AE708 assignment 2

184104002, Potluri Vachan Deep

October 26, 2022

Notes

Points 2-6 are taken from slide 20 of the file `Rocket_Performance_Parameters_Nozzles.pdf`.

1. Simulations are performed using OpenFOAM-v9.
2. All nozzles use throat radius $R_t = 1$ m and a converging section as a 75° arc of radius $1.5R_t$.
3. All nozzles use $\varepsilon = 4 = (R_e/R_t)^2$.
4. Just after the throat, all nozzles have an arc of angle θ_n and radius $0.4R_t$. For Rao nozzle(s), $\theta_n = 30°$ while $\theta_n = 15°$ for conical nozzle.
5. Conical nozzle uses a half-cone angle of 15°.
6. Two Rao nozzles are considered with the diverging section length 60% ($K = 0.6$) and 80% ($K = 0.8$) of the conical nozzle.
7. All geometries are 3° bodies of revolution. Axisymmetric simulations are done in OpenFOAM (rhoCentralFoam).
8. Stagnation conditions: $p_0 = 287$ Pa, $T_0 = 1$ K and $\rho_0 = 1\,\mathrm{kg\,m^{-3}}$.
9. Initial conditions: discontinuity at throat with stagnation conditions imposed on left, and exit pressure and temperature on right (without velocity). For off design condition, stagnation temperature is imposed on the right part too.
10. Boundary conditions
    - Inlet: `totalPressure` for p, `totalTemperature` for T, `zeroGradient` for $\mathbf{U}$.
    - Outlet
      - Design condition: `zeroGradient` for p, T and $\mathbf{U}$.
      - Off design: `fixedValue` for p, `zeroGradient` for T and $\mathbf{U}$.
    - Wall: `zeroGradient` for p and T, `slip` for $\mathbf{U}$.
    - Azimuthal faces: `wedge` for p, T and $\mathbf{U}$.
11. All simulations use end time 1 s. Off design simulation uses end time 16 s.

Listing 1: Python code for designing Rao nozzle

```
# See https://moodle.iitb.ac.in/pluginfile.php/336817/mod_assign/introattachment/0/
    Bell_nozzle_design_methodology.pdf?forcedownload=1

import numpy as np
import matplotlib.pyplot as plt

# Parameters
R_t = 1 # throat radius
epsilon = 4 # exit area to throat area ratio
R_e = R_t*(epsilon**0.5) # exit radius
theta_n = 30*np.pi/180 # initial nozzle contour angle after circular segment
```

```
Rc_div = 0.4*R_t # curvature radius after divergent section
x_n = Rc_div*np.sin(theta_n) # x coordinate of point N (x axis starts at throat)
y_n = R_t + Rc_div*(1-np.cos(theta_n)) # y coordinate of point N (y axis starts from
    nozzle axis)
theta_conical = 15*np.pi/180
L_conical = (R_e-R_t)/np.tan(theta_conical) # conical nozzle length with 15 degree cone
    angle
K = 0.8 # length of Rao nozzle relative to conical nozzle with 15 degree cone angle

# Solve for coefficients
coeff_matrix = np.array([
    [2*y_n, 1, 0],
    [R_e**2, R_e, 1],
    [y_n**2, y_n, 1]
])
rhs_vector = np.array([1/np.tan(theta_n), K*L_conical, x_n])
solution = np.linalg.solve(coeff_matrix, rhs_vector)
np.set_printoptions(formatter={'float': '{: 0.3e}'.format})
print("Solution coefficients: {}".format(solution))

# Plot for comparison
L_rao = solution[0]*R_e**2 + solution[1]*R_e + solution[2] # length of Rao nozzle
fig, ax = plt.subplots(1,1)
y = np.linspace(y_n, R_e)
x_rao = solution[0]*y**2 + solution[1]*y + solution[2]
x_conical = (y-y_n)/np.tan(theta_conical) + x_n
ax.plot(x_rao, y, "b-", label="Rao")
ax.plot(x_conical, y, "r--", label="Conical")
ax.grid()
ax.legend()
plt.show()

# Data output (from the point N to the exit)
## pre-throat: circular arc profile
## post-throat: smaller circular arc followed by conical/rao profile
# np.savetxt("rao_profile.dat", np.vstack((x_rao, y)).T)
```

Listing 2: Gmsh code for generating Rao nozzle mesh

```
/*
Point numbering convention:
    0xx: nozzle contour (converging and diverging), except parabolic profile
    1xx: temporary points
    2xx: parabolic profile spline
Line numbering convention:
    0xx: nozzle contours, including parabolic profile spline
Curve loops, surface convention:
    0xx: all
*/

R_t = 1; // throat radius
theta_n = 30*Pi/180; // angle at point N
theta_i = 75*Pi/180; // angle at inlet section
epsilon = 4; // exit area to throat area ratio
```

```
R_e = R_t*Sqrt(epsilon); // exit radius
Rc_div = 0.4*R_t; // radius of curvature just after throat in diverging section
Rc_conv = 1.5*R_t; // radius of curvature of converging section

a = 1.280e+00; // coefficients for parabola
b = -9.650e-01;
c = -2.041e-01;

n_parabola = 100; // number of points to be defined on parabola

x_n = Rc_div*Sin(theta_n); // coordinates of point N
y_n = R_t + Rc_div*(1-Cos(theta_n));
x_i = -Rc_conv*Sin(theta_i); // coordinates of inlet section
y_i = R_t + Rc_conv*(1-Cos(theta_i));
x_e = a*R_e^2 + b*R_e + c; // x coordinate of exit section
revolution_angle = 3*Pi/180; // body's revolution angle

// converging section (circular arc)
Point(100) = {0, R_t+Rc_conv, 0}; // arc center
Point(1) = {x_i, y_i, 0};
Point(2) = {0, R_t, 0};
Point(3) = {x_i, 0, 0};
Point(4) = {0, 0, 0};
Circle(1) = {1, 100, 2};
Line(2) = {3,4};
Line(3) = {3,1};
Line(4) = {4,2};
Curve Loop(1) = {2,4,-1,-3};

// circular arc after throat
Point(101) = {0, R_t+Rc_div, 0}; // arc center
Point(5) = {x_n, y_n, 0};
Point(6) = {x_n, 0, 0};
Circle(5) = {2, 101, 5};
Line(6) = {4,6};
Line(7) = {6,5};
Curve Loop(2) = {6,7,-5,-4};

// parabolic portion
/// define points on the parabolic section
For i In {1:n_parabola}
    y = y_n + (R_e-y_n)*i/n_parabola;
    Point(200+i-1) = {a*y^2 + b*y + c, y, 0};
EndFor
BSpline(8) = {5,200:200+n_parabola-1};
Point(7) = {x_e, 0, 0};
Line(9) = {6,7};
Line(10) = {7,200+n_parabola-1};
Curve Loop(3) = {9,10,-8,-7};



// plane surfaces, transfinite and recombine; extrude and physical entities
Transfinite Curve{1} = 41 Using Bump 0.1;
```

```
Transfinite Curve{2} = 41 Using Bump 0.15;
Transfinite Curve{5,6} = 21;
Transfinite Curve{8,9} = 31 Using Progression 1.1;
Transfinite Curve{3,4,7,10} = 31;
For i In {1:3}
    Plane Surface(i) = {i};
    Transfinite Surface{i};
    Recombine Surface{i};
    out~{i}[] = Extrude{
        {1,0,0},
        {0,0,0},
        revolution_angle
    }{
        Surface{i};
        Layers{1};
        Recombine;
    };
EndFor
// rotate all entities for symmetry about xy plane
Rotate{
    {1,0,0},
    {0,0,0},
    -0.5*revolution_angle
}{
    Point{:}; Curve{:}; Surface{:}; Volume{:};
}
Physical Volume("volume", 1) = {};
Physical Surface("back", 1) = {};
Physical Surface("front", 2) = {};
Physical Surface("inflow", 3) = {};
Physical Surface("outflow", 4) = {};
Physical Surface("wall", 5) = {};
For i In {1:3}
    Physical Volume(1) += {out~{i}[1]};
    Physical Surface(1) += {i};
    Physical Surface(2) += {out~{i}[0]};
    Physical Surface(5) += {out~{i}[3]};
EndFor
Physical Surface(3) += {out_1[4]};
Physical Surface(4) += {out_3[2]};

Mesh 3;
```

Listing 3: Gmsh code for generating Conical nozzle mesh

```
/*
Point numbering convention:
    0xx: nozzle contour (converging and diverging), except parabolic profile
    1xx: temporary points
    2xx: parabolic profile spline
Line numbering convention:
    0xx: nozzle contours, including parabolic profile spline
Curve loops, surface convention:
    0xx: all
```

```
*/

R_t = 1; // throat radius
theta_conical = 15*Pi/180; // half cone angle
theta_n = theta_conical;
theta_i = 75*Pi/180; // angle at inlet section
epsilon = 4; // exit area to throat area ratio
R_e = R_t*Sqrt(epsilon); // exit radius
Rc_div = 0.4*R_t; // radius of curvature just after throat in diverging section
Rc_conv = 1.5*R_t; // radius of curvature of converging section

x_n = Rc_div*Sin(theta_n); // coordinates of point N
y_n = R_t + Rc_div*(1-Cos(theta_n));
x_i = -Rc_conv*Sin(theta_i); // coordinates of inlet section
y_i = R_t + Rc_conv*(1-Cos(theta_i));
x_e = x_n + (R_e-y_n)/Tan(theta_conical); // x coordinate of exit section
revolution_angle = 3*Pi/180; // body's revolution angle

// converging section (circular arc)
Point(100) = {0, R_t+Rc_conv, 0}; // arc center
Point(1) = {x_i, y_i, 0};
Point(2) = {0, R_t, 0};
Point(3) = {x_i, 0, 0};
Point(4) = {0, 0, 0};
Circle(1) = {1, 100, 2};
Line(2) = {3,4};
Line(3) = {3,1};
Line(4) = {4,2};
Curve Loop(1) = {2,4,-1,-3};

// circular arc after throat
Point(101) = {0, R_t+Rc_div, 0}; // arc center
Point(5) = {x_n, y_n, 0};
Point(6) = {x_n, 0, 0};
Circle(5) = {2, 101, 5};
Line(6) = {4,6};
Line(7) = {6,5};
Curve Loop(2) = {6,7,-5,-4};

// conical portion
/// define points on the parabolic section
Point(7) = {x_e, R_e, 0};
Point(8) = {x_e, 0, 0};
Line(8) = {5,7};
Line(9) = {6,8};
Line(10) = {8,7};
Curve Loop(3) = {9,10,-8,-7};



// plane surfaces, transfinite and recombine; extrude and physical entities
Transfinite Curve{1} = 41 Using Bump 0.1;
Transfinite Curve{2} = 41 Using Bump 0.15;
Transfinite Curve{5,6} = 11;
```

```
Transfinite Curve{8,9} = 41 Using Progression 1.1;
Transfinite Curve{3,4,7,10} = 31;
For i In {1:3}
    Plane Surface(i) = {i};
    Transfinite Surface{i};
    Recombine Surface{i};
    out~{i}[] = Extrude{
        {1,0,0},
        {0,0,0},
        revolution_angle
    }{
        Surface{i};
        Layers{1};
        Recombine;
    };
EndFor
// rotate all entities for symmetry about xy plane
Rotate{
    {1,0,0},
    {0,0,0},
    -0.5*revolution_angle
}{
    Point{:}; Curve{:}; Surface{:}; Volume{:};
}
Physical Volume("volume", 1) = {};
Physical Surface("back", 1) = {};
Physical Surface("front", 2) = {};
Physical Surface("inflow", 3) = {};
Physical Surface("outflow", 4) = {};
Physical Surface("wall", 5) = {};
For i In {1:3}
    Physical Volume(1) += {out~{i}[1]};
    Physical Surface(1) += {i};
    Physical Surface(2) += {out~{i}[0]};
    Physical Surface(5) += {out~{i}[3]};
EndFor
Physical Surface(3) += {out_1[4]};
Physical Surface(4) += {out_3[2]};

Mesh 3;
```

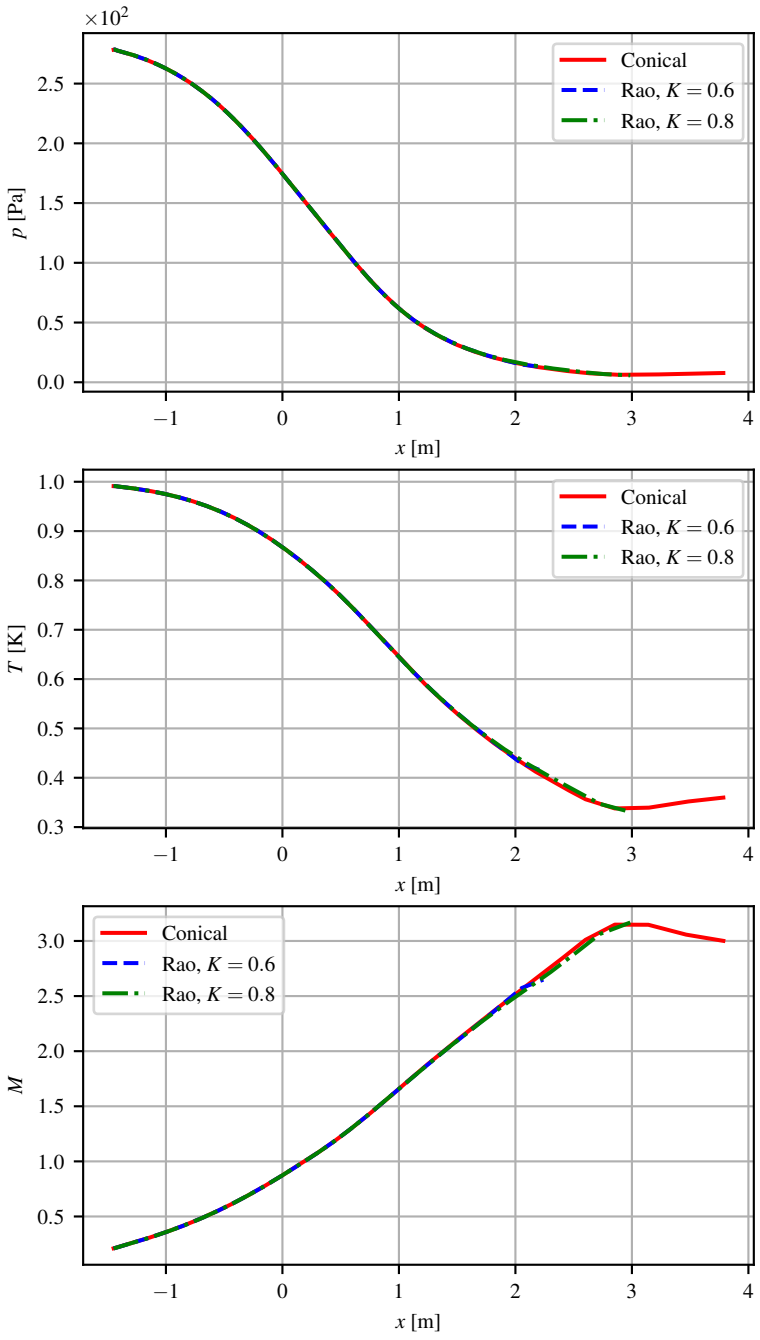


Figure 1: Pressure, Temperature and Mach number variation along the nozzle.

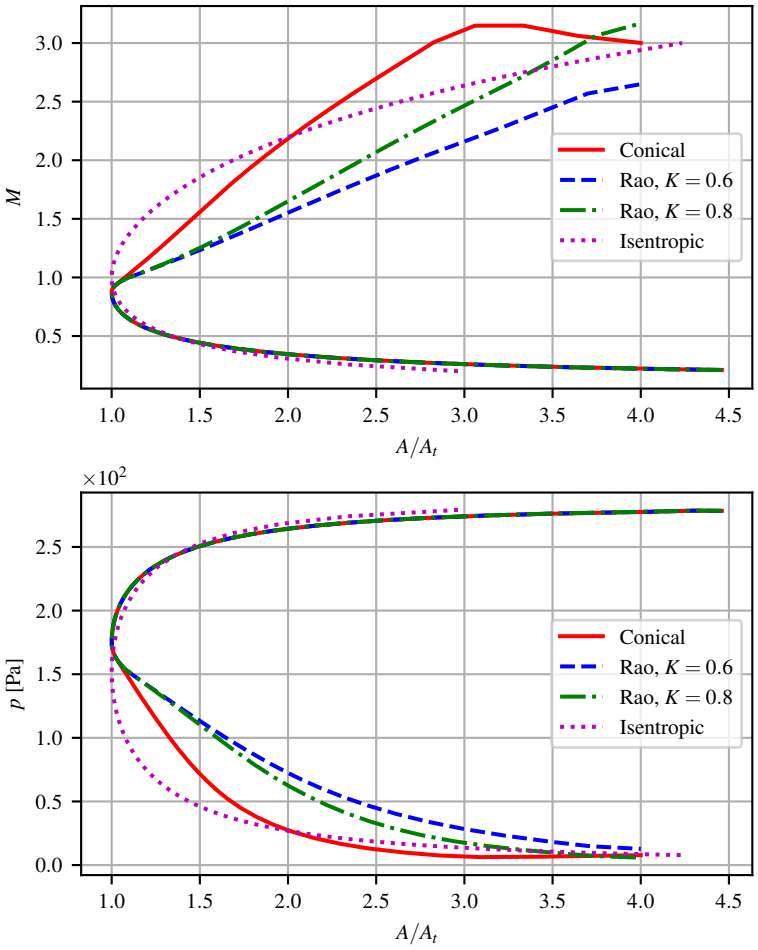
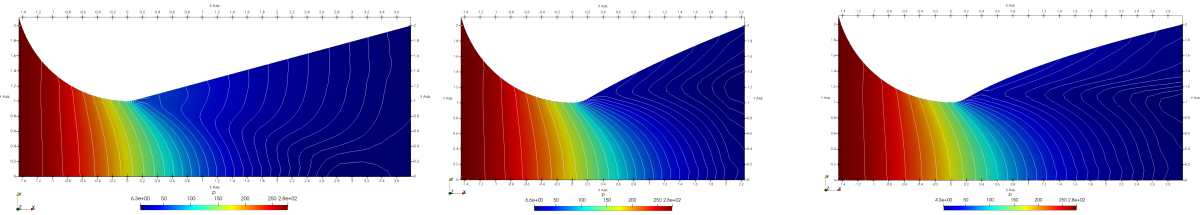


Figure 2: Mach number and pressure variation with the area ratio.



(a) Conical (b) Rao, $K = 0.6$ (c) Rao, $K = 0.8$

Figure 3: Pressure surface plot with 30 Mach number contours. Clearly, the Mach number contours are not radial and there is significant multidimensional nature to the flow. Quasi-1D analysis will fail here.