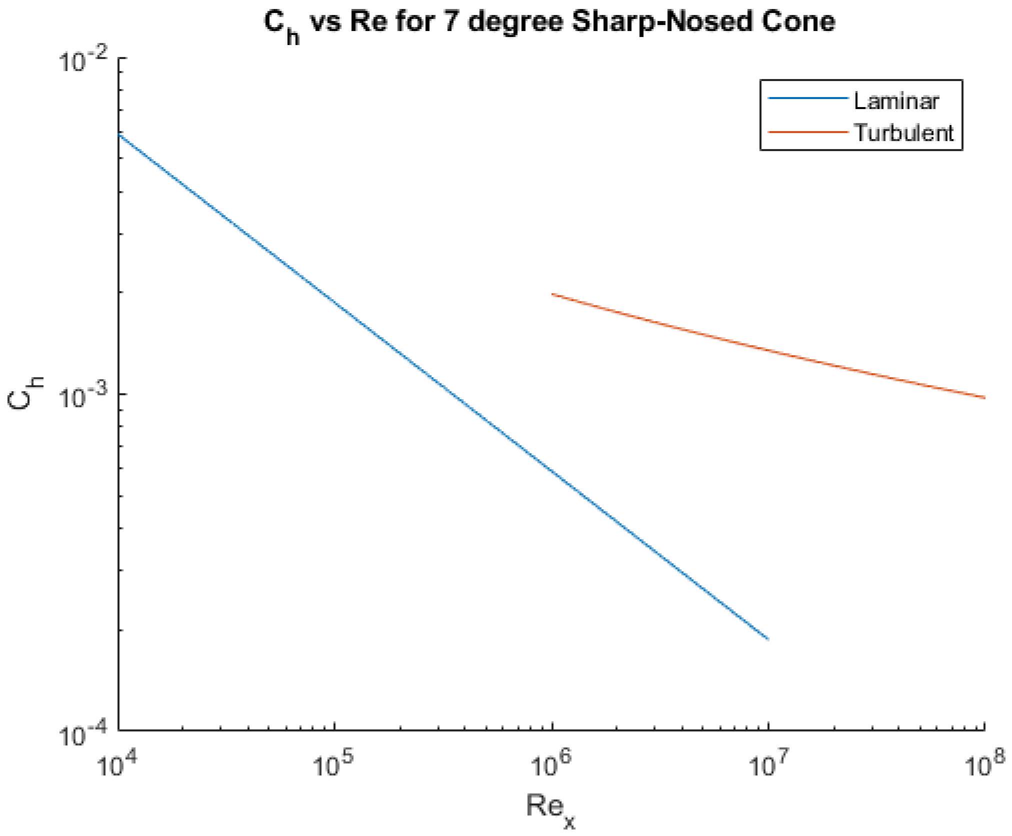
```
% Benjamin Stutzke
% ENAE 488P
% Homework 5
```

Question 1

```
% For edge conditions, use results for conical shock from
% https://devenport.aoe.vt.edu/aoe3114/calc.html

M1 = 8;
T1 = 220;
theta = deg2rad(7);
gamma = 1.4;
Pr = 0.75;
Tw = 300;
Re = 10^4:100:10^7;

% Finding post-shock conditions (= edge conditions)
% From https://devenport.aoe.vt.edu/aoe3114/calc.html,
Me =  6.83383405;
Tc_T1 =    1.33458957;
Te = Tc_T1 * T1;

% Laminar Solution
% Solving for reference T with Eckert
Tstar = (0.5 + 0.039*(Me^2) + 0.5*(Tw/Te))*Te;

% Sutherland's Law for viscosity
S = 110;
Tref = 273;
muref = 1.716e-5;

mustar = muref*((Tstar/Tref)^(3/2))*((Tref+S)/(Tstar+S));
mue = muref*((Te/Tref)^(3/2))*((Tref+S)/(Te+S));
muw = muref*((Tw/Tref)^(3/2))*((Tref+S)/(Tw+S));

% Chapman-Rubesin
% Ideal + constant pressure allows,
rhostar_rhoe = Te/Tstar;
Cstar = rhostar_rhoe * (mustar/mue);

% Stanton Number
Cf = 0.664.*sqrt(Cstar./Re);
Cf = Cf .* sqrt(3);

Ch = Cf./(2*(Pr^(2/3)));

figure(1);
hold on;
plot(Re,Ch);
set(gca, 'YScale', 'log');
set(gca, 'XScale', 'log');

% Turbulent Solution
Re_turb = 10^6:100:10^8;
Taw_Te = 1 + ((Me^2)*sqrt(Pr)*(gamma-1)/2);
Taw = Taw_Te * Te;

a = sqrt(((gamma-1)/2)*(Me^2)*(Te/Tw));
b = (Taw/Tw) - 1;

A = (2*(a^2) - b)/sqrt((b^2)+(4*(a^2)));
B = b/sqrt((b^2) + 4*(a^2));

S = sqrt(Taw_Te - 1)/(asin(A) + asin(B));
temp5 = ((mue/muw)*sqrt(Te/Tw)*0.06).*Re_turb./S;
temp6 = (S.*log(temp5)).^2;
Cf_turb = 0.455./temp6;

m = 1/6;
Cf_turb = Cf_turb .* ((2 + m)^(m/(m+1)));

Ch_turb = Cf_turb./(2*(Pr^(2/3)));

plot(Re_turb, Ch_turb);
title("C_h vs Re for 7 degree Sharp-Nosed Cone");
xlabel("Re_x");
ylabel("C_h");
legend("Laminar", "Turbulent");
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored.
```



Question 2

```
clear;

R = 6.04;
V1 = 3000;
T1 = 250;
rho1 = 0.0039;
gamma = 1.3;
Pr = 0.75;
Tw = 1500;
cp = 1000;

% First find post-normal shock conditions
a1 = sqrt(gamma*287*T1);
M1 = V1/a1;

temp7 = (gamma-1)*(M1^2) + 2;
temp8 = 2*gamma*(M1^2) - gamma + 1;
M2 = sqrt(temp7/temp8);

temp1 = (2*gamma*(M1^2) - gamma + 1);
temp2 = (((gamma-1)*(M1^2)) + 2);
temp3 = ((gamma+1)^2)*(M1^2);
T2_T1 = temp1*temp2/temp3;
```

```
T2 = T2_T1*T1;

a2 = sqrt(gamma*287*T2);
V2 = a2*M2;

rho2_rho1 = V1/V2;
rho2 = rho2_rho1 * rho1;

% Isentropic to stagnation conditions
rho0_rho2 = (1+((gamma-1)/2)*M2^2)^(1/(gamma-1));
rho0 = rho0_rho2 * rho2;

T0_T2 = (rho0_rho2)^(gamma-1);
T0 = T0_T2*T2;

% Find viscosity at stagnation
S = 110;
Tref = 273;
muref = 1.716e-5;

mu0 = muref*((T0/Tref)^(3/2))*((Tref+S)/(T0+S));

Taw_Te = 1 + ((M2^2)*sqrt(Pr)*(gamma-1)/2);
Taw = Taw_Te * T0;

rhoe = rho0;
mue = mu0;

alpha = (V1/R)*sqrt((2*rho1)/rhoe);
haw = cp*Taw;
hw = cp*Tw;

qw = 0.763*(Pr^(-.6)).*sqrt(rhoe.*mue.*alpha).*(haw - hw);
fprintf("Stagnation point heating (Joules): ");
disp(qw);

% At 1/10 scale, only the nose radius changes (heating goes up)
R_scale = R * 1/10;
alpha_scale = (V1/R_scale)*sqrt((2*rho1)/rhoe);
qw_scale = 0.763*(Pr^(-.6)).*sqrt(rhoe.*mue.*alpha_scale).*(haw - hw);

fprintf("Stagnation point heating for 1/10 scale (Joules): ");
disp(qw_scale);
```

```
Stagnation point heating (Joules):    5.6894e+04

Stagnation point heating for 1/10 scale (Joules):    1.7991e+05
```

Published with MATLAB® R2022b