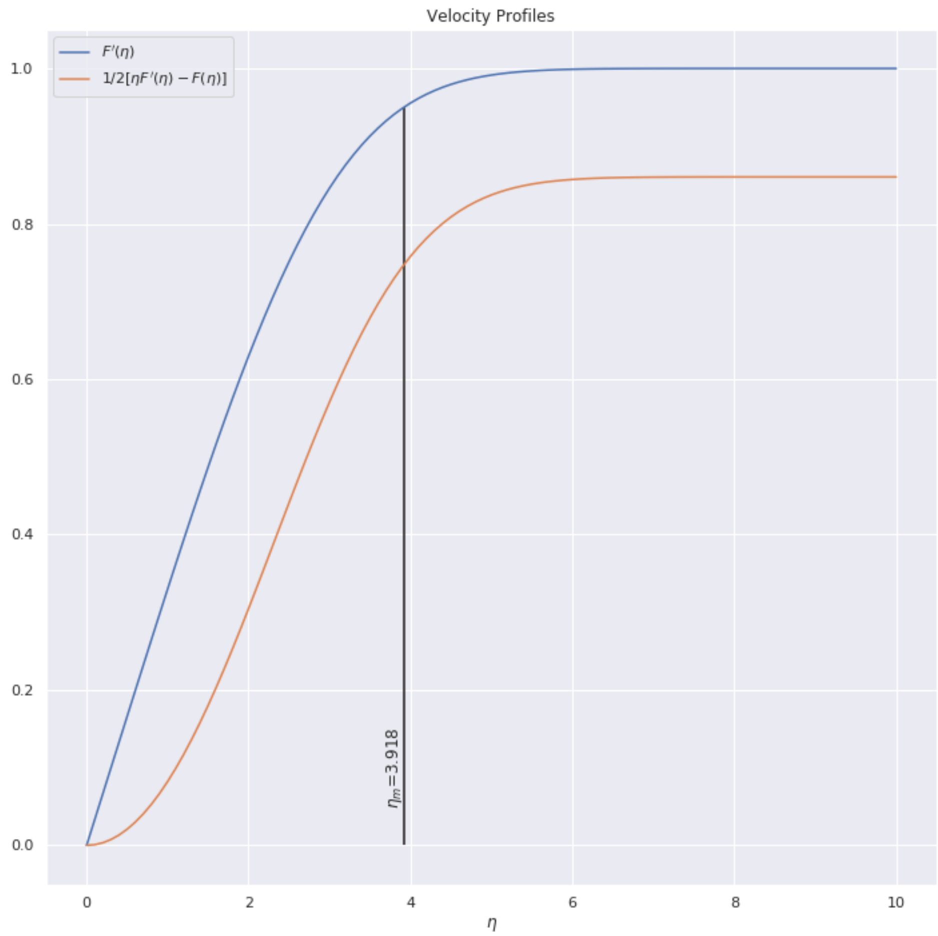

Figure 3: Dimensionless vertical and horizontal velocities.

In the above plot we see our nondimensionalized vertical and horizontal velocity profiles in orange and blue respectively. The vertical speed lags behind the horizontal speed as it is induced by the interactions between the horizontally slowed fluid near the plate and fluid with constant speed in the far field. In addition, we see a vertical black line associated with the blue curve labeled as $\eta_m = 3.918$. This indicates the thickness of the momentum boundary layer where $\frac{u}{U_\infty} = 0.95$, meaning that the horizontal velocity is 95% of its far field value.

3.4 Temperature Profile:

As with the velocity profiles we are also interested in the dynamics of the change in temperature. Again we look at a nondimensionalized form of the temperature which conveniently is already taken care of in the form of $G(\eta, Pr)$. We note that because G is also a function of the Prandtl number, we have multiple curves depending on this value.

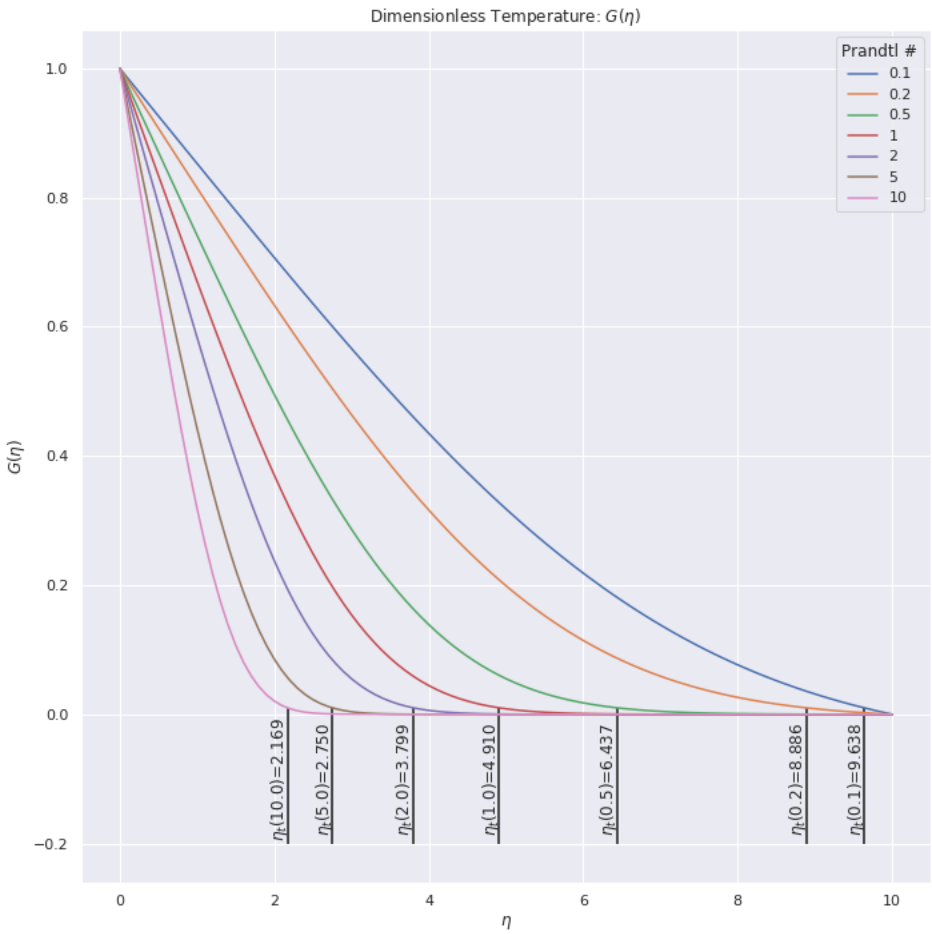

Figure 4: The values of $G(\eta)$ given different Prandtl numbers.

We see a variety of similar curves in the plot above, each of our dimensionless temperature G for different Prandtl numbers. On each curve we have also indicated the η_t value corresponding to the thickness of the thermal boundary layer.

3.5 Boundary Layer Thickness:

We are interested in examining the growth of both the momentum and thermal boundary layers along the plate. The way we have defined our similarity transform in (4) gives us a relation between η, x, and y. With accurate η values for the momentum and thermal boundaries found in 3.2 we can use this relation to define a function for the boundary layer height (thickness) in terms of the position on the plate. We define the following functions noting that η_t is also a function of the Prandtl number Pr.:

$$\delta_m(x) = y(x) = \eta_m \sqrt{\frac{\nu x}{U_\infty}} \tag{13}$$

$$\delta_t(x) = y(x) = \eta_t \sqrt{\frac{\nu x}{U_\infty}} \tag{14}$$

The figure below indicates the physical interpretation of $\delta_m(x)$. The interpretation of $\delta_t(x)$ is similar and therefore omitted:

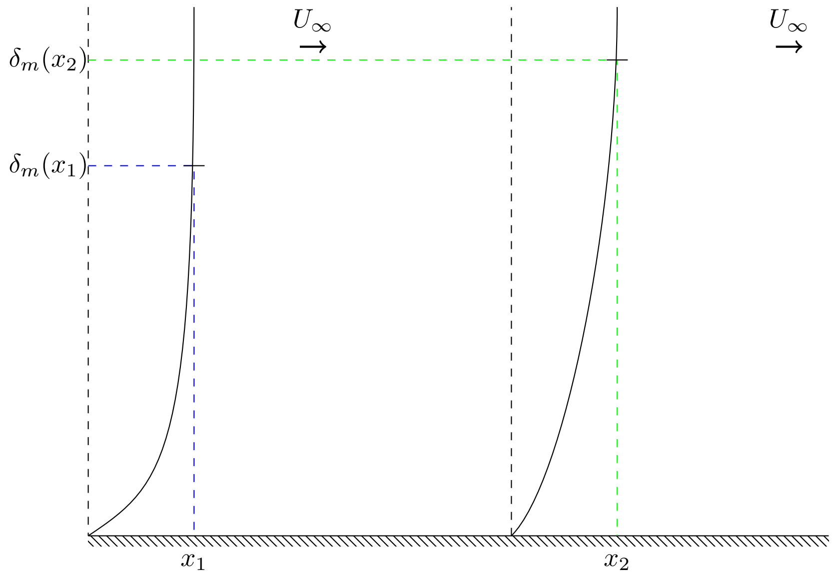

Figure 5: The region with decreased speed is referred to as the momentum boundary layer, and $\delta_m(x)$ is the function that represents its rising thickness as the liquid moves farther to the right.

Instead of looking at the functions δ_m and δ_t themselves, we will instead examine a dimensionless grouping as a function of the ratio of the distance across the plate to the length of the plate itself. We multiply our equations by a constant factor where we note that L is the length of the plate and $Re = \frac{U_\infty L}{\nu}$ is Reynolds number

$$\frac{\delta_m \sqrt{Re}}{L} = \eta_m \sqrt{\frac{x}{L}} \tag{15}$$

$$\frac{\delta_t \sqrt{Re}}{L} = \eta_t \sqrt{\frac{x}{L}} \tag{16}$$

Taking $\frac{x}{L} = z$ we can look at our boundary layer thicknesses along the entirety of the plate ($z \in [0, 1]$) where $z = 0$ indicates the leading edge of the plate and $z = 1$ indicates the far edge. Below we see plots of these dimensionless groupings for different Prandtl numbers where appropriate.

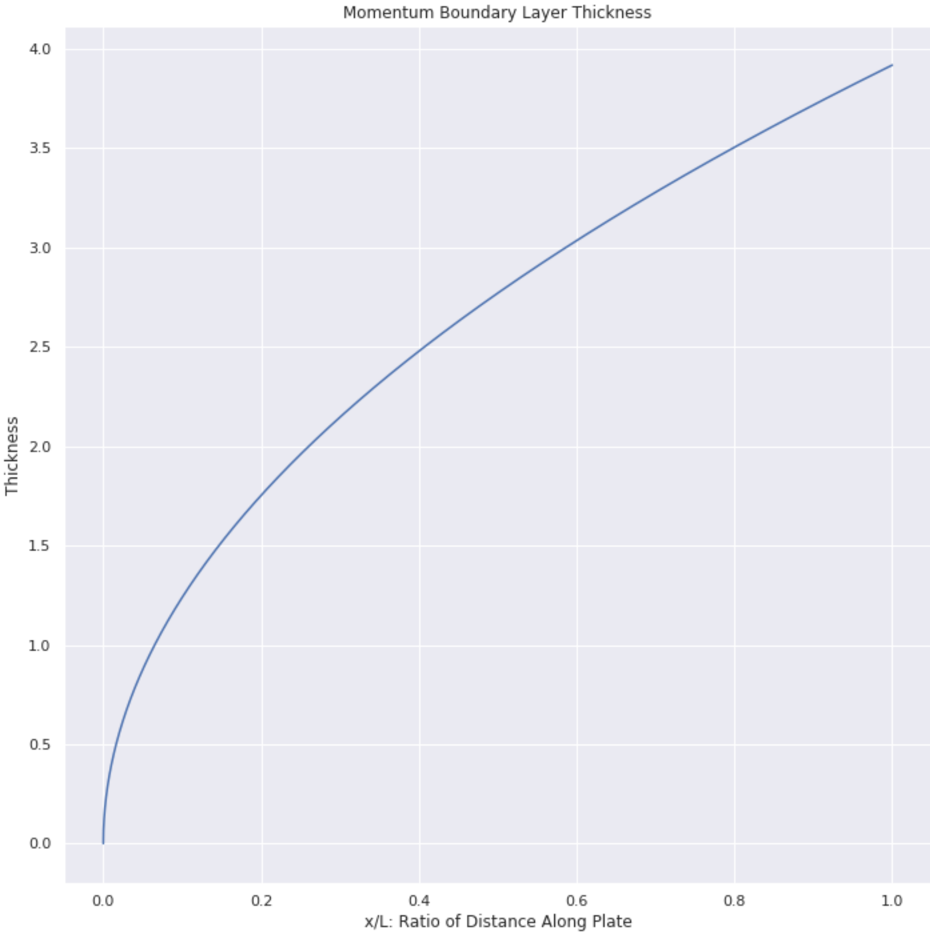

Figure 6: Momentum boundary layer thickness.

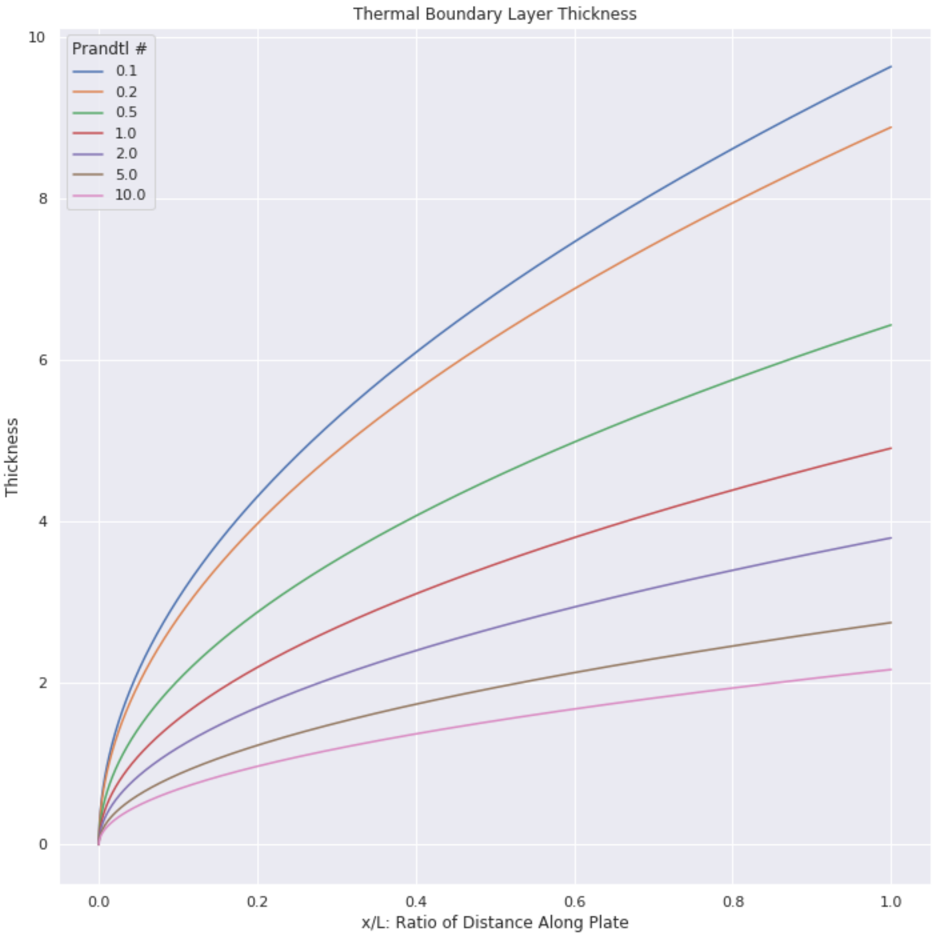

Figure 7: Thermal boundary layer thickness for various Prandtl numbers.

4 Discussion:

We have presented a mathematical model that describes the velocity and temperature dynamics of an arbitrary fluid with laminar flow across a flat plate. Using conservation of momentum and energy, a set of coupled PDE's can be derived that upon a similarity transformation yields a pair of partially coupled ODE's. Having numerically solved these equations, we then sought to examine the properties and evolution of the velocity and thermal boundary layers. We identified the thicknesses of our boundary layers and examined the velocity and temperature profiles all in the frame of our transformed coordinate system. We then used this to identify the exact evolution of these layers in the frame of our original Cartesian coordinates. The results presented in section 3 confirm several physical aspects of the problem and illuminate others as well.

It is worth noting that we choose to nondimensionalize the variables in our analysis. This approach yields two key benefits. First, since the original problem involves PDE's that are difficult to solve, nondimensionalization helps us consolidate similar variables into a single variable. For example, although the velocity and temperature depend on both x and y coordinates, by consolidating spatial information into one variable η, we are able to transform PDEs into ODEs and significantly reduce the mathematical complexity of the problem. Second, it also serves as a normalization process that enables us to plot arbitrary lengths easily and hence helps illustrating the relationship between variables for a non-specific problem.

Figure 3 shows the velocity profile for both the horizontal and vertical velocities of the fluid. From the problem statement, it may not be apparent why there is an induced vertical velocity as η increases. This vertical velocity is a result of conservation of momentum. The fluid encounters the plate at $x = 0$ with a horizontal free stream velocity of U_∞ and immediately starts sticking in place to the plate ($y = 0$). Since the horizontal velocity at the plate decreases from the initial free stream velocity, a velocity must be induced for the total momentum of the fluid to be conserved. Because the plate is preventing fluid from moving downwards, an induced upward velocity is necessary to satisfy conservation of momentum. Notably, we see that both components of the velocity asymptote for large values of η. Given that η is defined in terms of x and y, this means that for large values of y and small values of x (where y is not also small), the velocity is constant. For the horizontal velocity this shows that far above the plate or close to its leading edge at $x = 0$, the fluid is at its free stream horizontal velocity. This follows because our transformed equation that represents a nondimensionalized version of the horizontal velocity ($F'(\eta) = \frac{u}{U_\infty}$) asymptotes to 1. Furthermore, this agrees with the boundary conditions specified in section 2 because η approaches infinity as $x \to 0$ or $y \to \infty$. For the induced vertical velocity it is less immediately clear why it asymptotes to its particular value, or any value at all for that matter. We will not comment on why it achieves a specific value as it is beyond the scope of our investigation, but we will note that because the horizontal velocity is asymptotic the vertical velocity must also be. This is in order to maintain conservation of momentum because if the vertical velocity kept increasing this would mean the fluid was gaining momentum.

In (6), we defined the Prandtl number as the ratio of the viscosity and thermal diffusivity of a given fluid. In Figure 4 we see that for large Prandtl numbers the temperature of the fluid returns to its free stream value quickly and thus η_t is small. For small Prandtl numbers the opposite is true. A fluid with a large Prandtl number has a relatively large viscosity as compared to its thermal diffusivity. A small thermal diffusivity means heat doesn't diffuse very well, and thus doesn't diffuse well vertically. This means heat transfer from the plate to the liquid only covers a relatively small vertical distance into the liquid body. This results in a small vertical distance before the temperature has transitioned from T_w back to T_∞ and thus η_t – the boundary layer thickness – is small, relative to other Prandtl numbers. Conversely, consider a fluid with a small Prandtl number and therefore a relatively small viscosity as compared to the thermal diffusivity.

In this case, heat diffuses upwards well meaning heat has propagated from the plate far into the liquid. Therefore, there is a large vertical distance over which the temperature changes from T_w back to T_∞. This large vertical distance (large y) corresponds to a large η_t (larger boundary layer thickness) relative to other Prandtl numbers.

Figures 6 and 7 show the thermal and momentum boundary layer thicknesses scaled by a constant value ($\frac{\sqrt{Re}}{L}$) over the dimensionless distance ($\frac{x}{L}$) along the plate. As noted earlier this is useful when working with a problem such as ours that does not specify a fluid. The functions, which are $\mathcal{O}(\sqrt{x})$, increase most rapidly around $\frac{x}{L} = 0$ and increase at a slower rate for greater $\frac{x}{L}$. We first consider the thermal boundary layer shown in Figure 7. The explanation for why the layer grows faster for smaller x lies in the way heat diffuses up from the plate. At the leading corner of the plate the fluid is at its free stream temperature T_∞; however, just a small distance beyond this, where the fluid is in contact with the plate, it is now at temperature T_w. Because the boundary layer starts out at zero thickness, heat from the plate diffuses upwards and quickly changes the temperature of the fluid above it. Thus, the boundary layer increases quickly at first. Farther along the plate the boundary layer has grown and heat takes longer to diffuse all the way up to alter the temperature of the fluid. This is further evidenced by the variation in Figure 7 with respect to the Prandtl number. We see that the boundary layer increases fastest for the smallest Prandtl number. As discussed earlier, a small Prandtl number indicates a high thermal diffusivity, which means heat is able to quickly travel vertically through the fluid and alter its temperature. The lower the thermal diffusivity, the harder it is for heat to travel through the liquid and change the boundary layer. Importantly the plate is the only source of heat (or sink) and its temperature is kept constant. For distances far along the plate, the plate has to affect fluid very far away from itself which results in the slower growing boundary layer. Considering the momentum boundary layer in Figure 6 we see a similar argument applies for showing why the slope of the momentum boundary layer thickness decreases along the length of the plate. In this case it is the "sticking" effect that the plate has on the fluid instead of the diffusion of heat that causes the slow in growth. As the fluid just passes the leading corner of the plate the bottom layer of fluid near $y = 0$ begins to stick to the plate. This interaction has an immediate effect on the fluid vertically above it and so the boundary layer quickly begins to grow. However, farther along the plate where the boundary layer has already grown, the fluid interactions with the plate have less of an effect on the fluid much farther up near the end of the boundary layer. This means that the boundary layer grows more slowly as the driving force (the plate) has less of an effect farther from it.

5 References:

[BFB19] Richard L Burden, Douglas J Faires, and Annette M Burden. *Numerical Analysis*. 10th ed. Cengage, 2019.

6 Appendix:

6.1 Code:

```
%APPM 4650 Summer 2020 Project 1
clear all;
close all;
format long;

start = 0;
finish = 10;
h = 0.1;
x = [start:h:finish];

Pr_numbers = [0.1,0.2,0.5,1,2,5,10];
Pr_numbers_legend = [];

y3init = findy3init(@y1prime,@y2prime,@y3prime,0,0,0,1,0,10,0.1,1e-15);
fprintf('Using the garden hose method, y3(0) = %f \n', y3init);
[y1,y2,y3] = rk4three(@y1prime,@y2prime,@y3prime,0,0,y3init,start,finish,h);
[etaM,~] = findeta(x(find(y2 > 0.95,1)-3),x(find(y2 > 0.95,1)-2),x(find(y2 >
    0.95,1)-1),x(find(y2 > 0.95,1)), ...
    x(find(y2 > 0.95,1)+1),x(find(y2 > 0.95,1)+2),y2(find(y2 > 0.95,1)-3),y2(find(y2 >
        0.95,1)-2), ...
    y2(find(y2 > 0.95,1)-1),y2(find(y2 > 0.95,1)),y2(find(y2 > 0.95,1)+1),y2(find(y2 >
        0.95,1)+2),x(find(y2 > 0.95,1)-1), ...
    x(find(y2 > 0.95,1)),0.95,1e-12);
fprintf('The interpolated value for etaM is %f \n', etaM);

eta_thermals = [];
figure(1);
hold on;
for Pr = Pr_numbers
    y5init =
        findy5init(@y1prime,@y2prime,@y3prime,@y4prime,@y5prime,0,0,y3init,1,-1,0.5,Pr,start,finish,h,1e-15);
    fprintf('Using the garden hose method, y5(0) = %f for Pr = %f \n',y5init,Pr);

    [y1,y2,y3,y4,y5] =
        rk4five(@y1prime,@y2prime,@y3prime,@y4prime,@y5prime,0,0,y3init,1,y5init,Pr,start,finish,h);
```

```
    plot(x,y4);

    [etaT,fateta] = findeta(x(find(y4 < 0.01,1)-3),x(find(y4 < 0.01,1)-2),x(find(y4 <
        0.01,1)-1),x(find(y4 < 0.01,1)), ...
       x(find(y4 < 0.01,1)+1),x(find(y4 < 0.01,1)+2),y4(find(y4 < 0.01,1)-3),y4(find(y4 <
           0.01,1)-2), ...
       y4(find(y4 < 0.01,1)-1),y4(find(y4 < 0.01,1)),y4(find(y4 < 0.01,1)+1),y4(find(y4 <
           0.01,1)+2), ...
       x(find(y4 < 0.01,1)-1),x(find(y4 < 0.01,1)),0.01,1e-12);
    fprintf('For Pr = %f, the sixth-order Nevilles interpolated value for etaT is etaT =
        %f \n',Pr,etaT);
    plot([etaT,etaT],[-0.2,fateta]);
    eta_thermals = [eta_thermals, etaT];
    Pr_numbers_legend = [Pr_numbers_legend,compose('G(eta) for Pr = %f',Pr),compose('etaT
        for Pr = %f',Pr)];
end
title('G(\eta) vs. \eta for various Prandtl Numbers');
xlabel('\eta');
ylabel('G(\eta)');
legend(Pr_numbers_legend);
hold off;

figure(2);
hold on;
scatter(Pr_numbers,eta_thermals);
title('\eta Thermal vs. Pr number');
xlabel('Prandtl Number');
ylabel('\eta Thermal');
axis([0 max(Pr_numbers) 0 max(eta_thermals)]);
hold off;

y5init =
    findy5init(@y1prime,@y2prime,@y3prime,@y4prime,@y5prime,0,0,y3init,1,-1,1,5,start,finish,h,1e-15);
[y1,y2,y3,y4,y5] =
    rk4five(@y1prime,@y2prime,@y3prime,@y4prime,@y5prime,0,0,y3init,1,y5init,5,start,finish,h);
figure(3); %problem 2 for Pr = 5
hold on;
plot(start:h:finish,y2);
plot(start:h:finish,(1/2).*((start:h:finish).*y2 - y1));
```

```
plot([etaM,etaM],[y2(1),y2(end)]);
legend('Fprime(\eta)','(1/2)*(x*Fprime - F)','\eta Momentum');
title('Velocity Profiles');
xlabel('\eta');
hold off;

xoverL = 0:0.0001:10;
figure(4);
Pr_legend = [];
hold on;
for i = 1:1:length(Pr_numbers)
   plot(xoverL,eta_thermals(i).*sqrt(xoverL));
   Pr_legend = [Pr_legend,compose('Pr = %f',Pr_numbers(i))];
end
legend(Pr_legend);
xlabel('x/L');
ylabel('(\delta/L)*sqrt(Re)');
title('(deltaT/L)*sqrt(Re)');
hold off;

function [eta,fateta] =
    findeta(x0,x1,x2,x3,x4,x5,fx0,fx1,fx2,fx3,fx4,fx5,mineta,maxeta,target,tol)
   diff = 10; %distance between most recent interpolation value at eta and target
   increment = (maxeta - mineta)/10;
   while diff > tol
      fatmin = neville(x0,x1,x2,x3,x4,x5,fx0,fx1,fx2,fx3,fx4,fx5,mineta);
      fatmax = neville(x0,x1,x2,x3,x4,x5,fx0,fx1,fx2,fx3,fx4,fx5,maxeta);
      prod = (target-fatmin)*(target-fatmax);
      assert(prod < 0);

      while (target-fatmin)*(target-fatmax) < 0 && mineta < maxeta
         fatmin = neville(x0,x1,x2,x3,x4,x5,fx0,fx1,fx2,fx3,fx4,fx5,mineta);
         mineta = mineta + increment;
      end
      mineta = mineta - 2*increment;
      fatmin = neville(x0,x1,x2,x3,x4,x5,fx0,fx1,fx2,fx3,fx4,fx5,mineta);
      while (target-fatmin)*(target-fatmax) < 0 && mineta < maxeta
         fatmax = neville(x0,x1,x2,x3,x4,x5,fx0,fx1,fx2,fx3,fx4,fx5,maxeta);
         maxeta = maxeta - increment;
```

```
        end
        maxeta = maxeta + 2*increment;
        fatmax = neville(x0,x1,x2,x3,x4,x5,fx0,fx1,fx2,fx3,fx4,fx5,maxeta);
        increment = increment / 10;

        diff = abs(target-((fatmax + fatmin)/2));
    end
    eta = (mineta + maxeta)/2;
    fateta = (fatmax + fatmin)/2;
end

function y = neville(x0,x1,x2,x3,x4,x5,fx0,fx1,fx2,fx3,fx4,fx5,x)
    Q00 = fx0;
    Q10 = fx1;
    Q20 = fx2;
    Q30 = fx3;
    Q40 = fx4;
    Q50 = fx5;

    Q11 = ((x-x0)*Q10 - (x-x1)*Q00)/(x1-x0);
    Q21 = ((x-x1)*Q20 - (x-x2)*Q10)/(x2-x1);
    Q31 = ((x-x2)*Q30 - (x-x3)*Q20)/(x3-x2);
    Q41 = ((x-x3)*Q40 - (x-x4)*Q30)/(x4-x3);
    Q51 = ((x-x4)*Q50 - (x-x5)*Q40)/(x5-x4);

    Q22 = ((x-x0)*Q21 - (x-x2)*Q11)/(x2-x0);
    Q32 = ((x-x1)*Q31 - (x-x3)*Q21)/(x3-x1);
    Q42 = ((x-x2)*Q41 - (x-x4)*Q31)/(x4-x2);
    Q52 = ((x-x3)*Q51 - (x-x5)*Q41)/(x5-x3);

    Q33 = ((x-x0)*Q32 - (x-x3)*Q22)/(x3-x0);
    Q43 = ((x-x1)*Q42 - (x-x4)*Q32)/(x4-x1);
    Q53 = ((x-x2)*Q52 - (x-x5)*Q42)/(x5-x2);

    Q44 = ((x-x0)*Q43 - (x-x4)*Q33)/(x4-x0);
    Q54 = ((x-x1)*Q53 - (x-x5)*Q43)/(x5-x1);

    Q55 = ((x-x0)*Q54 - (x-x5)*Q44)/(x5-x0);
    y = Q55;
```

```
end

function y3init =
    findy3init(y1prime,y2prime,y3prime,y1init,y2init,y3min,y3max,start,finish,h,tol)
    diff = 1; %distance between y2(last) and 1
    increment = (y3max - y3min)/10;
    while diff > tol
        [~,y2atmin,~] =
            rk4three(y1prime,y2prime,y3prime,y1init,y2init,y3min,start,finish,h);
        y2assmin = y2atmin(end);
        [~,y2atmax,~] =
            rk4three(y1prime,y2prime,y3prime,y1init,y2init,y3max,start,finish,h);
        y2assmax = y2atmax(end);
        prod = (1-y2assmin)*(1-y2assmax);
        assert(prod < 0);

        while (1-y2assmin)*(1-y2assmax) < 0 && y3min < y3max
            [~,y2atmin,~] =
                rk4three(y1prime,y2prime,y3prime,y1init,y2init,y3min,start,finish,h);
            y2assmin = y2atmin(end);
            y3min = y3min + increment;
        end
        y3min = y3min - 2*increment;
        [~,y2atmin,~] =
            rk4three(y1prime,y2prime,y3prime,y1init,y2init,y3min,start,finish,h);
        y2assmin = y2atmin(end);
        while (1-y2assmin)*(1-y2assmax) < 0 && y3min < y3max
            [~,y2atmax,~] =
                rk4three(y1prime,y2prime,y3prime,y1init,y2init,y3max,start,finish,h);
            y2assmax = y2atmax(end);
            y3max = y3max - increment;
        end
        y3max = y3max + 2*increment;
        increment = increment / 10;

        diff = abs(1-((y2assmax + y2assmin)/2));
    end
    y3init = (y3max + y3min)/2;
end
```

```
function [y1,y2,y3] =
    rk4three(y1prime,y2prime,y3prime,y1init,y2init,y3init,start,finish,h)
    y1 = y1init;
    y2 = y2init;
    y3 = y3init;

    for x = start:h:finish - h
        k11 = h*y1prime(x,y1(end),y2(end),y3(end));
        k12 = h*y2prime(x,y1(end),y2(end),y3(end));
        k13 = h*y3prime(x,y1(end),y2(end),y3(end));

        k21 = h*y1prime(x+(h/2),y1(end)+(k11/2),y2(end)+(k12/2),y3(end)+(k13/2));
        k22 = h*y2prime(x+(h/2),y1(end)+(k11/2),y2(end)+(k12/2),y3(end)+(k13/2));
        k23 = h*y3prime(x+(h/2),y1(end)+(k11/2),y2(end)+(k12/2),y3(end)+(k13/2));

        k31 = h*y1prime(x+(h/2),y1(end)+(k21/2),y2(end)+(k22/2),y3(end)+(k23/2));
        k32 = h*y2prime(x+(h/2),y1(end)+(k21/2),y2(end)+(k22/2),y3(end)+(k23/2));
        k33 = h*y3prime(x+(h/2),y1(end)+(k21/2),y2(end)+(k22/2),y3(end)+(k23/2));

        k41 = h*y1prime(x+h,y1(end)+k31,y2(end)+k32,y3(end)+k33);
        k42 = h*y2prime(x+h,y1(end)+k31,y2(end)+k32,y3(end)+k33);
        k43 = h*y3prime(x+h,y1(end)+k31,y2(end)+k32,y3(end)+k33);

        y1 = [y1, y1(end)+(k11+2*k21+2*k31+k41)/6];
        y2 = [y2, y2(end)+(k12+2*k22+2*k32+k42)/6];
        y3 = [y3, y3(end)+(k13+2*k23+2*k33+k43)/6];
    end
end

function y5init =
    findy5init(y1prime,y2prime,y3prime,y4prime,y5prime,y1init,y2init,y3init,y4init,y5min,y5max,...
r,start,finish,h,tol)
    diff = 1; %distance between y4(last) and 0
    increment = (y5max - y5min)/10;
    while diff > tol
        [~,~,~,y4atmin,~] =
            rk4five(y1prime,y2prime,y3prime,y4prime,y5prime,y1init,y2init,y3init,y4init,y5min,...
           Pr,start,finish,h);
```

```
        y4assmin = y4atmin(end);
        [~,~,~,y4atmax,~] =
            rk4five(y1prime,y2prime,y3prime,y4prime,y5prime,y1init,y2init,y3init,y4init,y5max,...
           Pr,start,finish,h);
        y4assmax = y4atmax(end);
        prod = y4assmin*y4assmax;
        assert(prod < 0);

        while y4assmin*y4assmax < 0 && y5min < y5max
            [~,~,~,y4atmin,~] =
                rk4five(y1prime,y2prime,y3prime,y4prime,y5prime,y1init,y2init,y3init,y4init,y5min,...
               Pr,start,finish,h);
            y4assmin = y4atmin(end);
            y5min = y5min + increment;
        end
        y5min = y5min - 2*increment;
        [~,~,~,y4atmin,~] =
            rk4five(y1prime,y2prime,y3prime,y4prime,y5prime,y1init,y2init,y3init,y4init,y5min,Pr,...
           start,finish,h);
        y4assmin = y4atmin(end);
        while y4assmin*y4assmax < 0 && y5min < y5max
            [~,~,~,y4atmax,~] =
                rk4five(y1prime,y2prime,y3prime,y4prime,y5prime,y1init,y2init,y3init,y4init,y5max,...
               Pr,start,finish,h);
            y4assmax = y4atmax(end);
            y5max = y5max - increment;
        end
        y5max = y5max + 2*increment;
        increment = increment / 10;

        diff = (abs(y4assmax) + abs(y4assmin))/2;
    end
    y5init = (y5max + y5min)/2;
end

function [y1,y2,y3,y4,y5] =
     rk4five(y1prime,y2prime,y3prime,y4prime,y5prime,y1init,y2init,y3init,y4init,y5init,Pr,...
    start,finish,h)
    y1 = y1init;
```

```
y2 = y2init;
y3 = y3init;
y4 = y4init;
y5 = y5init;

for x = start:h:finish - h
    k11 = h*y1prime(x,y1(end),y2(end),y3(end));
    k12 = h*y2prime(x,y1(end),y2(end),y3(end));
    k13 = h*y3prime(x,y1(end),y2(end),y3(end));
    k14 = h*y4prime(x,y1(end),y2(end),y3(end),y4(end),y5(end));
    k15 = h*y5prime(x,y1(end),y2(end),y3(end),y4(end),y5(end),Pr);

    k21 = h*y1prime(x+(h/2),y1(end)+(k11/2),y2(end)+(k12/2),y3(end)+(k13/2));
    k22 = h*y2prime(x+(h/2),y1(end)+(k11/2),y2(end)+(k12/2),y3(end)+(k13/2));
    k23 = h*y3prime(x+(h/2),y1(end)+(k11/2),y2(end)+(k12/2),y3(end)+(k13/2));
    k24 =
        h*y4prime(x+(h/2),y1(end)+(k11/2),y2(end)+(k12/2),y3(end)+(k13/2),y4(end)+(k14/2),y5(end)+(k15/2)
    k25 =
        h*y5prime(x+(h/2),y1(end)+(k11/2),y2(end)+(k12/2),y3(end)+(k13/2),y4(end)+(k14/2),y5(end)+(k15/2)

    k31 = h*y1prime(x+(h/2),y1(end)+(k21/2),y2(end)+(k22/2),y3(end)+(k23/2));
    k32 = h*y2prime(x+(h/2),y1(end)+(k21/2),y2(end)+(k22/2),y3(end)+(k23/2));
    k33 = h*y3prime(x+(h/2),y1(end)+(k21/2),y2(end)+(k22/2),y3(end)+(k23/2));
    k34 =
        h*y4prime(x+(h/2),y1(end)+(k21/2),y2(end)+(k22/2),y3(end)+(k23/2),y4(end)+(k24/2),y5(end)+(k25/2)
    k35 =
        h*y5prime(x+(h/2),y1(end)+(k21/2),y2(end)+(k22/2),y3(end)+(k23/2),y4(end)+(k24/2),y5(end)+(k25/2)

    k41 = h*y1prime(x+h,y1(end)+k31,y2(end)+k32,y3(end)+k33);
    k42 = h*y2prime(x+h,y1(end)+k31,y2(end)+k32,y3(end)+k33);
    k43 = h*y3prime(x+h,y1(end)+k31,y2(end)+k32,y3(end)+k33);
    k44 = h*y4prime(x+h,y1(end)+k31,y2(end)+k32,y3(end)+k33,y4(end)+k34,y5(end)+k35);
    k45 =
        h*y5prime(x+h,y1(end)+k31,y2(end)+k32,y3(end)+k33,y4(end)+k34,y5(end)+k35,Pr);

    y1 = [y1, y1(end)+(k11+2*k21+2*k31+k41)/6];
    y2 = [y2, y2(end)+(k12+2*k22+2*k32+k42)/6];
    y3 = [y3, y3(end)+(k13+2*k23+2*k33+k43)/6];
    y4 = [y4, y4(end)+(k14+2*k24+2*k34+k44)/6];
```

```
        y5 = [y5, y5(end)+(k15+2*k25+2*k35+k45)/6];
    end
end

function f = y1prime(x,y1,y2,y3)
    f = y2;
end

function f = y2prime(x,y1,y2,y3)
    f = y3;
end

function f = y3prime(x,y1,y2,y3)
    f = -(y1*y3)/2;
end

function f = y4prime(x,y1,y2,y3,y4,y5)
    f = y5;
end

function f = y5prime(x,y1,y2,y3,y4,y5,Pr)
    f = -(y1*y5*Pr)/2;
end
```