Table of Contents


```
function Final()
```

Final

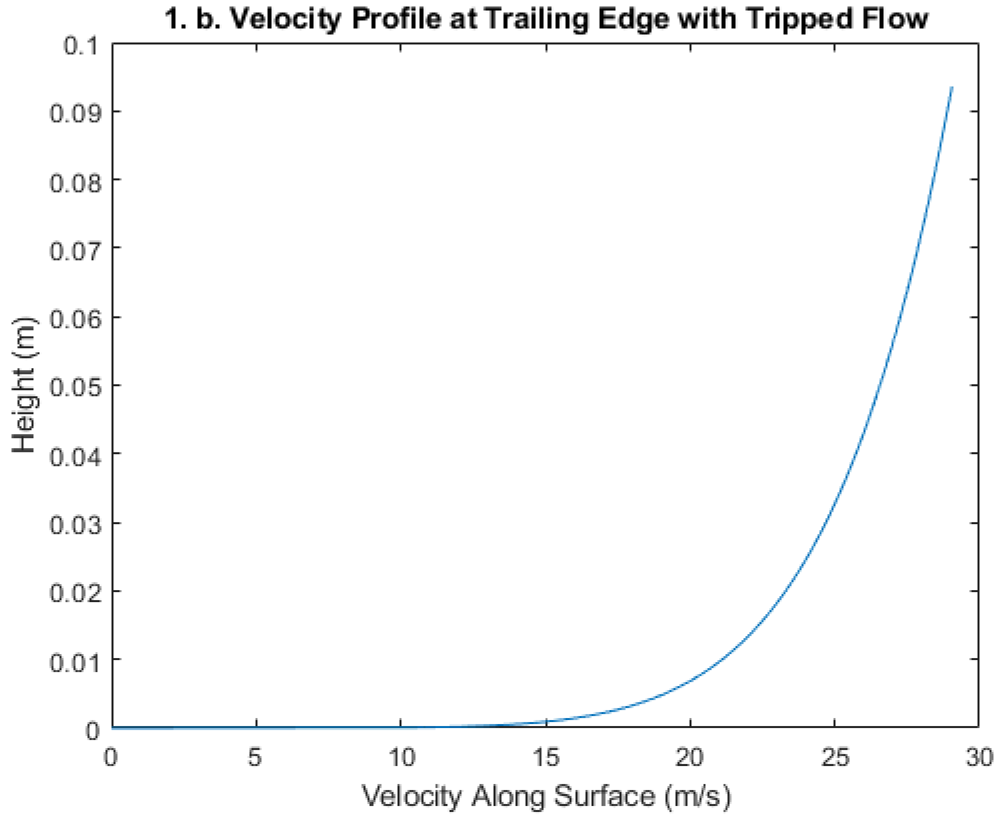
Liam Hood

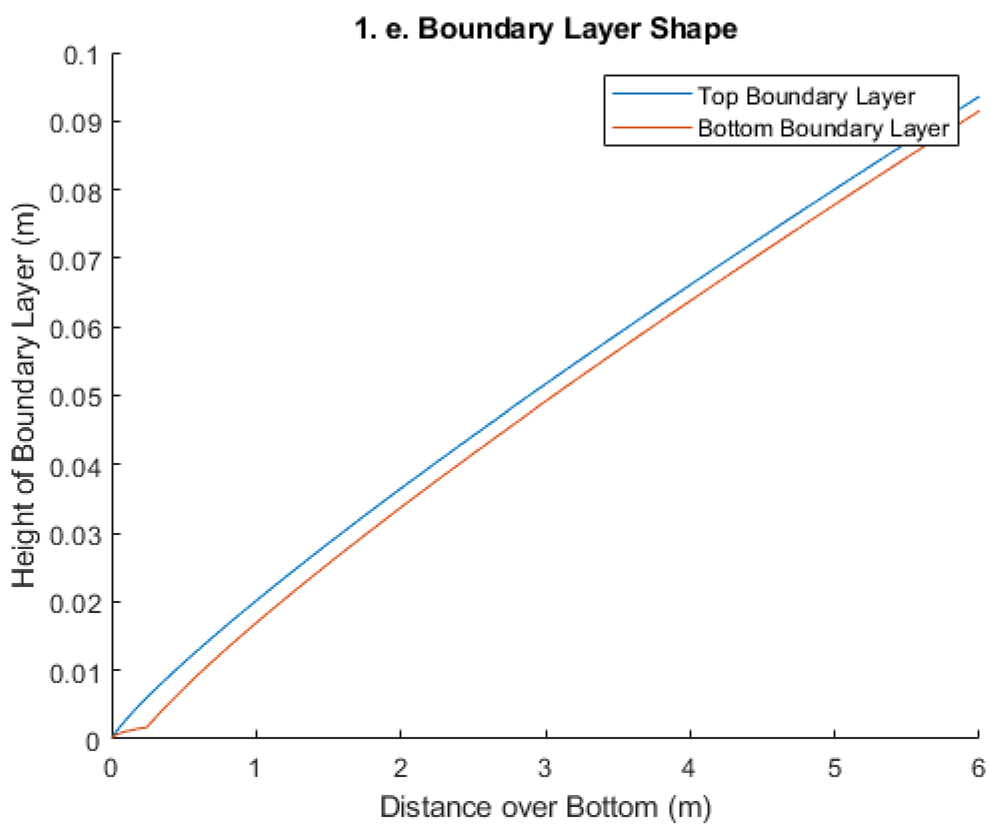
```
clear ; close all ; clc ;
n = 1e3 ;
rho = 1.225 ; % From Wikipedia
mu = 17.89e-6 ; % From Engineering toolbox
```

Answers to 1

```
Problem1(n,rho,mu)
```

```
1
a
The viscous on the top in the normal orientation is 17.6394 N
The viscous drag on the top in the wide and short orientation is21.974
 N
Drag is higher in the wide and short orientation. The coefficient of
 friction
decreases as Reynolds number increases so the friction force is higher
 if the
flow must go over the same area without the Reynolds number increasing
 as much
c
Boundary layer height on the bottom at the back is 0.09149 m
While the boundary layer on the top at the back is 0.093604 m
d
The total viscous drag is 35.0337 N
f
The boundary would be similar on the top in some spots. The cockipit
 would
change this with the pressure gradient at the front and rear. The
 fairings over
the wheels would change the boundary layer as it would add another
 dimension to
the flow.
```

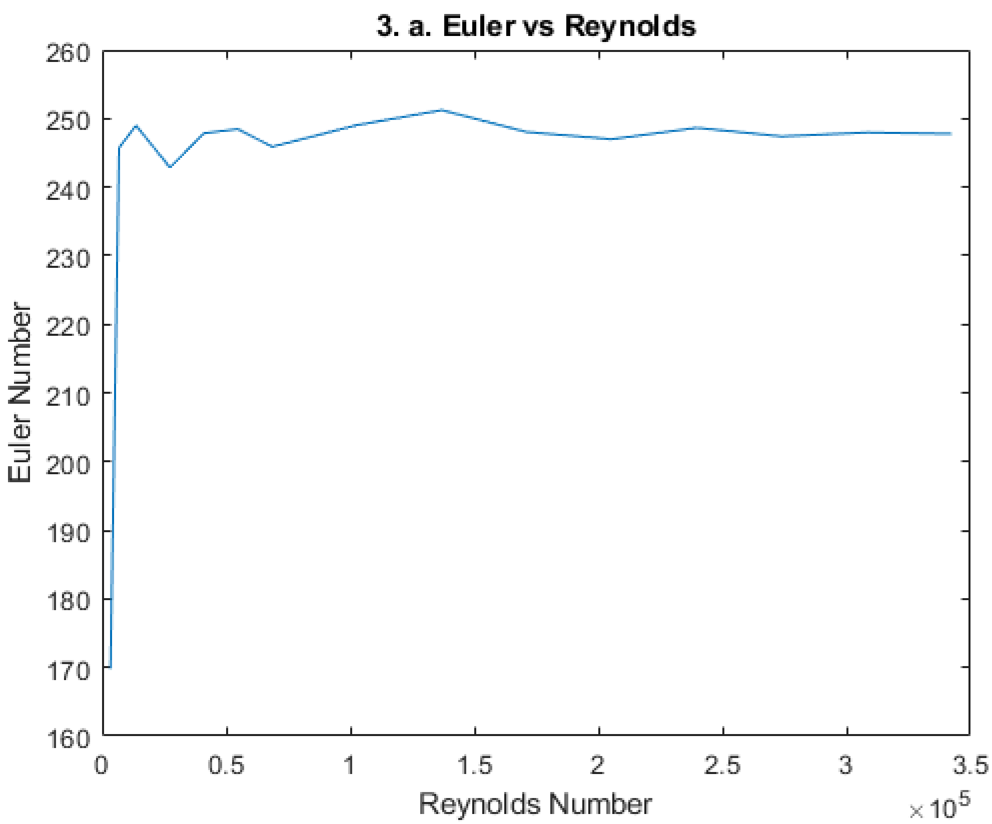

1. b. Velocity Profile at Trailing Edge with Tripped Flow
Height (m)
Velocity Along Surface (m/s)
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
5
10
15
20
25
30


1. e. Boundary Layer Shape
Top Boundary Layer
Bottom Boundary Layer
Height of Boundary Layer (m)
Distance over Bottom (m)
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
1
2
3
4
5
6

Answers to 3

```
Problem3(rho,mu)
```

3
a
The Euler number quickly appears to reach a constant value of around 253
It doesn't seem to matter how high the Reynolds number gets
b
The pressure drop at 80 m/s is 1943273.8765 Pa
c
If focusing on seperation point it could be more accurately found on a larger model.
A larger model would be good if a big but slow wind tunnel is all that is available.
Large model also good for studying a very small feature of a larger object.



Method for Solving Problems

```
function Problem1(n,rho,mu)
    % Constants
    L = 6 ;
    W = 2 ;
```

```
    v = 65*(1609/(60*60)) ;
    A = L*W ;
    x = linspace( 0 , L , n ) ;

    % Reynolds number
    Rex = ( rho*v*x )/( mu ) ;
    Rex(1) = 1 ;
        % Friction Top normal orientation
        CfTop = .074/(Rex(n)^(1/5)) ; % Average Coefficient of
friction
        FrictionTop = .5*rho*v^2*CfTop*A ;
        disp( '1' )
        disp( 'a' )
        disp([ 'The viscous on the top in the normal orientation is '
, num2str( FrictionTop ) , ' N' ])
        % Info for wide and short orientation
        Rel_ws = ( rho*v*W )/( mu ) ; % Reynolds number using the old
width as characteristic length
        CfTop_ws = .074/(Rel_ws^(1/5)) ; % Average Coefficient of
friction
        FrictionTop_ws = .5*rho*v^2*CfTop_ws*A ;
        disp([ 'The viscous drag on the top in the wide and short
orientation is' , num2str( FrictionTop_ws ) , ' N' ])
        disp( 'Drag is higher in the wide and short orientation. The
coefficient of friction' )
        disp( 'decreases as Reynolds number increases so the friction
force is higher if the' )
        disp( 'flow must go over the same area without the Reynolds
number increasing as much' )

    % Shape of top boundary layer
    DeltaTop = .16.*x./(Rex.^(1/7)) ;
    y = linspace( 0 , DeltaTop(n) , n ) ;
    % velocity of boundary layer on top
    u = v.*(y./DeltaTop(n)).^(1/7) ;

    % Velocity profile on top
    figure
    plot( u , y )
    title( '1. b. Velocity Profile at Trailing Edge with Tripped Flow'
)
    xlabel( 'Velocity Along Surface (m/s)' )
    ylabel( 'Height (m)' )

    % Boundary layer shape on bottom
    % Bottom
    for ii = 1:n
        if Rex(ii) < 5e5
            DeltaBottom(ii) = 4.91*x(ii)/(Rex(ii)^(1/2)) ;
            xTran = x(ii) ;
            DeltaLam = DeltaBottom(ii) ;
        else
            DeltaBottom(ii) = .16*(x(ii)-xTran)/(Rex(ii)^(1/7)) +
DeltaLam ;
```

```
        end
    end
    disp( 'c' )
    disp([ 'Boundary layer height on the bottom at the back is ' ,
 num2str(DeltaBottom(n)) , ' m' ])
    disp([ 'While the boundary layer on the top at the back is ' ,
 num2str(DeltaTop(n)) , ' m' ])
    % plot of boundary layer shapes
    figure
    hold on ;
    plot( x , DeltaTop )
    plot( x , DeltaBottom )
    xlabel( 'Distance over Bottom (m)' )
    ylabel( 'Height of Boundary Layer (m)' )
    title( '1. e. Boundary Layer Shape' )
    legend( 'Top Boundary Layer' , 'Bottom Boundary Layer' )
    hold off;

    % Friction

    % Top
    %     CfTop = .074/(Rex(n)^(1/5)) ;
    %     FrictionTop = .5*rho*v^2*CfTop*A ;

        % Bottom
        CfBLam = 1.33/(5e5^(1/2)) ;
        FrictionBLam = .5*rho*v^2*CfBLam*2*xTran ;
        CfBTurb = .074/(Rex(n)^(1/5)) ;
        FrictionBTurb = .5*rho*v^2*CfBTurb*2*(6-xTran) ;
        FrictionBottom = FrictionBLam + FrictionBTurb ;

        TotalFriction = FrictionTop + FrictionBottom ;
        disp( 'd' )
        disp([ 'The total viscous drag is ' ,
 num2str(TotalFriction) , ' N' ])

        disp( 'f' )
        disp( 'The boundary would be similar on the top in some spots.
 The cockipit would' )
        disp( 'change this with the pressure gradient at the front and
 rear. The fairings over')
        disp( 'the wheels would change the boundary layer as it would
 add another dimension to ')
        disp( 'the flow. ' )
end

function Problem3(rho,mu)
    disp( '3' )
    v = [ .5 1 2 4 6 8 10 15 20 25 30 35 40 45 50 ] ;
    dP = [ 52 301 1220 4760 10930 19480 30120 68640 123100 189898
 272300 373120 484900 615100 758800 ] ;
    diameter = .1 ;

    Eu = dP./(rho.*v.^2) ;
```

```
    Re = rho.*v.*diameter./mu ;

    figure
    plot( Re , Eu )
    title( '3. a. Euler vs Reynolds' )
    xlabel( 'Reynolds Number' )
    ylabel( 'Euler Number' )
    disp( 'a' )
    disp( 'The Euler number quickly appears to reach a constant value
 of around 253 ' )
    disp( 'It doesn''t seem to matter how high the Reynolds number
 gets' )

    Eu_80 = ( Eu( length(Eu) ) + Eu(length(Eu)-1) ) / 2 ;
    dP_80 = Eu_80*rho*80^2 ;
    disp( 'b' )
    disp([ 'The pressure drop at 80 m/s is ' , num2str( dP_80 ) , '
 Pa' ])
    disp( 'c' )
    disp( 'If focusing on seperation point it could be more accurately
 found on a larger model.' )
    disp( 'A larger model would be good if a big but slow wind tunnel
 is all that is available.' )
    disp( 'Large model also good for studying a very small feature of
 a larger object.' )

end

end
```

Published with MATLAB® R2018b