

---

```

% Thomas Satterly
% AAE 537, HW 3
% Problem 3, Part iii

clear;
close all;

% Cp property for air, J/mol*K
cp_air_f = @(T) 27.453 + 6.1839 * (T / 1000) + 0.89932 * (T /
    1000)^2; % Air CP relation, T in K

% Area function (total area at percent along duct)
area = @(x, R, R_h) pi * ((R - (R_h * x))^2 - (R_h - (R_h * x))^2);

MW_air = 28.97; % g/mol
R_air = 287.058; % J/kg*K
gamma_air = 1.4;
R = 8314; % J / mol*K, Universal gas constant

% Calculate entrance stream properties first

% Core stream properties
T_1 = 1200; % K
mdot_1 = 28.7; % kg/s
M_1 = 0.4; % Mach
P_1 = 2 * 101325; % Pa
rho_1 = P_1 / (R_air * T_1); % kg/m^3
cp_1 = cp_air_f(T_1) / MW_air * 1000; % J/kg*K
gamma_1 = cp_1 / (cp_1 - R_air);
v_1 = M_1 * sqrt(gamma_1 * R_air * T_1); % m/s
Tt_1 = aeroBox.isoBox.calcStagTemp('mach', M_1, 'Ts', T_1, 'gamma',
    gamma_1); % K
Pt_1 = aeroBox.isoBox.calcStagPressure('mach', M_1, 'Ps',
    P_1, 'gamma', gamma_1); % Pa

% Core entrance area
A_1 = mdot_1 / (rho_1 * v_1); % m^2
A_star_1 = A_1 / aeroBox.isoBox.calcARatio(M_1, gamma_1);

% Core mass flow parameter
MFP_1 = sqrt(gamma_1 / R_air) * (M_1 / ((1 + ((gamma_1 - 1) / 2) *
    M_1^2)^((gamma_1 + 1) / (2 * (gamma_1 - 1)))));

% Fan stream properties
T_2 = 350; % K
mdot_2 = 83.9; % kg/s
M_2 = 0.408; % Mach
P_2 = 2 * 101325; % Pa
rho_2 = P_2 / (R_air * T_2); % kg/m^3
cp_2 = cp_air_f(T_2) / MW_air * 1000; % J/kg*K

```

---

---

```

gamma_2 = cp_2 / (cp_2 - R_air);
v_2 = M_2 * sqrt(gamma_2 * R_air * T_2); % m/s
Tt_2 = aeroBox.isoBox.calcStagTemp('mach', M_2, 'Ts', T_2, 'gamma',
    gamma_2); % K
Pt_2 = aeroBox.isoBox.calcStagPressure('mach', M_2, 'Ps',
    P_2, 'gamma', gamma_2); % Pa

% Fan entrance area
A_2 = mdot_2 / (rho_2 * v_2); % m^2
A_star_2 = A_2 / aeroBox.isoBox.calcARatio(M_2, gamma_2);

% Fan mass flow parameter
MFP_2 = sqrt(gamma_2 / R_air) * (M_2 / ((1 + ((gamma_2 - 1) / 2) *
    M_2^2)^((gamma_2 + 1) / (2 * (gamma_2 - 1)))));

% Total area
A_t = A_1 + A_2; % m^2

% Solve for the outer and hub radius
R = sqrt(A_t / pi * 16 / 15); % m
R_h = R / 4; % m

% Solve for lobe radius
r_lobe = R_h * (pi / (24 - 2 * pi)); % m

% Solve for mixer perimeter
P = 24 * pi * r_lobe + 24 * (R - R_h - 4 * r_lobe); % m

% Guess at L_n
L_n = 0.4; % m
Lerr = inf;
tolerance = 1e-1;
numSteps = 100;

step = 0.1;
lastDirection = 1;
mdot_1_save = mdot_1;
mdot_2_save = mdot_2;
while (abs(Lerr) > tolerance)
    clearvars results;
    dx = L_n / numSteps;
    mdot_1 = mdot_1_save;
    mdot_2 = mdot_2_save;
    dmdot_1 = mdot_1 / numSteps;
    dmdot_2 = mdot_2 / numSteps;
    dmdot_mix = 0;
    T_oldMix = T_1; % just a starting point, number doesn't really
    matter
    v_oldMix = 0;
    x = 0;
    b = 0;
    i = 0;
    while b < r_lobe
        i = i + 1;

```

---

---

```

    % Using last known state of core and fan, find the increase in
    % mixture area
    r = v_2 / v_1;
    s = rho_2 / rho_1;

    % Find M_r just for fun
    M_r = (2 * M_1 * (1 - r)) / (1 + (1 / sqrt(s)));

    lambda_s = ((1 - r) * (1 + sqrt(s))) / (2 * (1 + r *
sqrt(s)));
    db_dx_i = 0.161 * lambda_s;
    v_c = (v_1 + sqrt(s) * v_2) / (1 + sqrt(s));
    M_c = (v_1 - v_c) / v_1;
    db_dx_c = (0.2 + 0.8 * exp(-3 * M_c^2)) * db_dx_i;
    b = b + db_dx_c * dx;
    A_mix = b * P * 2; % Cross section area of mix stream tube,
m^2

    x = x + dx;
    if (b >= r_lobe) % If fully mixed, exit the loop
        break;
    end

    % Mix the streams together.
    dmdot_total = dmdot_1 + dmdot_2 + dmdot_mix;
    v_mix = (dmdot_1 * v_1 + dmdot_2 * v_2 + dmdot_mix *
v_oldMix) / dmdot_total;

    % Know a few parameters b/c mixing identical substance
    R_mix = R_air;
    MW_mix = MW_air;

    % Iterate to find the new cp_mix
    T_mix = (T_1 + T_2 + T_oldMix) / 3; % Starting guess
    err = inf;
    T_step = 10;
    lastDir = 1;
    while abs(err) > 1e-3
        % Calculate cps of everything
        cp_1 = cp_air_f(T_1) / MW_air * 1000;
        cp_2 = cp_air_f(T_2) / MW_air * 1000;
        cp_oldMix = cp_air_f(T_oldMix) / MW_air * 1000;

        % Calculate cp_mix on a mass basis
        cp_mix = (dmdot_1 / dmdot_total) * cp_1 + ...
            (dmdot_2 / dmdot_total) * cp_2 + ...
            (dmdot_mix / dmdot_total) * cp_oldMix;

        % Calculate energy balance error
        err = dmdot_1 * (cp_1 * T_1 + v_1^2 / 2) + ... % Core
stream tube
            dmdot_2 * (cp_2 * T_2 + v_2^2 / 2) + ... % Fan stream
tube

```

---

---

```

        dmdot_mix * (cp_oldMix * T_oldMix + v_oldMix^2 / 2)
- ... % Last stream tube
        dmdot_total * (cp_mix * T_mix + v_mix^2 / 2);

    if sign(err) ~= lastDir
        T_step = T_step / 2;
    end
    lastDir = sign(err);

    if err > 0
        T_mix = T_mix + T_step;
    else
        T_mix = T_mix - T_step;
    end
end

% Found new flow, let's finish it off
gamma_mix = cp_mix / (cp_mix - R_mix);
M_mix = v_mix / sqrt(gamma_mix * R_mix * T_mix);

% Finally, solve for new parameters upon next step
mdot_1 = mdot_1 - dmdot_1;
mdot_2 = mdot_2 - dmdot_2;
A_left = area(x / L_n, R, R_h) - A_mix;
if A_left <= 0
    keyboard;
end

% Put that MF-P to use!
A_1_temp = (mdot_1 * sqrt(Tt_1)) / (MFP_1 * Pt_1);
A_2_temp = (mdot_2 * sqrt(Tt_2)) / (MFP_2 * Pt_2);
A_temp = A_1_temp + A_2_temp;
A_1 = A_left * (A_1_temp / A_temp);
A_2 = A_left * (A_2_temp / A_temp);

if (A_1 < A_star_1)
    A_1 = A_star_1;
    M_1 = 1;
else
    M_1 = aeroBox.isoBox.machFromAreaRatio(A_1 / A_star_1,
gamma_1, 0);
end

if (A_2 < A_star_2)
    A_2 = A_star_2;
    M_2 = 1;
else
    M_2 = aeroBox.isoBox.machFromAreaRatio(A_2 / A_star_2,
gamma_2, 0);
end

%           M_1 = aae537.hw3.p3.machFromMFP('MFP', MFP_1, 'gamma',
gamma_1, 'R', R_air);

```

---

---

```

%           M_2 = aae537.hw3.p3.machFromMFP('MFP', MFP_2, 'gamma',
gamma_2, 'R', R_air);

           P_1 = aeroBox.isoBox.calcStaticPressure('mach', M_1, 'gamma',
gamma_1, 'Pt', Pt_1);
           P_2 = aeroBox.isoBox.calcStaticPressure('mach', M_2, 'gamma',
gamma_2, 'Pt', Pt_2);
%           P_1 = 2 * 101325 - (101325 * (x / L_n));
%           P_2 = P_1;

           M_1 = aeroBox.isoBox.machFromPressureRatio('Prat', P_1 /
Pt_1, 'gamma', gamma_1);
           M_2 = aeroBox.isoBox.machFromPressureRatio('PRat', P_1 /
Pt_1, 'gamma', gamma_2);

           T_1 = aeroBox.isoBox.calcStaticTemp('mach', M_1, 'gamma',
gamma_1, 'Tt', Tt_1);
           T_2 = aeroBox.isoBox.calcStaticTemp('mach', M_2, 'gamma',
gamma_2, 'Tt', Tt_2);

           v_1 = M_1 * sqrt(gamma_1 * R_air * T_1);
           v_2 = M_2 * sqrt(gamma_2 * R_air * T_2);
           rho_1 = mdot_1 / (A_1 * v_1);
           rho_2 = mdot_2 / (A_2 * v_2);
           rho_mix = dmdot_total / (A_mix * v_mix);

% Update mixture params for next round
dmdot_mix = dmdot_total;
v_oldMix = v_mix;
T_oldMix = T_mix;

% Log parameters
results(i).x = x;
results(i).mdot_1 = mdot_1;
results(i).mdot_2 = mdot_2;
results(i).mdot_mix = dmdot_total;
results(i).A_mix = A_mix;
results(i).A_1 = A_1;
results(i).A_2 = A_2;
results(i).A_total = A_mix + A_1 + A_2;
results(i).A_real = area(x / L_n, R, R_h);
results(i).v_1 = v_1;
results(i).v_2 = v_2;
results(i).v_mix = v_mix;
results(i).rho_1 = rho_1;
results(i).rho_2 = rho_2;
results(i).rho_mix = rho_mix;
results(i).P_1 = P_1;
results(i).P_2 = P_2;
results(i).b = b;
results(i).T_1 = T_1;
results(i).T_2 = T_2;
results(i).T_mix = T_mix;

```

---

---

```

        results(i).M_1 = M_1;
        results(i).M_2 = M_2;
        results(i).M_mix = M_mix;
        results(i).M_c = M_c;
        results(i).db_dx_c = db_dx_c;
        results(i).M_r = M_r;
        results(i).P = P_1;
        results(i).P_2 = P_2;
    end

    Lerr = (L_n - x) / x;

    if sign(Lerr) ~= lastDirection
        step = step / 2;
    end
    lastDirection = sign(Lerr);

    if Lerr > 0
        L_n = L_n - step;
    else
        L_n = L_n + step;
    end
end

figure;
hold on;
plot([results.x], [results.v_1]);
plot([results.x], [results.v_2]);
plot([results.x], [results.v_mix]);
legend('Core', 'Fan', 'Mix');
xlabel('x (m)');
ylabel('v (m/s)');
title('Stream Tube Velocity');

figure;
hold on;
plot([results.x], [results.rho_1]);
plot([results.x], [results.rho_2]);
plot([results.x], [results.rho_mix]);
legend('Core', 'Fan', 'Mix');
xlabel('x (m)');
ylabel('\rho (kg/m^3)');
title('Stream Tube Density');

figure;
hold on;
plot([results.x], [results.A_1]);
plot([results.x], [results.A_2]);
plot([results.x], [results.A_mix]);
plot([results.x], [results.A_1] + [results.A_2] + [results.A_mix]);
plot([results.x], [results.A_real]);
xlabel('x');
ylabel('Stream Area (m^2)');

```

---

---

```

legend('Core', 'Fan', 'Mix', 'Total', 'Linear Duct');
title('Stream Tube Area');

figure;
hold on;
plot([results.x], [results.b]);
xlabel('x');
ylabel('b');
title('b v. x');

figure;
hold on;
plot([results.x], [results.db_dx_c]);
xlabel('x');
ylabel('db/dx_c');
title('db/dx_c v. x');

figure;
hold on;
plot([results.x], [results.T_1]);
plot([results.x], [results.T_2]);
plot([results.x], [results.T_mix]);
xlabel('x');
ylabel('T (K)');
legend('Core', 'Fan', 'Mix');
title('Static Temperature v. x');

figure;
hold on;
plot([results.x], [results.M_1]);
plot([results.x], [results.M_2]);
plot([results.x], [results.M_mix]);
xlabel('x');
ylabel('Mach');
legend('Core', 'Fan', 'Mix');
title('Stream Mach v. x');

figure;
hold on;
plot([results.x], [results.M_c]);
xlabel('x');
ylabel('M_c');
title('M_c v. x');

figure;
hold on;
plot([results.x], [results.M_r]);
xlabel('x');
ylabel('M_r');
title('M_r v. x');

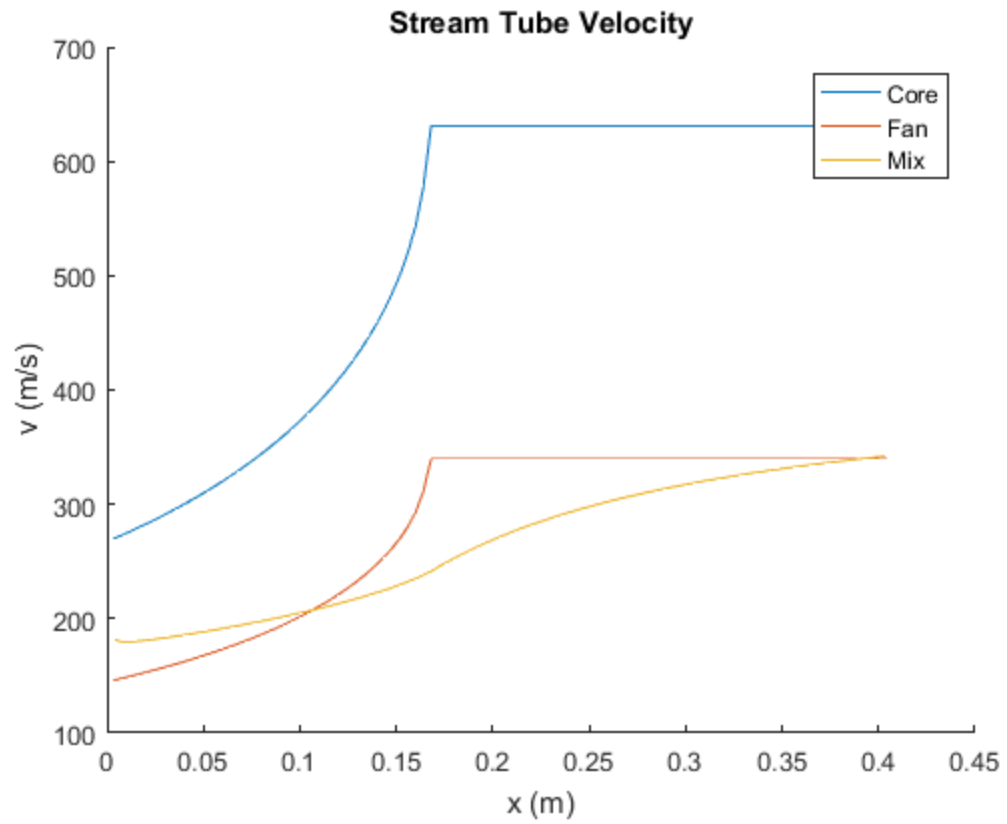
figure;
hold on;
plot([results.x], [results.P_1]);

```

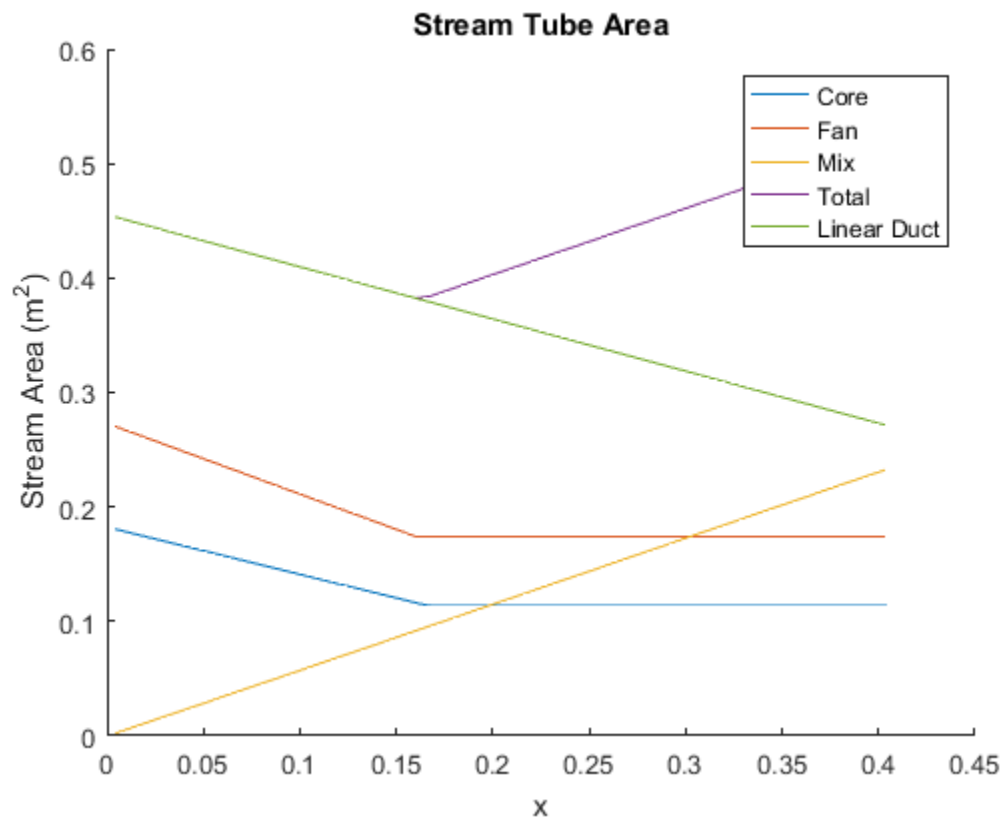
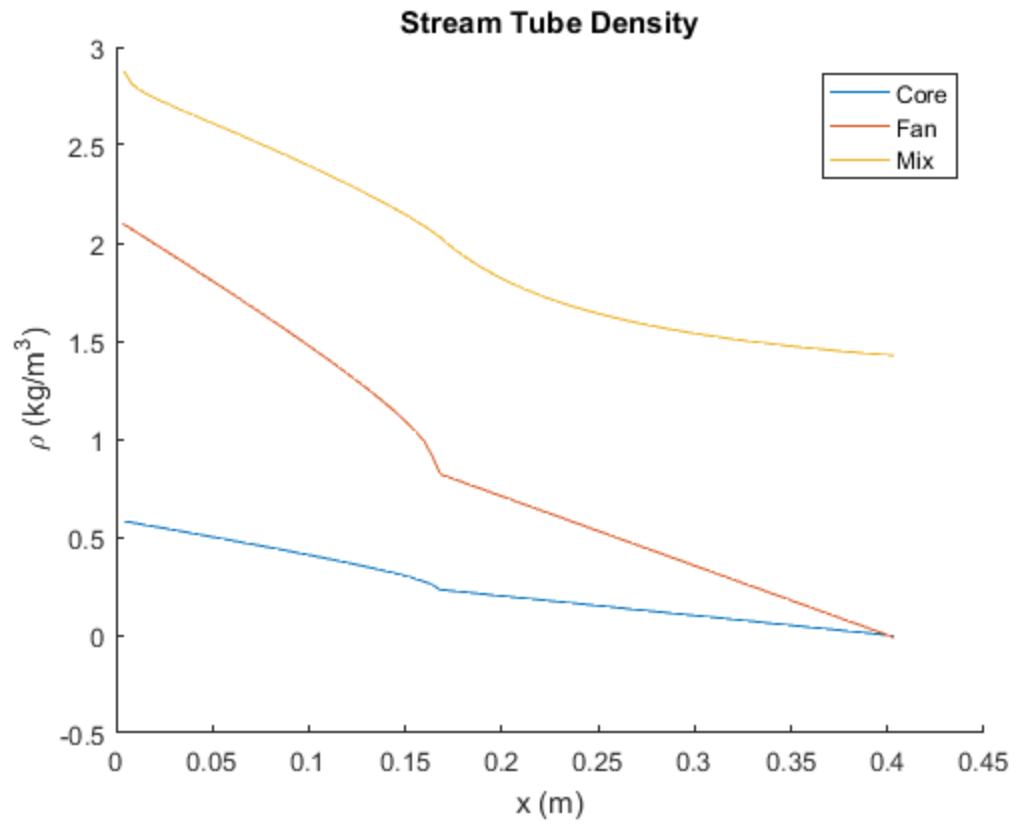
---

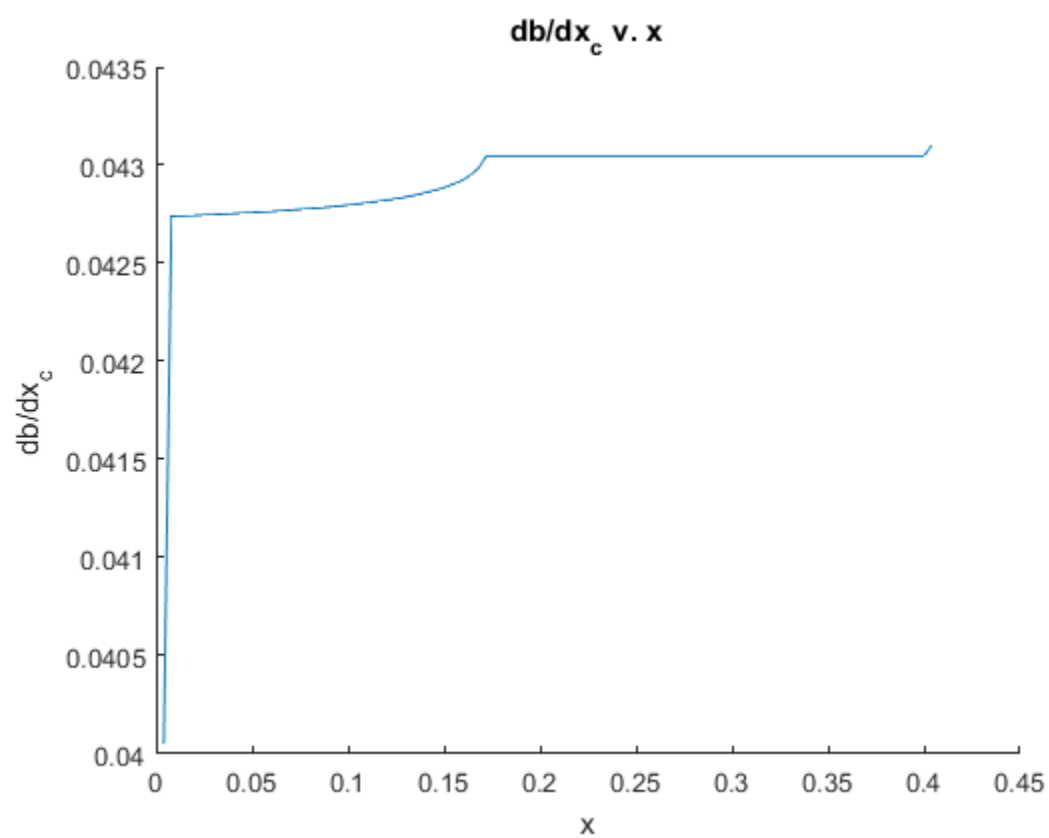
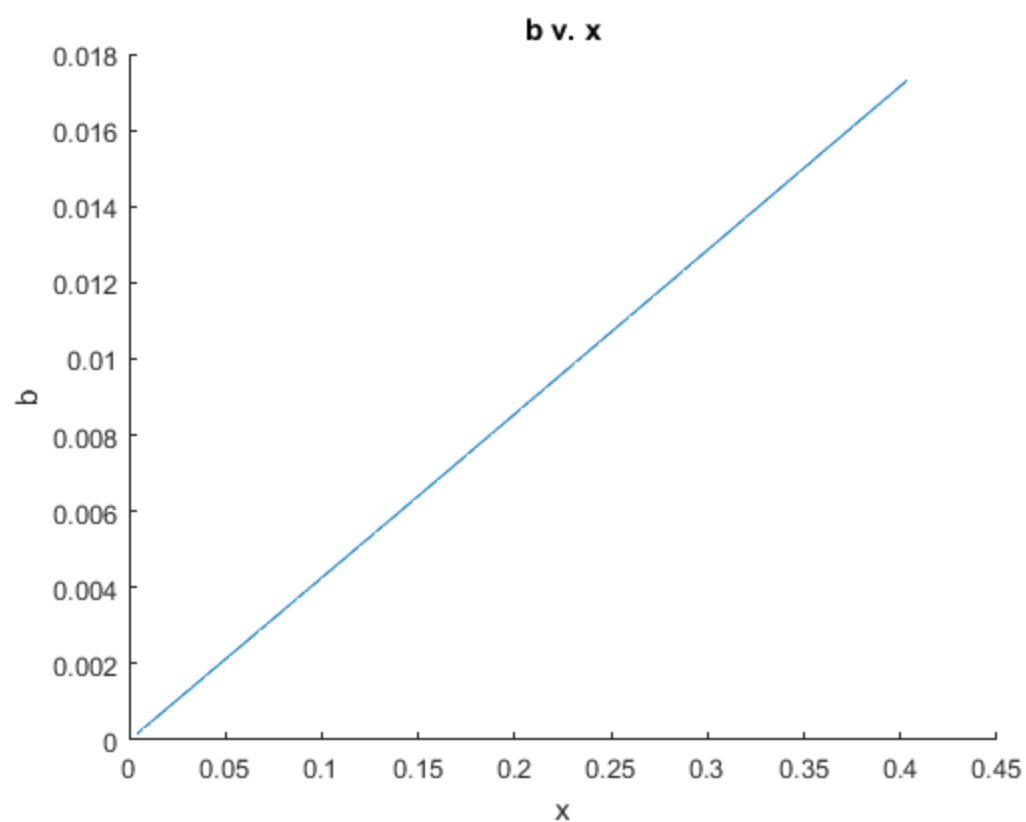
---

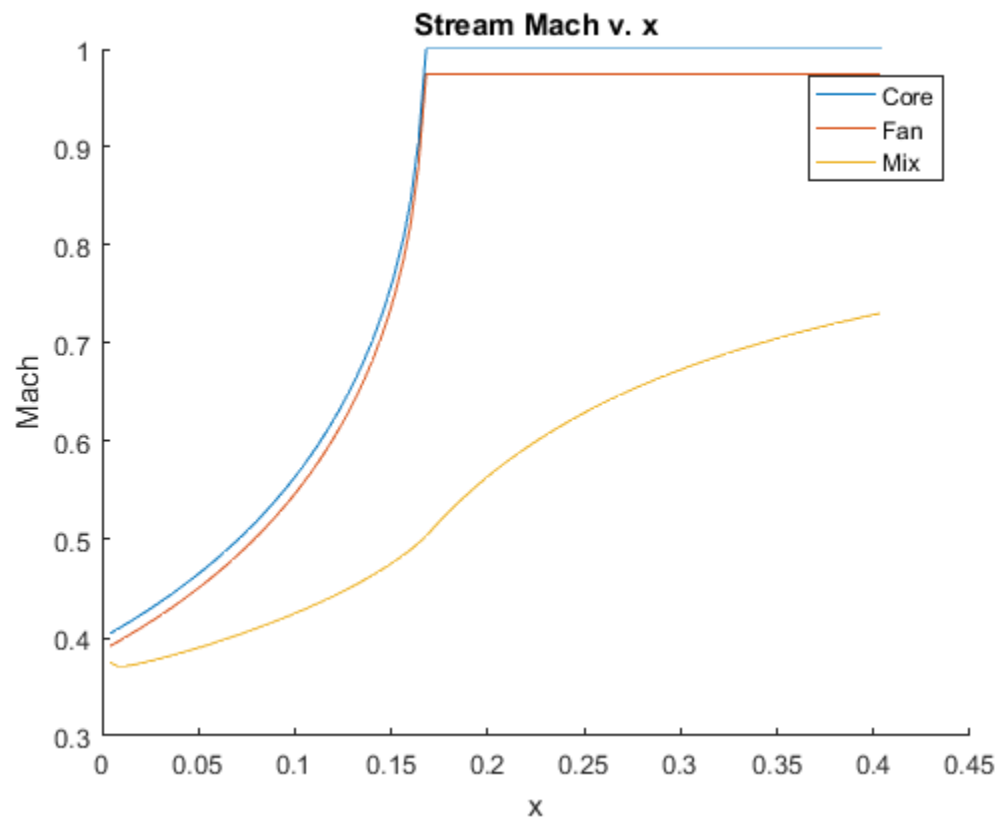
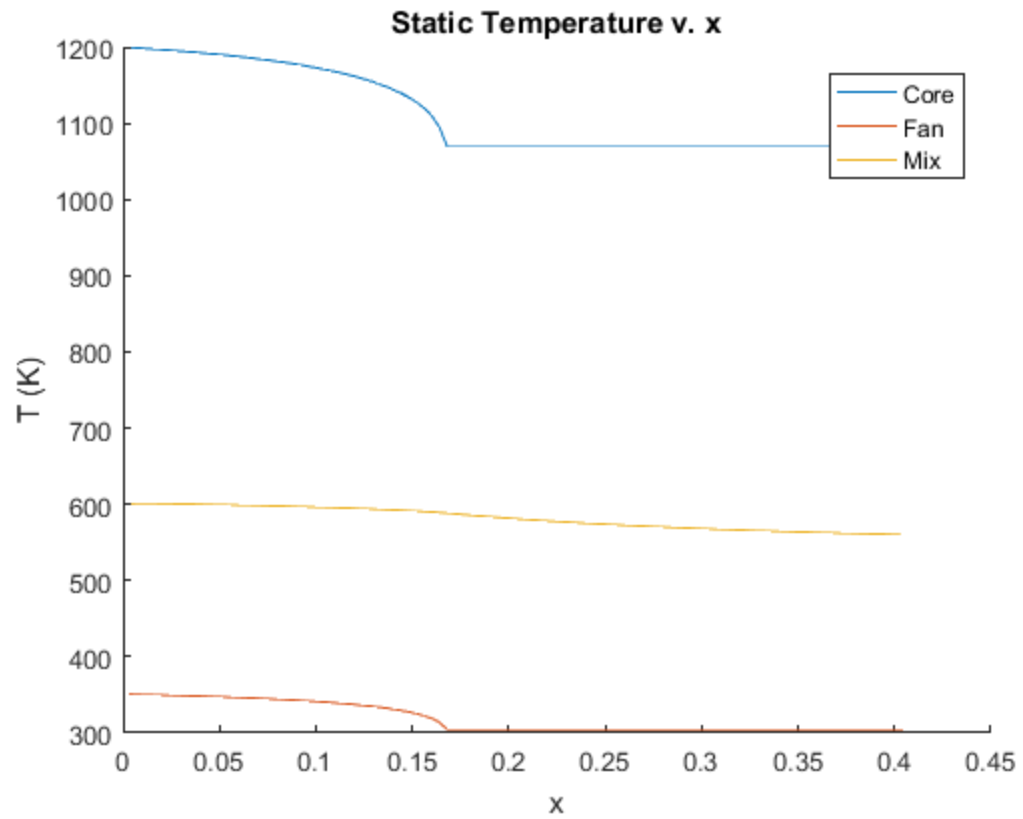
```
plot([results.x], [results.P_2]);  
xlabel('x');  
ylabel('Static Pressure (Pa)');  
legend('Core', 'Fan');  
title('Stream Pressure v. x');
```

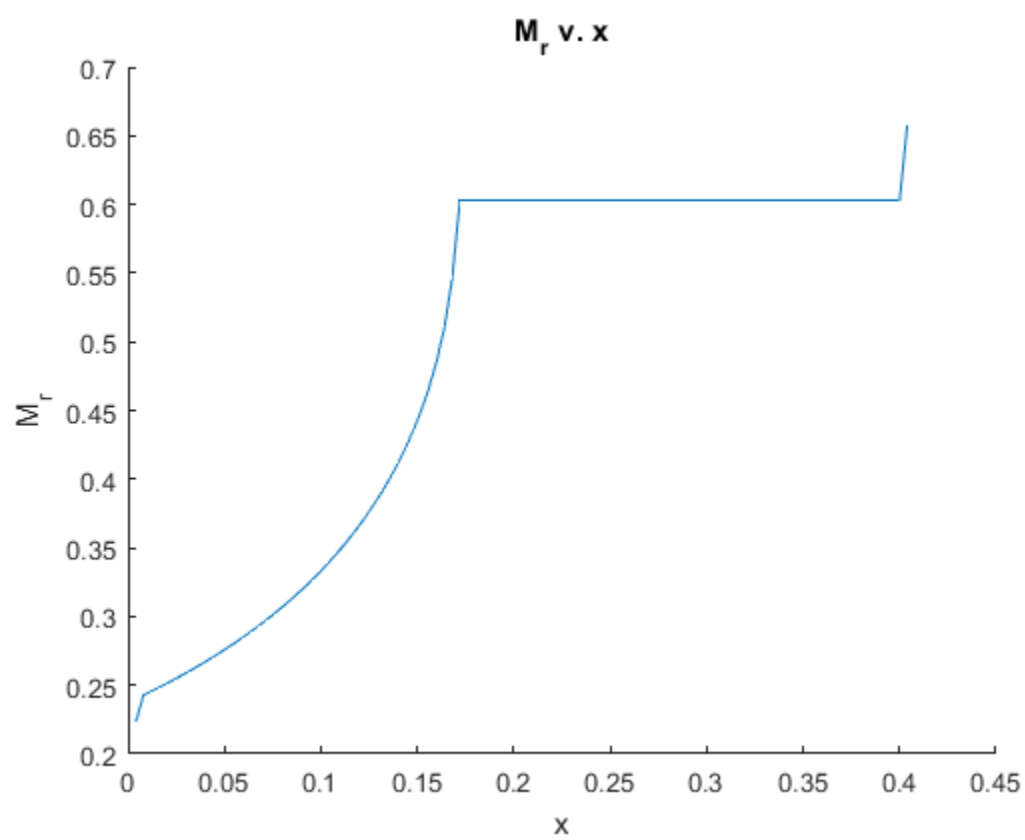
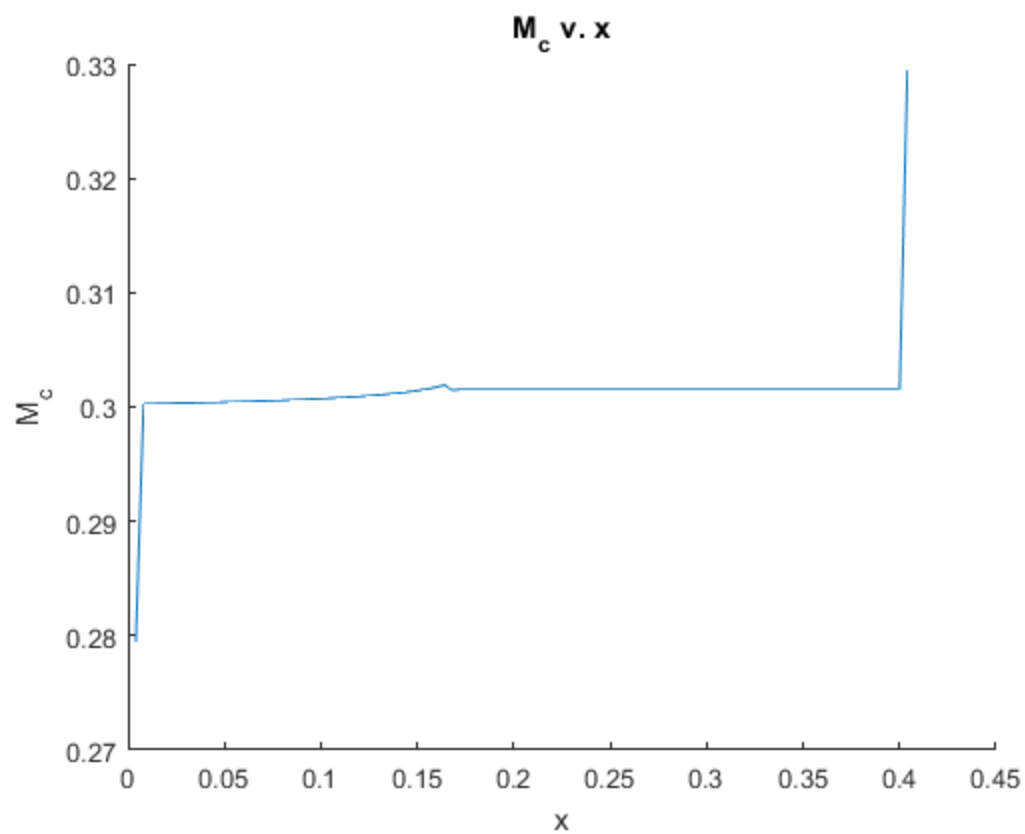


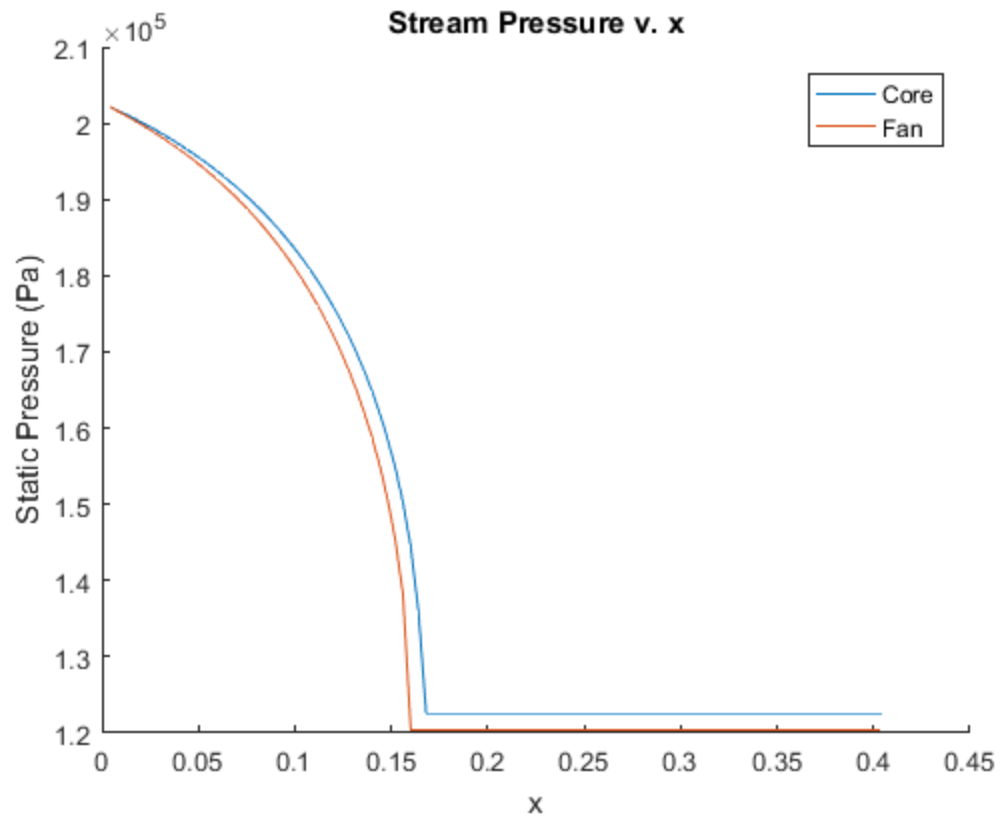












*Published with MATLAB® R2016a*