Friedrich-Alexander University of Erlangen

LSTM Institute of Fluid Mechanics

Dr.-Ing. M. Münsch, S. Saha

Report for Practical Sheet 2

Students: Franciskus Xaverius Erick, 22811754, franciskus.erick@fau.de
Serouj Djabraian, 22849126, serouj.d.djabraian@fau.de

Submitted: Erlangen, January 11th 2021

Table of Contents

Table of Figures

1.1 Discretized Boundary Layer Equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \frac{1}{Re}\frac{\partial^2 u}{\partial y^2} \tag{2}$$

a) Equation (1) is discretized with 2 different schemes. Backward Difference Scheme for the $\frac{\partial v}{\partial y}$ term and a modified Forward Difference Scheme for the $\frac{\partial u}{\partial x}$ term, whereby the average value between forward difference at point y = j+1 and at point y = j is determined. By using backward difference scheme, an iterative formula for v with other previously determined known values could be formed. The discretized equation is represented as follows.

$$\frac{1}{2}\left(\frac{\mathrm{u}_{i+1,j} - \mathrm{u}_{i,j}}{\Delta x} + \frac{\mathrm{u}_{i+1,j-1} - \mathrm{u}_{i,j-1}}{\Delta x}\right) + \frac{\mathrm{v}_{i+1,j} - \mathrm{v}_{i-1,j}}{\Delta y} = 0 \tag{3}$$

The iterative formula for v is shown as follows.

$$\mathrm{v}_{i+1,j} = \mathrm{v}_{i-1,j} - \frac{1}{2}\Delta y\left(\frac{\mathrm{u}_{i+1,j} - \mathrm{u}_{i,j}}{\Delta x} + \frac{\mathrm{u}_{i+1,j-1} - \mathrm{u}_{i,j-1}}{\Delta x}\right) + \tau_1 + \tau_2 \tag{4}$$

Whereby τ_1 is the error term of the order $O(\Delta x)$ from the modified forward difference scheme and τ_2 is the error term of the order $O(\Delta y)$ from the backward difference scheme.

b) For equation (2), the central difference scheme is used to discretize the partial derivatives with respect to y. The forward difference scheme is used to discretize the partial derivative term with respect to x. The discretized equation is represented as follows.

$$\mathrm{u}_{i,j}\frac{\mathrm{u}_{i+1,j}-\mathrm{u}_{i,j}}{\Delta x}+\frac{1}{2}\mathrm{v}_{i,j}\frac{\mathrm{u}_{i,j+1}-\mathrm{u}_{i,j-1}}{\Delta y}=\frac{1}{Re}\frac{\mathrm{u}_{i,j-1}-2\mathrm{u}_{i,j}+\mathrm{u}_{i,j+1}}{\Delta y^2} \tag{5}$$

By rearranging the equation, an iterative formula for the term $\mathrm{u}_{i,j+1}$ with respect to already known terms from x = i can thus be formed.

$$\mathrm{u}_{i+1,j}=\mathrm{u}_{i,j}+\frac{1}{Re}\frac{\Delta x}{\Delta y^2}\frac{\mathrm{u}_{i,j-1}-2\mathrm{u}_{i,j}+\mathrm{u}_{i,j+1}}{\mathrm{u}_{i,j}}-\frac{\Delta x}{\Delta y}\frac{\mathrm{v}_{i,j}}{\mathrm{u}_{i,j}}\frac{\mathrm{u}_{i,j+1}-\mathrm{u}_{i,j-1}}{2} \tag{6}$$

The chosen scheme is explicit as the next iterative term solely dependent on values from the current space step at x = i. The accuracy of the scheme is of the order 1 with the combined error term of the following order $(\Delta x^2)+O(\Delta x)+O(\Delta y)$. As this is an explicit scheme, it is conditionally stable with the following condition:

$$\Delta x \le \frac{1}{2}\,Re\mathrm{u}_{i,j}\Delta y^2$$

1.2 Results and Discussion

c) The Octave code for solving the boundary layer equation is attached in the appendix. The solution domain is x = [0,1] and y =[0,0.1]. Δx is taken to be 0.0005 (2000 discretization points in x) and Δy is taken to be 0.00125 (80 discretization points in y). These step sizes are taken to satisfy the stability condition as well as to achieve accurate result with reasonable computational time.

d) The numerical results of u and v of the whole domain is represented in the following figures.

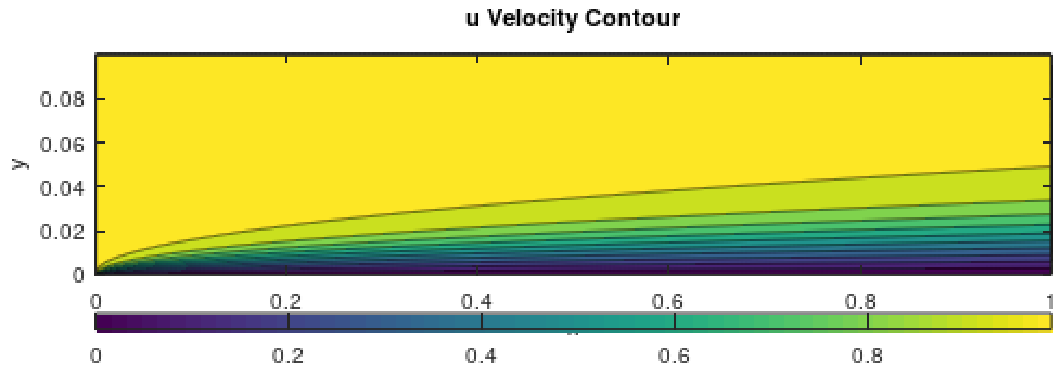

Figure 1: u velocity contour over the whole simulation domain

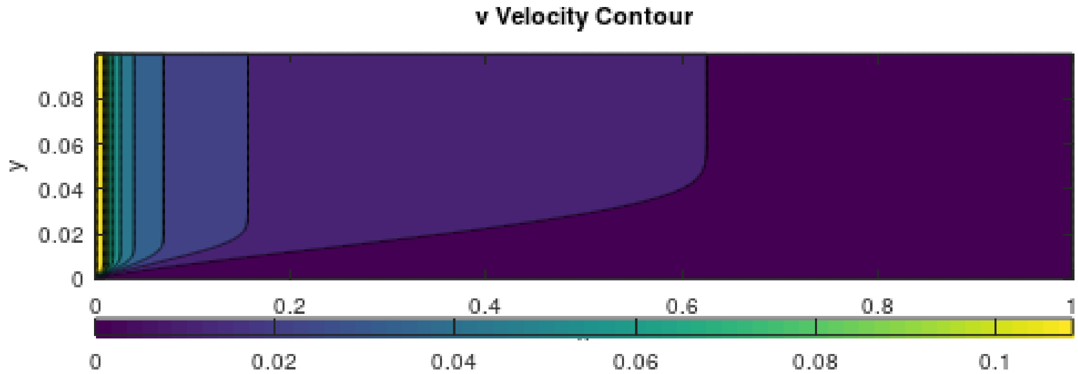

Figure 2 : v velocity contour over the whole simulation domain

From the numerical results of the u-velocity contour in figure 1, we can clearly observe the formation of the boundary layer over the flat plate, where the boundary layer thickness increases over x, reaching a maximum value of y=0.05 at x=1 which was already given in our problem statement. The thickness of the boundary layer seems to be similar to the displacement function of boundary layer, which is proportional to the square root of x. The values of the boundary conditions are also clearly illustrated in the figures. The value of u is 0 at the bottom of the plate, reflecting the boundary condition u = v = 0 at the bottom of the plate due to no slip condition. The value of u is 1 at the boundary, reflecting the fact that u reaches the free flow velocity at y values greater than the boundary layer. The u velocity contour is uniform across x, which is also expected as the u velocity profile is dependent on the similarity profile η.

For the v-velocity contour in figure 2, we can observe that the v-velocity is highest for only the very first nodes starting from the inlet, then all other values are subject to a dramatic decrease in the x-direction. This might be due to the change of u velocity component at the leading edge of the boundary layer. The boundary condition v = 0 due to no slip condition is also shown in the figure

e) The finite difference method is compared to the Runge Kutta Method. The following figures represent the u and v velocity profile against η as obtained from the RK4 Method.

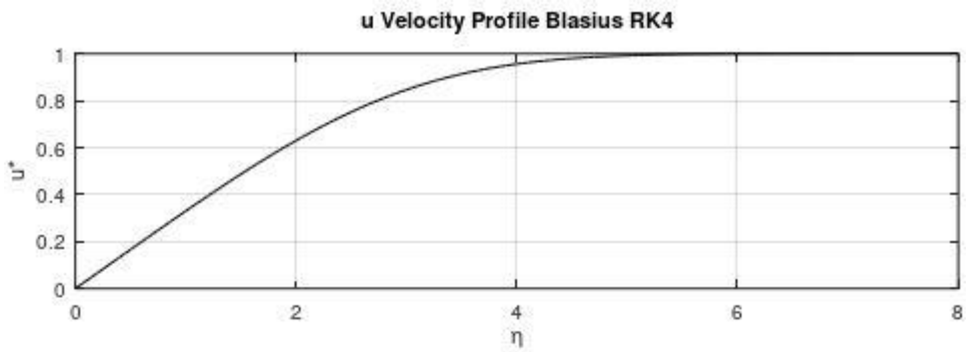

Figure 3 : u Velocity Profile obtained from RK4 method

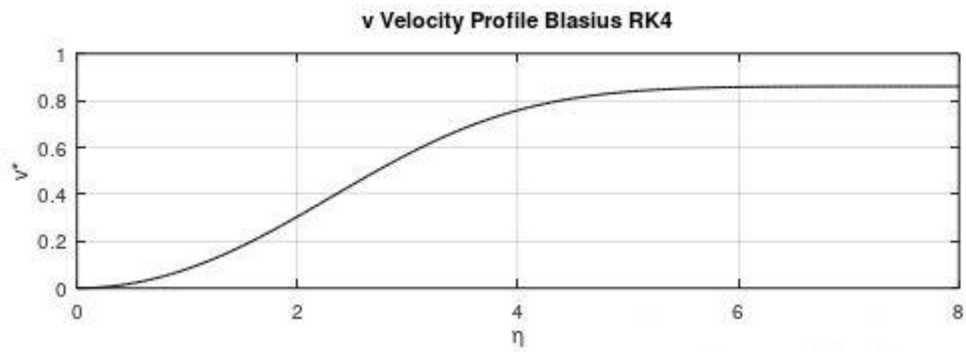

Figure 4: v Velocity Profile obtained from RK4 method

Similarly, the u and v velocity profiles from the finite difference method at x = 0.0005 and x = 0.5 are also plotted against η in the following figures.

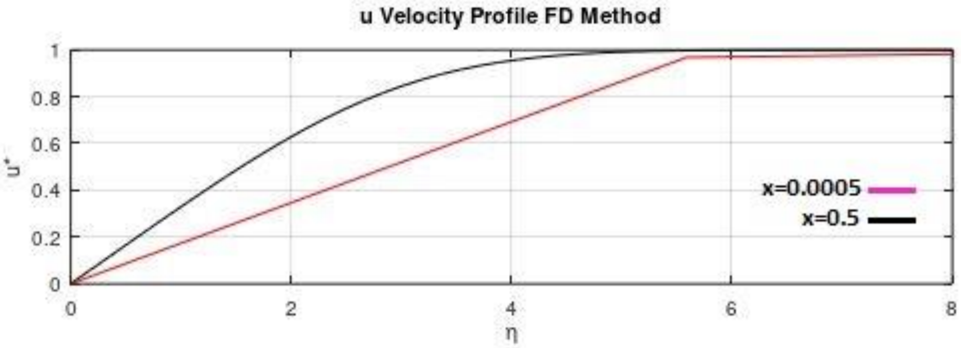

Figure 5 : u Velocity Profile from Finite Difference Method

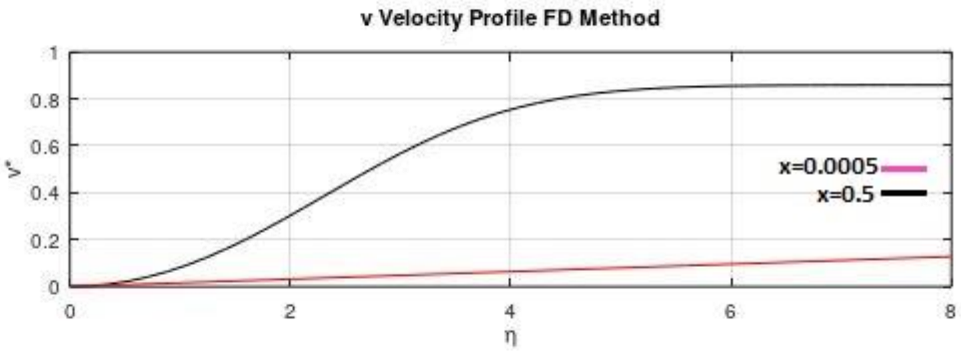

Figure 6 : v Velocity Profile from Finite Difference Method

At x = 0.5, there is a good agreement between both and u and v velocity profiles obtained from finite difference method and Runge Kutta Method. The value of u rises almost linearly and slowed down to an asymptote roughly at the value of η = 5 where the boundary layer is reached. u reaches the value of 1 as it reaches the free flow velocity. The v velocity profile has a shape similar to a sigmoid function until it roughly reaches the value of η = 5 where the velocity value of roughly 0.85 is reached. However, this is not the case at x = 0.0005, which is the first node at x near the leading edge as shown in the figures above. This might be due to the fact that the boundary layer at the leading edge is very thin, with δ being around the magnitude of Δx and Δy. As such, there are simply not enough nodes in the y direction where the velocity values are iteratively solved. Furthermore, the assumptions used to simplify the Navier-Stokes equations into equation(1) and equation(2) of the boundary layer equations are no longer valid at the leading edge. The change of u and v in the x-direction from the inlet into the leading region of the boundary layer is no longer negligible. This is also further proven by analyzing the u and v velocity profiles at increasing x values, where the velocity profiles from finite difference converge to the velocity profiles from the Runge-Kutta solution. We can conclude that the proposed numerical scheme is able to satisfactorily simulate the boundary layer equation, however the solutions from the leading edge are not valid.

Literature References

[1] M. Münsch, *Numerical Methods in Thermo-Fluid Dynamics I ; Simplification of Conservation Eqs. up to Convsistence, Stability and Convergence* , Numerical Methods in Thermo-Fluid Dynamics I, **2020**, Friedrich-Alexander Universität Erlangen, Microsoft Power Point Presentation

[2] M. Münsch, S. Saha, N. Chen, *Numerical Methods in Thermo-Fluid Dynamics I: Deliverable Task Sheet 2*, **2020**, Friedrich-Alexander Universität Erlangen,Practical Task Sheet.

[3] P. Lermusiaux, *Finite Difference*, 2.29 Numerical Fluid Mechanics, **2015**, Massachusetts Institute of Technology, Massachusetts, Lecture Notes, Retrieved from: https://ocw.mit.edu/courses/mechanical-engineering/2-29-numerical-fluid-mechanics-spring-2015/lecture-notes-and-references/MIT2_29S15_Lecture14.pdf

Appendix

```
clc
clear all
close all
%initializing dimensions for discretization in space
nx = 2001;                %number of nodes in x
ny = 81;                 %number of nodes in y
L = 1;                    %length
H = 0.1;                  %height
dx = L/(nx-1);
dy = H/(ny-1);
Re = 1e4;                 %Reynold's number
%initializing meshgrid
x = linspace(0,L,nx);
y = linspace(0,H,ny);
[X,Y]=meshgrid(x,y);

%% Initializing v and u vectors
v = zeros(ny,nx);
u = zeros(ny,nx);

%% Setting up boundary conditions
u(:,1) = 1; % u at inlet
v(:,1) = 0; % v at inlet
u(1,:) = 0; % No slip condition at the bottom
v(1,:) = 0; % No slip condition at the bottom
u(ny,:) = 1; % Free outer flow at the top

%% Numerical Solution using Finite Difference Method
for i = 1:nx-1
    %determining u(i+1,j) from the derived FD equation
    for j = 2:ny-1
        usol(j) = u(j,i) + dx/(Re*u(j,i)*(dy^2))*(u(j+1,i)-2*u(j,i) +u(j-
1,i)) - dx*v(j,i)/(2*dy*u(j,i))*(u(j+1,i)-u(j-1,i));
    end
    u(2:ny-1,i+1) = usol(2:end);
    % determining v from the derived FD equation
    for j = 2:ny
        v(j,i+1) = v(j-1,i+1) - dy/2/dx*(u(j,i+1)-u(j,i)+u(j-1,i+1)-u(j-
1,i));
    end
end

%% Numerical Solution of Blasius Equation Using Runge-Kutta
h = 0.05
f1 = @(y2) y2;
f2 = @(y3) y3;
f3 = @(y1, y3) -y1*y3/2;
eta = 0:h:10;
%initial conditions of f and f'. Best f'' initial guess determined by
Newton Raphson Method
y1(1) = 0;
y2(1) = 0;
y3(1) = 0.33230;
for i = 1:(length(eta)-1)
  a = h.*[f1(y2(i)), f2(y3(i)), f3(y1(i), y3(i))];
```

```
  b = h.*[f1(y2(i)+a(2)/2), f2(y3(i)+a(3)/2), f3( y1(i)+a(1)/2,
y3(i)+a(3)/2)];
  c = h.*[f1(y2(i)+b(2)/2), f2(y3(i)+b(3)/2), f3( y1(i)+b(1)/2,
y3(i)+b(3)/2)];
  d = h.*[f1(y2(i)+c(2)), f2(y3(i)+c(3)), f3(y1(i)+c(1), y3(i)+c(3))];
  y3(i+1) = y3(i)+ 1/6*(a(3)+2*b(3)+2*c(3)+d(3));
  y2(i+1) = y2(i)+ 1/6*(a(2)+2*b(2)+2*c(2)+d(2));
  y1(i+1) = y1(i)+ 1/6*(a(1)+2*b(1)+2*c(1)+d(1));
end

%% Plotting
%contour of u*
figure,
h1 = subplot(321);
set(h1,'XLim',[0 L],'YLim',[0 H]);
contourf(h1,X,Y,u, [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 0.99]);
colorbar('peer',h1,'SouthOutside');
title(h1,'u* Velocity Contour');
xlabel(h1,'x'),ylabel(h1,'y');
%contour of v*
h2 = subplot(322);
set(h2,'XLim',[0 L],'YLim',[0 H]);
contourf(h2,X,Y,v);
colorbar('peer',h2,'SouthOutside');
title(h2,'v* Velocity Contour');
xlabel(h2,'x'),ylabel(h2,'y ');

%u velocity profile FD at x = 0.5 and x = 0.0005
h3 = subplot(323);
%delta and eta at x = 0.5
delta_middle = 0.01 / sqrt(2);
eta_middle = Y(:,1001)/delta_middle;
%delta and eta at x = 0.0005
delta_lead = 0.01 / sqrt(2000);
eta_lead = Y(:,2)/delta_lead;
plot(eta_middle, u(:,1001),'-k', eta_lead, u(:,2),'-r');
xlim([0 8]);
grid on;
title(h3,'u Velocity Profile FD Method');
xlabel(h3,'{\eta}'),ylabel(h3,'u*[-]');

% v velocity profile FD at x = 0.5 and x = 0.0005
h4 = subplot(324);
plot(eta_middle, v(:,1001)/(delta_middle * 2),'-k', eta_lead,
v(:,2)/(delta_lead * 2000),'-r');
xlim([0 8]);
grid on;
title(h4,'v Velocity Profile FD Method');
xlabel(h4,'{\eta}'),ylabel(h4,'v*[-]');

% u velocity profile Blasius RK4
h5 = subplot(325);
u_blasius = y2;
plot( eta, u_blasius, '-k');
xlim([0 8]);
ylim([0 1]);
grid on;
title(h5,'u Velocity Profile Blasius RK4');
xlabel(h5,'{\eta}'),ylabel(h5,'u*[-]');
```

```
% v velocity profile Blasius RK4
h6 = subplot(326);
v_blasius = ( eta .* y2 - y1 )/2;
plot( eta, v_blasius, '-k');
xlim([0 8]);
grid on;
title(h6,'v Velocity Profile Blasius RK4');
xlabel(h6,'{\eta}'),ylabel(h6,'v*[-]');
```