
Table of Contents

Roshan Jaiswal-Ferri	1
Workspace Prep	1
NASA CEA	1
Cp (kJ/kg-K) vs Temperature (K) Polyfits for N2, H2O, H2, CO, CO2	2
CF Plots	4

Roshan Jaiswal-Ferri

%Aero 402 Homework 2: 10/13/25

Workspace Prep

```
format long      %Allows for more accurate decimals
close all;       %Clears all
clear all;       %Clears Workspace
clc;             %Clears Command Window
```

NASA CEA

```
g = 9.8;
A = [1, 2, 5, 10, 100];
Cf = [0.6636, 1.2157, 1.4872, 1.6253, 1.8976];
Isp = [1562.2, 2861.9, 3501.0, 3826.0, 4467.1];
Isp = Isp ./ g;

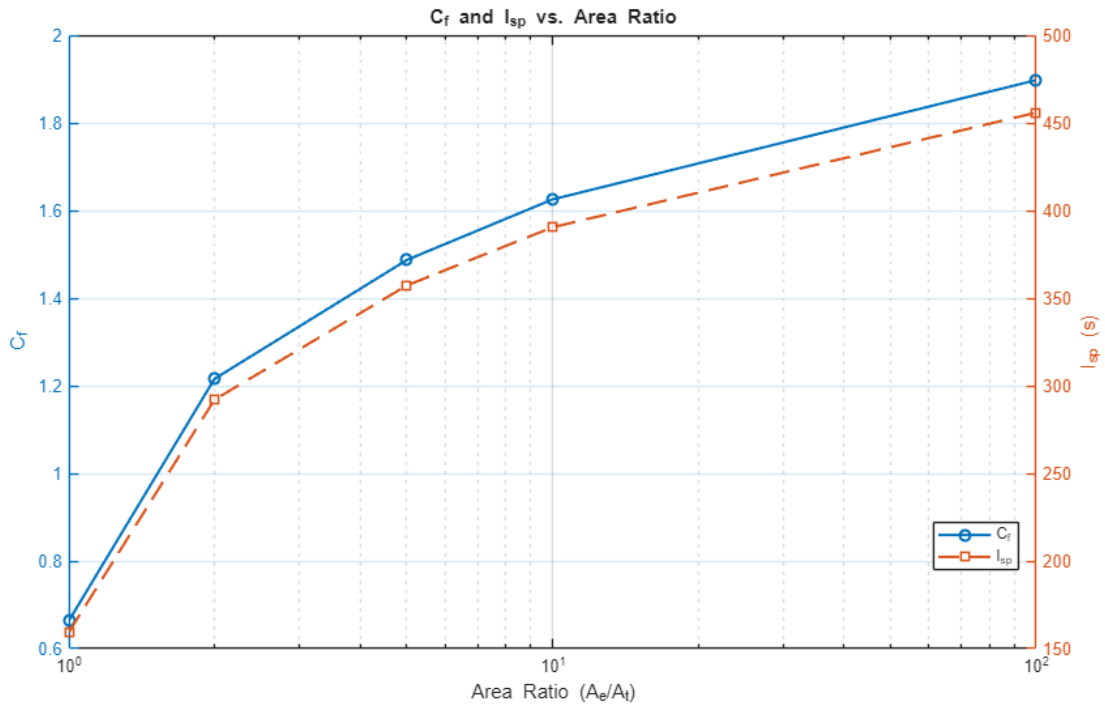
figure
yyaxis left
semilogx(A, Cf, '-o', 'LineWidth', 1.5)
ylabel('C_f')
grid on

yyaxis right
semilogx(A, Isp, '--s', 'LineWidth', 1.5)
ylabel('I_s_p (s)')

xlabel('Area Ratio (A_e/A_t)')
title('C_f and I_s_p vs. Area Ratio')
legend('C_f', 'I_s_p', 'Location', 'best')

fprintf(['These plots seem to show what is expected from in class and \n' ...
        'the slides\n'])

These plots seem to show what is expected from in class and
the slides
```



Cp (kJ/kg-K) vs Temperature (K) Polyfits for N2, H2O, H2, CO, CO2

```
% Data (ChatGPT Helped format and take data from screenshots of engtoolbox)
gases = struct([]);
```

```
% N2
```

```
gases(end+1).name = 'N2';
gases(end).T = [175 200 225 250 275 300 325 350 375 400 450 500 550 600 650
700 ...
750 800 850 900 950 1000 1050 1100 1150 1200 1250 1300 1350 1400 ...
1500 1600 1700 1800 1900 2000 2100 2200 2300 2400 2500 2600 2700 ...
2800 2900 3000 3500 4000];
gases(end).Cp = [1.039 1.039 1.039 1.039 1.039 1.040 1.040 1.041 1.042
1.044 ...
1.049 1.056 1.065 1.075 1.086 1.098 1.110 1.122 1.134 1.146 ...
1.157 1.167 1.177 1.187 1.196 1.204 1.212 1.219 1.226 1.232 ...
1.244 1.254 1.263 1.271 1.278 1.284 1.290 1.295 1.300 1.304 ...
1.307 1.311 1.314 1.317 1.320 1.323 1.333 1.342];
```

```
% H2O
```

```
gases(end+1).name = 'H2O';
gases(end).T = [175 200 225 250 275 300 325 350 375 400 450 500 550 600 650
700 ...
750 800 850 900 950 1000 1050 1100 1150 1200 1250 1300 1350 1400 ...
1500 1600 1700 1800 1900 2000 2100 2200 2300 2400 2500 2600 2700 ...
2800 2900 3000 3500 4000 4500 5000 5500 6000];
```

```

gases(end).Cp = [1.850 1.851 1.852 1.854 1.856 1.859 1.871 1.880 1.889
1.904 ...
    1.944 1.984 2.015 2.047 2.080 2.113 2.147 2.182 2.218 2.252 ...
    2.287 2.323 2.358 2.392 2.425 2.458 2.490 2.521 2.552 2.581 ...
    2.609 2.662 2.711 2.756 2.798 2.836 2.872 2.904 2.934 2.962 ...
    2.987 3.011 3.033 3.053 3.072 3.090 3.163 3.217 3.258 3.292 ...
    3.322 3.350];

% H2
gases(end+1).name = 'H2';
gases(end).T = [175 200 225 250 275 300 325 350 375 400 450 500 550 600 650
700 ...
    750 800 850 900 950 1000 1050 1100 1150 1200 1250 1300 1350 1400 ...
    1500 1600 1700 1800 1900 2000 2100 2200 2300 2400 2500 2600 2700 ...
    2800 2900 3000 3500 4000 4500 5000 5500 6000];
gases(end).Cp = [13.12 13.53 13.83 14.05 14.20 14.31 14.38 14.43 14.46
14.48 ...
    14.50 14.51 14.53 14.55 14.57 14.60 14.65 14.71 14.77 14.83 ...
    14.90 14.96 15.03 15.10 15.15 15.25 15.34 15.44 15.54 15.64 ...
    15.77 16.02 16.23 16.44 16.64 16.83 17.01 17.18 17.35 17.50 ...
    17.65 17.80 17.93 18.06 18.17 18.28 18.39 19.00 19.39 19.83 ...
    20.23 20.61];

% CO
gases(end+1).name = 'CO';
gases(end).T = [175 200 225 250 275 300 325 350 375 400 450 500 550 600 650
700 ...
    750 800 850 900 950 1000 1050 1100 1150 1200 1250 1300 1350 1400 ...
    1500 1600 1700 1800 1900 2000 2100 2200 2300 2400 2500 2600 2700 ...
    2800 2900 3000 3500 4000 4500 5000 5500 6000];
gases(end).Cp = [1.039 1.039 1.039 1.039 1.040 1.040 1.041 1.041 1.043
1.044 ...
    1.048 1.054 1.065 1.075 1.087 1.100 1.113 1.126 1.139 1.151 ...
    1.164 1.177 1.190 1.203 1.212 1.220 1.227 1.234 1.240 1.246 ...
    1.257 1.267 1.275 1.282 1.288 1.294 1.299 1.304 1.308 1.311 ...
    1.315 1.318 1.321 1.324 1.326 1.329 1.339 1.346 1.353 1.359 ...
    1.365 1.370];

% CO2
gases(end+1).name = 'CO2';
gases(end).T = [175 200 225 250 275 300 325 350 375 400 450 500 550 600 650
700 ...
    750 800 850 900 950 1000 1050 1100 1150 1200 1250 1300 1350 1400 ...
    1500 1600 1700 1800 1900 2000 2100 2200 2300 2400 2500 2600 2700 ...
    2800 2900 3000 3500 4000 4500 5000 5500 6000];
gases(end).Cp = [0.709 0.735 0.763 0.791 0.819 0.848 0.871 0.895 0.918
0.939 ...
    0.961 1.014 1.046 1.075 1.105 1.126 1.148 1.168 1.187 1.204 ...
    1.220 1.234 1.247 1.259 1.270 1.280 1.290 1.298 1.306 1.313 ...
    1.326 1.338 1.343 1.356 1.364 1.371 1.377 1.383 1.388 1.393 ...
    1.397 1.401 1.404 1.408 1.411 1.414 1.427 1.437 1.446 1.455 ...
    1.465 1.476];

n = 2;

```

```

T0 = 298;
dHrxn = 246.007; % kJ/mol
M =
struct('N2',0.0280134,'H2O',0.01801528,'H2',0.00201588,'CO',0.0280101,'CO2',0
.0440095);
nu = struct('N2',0.5,'H2O',0.836,'H2',0.664,'CO',0.836,'CO2',0.164);

% Fit Cp(T) & Integrat
coeffs = struct();
for k = 1:numel(gases)
    species = gases(k).name;
    p = polyfit(gases(k).T, gases(k).Cp, n);
    q_mass = polyint(p, 0);
    Mm = M.(species);
    q_molar = q_mass * Mm;
    coeffs.(matlab.lang.makeValidName(species)).q_molar = q_molar;
end

% Solve for Tc
F_mix = @(Tc) 0;
speciesList = fieldnames(nu);
for i = 1:numel(speciesList)
    s = speciesList{i};
    q_i = coeffs.(matlab.lang.makeValidName(s)).q_molar;
    F_mix = @(Tc) F_mix(Tc) + nu.(s)*(polyval(q_i,Tc) - polyval(q_i,T0));
end
F_root = @(Tc) F_mix(Tc) - dHrxn;

Tmin_all = T0;
Tmax_all = max(arrayfun(@(g) max(g.T), gases));
Tc_mix = fzero(F_root, [Tmin_all, Tmax_all]);

disp(['Mixture adiabatic temperature Tc: ', num2str(Tc_mix), ' K']);

Mixture adiabatic temperature Tc: 2484.1974 K

```

CF Plots

```

gamma = 1.20;
plp3 = [2 10 100 1000 Inf]; % p1/p3
p0pc = 1./plp3; p0pc(isinf(p0pc)) = 0;

Me = linspace(1.001,10,3000);
t = 1 + (gamma-1)/2.*Me.^2;
pepc = t.^(-gamma/(gamma-1)); % p_e/p_c
eps = (1./Me).*((2./(gamma+1)).*t).^((gamma+1)/(2*(gamma-1))); % epsilon:
A_e/A_t
m = eps>=1 & eps<=100; pepc = pepc(m); eps = eps(m);

figure;
hold on
grid on
box on

```

```

set(gca, 'XScale', 'log')

for k = 1:numel(p0pc)
    CF = C_f(gamma, pepc, eps, p0pc(k));
    CF(CF < 0) = NaN; % no plot below zero
    sep = pepc < 0.4*p0pc(k); %flow sep

    if isinf(plp3(k))
        lbl = 'p_1/p_3=\infty';
    else
        lbl = sprintf('p_1/p_3=%g', plp3(k));
    end

    plot(eps(~sep), CF(~sep), 'LineWidth', 1.6, 'DisplayName', lbl);

    % dashed part is flow separation
    if any(sep)
        plot(eps(sep), CF(sep), '--', 'LineWidth', 1.6);
    end
end

% Peak C_F line
% max_eps = zeros(size(p0pc));
% max_CF = zeros(size(p0pc));
for k = 1:numel(p0pc)
    CF = C_f(gamma, pepc, eps, p0pc(k));
    CF(CF < 0) = NaN;
    [max_CF(k), idx] = max(CF);
    max_eps(k) = eps(idx);
end
plot(max_eps, max_CF, 'ko-', 'LineWidth', 1.5, 'MarkerFaceColor', 'y', ...
     'DisplayName', 'Peak C_F line');

xlabel('Area ratio A_e/A_t (log scale)');
ylabel('Thrust coefficient C_F');
title('C_F vs A_e/A_t (\gamma = 1.20)');
legend('Location', 'southeast', 'AutoUpdate', 'off');
xlim([1 100]);

fprintf(['\nOptimum thrust coefficeints are where the exit pressure is\n' ...
        ' equal to the ambient pressure and therefore ideally expanded. \n' ...
        'The nozzle is under expanded above the maximum (or on the left)\n' ...
        'and is over expanded below the maximum (or on the right) of the \n' ...
        'max Cf line. The dashed lines are where flow separation occurs. \n\n'
        ...
        'An expansion ratio of one (eps = 1) means there will be no expansion
\n' ...
        'at all because exit area will equal throat area. The flow will be
choked'])

function CF = C_f(gamma, pe_pc, Ae_At, p0_pc)
CF = sqrt( (2.*gamma.^2./(gamma-1)) .* (2./(gamma+1)).^((gamma+1)./(
gamma-1)) ...
        .* (1 - pe_pc.^((gamma-1)./gamma)) ) ...

```

```

        + (pe_pc - p0_pc) .* Ae_At;
end

% Didnt use this one, replaced with ratios above

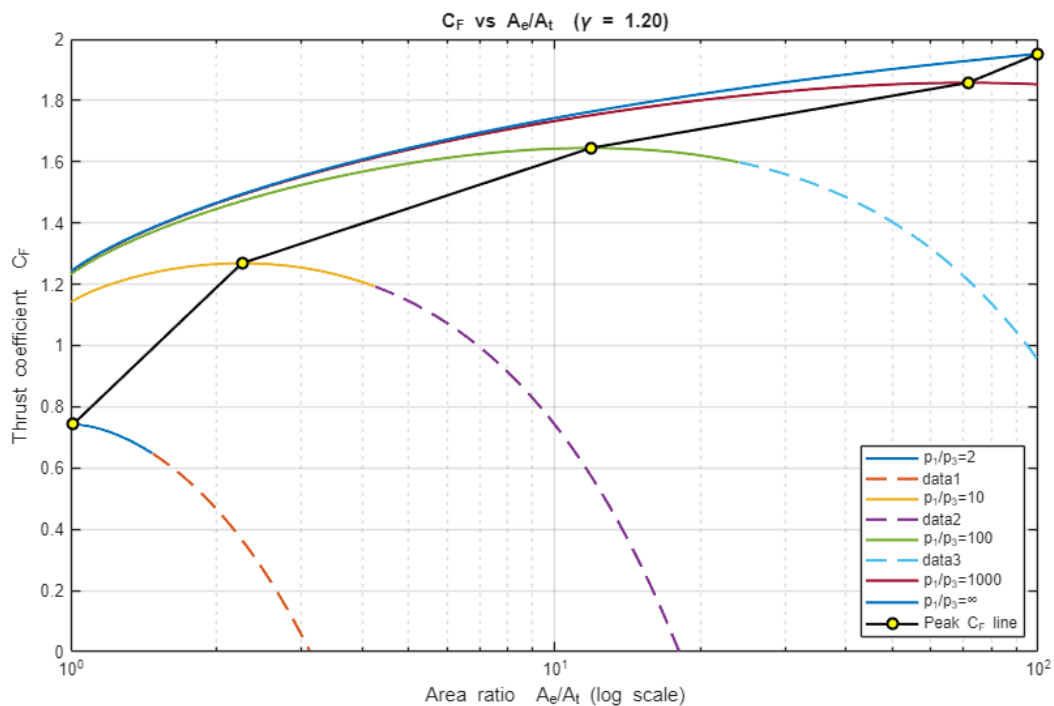
function CF = C_fold(gamma, pe, pc, Ae, At, p0)

CF = sqrt( (2.*gamma.^2./(gamma-1)) .* (2./(gamma+1)).^((gamma+1)./(
(gamma-1)) ...
        .* ( 1 - (pe./pc).^((gamma-1)./gamma) ) ) ...
        + ((pe - p0)./pc) .* (Ae./At);
end

```

Optimum thrust coefficeints are where the exit pressure is equal to the ambient pressure and therefore ideally expanded. The nozzle is under expanded above the maximum (or on the left) and is over expanded below the maximum (or on the right) of the max Cf line. The dashed lines are where flow seperation occurs.

An expansion ratio of one ($\epsilon = 1$) means there will be no expansion at all because exit area will equal throat area. The flow will be choked



Published with MATLAB® R2024b