Biofluids Dynamics Modeling Project

Cooper Randeen[1]

Advisor: Dr. Matthew Rickard

Department of Biomedical Engineering California Baptist University, Riverside, CA 92504, USA

ABSTRACT

This project aimed to understand how velocity profiles and flow rates are determined using finite-volume methods using MATLAB. A 170cm medical tubing was modeled, consisting of 5mm and 3mm inner-diameter sections separated by a sharp entrance. Laminar flow equations were derived directly using $\sum F = ma$. The pipe radius was split into 0.5mm increments for the 5mm diameter and 0.3mm increments for the 3mm diameter and the resulting algebraic system was solved in MATLAB to obtain the velocity field. Boundary conditions were considered with the no-slip condition at the wall and zero shear at the centerline. Pressures at the junction between the two pipes were iteratively solved until flow rates matched and the expected pressure drop due to minor loss and the difference in pressure were equal.

Key Words: *MATLAB, Flow Rate, Laminar.*

1. INTRODUCTION

This project focused on developing a deeper understanding of how velocity profiles and flow rates are determined using finite-volume methods, such as those implemented in CFD software. A 170cm medical tubing system was modeled, incorporating an empirical sharp-edge entrance loss that divided 5mm and 3mm diameter tubing. The model was evaluated both numerically and analytically using MATLAB. To reinforce fundamental fluid mechanics concepts, the laminar flow equations were derived manually rather than relying on standard analytical solutions.

2. METHODS

In the first portion of this project, the finite-volume method was applied to a circular pipe cross-section which was divided into thin cylindrical shells. In dividing the pipe into cylindrical shells, a continuous fluid region is created into a finite number of control volumes. Dividing into control volumes allows each control volume to be solved individually to create a full velocity gradient – exactly how CFD software's function.

The horizontal pipe was modeled to have a 5mm diameter with the velocity profile fully developed, circular, steady, incompressible, zero gravity, and Newtonian. With these assumptions the pipe flow was made symmetric about the centerline causing only the radial distance from the wall to the centerline being modeled. The 2.5mm radius is divided into equal radial increments of 0.5 mm, which produces six control volumes between the wall and the axis of the tube. Each of these volumes has a top face (farther from the center) and a bottom face (closer to the center), and the shear stress at each face is computed using the velocity difference between the adjacent nodes separated by 0.5 mm. The outermost donut touches the pipe wall, where the no-slip condition fixes the velocity to zero. The innermost donut touches the centerline, where the shear stress must be zero because the velocity gradient is zero at that point. The same steps were repeated for the 3mm diameter pipe with a 0.3mm separation between nodes.

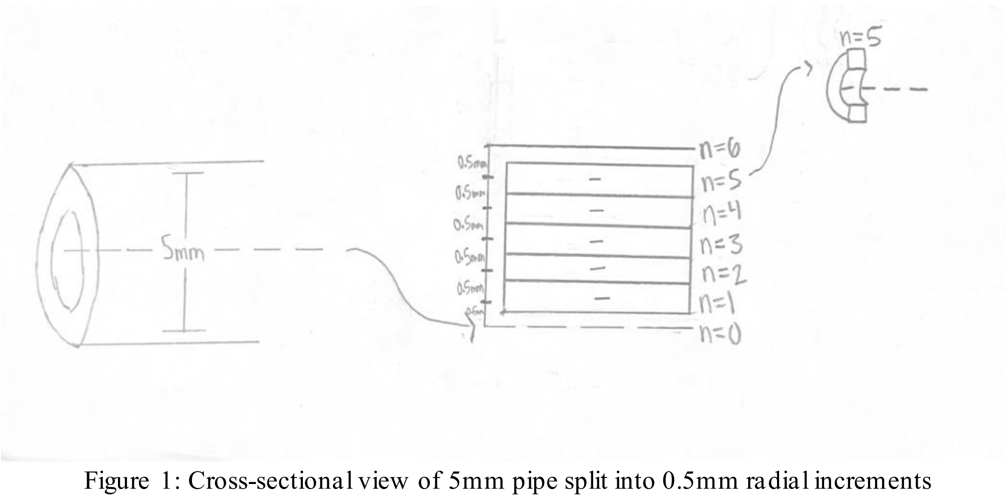

Figure 1: Cross-sectional view of 5mm pipe split into 0.5mm radial increments

With this information, laminar flow equations were derived using $\sum F = ma$. Considering the X direction to be positive the derivation of the equation is,

$$\sum F_x = 0 = A_L P_L - A_R P_R - [-\mu \frac{u(n+1) - u(n)}{dr} A_T] + [-\mu \frac{u(n) - u(n-1)}{dr} A_B]$$

With these assumptions of net force being equal to 0, acceleration is also equal to 0 meaning velocity is constant. In order for the sum of the forces to equal 0 the pressure drop must overcome the viscous friction which can be seen with the pressure force equaling the wall shear force.

To get shear stress, the velocity profile must be taken above the point, at the point, and below the point being analyzed – this creates e(n), f(n), and g(n). At the top of the cylinder shear stress was calculated by taking the velocity of the point being analyzed and then the point below that with the difference in radius. The cylindrical shell being analyzed also gives the velocity at the point through the equation:

$$d(n) = u(n+1)e(n) - u(n)f(n) + u(n-1)g(n)$$

This system of equations was then expressed in matrix form, where each row represents the force balance on a single cylindrical shell. Boundary conditions were applied directly in the matrix: the no-slip condition at the wall fixes $u(6) = 0$, and the symmetry condition at the centerline enforces $u(1) = u(2)$, which is reflected in the first row of the matrix.

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ g(2) & -f(2) & e(2) & 0 & 0 & 0 \\ 0 & g(2) & -f(2) & e(2) & 0 & 0 \\ 0 & 0 & g(2) & -f(2) & e(2) & 0 \\ 0 & 0 & 0 & g(2) & -f(2) & e(2) \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u(1) \\ u(2) \\ u(3) \\ u(4) \\ u(5) \\ u(6) \end{bmatrix} = \begin{bmatrix} 0 \\ d(2) \\ d(3) \\ d(4) \\ d(5) \\ 0 \end{bmatrix}$$

With the equations established, MATLAB code was written to compute the velocity gradient. Because the pressures at the end of the larger pipe (P2) and at the start of the smaller pipe (P3) were unknown, initial estimates were assigned so the numerical solution could be carried out. After the code was written, the velocity profiles for both pipe diameters were calculated and plotted. To validate the numerical results, an analytical expression for the velocity distribution was derived using calculus:

$$u(r) = -\frac{R^2}{4\mu}\frac{dP}{dx}(1 - \frac{r^2}{R^2})$$

This analytical solution was then plotted alongside the numerical results to assess the accuracy of the computational method. Once the velocity fields were obtained, the pressure drop across the tubes was determined using:

$$\Delta P = \frac{1}{2}\rho V_2^2 - \frac{1}{2}\rho V_1^2 + \rho g h_{fitting}$$

where

$$h_{fitting} = C\frac{V_2^2}{2g}$$

C = 0.5 (Sharp Entrance)

This allowed an error to be calculated using the equation:

$$Error = P2 - P3 - \Delta P$$

After error is calculated, P2 and P3 pressures could be calculated as their actual values. This must be done to satisfy the continuity equation. P2 was first set to 999Pa and P3 was then altered to achieve the exact same flow rate. After, the error value was analyzed to see whether P2 needed to increase or decrease to lower error. P3 was then altered again to achieve the same flow rate. These steps were repeated until error was negligible, and flow rates were identical. At this point, the exact pressure values at P2 and P3 were obtained.

3. RESULTS AND DISCUSSION

The numerical velocity profiles for both the 5 mm and 3 mm inner-diameter pipes were first obtained using the finite-volume formulation described previously. The fully developed laminar flow assumption resulted in the expected parabolic-type distribution, with zero velocity at the pipe wall and maximum velocity at the centerline. Figure 2 compares the numerical solution to the analytical profile derived from the differential equation.

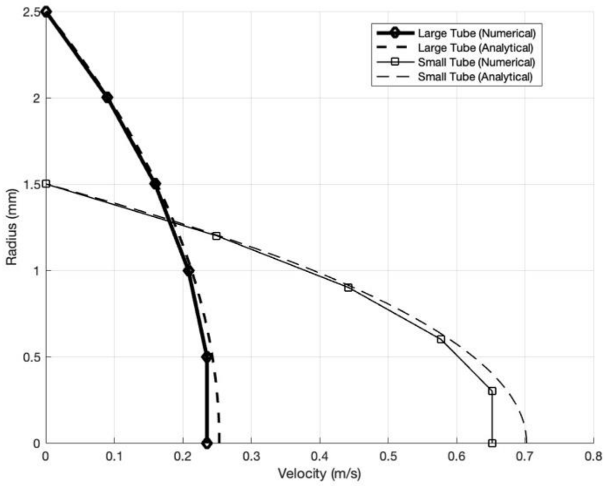

Figure 2: Relationship between velocity and radius for pipe diameters

Based on the assumptions of steady and incompressible flow, the fluid must satisfy the conservation of mass. This requirement is reflected in the velocity profiles, where the area under the curve (representing volumetric flow rate) is identical for both the large and small tubes.

Although the maximum velocity is higher in the smaller tube, the velocity gradient (slope) is steeper in the larger tube because the change in velocity occurs over a larger radial distance. In the small tube, the flow is more accelerated at the centerline, but the profile transitions to zero velocity more gradually because the radius is smaller. This behavior is consistent with laminar flow, where the velocity distribution is always parabolic, but its curvature depends on the pipe radius and pressure gradient. Pressures and the velocity gradient were calculated as follows:

P1 = 1000Pa
P2 = 855.6Pa
P3 = 777.8Pa
P4 = 0Pa
Velocity Gradient of 5mm Tube = [0.2354, 0.2354, 0.2083, 0.1597, 0.09014, 0]
Velocity Gradient of 3mm Tube = [0.6521, 0.6521, 0.5772, 0.4423, 0.2497, 0]

CONCLUSION

This project demonstrated how velocity profiles are constructed using finite-volume methods through the use of MATLAB. The numerical velocity profiles produced by the MATLAB implementation showed agreement with the analytical parabolic solution, confirming that the finite-volume approach correctly captured the flow behavior.

REFERENCES

M. Rickard, "Biofluid Dynamics Modeling Project," course handout for EGR 363 – Biofluid Mechanics and Relevant Technologies,
California Baptist University, Fall 2025.

APPENDIX

```
%% large tube
QL = 0;
Qs = 0;
p = 1000; % kg/m^3 rho
L = 1;
u = 0.00089;
P1 = 1000; % Pa
P2 = 855.6; % Pa
P3 = 777.8;
Rl = 0.0025; %m %big R
drL = 0.0005;%m
r = [0 drL drL*2 drL*3 drL*4 Rl];

for n = 2:5
   e(n) = (r(n+1)+r(n))/2;
   f(n) = (r(n+1)+2*r(n)+r(n-1))/2;
   g(n) = (r(n) + r(n-1))/2;
end
n = 6;
A = zeros(n, n);


A(1,1) = 1;
A(1,2) = -1;
A(6,6) = 1;

for i = 2:n-1
   A(i, i-1)   = g(i);
   A(i, i) = -f(i);
   A(i, i+1) = e(i);
end

d = zeros(6,1);
for i = 1:n
   d(i) = -((P1-P2)/(u*L))*(r(i)*(drL/L)^2);
end
d(n) = 0;
A = inv(A);
uL = A*d;

rL_exact = [0:0.000001:Rl];
for i = 1:1:length(rL_exact)
   ul_exact(i) = (-(Rl^2*(P2-P1)/(4*u*L))*(1-(rL_exact(i)^2/Rl^2)));

end

Ls = 0.7;
P4 = 0; %Pa
Rs = 0.0015; %m %big R
drs = 0.0003;%m
rs = [0 drs drs*2 drs*3 drs*4 Rs];

for n = 2:5
   e(n) = (rs(n+1)+rs(n))/2;
   f(n) = (rs(n+1)+2*rs(n)+rs(n-1))/2;
   g(n) = (rs(n) + rs(n-1))/2;
end
n = 6;
A = zeros(n, n);


A(1,1) = 1;
A(1,2) = -1;
A(6,6) = 1;
```

```
for i = 2:n-1
    A(i,  i-1)   = g(i);
    A(i,  i) = -f(i);
    A(i,  i+1) = e(i);
end

d = zeros(6,1);
for i = 1:n
    d(i) = -((P3-P4)/(u*Ls))*(rs(i)*(drs)^2);
end
d(n) = 0;
A = inv(A);
us = A*d;

rs_exact = [0:0.000001:Rs];
for i = 1:1:length(rs_exact)
    us_exact(i) = (-(Rs^2*(P4-P3)/(4*u*Ls))*(1-(rs_exact(i)^2/Rs^2)));

end

for i = 2:5
    QL(i)  = uL(i) * drL * 2*pi*r(i);
    Qs(i)  = us(i) * drs * 2*pi*rs(i);
end

coreL = uL(1)*pi*(r(2)/2)^2;
coreS = us(1)*pi*(rs(2)/2)^2;

QL = coreL + sum(QL)
Qs = coreS + sum(Qs)

V1  = QL/(pi*(r(6))^2);
V2  = Qs/(pi*(rs(6))^2);

hf_fitting = (0.5*(V2^2))/(2*9.81);
delta_p = (((V2^2)*p)/2)-(((V1^2)*p)/2)  + (hf_fitting*p*9.81);


error = P2-P3-delta_p



figure(1)
hold on

plot(uL, r*1000, '-dk', 'LineWidth',  3, 'MarkerSize',  7, 'DisplayName', 'Large Tube (Numerical)');
plot(ul_exact, rL_exact*1000, '--k', 'LineWidth', 2, 'DisplayName', 'Large Tube (Analytical)');


plot(us, rs*1000, '-sk', 'LineWidth',  1, 'MarkerSize', 7, 'DisplayName', 'Small Tube (Numerical)');
plot(us_exact, rs_exact*1000, '--k', 'LineWidth', 1, 'DisplayName', 'Small Tube (Analytical)');

xlabel('Velocity  (m/s)');
ylabel('Radius (mm)');
legend('Location', 'best');
grid  on;
```