

---

## Table of Contents

Question 1 .....	1
Question 2 .....	3
Question 3 .....	5

## Question 1

```
%a) Calculating and plotting gamma
T = 300:0.1:4000;
gamma_perf = 7/5;
theta_v = 3390;

s1 = exp(theta_v./T) ./ (exp(theta_v./T)-1).^2;
s2 = (gamma_perf-1) .* ((theta_v./T).^2);
s3 = (1+(s1 .* s2)).^-1;
gamma = 1+(gamma_perf-1).*s3;

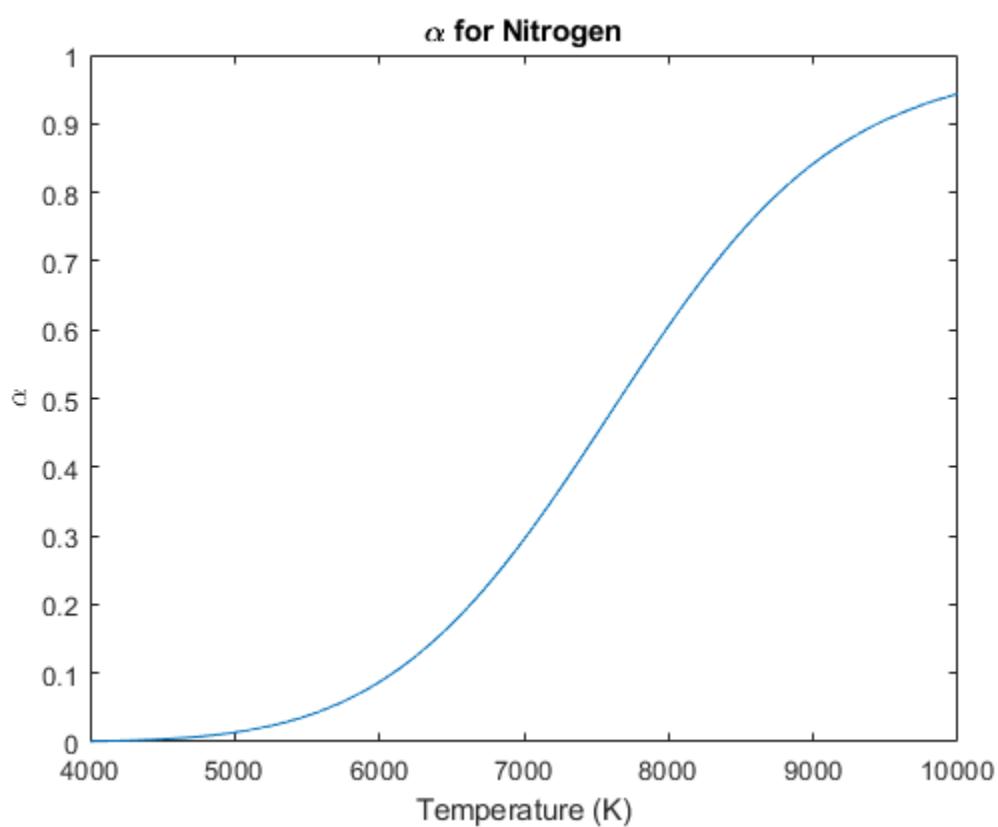
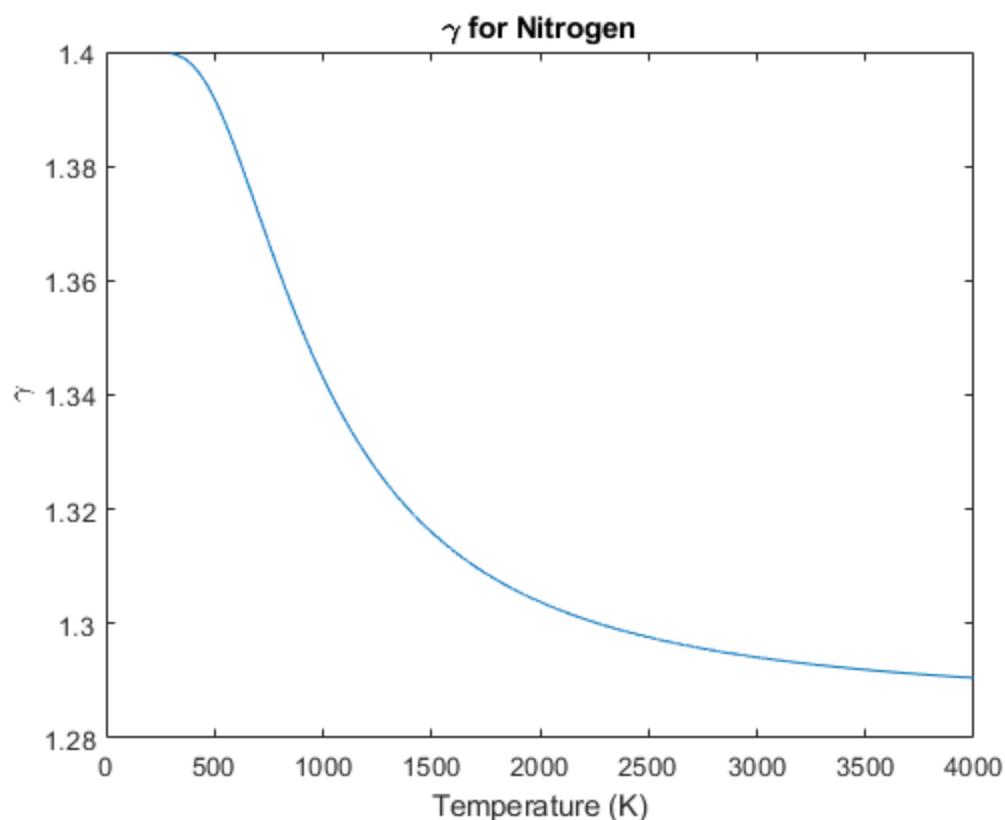
figure(1);
grid on;
plot(T, gamma);
xlabel('Temperature (K)');
ylabel('\gamma');
title('\gamma for Nitrogen');

%b) Calculating and plotting alpha
T = 4000:0.1:10000;
rho = 0.1;
rho_d = 130000;
theta_d = 113200;

s = (rho_d/rho).*exp(-theta_d./T);
alpha = 1:1:length(T);

for k=1:length(T)
    r = roots([1 s(k) -s(k)]);
    alpha(k) = r(r>=0);
end

figure(2);
grid on;
plot(T, alpha);
xlabel('Temperature (K)');
ylabel('\alpha');
title('\alpha for Nitrogen');
```



---

## Question 2

```
gamma = 1.4;
R = 287;
p0 = 5e6;
T0 = 750;
dstar = 20e-3;
Te = 50;

% Solving for Mach number at exit
syms x
Me = vpasolve(((Te/T0)^-1) == 1 + ((gamma-1)/2)*(x^2), x);
Me = Me( Me > 0)

% Solving for exit diameter
c1 = (1/(Me^2));
c2 = (2/(gamma+1));
c3 = 1+((gamma-1)/2)*(Me^2));
c4 = c3*c2;
c5 = c4^((gamma+1)/(gamma-1));
c6 = c5*c1;

de = vpasolve( (((x/2)^2)/((dstar/2)^2))^2 == c6, x);
de = de( de > 0) % meters

% Solving for exit velocity
ae = sqrt(gamma*R*Te)
ve = double(Me*ae) % m/s

% Solving for exit pressure
pe = ((1 + ((gamma-1)/2)*(Me^2)))^(-gamma/(gamma-1)) * p0 % Pa

% Solving for exit density
rho0 = p0/(R*T0)
rhoe = ((1 + ((gamma-1)/2)*(Me^2)))^(-1/(gamma-1)) * rho0 % kg/m3

% Approximations

p_ratio_approx = (((gamma-1)/2)*(Me^2))^(-gamma/(gamma-1));
pe_approx = p_ratio_approx * p0

rho_ratio_approx = (((gamma-1)/2)*(Me^2))^(-1/(gamma-1));
rhoe_approx = rho_ratio_approx * rho0

a0 = sqrt(gamma*R*T0)
ve_approx = sqrt(2/(gamma-1)) * a0

Me =
```

8.3666002653407554797817202578519

---

```
de =  
0.3055768890218131386927105425566  
  
ae =  
141.7392  
  
ve =  
1.1859e+03  
  
pe =  
382.51687369949796396832250862048  
  
rho0 =  
23.2288  
  
rhoe =  
0.026656228132369196095353484921288  
  
pe_approx =  
486.99205887832431611616887915407  
  
rhoe_approx =  
0.031674280252248736007555699457175  
  
a0 =  
548.9535  
  
ve_approx =  
1.2275e+03
```

---

## Question 3

```
% a) Solving for Coefficients of Pressure across upper and lower
surface
M1 = 5:0.1:15;
theta = deg2rad(10);
gamma = 1.4;

% Solving the B-Theta-M equation for beta
beta = M1;
syms b;
for k=1:length(beta)
    beta(k) = vpasolve(tan(theta) ==
2*cot(b)*(((M1(k)^2)*(sin(b)^2))-1)/(((M1(k)^2)*(gamma +
cos(2*b))+2)), b, 0:(pi/2));
end

% Plugging beta into 2.30 to solve for Cp lower
Cp_l = (4/(gamma+1)).*((sin(beta).^2)-(1./(M1.^2)));

% Solving the P-M expression for mul
mul = sqrt((gamma+1)/(gamma-1)).*atan(sqrt((gamma-1)/(gamma
+1)).*((M1.^2)-1))-atan(sqrt((M1.^2)-1));

% Solving for mu2 by using turning angle and mul
mu2 = theta + mul;

% Numerically solving for M2 by using P-M expression
M2 = M1;
syms m2;
for k=1:length(M2)
    M2(k) = abs(vpasolve(mu2(k) == (sqrt((gamma+1)/
(gamma-1))*atan(sqrt((gamma-1)/(gamma+1))*(m2^2)-1))-atan(sqrt((m2^2)-1))), m2));
end

% Using M2, M1 to solve for Cp upper across P-M expansion
pratio_upper = (((1+((gamma-1)/2).* (M1.^2)))./
(1+((gamma-1)/2).* (M2.^2))).^(gamma/(gamma-1));
Cp_u = (pratio_upper - 1).* (2./ (gamma.* (M1.^2)));

% Oblique Shock Approximation (2.30)
Cp_l_approx1 = (4/(gamma+1)).*(sin(beta).^2);

% Oblique Shock Approximation (2.43)
Cp_l_approx2 = (2*theta*theta).*(((gamma+1)/4) + (((gamma
+1)/4)^2)+(1./((M1.^2).* (theta.^2))).^(1/2));

% P-M Expansion Approximation (2.61)
pratio_upper_approx = (1+((gamma-1)/2).*M1.*-theta)).^(2*gamma)/
(gamma-1);
Cp_u_approx = (pratio_upper_approx - 1).* (2./ (gamma.* (M1.^2))');
```

---

```

figure(3);
grid on;
plot(M1, Cp_u);
xlabel('Mach Number');
ylabel('Cp_u');
title('Cp_u (across P-M expansion) vs Mach Number');

figure(4);
grid on;
plot(M1, Cp_l);
xlabel('Mach Number');
ylabel('Cp_l');
title('Cp_l (across oblique shock) vs Mach Number');

figure(5);
grid on;
plot(M1, Cp_u./Cp_l);
xlabel('Mach Number');
ylabel('Cp_u/Cp_l');
title('Cp_u/Cp_l vs Mach Number');

figure(6);
grid on;
hold on;
plot(M1, Cp_l);
plot(M1, Cp_l_approx1);
plot(M1, Cp_l_approx2);
xlabel('Mach Number');
ylabel('Cp_l');
legend('Cp_l (exact)', 'Cp_l approximation from (2.30)', 'Cp_l
approximation from (2.43)');
title('Comparison of Exact Value and Approximations of C_p Across
Oblique Shock');

figure(7);
grid on;
hold on;
plot(M1, Cp_u);
plot(M1, Cp_u_approx);
xlabel('Mach Number');
ylabel('Cp_u');
legend('Cp_u (exact)', 'Cp_u approximation from (2.61)');
title('Comparison of Exact Value and Approximation of C_p Across P-M
Expansion');

% b) Plotting Coefficient of Lift

% Calculating Cl approximation
c1 = (gamma+1)/2;
c2 = sqrt( (((gamma+1)/2)^2) + ((2./(theta.*M1)).^2) );
c3 = 2./(gamma.*(M1.^2).* (theta.^2));
c4 = 1 - ((1-((gamma-1)/2).*M1.*theta)).^( (2*gamma)/(gamma-1) );
Cl_approx = (theta.^2).*(c1+c2+(c3.*c4));

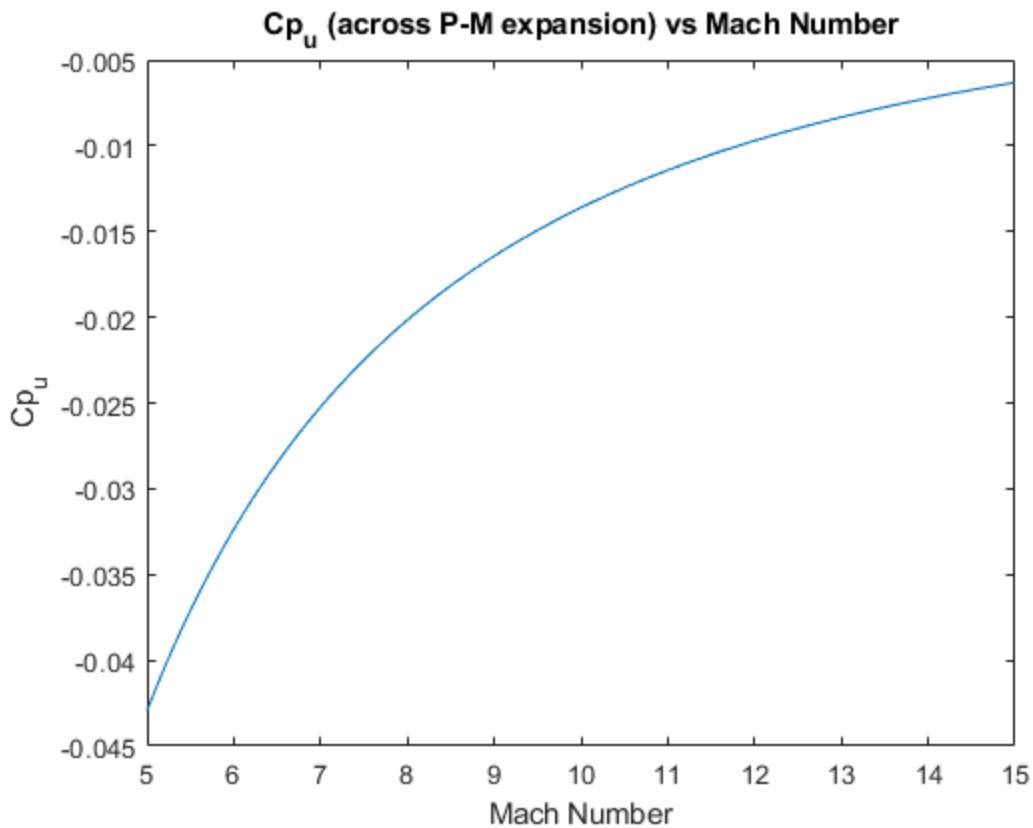
```

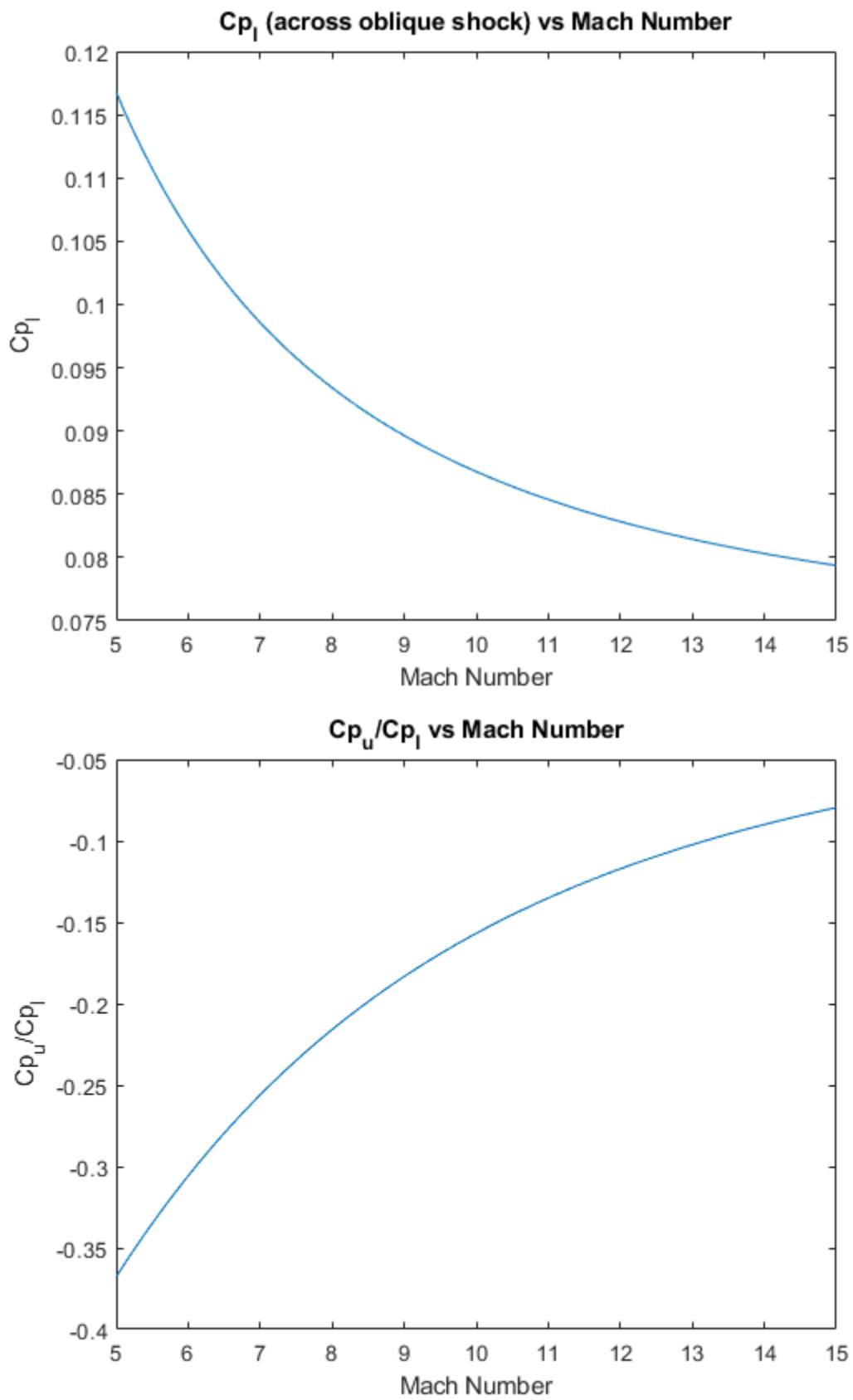
---

---

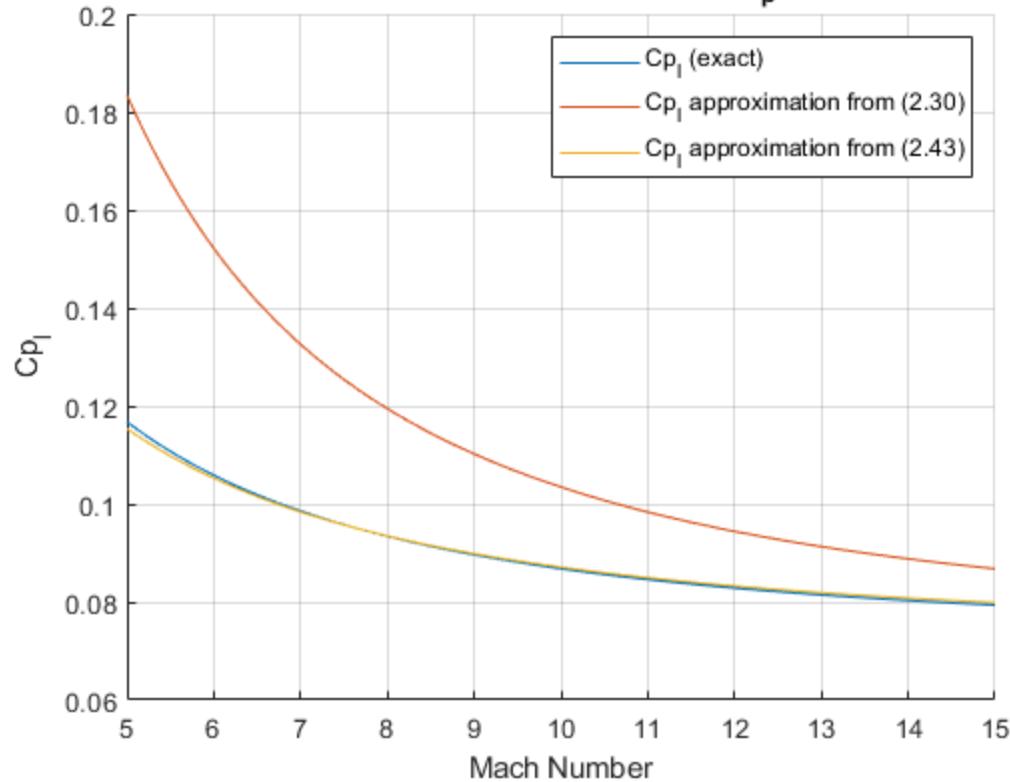
```
% Calculating Cl exactly
Cl = Cp_l - Cp_u;

figure(8);
grid on;
hold on;
plot(M1, Cl);
plot(M1, Cl_approx);
xlabel('Mach Number');
ylabel('C_L');
legend('C_L(exact)', 'C_L (approximation)');
title('Comparison of Exact Value and Approximation of C_L');
```

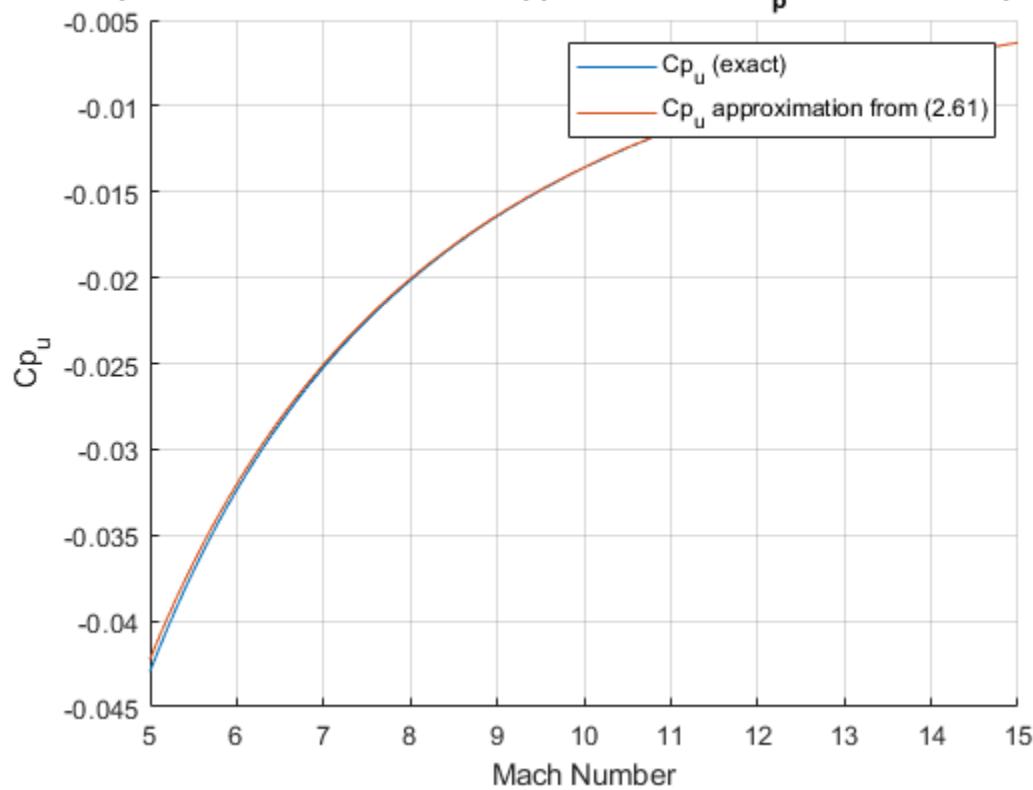


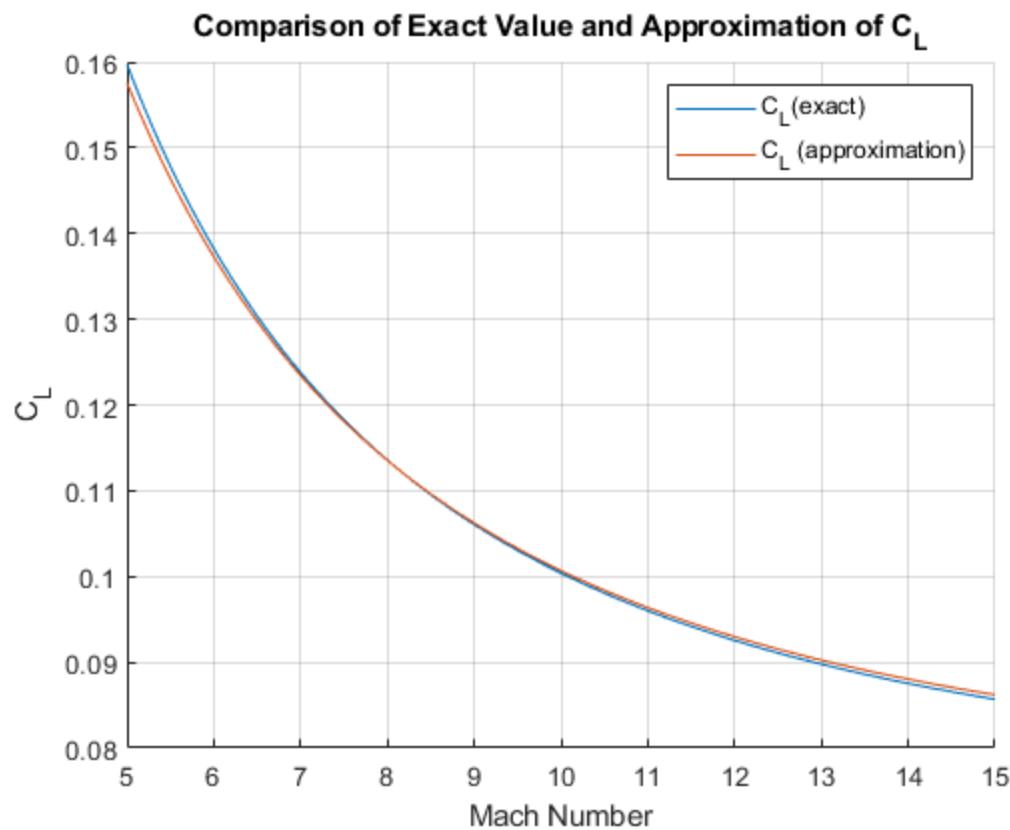


**Comparison of Exact Value and Approximations of  $C_p$  Across Oblique Shock**



**Comparison of Exact Value and Approximation of  $C_p$  Across P-M Expansion**





Published with MATLAB® R2020a