AE 6009

Computer Problem

Sijian Tan

2022.12.1

I. Introduction

Based on the discussion in the class, the similarity solution for Hiemenz flow could be expressed as by a non-dimensional stream function $F(\eta)$ satisfying the ordinary differential equation:

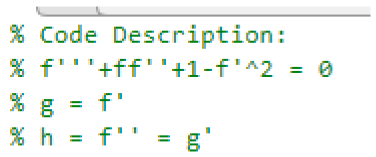
$$F''' + FF'' + 1 - F'^2 = 0$$

The objective of this project is to solve this equation using numerical methods (Matlab Code), show the results for F and its derivatives in both tabulated and graphical forms and explain the physical interpretations of the results.

The learning process of Fortran will also be shown at the end of the report.

II. Methods

For this project, a Matlab code is written to solve the equation, create the plot and save data into an excel file. In the code, different variables are defined to solve the problem:

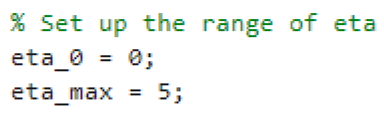


```
% Code Description:
% f'''+ff''+1-f'^2 = 0
% g = f'
% h = f'' = g'
```

The problem is known as a *two-point* boundary value problem because the boundary conditions are imposed at two different locations as shown below:

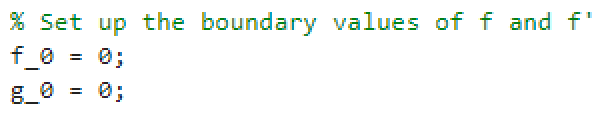
$$F(0) = 0, F'(0) = 0, F'(\infty) = 1$$

The goal is to use initial values of F and its derivatives for integration to get the values of $F'(\infty)$, check whether that value is close to 1, and finally choose the best initial value of F'' to make it satisfied. Based on the discussion from *White*, $\eta = 4.8$ is sufficient. In the code, 5 is chosen as the maximum value:



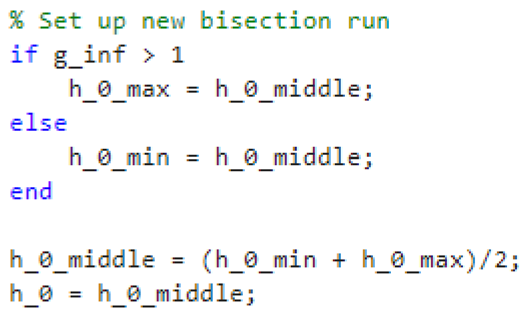
```
% Set up the range of eta
eta_0 = 0;
eta_max = 5;
```

The initial values of F, F' are set up as below:



```
% Set up the boundary values of f and f'
f_0 = 0;
g_0 = 0;
```

Bisection algorithm is used to determine the initial value of F'', as shown below:



```
% Set up new bisection run
if g_inf > 1
    h_0_max = h_0_middle;
else
    h_0_min = h_0_middle;
end

h_0_middle = (h_0_min + h_0_max)/2;
h_0 = h_0_middle;
```

If the estimated $F'(\infty)$ value is greater than 1, then the middle point will be the upper bound, otherwise the middle point will be the lower bound.

After setting up the initial values, the integration method starts. The algorithm is: firstly, transfer an ODE of order n into a system of n coupled first order equations, as shown below:

$$\frac{dF}{d\eta} = H$$

$$\frac{dH}{d\eta} = G$$

$$\frac{dG}{d\eta} = H^2 - FG - 1$$

Then, divide the range of integration into many smaller intervals, try to approximate the slope of the curve within each interval. Finally, apply the numerical integration with the initial values by "finding the area under a curve between prescribed end points". In this project, *fourth-order Runge-Kutta* method is chosen:

$$y_{n+1} = y_n + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 + O(h^5)$$

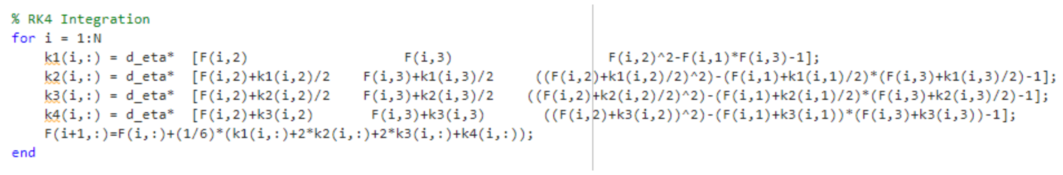
$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right)$$

$$k_3 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right)$$

$$k_4 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_3\right)$$

In the code, RK4 method is implemented as below:

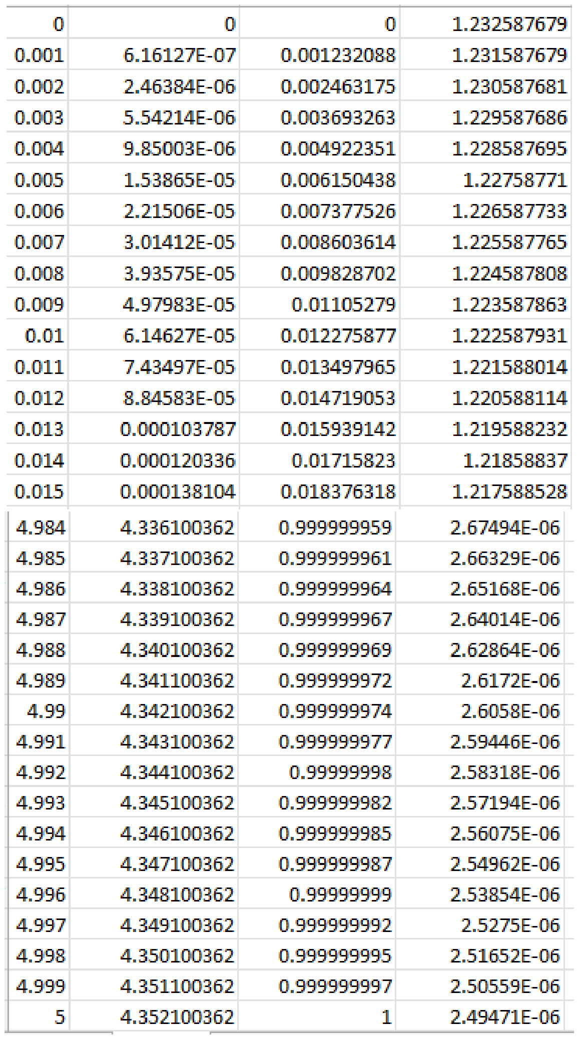


```
% RK4 Integration
for i = 1:N
    k1(i,:) = d_eta*  [F(i,2)                   F(i,3)                   F(i,2)^2-F(i,1)*F(i,3)-1];
    k2(i,:) = d_eta*  [F(i,2)+k1(i,2)/2    F(i,3)+k1(i,3)/2     ((F(i,2)+k1(i,2)/2)^2)-(F(i,1)+k1(i,1)/2)*(F(i,3)+k1(i,3)/2)-1];
    k3(i,:) = d_eta*  [F(i,2)+k2(i,2)/2    F(i,3)+k2(i,3)/2    ((F(i,2)+k2(i,2)/2)^2)-(F(i,1)+k2(i,1)/2)*(F(i,3)+k2(i,3)/2)-1];
    k4(i,:) = d_eta*  [F(i,2)+k3(i,2)       F(i,3)+k3(i,3)       ((F(i,2)+k3(i,2))^2)-(F(i,1)+k3(i,1))*(F(i,3)+k3(i,3))-1];
    F(i+1,:)=F(i,:)+(1/6)*(k1(i,:)+2*k2(i,:)+2*k3(i,:)+k4(i,:));
end
```

The error is calculated as the difference between 1 and the last value of F, and the iteration will last until the error is lower than the tolerance value, which is 10^{-10}.

III. Results and Discussion

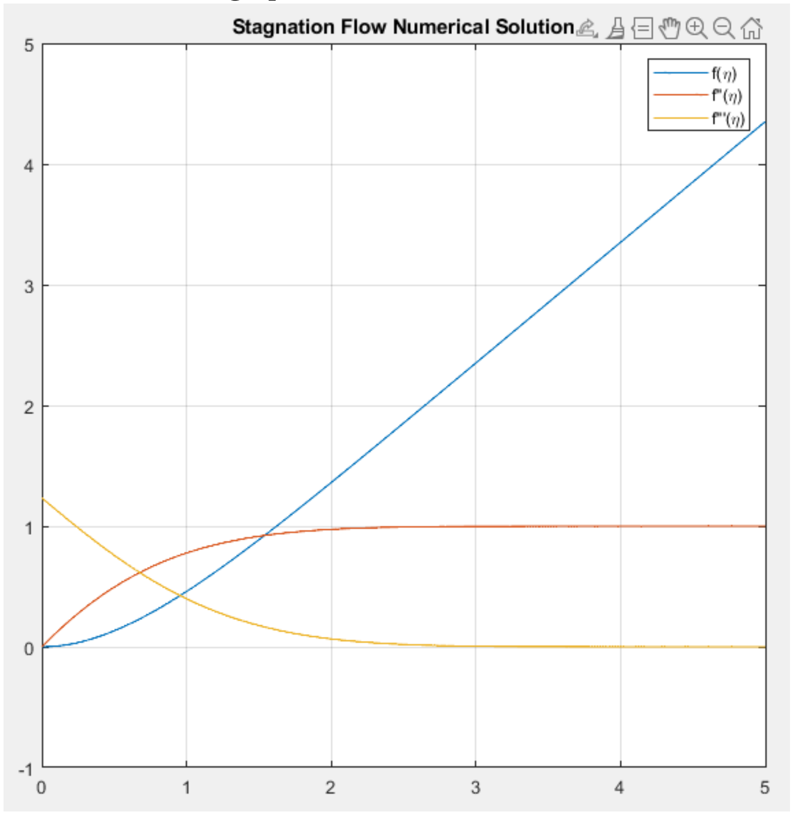
There are 2 deliverables for this project. The first one is the data tabulated in an excel file:



0	0	0	1.232587679
0.001	6.16127E-07	0.001232088	1.231587679
0.002	2.46384E-06	0.002463175	1.230587681
0.003	5.54214E-06	0.003693263	1.229587686
0.004	9.85003E-06	0.004922351	1.228587695
0.005	1.53865E-05	0.006150438	1.22758771
0.006	2.21506E-05	0.007377526	1.226587733
0.007	3.01412E-05	0.008603614	1.225587765
0.008	3.93575E-05	0.009828702	1.224587808
0.009	4.97983E-05	0.01105279	1.223587863
0.01	6.14627E-05	0.012275877	1.222587931
0.011	7.43497E-05	0.013497965	1.221588014
0.012	8.84583E-05	0.014719053	1.220588114
0.013	0.000103787	0.015939142	1.219588232
0.014	0.000120336	0.01715823	1.21858837
0.015	0.000138104	0.018376318	1.217588528
4.984	4.336100362	0.999999959	2.67494E-06
4.985	4.337100362	0.999999961	2.66329E-06
4.986	4.338100362	0.999999964	2.65168E-06
4.987	4.339100362	0.999999967	2.64014E-06
4.988	4.340100362	0.999999969	2.62864E-06
4.989	4.341100362	0.999999972	2.6172E-06
4.99	4.342100362	0.999999974	2.6058E-06
4.991	4.343100362	0.999999977	2.59446E-06
4.992	4.344100362	0.99999998	2.58318E-06
4.993	4.345100362	0.999999982	2.57194E-06
4.994	4.346100362	0.999999985	2.56075E-06
4.995	4.347100362	0.999999987	2.54962E-06
4.996	4.348100362	0.99999999	2.53854E-06
4.997	4.349100362	0.999999992	2.5275E-06
4.998	4.350100362	0.999999995	2.51652E-06
4.999	4.351100362	0.999999997	2.50559E-06
5	4.352100362	1	2.49471E-06

The first column is the value of eta, the second column is the value of F, the third column is the value of F' and the fourth column is the value of F''. Based on the final iteration of bisection method, the best guess value of $F''(0)$ is 1.2326, which can satisfy the requirements $F'(\infty) = 1$.

The other deliverable is the graph of F and its derivatives:



The graph visualizes the result pretty well, when eta approaches to 5, F'' value is approaching to 1.

Physical interpretation:

1. Based on the discussion from class:

$$\eta = y\sqrt{B/\upsilon}$$

$$F(\eta) = \frac{\psi}{x\sqrt{B\upsilon}}$$

   So η is acting as a normalized “y” value and $F(\eta)$ here is acting as a normalized stream function.

2. In addition,

$$u = \frac{\partial\psi}{\partial y} = BxF'$$

$$v = -\frac{\partial\psi}{\partial x} = -\sqrt{B\upsilon}F$$

   Therefore, F' can be used to express the scale of u, and F could be used to express the scale of v. Therefore, at the wall situation ($\eta = 0$), due to no-slip assumption:

$$u = 0 \rightarrow F'(0) = 0$$

Due to the impermeability at the wall:

$$v = 0 \rightarrow F(0) = 0$$

When η is large enough, the flow is close to free stream. Based on the discussion in the class

$$u = Bx \rightarrow F'(\infty) = 1$$

3. For the term F'', it can be regarded as the derivative of u. When at the wall, this term is related to the wall shear stress:

$$\tau_w = \mu \frac{\partial u}{\partial y}$$

Actually in the lecture of boundary layers, the Blasius solution shows that:

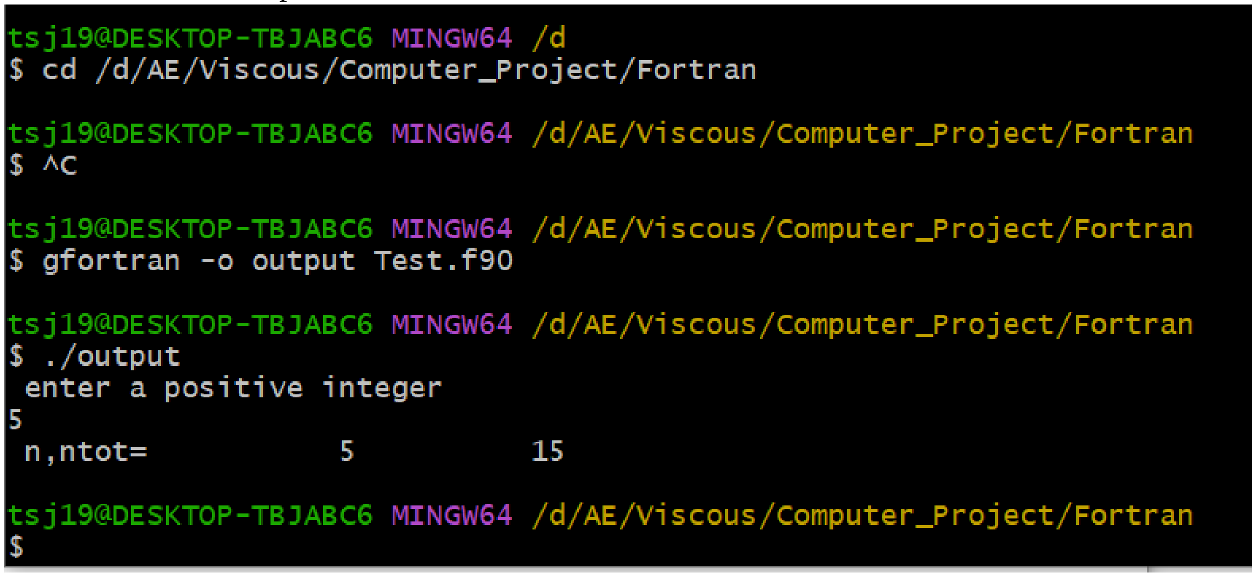
$$\tau_w = \mu \frac{\partial u}{\partial y}\Big|_{y=0} = \mu \left[\frac{\partial u}{\partial \eta}\frac{\partial \eta}{\partial y}\right]_{y=0} = \mu U_\infty f''(0)\sqrt{\frac{U_\infty}{\nu x}}$$

Which can be used to calculate the drag coefficients.

4. The trends in the graph match the physical interpretations. As η increases, u (F') keeps increasing until it reaches free stream velocity, and the wall shear stress (F'') keeps decreasing because it is getting further away from the wall.

IV. Learning Progress of Fortran

Installed Fortran compiler and also ran the test code:



```
tsj19@DESKTOP-TBJABC6 MINGW64 /d
$ cd /d/AE/Viscous/Computer_Project/Fortran

tsj19@DESKTOP-TBJABC6 MINGW64 /d/AE/Viscous/Computer_Project/Fortran
$ ^C

tsj19@DESKTOP-TBJABC6 MINGW64 /d/AE/Viscous/Computer_Project/Fortran
$ gfortran -o output Test.f90

tsj19@DESKTOP-TBJABC6 MINGW64 /d/AE/Viscous/Computer_Project/Fortran
$ ./output
 enter a positive integer
5
 n,ntot=           5          15

tsj19@DESKTOP-TBJABC6 MINGW64 /d/AE/Viscous/Computer_Project/Fortran
$
```

V. Appendix

Matlab Code:

```
% Code Description:
```

```
% f'''+ff''+1-f'^2 = 0
% g = f'
% h = f'' = g'

% Set up the range of eta
eta_0 = 0;
eta_max = 5;

% Set up the boundary values of f and f'
f_0 = 0;
g_0 = 0;

% Set up the initial value for bisection method
h_0_min = 1;
h_0_max = 2;
h_0_middle = (h_0_min + h_0_max)/2;
h_0 = h_0_middle;

% Set up the iteration pointer of RK4 integration
d_eta = 0.001;
N = (eta_max - eta_0)/ d_eta;
eta = linspace(eta_0, eta_max, N+1);

% Set up the initial value of error
err= 1;

% Iteration until error smaller than tolerance
while err > 10^-10

    % Include all derivatives into a vector
    F = zeros(N+1, 3);
    F(1,:) = [f_0 g_0 h_0];


    % RK4 Integration
    for i = 1:N
        k1(i,:) = d_eta*  [F(i,2)                   F(i,3)
F(i,2)^2-F(i,1)*F(i,3)-1];
        k2(i,:) = d_eta*  [F(i,2)+k1(i,2)/2    F(i,3)+k1(i,3)/2
((F(i,2)+k1(i,2)/2)^2)-(F(i,1)+k1(i,1)/2)*(F(i,3)+k1(i,3)/2)-1];
        k3(i,:) = d_eta*  [F(i,2)+k2(i,2)/2    F(i,3)+k2(i,3)/2
((F(i,2)+k2(i,2)/2)^2)-(F(i,1)+k2(i,1)/2)*(F(i,3)+k2(i,3)/2)-1];
        k4(i,:) = d_eta*  [F(i,2)+k3(i,2)       F(i,3)+k3(i,3)
((F(i,2)+k3(i,2))^2)-(F(i,1)+k3(i,1))*(F(i,3)+k3(i,3))-1];
        F(i+1,:)=F(i,:)+(1/6)*(k1(i,:)+2*k2(i,:)+2*k3(i,:)+k4(i,:));
    end

    % Get the value of f'(inf)
    g_inf = F(end,2);

    % Error judgment
    err = abs(1-g_inf);

    % Set up new bisection run
    if g_inf > 1
```

```
        h_0_max = h_0_middle;
    else
        h_0_min = h_0_middle;
    end

    h_0_middle = (h_0_min + h_0_max)/2;
    h_0 = h_0_middle;


end

% write data to excel
writematrix(F, "Output.xlsx")

%% plotting
f1=figure;
f1.Units = 'normalized';
f1.Position = [0.1 0.1 0.8 0.6];
plot(eta,F,'LineWidth',1);
xlim([0 5]);
ylim([-1 5]);
axis square
yticks(-1:1:9);
yticklabels({'-1','0','1','2','3','4','5','6','7','8','9'});
xticks(0:1:10);
xticklabels({'0','1','2','3','4','5','6','7','8','9','10'});
grid on
title('Stagnation Flow Numerical Solution');
xlabel('\eta');
legend('f(\eta)','f"(\eta)','f"''(\eta)');
```

F Table: (Not the complete table, complete one has 5000 rows due to high accuracy)

η	F	F’	F’’
0	0	0	1.2326
0.1	0.005996395	0.118264895	1.132830784
0.5	0.13358522	0.494649305	0.758307507
1	0.459227028	0.777865296	0.398012983
1.5	0.887329021	0.916168271	0.17695816
2	1.361974216	0.973216811	0.06582544
2.5	1.854428826	0.99285128	0.020227394
3	2.352556835	0.998424309	0.005078073
4	3.352109674	0.999958713	0.000168827
4.8	4.15210043	0.9999992	6.01967E-06
5	4.352100362	1	2.49471E-06