Scilab Textbook Companion for Modern Compressible Flow With Historical Perspective by John D Anderson[1]

Created by
Davis Jose
MCA
Mathematics
CUSAT
College Teacher
None
Cross-Checked by
None

July 31, 2019

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Modern Compressible Flow With Historical Perspective

Author: John D Anderson

Publisher: Mcgraw-hill Education

Edition: 3

Year: 2002

ISBN: 978-0072424430

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

Chapter 1

compressible flow some history and introductory thoughts

Scilab code Exa 1.1 `Fractional change in pressure`

```
clc
// Example 1.1.py
// Consider the low-speed flow of air over an airplane wing at standard
// sea level conditions the free-stream velocity far ahead of the wing
// is 100 mi/h. The flow accelerates over the wing, reaching a maximum
// velocity of 150 mi/h at some point on the wing. What is the percentage
// pressure change between this point and the free stream//


// Variable declaration
rho = 0.002377      // density at sea level (slug/ft^3)
p_1 = 2116.0        // pressure at sea level (lb/ft^2)
```

```
v_1 = 100.0          // velocity far ahead of the wing (mi/h)
v_2 = 150.0          // velocity at some point on the wing (mi/h)

// Calculations

u_1 = v_1 * 88.0/60.0 // converting v_1 in ft/s
u_2 = v_2 * 88.0/60.0 // converting v_2 in ft/s

delP = 0.5*rho*(u_2*u_2 - u_1*u_1) // p_1 - p_2 from Bernoulli's equation
fracP = delP/p_1 // fractional change in pressure with respect to freestream

// Result
printf("\n Fractional change in pressure is %.3f or %.1f percent", fracP, fracP*100)
```

Scilab code Exa 1.2 Total mass stored

```
clc
// Example 1.2.py
// A pressure vessel that has a volume of 10 m^3 is used to store high
// pressure air for operating a supersonic wind tunnel. If the air pressure
// and temperature inside the vessel are 20 atm and 300 K, respectively,
// what is the mass of air stored in the vessel//

// Variable declaration
V = 10               // volume of vessel (m^3)
p = 20.0             // pressure (atm)
T = 300              // temperature (K)
```

```
R = 287.0            // gas constant (J/Kg/K)

// Calculations
p = p * 101000.0     // units conversion to N/m^2
rho = p/R/T          // from ideal gas equation of
   state
m = V * rho          // total mass volume * density


// Result
printf("\n Total mass stored is %.1f Kg", m)
```

Scilab code Exa 1.3 Isothermal compressibility for air

```
clc
// Example 1.3.py
// Calculate the isothermal compressibility for air
   at a pressure of 0.5 atm.

// Variable declaration
p = 0.5             // pressure (atm)
p_si = 0.5*101325   // pressure (N/m^2)
p_eng = 0.5*2116    // pressure (lb/ft^2)

// Calculations
tau_atm = 1/p         // isothermal compressibility in
    atm^-1
tau_si = 1/p_si       // isothermal compressibility in
    m^2/N
tau_eng = 1/p_eng     // isothermal compressibility in
    ft^2/lb

// Result
printf("\n Isothermal compressibility for air at %.1
```

```
f atm is %.2f atm^-1 or %.2e m^2/N or %.2e ft^2/lb", p, tau_atm, tau_si, tau_eng)
```

Scilab code Exa 1.4 Total internal energy

```
clc
// Example 1.4.py
// For thre pressure vessel in Example 1.2, calculate the total internal
// energy of the gas stored in the vessel.

// Variable declaration from example 1.2
V = 10                  // volume of vessel (m^3)
p = 20.0                // pressure (atm)
T = 300                 // temperature (Kelvin)

R = 287.0               // gas constant (J/Kg/K)
gamma1 = 1.4             // ratio of specific heats for air

// Calculations
cv = R / (gamma1-1) // specific heat capacity at constant volume(J/Kg K)
e = cv * T              // internal energy per Kg (J/Kg)
p = p * 101000.0        // units conversion to N/m^2
rho = p/R/T             // from ideal gas equation of state (Kg/m^3)
m = V * rho             // total mass = volume * density (Kg)
E = m*e                 // total energy in J

// Result
printf("\n Total internal energy is %.2e J", E)
```

Scilab code Exa 1.5 Total change in entropy

```
clc
// Example 1.5.py
// Consider the air in the pressure vessel in Example 1.2. Let us now heat
// the gas in the vessel. Enough heat is added to increase the temperature
// to 600 K. Calculate the change in entropy of the air inside the vessel.

// Variable declaration from example 1.2
V = 10                  // volume of vessel (m^3)
p = 20.0                // pressure (atm)
T = 300.0               // initial temperature (K)
T2 = 600.0              // final temperature (Kelvin)
R = 287.0               // gas constant (J/Kg/K)
gamma1 = 1.4             // ratio of specific heats for air


// Calculations
p2_by_p = T2/T          // p2/p, at constant volume p/T = constant

cv = R / (gamma1-1) // specific heat capacity at constant volume (J/Kg K)
cp = cv + R             // specific heat capacity at constant pressure (J/Kg K)

p = p * 101000.0        // units conversion to N/m^2
rho = p/R/T             // from ideal gas equation of state (Kg/m^3)
m = V * rho             // total mass = volume * density
```

```
(Kg)

//
del_s = cp*log(T2/T) - R*log(p2_by_p) // change in entropy per unit mass (J/ Kg K)
total_del_s = m*del_s                 // total change in entropy (J/K)

// Result
printf("\n Total change in entropy is %.3e J/K", total_del_s)
```

Scilab code Exa 1.6 Pressure at the exit

```
clc
// Example 1.6.py
// Consider the flow through a rocket engine nozzle. Assume that the gas flow
// through the nozzle in an isentropic expansion of a calorically perfect gas.
// In the combustion chamber, the gas which results from the combustion of the
// rocket fuel and oxidizer is at a pressure and temperature of 15 atm and
// 2500 K, respectively, the molecular weight and specific heat at constant
// pressure of the combustion gas are 12 and 4157 J/kg K, respectively. The gas
// expands to supersonic speed through the nozzle, with temperature of 1350 K at
// the nozzle exit. Calculate the pressure at the exit.


// Variable declaration
```

```
pc = 15.0            // pressure combustion chamber (
    atm)
Tc = 2500.0          // temperature combustion chamber
    (K)
mol_wt = 12.0        // molecular weight (gm)
cp = 4157.0          // specific heat at constant
    pressure (J/Kg/K)

Tn  = 1350.0         // temperature at nozzle exit (K)

// Calculations
R = 8314.0/mol_wt   // gas constant = R_prime/mo_wt,
    R_prime = 8314 J/K
cv = cp - R          // specific heat at constant
    volume (J/Kg/K)
gamma1 = cp/cv        // ratio of specific heat

pn_by_pc = (Tn/Tc** gamma1/(gamma1-1)) // ratio of
    pressure for isentropic process** pn/pc

pn = pc * pn_by_pc // pn = pc * pn/pc

// Result
printf("\n Pressure at the exit is %.3f atm", pn)
```

Scilab code Exa 1.7 Total Drag per unit span

```
clc
// Example 1.7.py
// A flat plate with a chord length of 3 ft and an
    infinite span(perpendicular to
// the page in fig 1.5) is immersed in a Mach 2 flow
    at standard sea level
// conditions at an angle of attack of 10 degrees.
    The pressure distribution
```

```
// over the plate is as follows: upper surface, p2=constant=1132 lb/ft^2 lower
// surface, p3=constant=3568 lb/ft^2. The local shear stress is given by tau_w =
// 13/xeta^0.2, where tau_w is in pounds per square feet and xeta is the distance
// in feet along the plate from the leading edge. Assume the distribution of
// tau_w over the top and bottom surfaces is the same. Both the pressure and
// shear disributions are sketched qualitatively in fig. 1.5. Calculate the lift
// and drag per unit span on the plate.

//

// Variable declaration
M1 = 2.0          // mach number freestream
p1 = 2116.0       // pressure at sea level (in lb/ft^2)
l = 3.0           // chord of plate (in ft)
alpha = 10.0      // angle of attack in degrees

p2 = 1132.0       // pressure on the upper surface (in lb/ft^2)
p3 = 3568.0       // pressure on the lower surface (in lb/ft^2)

// Calculations

// assuming unit span

pds = -p2*l + p3*l    // integral p.ds from leading edge to trailing edge (in lb/ft)

L = pds*cos(alpha*%pi/180.0) // lift per unit span (in lb/ft), alpha is converted to radians

```

```
Dw = pds*sin(alpha*%pi/180.0) // pressure drag per unit span (in lb/ft), alpha is converted to radians

Df = 16.25 * (l** 4.0/5.0) // skin friction drag per unit span (in lb/ft)
                              // from integral tau.d(xeta)

Df = 2 * Df * cos(alpha*%pi/180.0) // since skin friction acts on both the side

D = Df + Dw                   // total drag per unit span (in lb/ft)
// Result
printf("\n Total Lift per unit span = %.0f lb", L)

printf("\n Total Drag per unit span = %.0f lb", D)
```

Chapter 3

one dimentional flow

Scilab code Exa 3.2 Velocity at the exit of the rocket nozzle

```
clc
// Example 3.2.py
// Return to Example 1.6, Calculate the Mach Number and velocity at the exit of the rocket
// nozzle.

// Variable declaration from example 1.6
pc = 15.0           // pressure combustion chamber (atm)
Tc = 2500.0         // temperature combustion chamber (K)
mol_wt = 12.0       // molecular weight (gm)
cp = 4157.0         // specific heat at constant pressure (J/Kg/K)

Tn  = 1350.0        // temperature at nozzle exit (K)

// Calculations
R = 8314.0/mol_wt   // gas constant = R_prime/mo_wt, R_prime = 8314 J/K
cv = cp - R         // specific heat at constant
```

```
   volume (J/Kg k)
gamma1 = cp/cv       // ratio of specific heat

pn_by_pc = (Tn/Tc** gamma1/(gamma1-1)) // ratio of
   pressure for isentropic process** pn/pc

Mn = (2/(gamma1-1)*((1/pn_by_pc**(gamma1-1)/gamma1)
   - 1)** 0.5) // Mach number at exit** from
   isentropic flow relation

an = (gamma1*R*Tn** 0.5)    // Speed of sound at exit
    (m/s)
Vn = Mn*an                   // Velocity at exit (m/s
   )


// Result
printf("\n Mach number at the exit of the rocket
   nozzle is %.3f",(Mn))

printf("\n Velocity at the exit of the rocket nozzle
    is %.1f m/s",(Vn))
```

Scilab code Exa 3.3 Percentage change in density

```
clc
// Example 3.3.py
// Return to Example 1.1, calculate the percentage
   density change between the given point
// on the wing and the free-stream, assuming
   compressible flow.

// Variable declaration from example 1.1
rho_1 = 0.002377    // density at sea level (slug/ft
   ^3)
```

```
T_1 = 519.0          // temperature at sea level (R)
v_1 = 100.0          // velocity far ahead of the wing (mi/h)
v_2 = 150.0          // velocity at some point on the wing (mi/h)
gamma1 = 1.4          // ratio of specific heat capacity for air
R = 1716.0           // gas constant (ft lbf/slug/R)

// Calculations
cp = gamma1*R/(gamma1-1) // specific heat capacity at constant pressure (ft lb/ slug / R)
u_1 = v_1 * 88.0/60.0   // converting v_1 in ft/s
u_2 = v_2 * 88.0/60.0   // converting v_2 in ft/s

T_2 = T_1 + (u_1*u_1 - u_2*u_2)/cp/2.0       // temperature at a point from energy equation  (R)

rho_2_by_rho_1 = ((T_2/T_1)** 1/(gamma1-1))// density ratio from isentropic flow relation

rho_2 = rho_2_by_rho_1 * rho_1               // density at the point (slug/ ft^3)

delrho = rho_1 - rho_2                       // change in density (slug/ ft^3)
fracrho = delrho/rho_1                       // fractional change in density

// Result
printf("\n Percentage change in density is %.1f",(fracrho*100))
```

Scilab code Exa 3.4 `downstream pressure`

```
clc
// Example 3.4.py
// A normal shock wave is standing in the test section of a supersonic wind tunnel.
// Upstream of the wave, M1 = 3, p1 = 0.5 atm, and T1 = 200 K. Find M2, p2, T2 and
// u2 downstream of the wave


// Variable declaration from example 1.1
M1 = 3.0            // upstream mach number
p1 = 0.5            // upstream pressure (atm)
T1 = 200.0          // upstream temperature (K)
R = 287.0           // gas constant (J/Kg/K)
gamma1 = 1.4         // ratio of specific heats for air

// Calculations

// from shock relation (Table A2) for M1 = 3.0
// subscript 2 means downstream of the shock
p2_by_p1 = 10.33      // p2/p1
T2_by_T1 = 2.679      // T2/T1
M2 = 0.4752           // M2

p2 = p2_by_p1 * p1         // downstream pressure (atm)
T2 = T2_by_T1 * T1         // downstream temperature (K)
a2 = (gamma1*R*T2** 0.5) // speed of sound downstream of the shock (m/s)
u2 = M2*a2                 // downstream velocity (m/s)


// Result
printf("\n M2 = %.4f",(M2))

printf("\n p2 = %.3f atm",(p2))
```

```
printf("\n T2 = %.1f K",(T2))

printf("\n u2 = %.1f m/s",(u2))
```

Scilab code Exa 3.6 Mach number

```
clc
// Example 3.6.py
// Consider a point in a supersonic flow where the
   static pressure is 0.4 atm. When
// a pitot tube is inserted in the  at this point,
   the pressure measured by the
// pitot tube is 3 atm. Calculate the mach number at
    this point.

// Variable declaration
p1 = 0.4         // static pressure (in atm)
po2 = 3.0        // pressure measured by the pitot
   tube (in atm)

//Calculations
// from table A2 for po2/p1 = 7.5
M1 = 2.35

// Results
printf("\n Mach number is %.2f",(M1))
```

Scilab code Exa 3.7 Change

```
clc
// Example 3.7.py
```

```
// For the normal shock that occurs in front of the pitot tube in Example 3.6,
// calculate the entropy change across the shock.


// Variable declaration
M1 = 2.34           // mach number from example 3.6
R = 1716.0          // gas constant (ft lbf/slug/R)

// Calculations
// subscript 2 means downstream of the shock

po2_by_po1 = 0.5615        // from shock table A2 for mach M1
//
dels = -R*log(po2_by_po1) // s2 − s1 (in lb/slug R)

// Result
printf("\n Change is entropy is %.1f lb/slug R", dels)
```

Scilab code Exa 3.10 `Heat require to choke the flow`

```
clc
// Example 3.10.py
// In example 3.9, how much heat per unit mass must be added to choke the flow//


// Variable declaration from example 3.9
To1 = 840                  // upstream total temperature (in K)
M1 = 3.0                   // upstream mach number
To1_by_Tostar = 0.6540 // To1/Tostar from Table A3
cp = 1004.5                // specific heat at constant
```

```
pressure for air (in J/Kg K)

// Calculations
Tostar = To1 / To1_by_Tostar // Tostar = To1 *
    Tostar/To1 (in K)

M2 = 1.0          // for choked flow
To2 = Tostar      // since M2 = 1.0

q = cp * (To2 - To1) // required heat = cp(To2 - To1
    ) (in J/kg)


// Result
printf("\n Heat require to choke the flow is %.2e J/
    kg", q)
```

Scilab code Exa 3.13 Length required to choke the flow

```
clc
// Example 3.13.py
// In example 3.12, what is the length of the duct
    required to choke the flow//


// Variable declaration from example 3.12
M1 = 3.0            // mach number
C1 = 0.5222         // C1 = 4*f*L1star/D
f = 0.005           // friction coefficient
D = 0.4             // diameter of pipe (in ft)

// Calculations
L1star = 0.5222 * D/4.0/f


```

```
// Result
printf("\n Length required to choke the flow is %.2f
    ft", L1star)
```

Chapter 4

oblique shock and expansion waves

Scilab code Exa 4.1 To2

```
clc
// Example 4.1.py
// A uniform supersonic stream with M1 = 3.0, p1 = 1 atm, T1 = 288 K encounters
// a compression corner which deflects the stream by an angle theta = 20 deg.
// Calculate the shock wave angle, and p2, T2, M2, po2 and To2 behind the shock
// wave.


// Variable declaration
M1 = 3.0            // upstream mach number
p1 = 1.0            // upstream pressure (in atm)
T1 = 288            // upstream temperature (in K)
theta = 20          // deflection (in degrees)

// Calculations
// subscript 2 means behind the shock
```

```

// from figure 4.5 from M1 = 3.0, theta = 20.0 deg.
beta1 = 37.5                    // shock angle (in degress)

// degree to radian conversion is done by multiplying by %pi/180
//
Mn1 = M1 * sin(beta1*%pi/180) // upstream mach number normal to the shock

// from Table A2 for Mn1 = 1.826
p2_by_p1 = 3.723              // p2/p1
T2_by_T1 = 1.551              // T2/T1
Mn2 = 0.6108
po2_by_po1 = 0.8011           // po2/po1

p2 = p2_by_p1 * p1            // p2 (in atm) = p2/p1 * p1
T2 = T2_by_T1 * T1            // T2 (in K) = T2/T1 * T1

M2 = Mn2/(sin((beta1-theta)*%pi/180)) // mach number behind the shock

// from A1 for M1 = 3.0
po1_by_p1 = 36.73
To1_by_T1 = 2.8

po2 = po2_by_po1 * po1_by_p1 * p1   // po2 (in atm) = po2/po1 * po1/p1 * p1
To1 =  To1_by_T1 * T1    // To2 (in atm) = To2/To1 * To1/T1 * T1
To2 = To1_by_T1 * T1    // To2 (in atm) = To2/To1 * To1/T1 * T1


// Result
```

```
printf("\n Shock wave angle %.2f degrees",(beta1))

printf("\n p2 = %.2f atm", p2)

printf("\n T2 = %.2f K", T2)

printf("\n M2 = %.2f ", M2)

printf("\n po2 = %.2f atm", po2)

printf("\n To2 = %.2f K", To2)
```

Scilab code Exa 4.2 comparison

```

clc
// Example 4.2.py
// In Example 4.1, the deflection angle is increased to theta = 30 degrees.
// Calculate the pressure and Mach number behind the wave, and compare these
// results with those of Example 4.1.


// Variable declaration
M1 = 3.0            // upstream mach number
p1 = 1.0            // upstream pressure (in atm)
T1 = 288            // upstream temperature (in K)
theta = 30          // deflection (in degrees)

// Calculations
// subscript 2 means behind the shock

// from figure 4.5 from M1 = 3.0, theta = 30.0 deg.
beta1 = 52.0                   // shock angle (in
```

```
degrees)

// degree to radian conversion is done by multiplying by %pi/180
//
Mn1 = M1 * sin(beta1*%pi/180) // upstream mach number normal to the shock

// from Table A2 for Mn1 = 2.364
p2_by_p1 = 6.276             // p2/p1
Mn2 = 0.5286

p2 = p2_by_p1 * p1           // p2 (in atm) = p2/p1 * p1
M2 = Mn2/(sin((beta1-theta)*%pi/180)) // mach number behind the shock


printf("\n Shock wave angle %.2f degrees",(beta1))

printf("\n p2 = %.3f atm", p2)

printf("\n M2 = %.2f ", M2)
printf("\n comparison")
```

Scilab code Exa 4.3 Shock wave angle

```

clc
// Example 4.3.py
// In Example 4.1, the free stream mach number is increased to 5.0.
// Calculate the pressure and Mach number behind the wave, and compare these
// results with those of Example 4.1.
```

```
// Variable declaration
M1 = 5.0            // upstream mach number
p1 = 1.0            // upstream pressure (in atm)
T1 = 288            // upstream temperature (in K)
theta = 20.0          // deflection (in degrees)

// Calculations
// subscript 2 means behind the shock

// from figure 4.5 from M1 = 5.0, theta = 20.0 deg.
beta1 = 30.0                    // shock angle (in degrees)

// degree to radian conversion is done by multiplying by %pi/180
//
Mn1 = M1 * sin(beta1*%pi/180) // upstream mach number normal to the shock

// from Table A2 for Mn1 = 2.5
p2_by_p1 = 7.125              // p2/p1
Mn2 = 0.513

p2 = p2_by_p1 * p1            // p2 (in atm) = p2/p1 * p1
M2 = Mn2/(sin((beta1-theta)*%pi/180)) // mach number behind the shock


printf("\n Shock wave angle %.2f degrees",(beta1))

printf("\n p2 = %.3f atm", p2)

printf("\n M2 = %.2f ", M2)
```

Scilab code Exa 4.5 Coefficient of pressure

```
clc
// Example 4.5.py
// Consider a 15 deg half angle wedge at zero angle
   of attack. Calculate the
// pressure coefficient on the wedge surface in a
   Mach 3 flow of air.


// Variable declaration
M1 = 3.0                // upstream mach number
theta = 15.0            // deflection (in degrees)
gamma1 = 1.4             // ratio of specific heats


// Calculations
// subscript 2 means behind the shock

// from figure 4.5 from M1 = 3.0, theta = 15.0 deg.
beta1 = 32.2                     // shock angle (in
   degress)

// degree to radian conversion is done by
   multiplying by %pi/180
//
Mn1 = M1 * sin(beta1*%pi/180) // upstream mach
   number normal to the shock

// from Table A2 for Mn1 = 1.6
p2_by_p1 = 2.82              // p2/p1

Cp = 2/(gamma1*M1*M1) * (p2_by_p1 - 1)

```

```
// Results
printf("\n Coefficient of pressure is %.3f",(Cp))
```

Scilab code Exa 4.6 Coefficient of drag

```
clc
// Example 4.6.py
// Consider a 15 deg half angle wedge at zero angle of attack in a Mach 3 flow of
// air. Calculate the drag coefficient. Assume that the pressure exerted over the
// base of the wedge, the base pressure, is equal to the free stream pressure.



// Variable declaration
M1 = 3.0              // upstream mach number
theta = 15.0          // deflection (in degrees)
gamma1 = 1.4          // ratio of specific heats


// Calculations
// subscript 2 means behind the shock

// from figure 4.5 from M1 = 3.0, theta = 15.0 deg.
beta1 = 32.2                   // shock angle (in degress)

// degree to radian conversion is done by multiplying by %pi/180
//
Mn1 = M1 * sin(beta1*%pi/180) // upstream mach number normal to the shock
```

```

// from Table A2 for Mn1 = 1.6
p2_by_p1 = 2.82              // p2/p1

cd1 = 4/(gamma1*M1*M1)*(p2_by_p1 - 1)*tan(theta*%pi/180)


// Results
printf("\n Coefficient of drag is %.3f",(cd1))
```

Scilab code Exa 4.7 Mach behind reflected shock M3

```
clc
// Example 4.7.py
// Consider a horizontal supersonic flow at Mach 2.8 with a static pressure and
// temperature of 1 atm and 519 R, respectively. This flow passes over a compr-
// ession corner with deflection angle of 16 degrees. The oblique shock generated
// at the corner propagates into the flow, and is incident on a horizontal wall,
// as shown in Fig. 4.15. Calculate the angle phi made by the reflected shock wave
// with respect to the wall, and the Mach number, pressure and temperature behind
// the reflected shock.


// Variable declaration
M1 = 2.8            // upstream mach number
p1 = 1.0            // upstream pressure (in atm)
T1 = 519.0          // upstream temperature (in R)
theta = 16.0        // deflection (in degrees)
```

```

// Calculations
// subscript 2 means behind the shock

// from figure 4.5 from M1 = 2.8, theta = 16.0 deg.
beta1_1 = 35.0                              // shock angle (in degress)

// degree to radian conversion is done by multiplying by %pi/180
//
Mn1 = M1 * sin(beta1_1*%pi/180)          // upstream mach number normal to the shock

// from Table A2 for Mn1 = 1.606
p2_by_p1 = 2.82                          // p2/p1
T2_by_T1 = 1.388                         // T2/T1
Mn2 = 0.6684


p2 = p2_by_p1 * p1                       // p2 (in atm) = p2/p1 * p1
T2 = T2_by_T1 * T1                       // T2 (in R) = T2/T1 * T1

M2 = Mn2/(sin((beta1_1-theta)*%pi/180)) // mach number behind the shock

// from figure 4.5 from M2 = 2.053, theta = 16.0 deg.
beta1_2 = 45.5                              // shock angle of reflected(in degress)

// degree to radian conversion is done by multiplying by %pi/180
Mn2 = M2 * sin(beta1_2*%pi/180)          // upstream mach number normal to the shock

```

```
// from Table A2 for Mn1 = 1.46
p3_by_p2 = 2.32                          // p3/p2
T3_by_T2 = 1.294                         // T3/T2
Mn3 = 0.7157


p3 = p3_by_p2 * p2                       // p3 (in atm) = p3/p2 * p2
T3 = T3_by_T2 * T2                       // T3 (in R) = T3/T2 * T2

phi = beta1_2 - theta                     // (in degrees)
M3 = Mn3/(sin((beta1_2-theta)*%pi/180)) // mach number behind the reflected shock




// Result
printf("\n phi %.2f degrees", phi)

printf("\n Pressure behind reflected shock, p3 = %.2f atm", p3)

printf("\n Temperature behind reflected shock, T3 = %.2f R", T3)

printf("\n Mach behind reflected shock, M3 = %.2f ", M3)
```

Scilab code Exa 4.8 Angle rearward

```
clc
// Example 4.8.py
```

```
// A uniform supersonic stream with M1 = 1.5, p1 = 1700 lb/ft^2, and T1 = 460.0 R
// encounters an expansion corner which deflects the stream by and angle theta_2
// = 20 degrees. Calculate M2, p2, T2, po2, To2, and the angles the forward and
// rearward Mach lines make with respect to the upstream flow direction.


// Variable declaration
M1 = 1.5             // upstream mach number
p1 = 1700.0          // upstream pressure (in lb/ft^2)
T1 = 460.0           // upstream temperature (in R)
theta_2 = 20.0       // deflection (in degrees)


// Calculations
// subscript 2 means after the expansion fan

// from Table A5 for M1 = 1.5
v1 = 11.91           // (in degrees)
mu1 = 41.81          // (in degrees)

v2 = v1 + theta_2

// from Table A5, for v2 = 31.91
M2 = 2.207           // Mach behind the expansion fan
mu2 = 26.95          // (in degrees)

// from Table A1 for M1 = 1.5
po1_by_p1 = 3.671 // po1/p1
To1_by_T1 = 1.45  // To1/T1

// from Table A1 for M2 = 2.207
po2_by_p2 = 10.81 // po2/p2
To2_by_T2 = 1.974 // To2/T2

```

```
p2 = 1/po2_by_p2 * po1_by_p1 * p1 // p2 (in lb/ft^2) = p2/po2 * po2/po1 * po1/p1 * p1 and po2 = po1
T2 = 1/To2_by_T2 * To1_by_T1 * T1 // T2 (in R) = T2/To2 * To2/To1 * To1/T1 * T1 and To2 = To1


angle_forward  = mu1                   // angle of forward ray (in degrees)
angle_rearward = mu2 - theta_2         // angle of backward ray (in degrees)

po2 = po1_by_p1 * p1          // po2 (in lb/ft^2) = po1/p1 * p1
To2  = To1_by_T1 * T1          // To2 (in R) = To1/T1 * T1
 po1 = po1_by_p1 * p1          // po2 (in lb/ft^2) = po1/p1 * p1
 To1 = To1_by_T1 * T1          // To2 (in R) = To1/T1 * T1

// Result
printf("\n M2 = %.3f", M2)

printf("\n p2 = %.2f lb/ft^2", p2)

printf("\n T2 = %.2f deg R", T2)

printf("\n po2 = %.2f lb/ft^2", po2)

printf("\n To2 = %.2f deg R", To2)

printf("\n Angle forward = %.2f degrees", angle_forward)

printf("\n Angle rearward = %.2f degrees", angle_rearward)
```

Scilab code Exa 4.9 Freestream mach number

```
clc
// Example 4.9.py
// Consider the arrangement shows in fig. 4.29. A 15 degree half angle diamond
// wedge airfoil is in supersonic flow at zero angle of attack. A pitot tube is
// inserted into the flow at the location shown in fig 4.29. The pressure measured
// by the Pitot tube is 2.596 atm. At point a on the backface, the pressure is 0.1
// atm. Calculate the freestream Mach number M1.

//

// Variable declaration
theta = 15.0     // wedge angle/deflection (in degrees)
po4 = 2.596      // measured pressure (in atm)
p3 = 0.1         // pressure at point a (in atm)

// Calculations

po4_by_p3 = po4/p3

// from Table A 2 for po4/p3 = 25.96
M3 = 4.45
v3 = 71.27
v2 = v3 - 2*theta

// from Table A 5, for v2 = 41.27 degrees
M2 = 2.6
// Mn2 = M2*sin((beta1-theta)*%pi/180)  @equation 1
```

```

// Guessing

// Guess 1
M1 = 4.0                                  // Guess for freestream number
beta1 = 27.0                               // from fig 4.5 (in degrees)
Mn1 = M1*sin(beta1*%pi/180)                 // mach number normal to shock

// from Table A2 for Mn1 = 1.816
Mn2 = 0.612
// but Mn2 from equation 1 is 0.54

// Guess 2
M1 = 4.5                                  // Guess for freestream number
beta1 = 25.5                               // from fig 4.5 (in degrees)
Mn1 = M1*sin(beta1*%pi/180)                 // mach number normal to shock

// from Table A2 for Mn1 = 1.937
Mn2 = 0.588
// but Mn2 from equation 1 is 0.47

// Guess 3
M1 = 3.5                                  // Guess for freestream number
beta1 = 29.2                               // from fig 4.5 (in degrees)
Mn1 = M1*sin(beta1*%pi/180)                 // mach number normal to shock

// from Table A2 for Mn1 = 1.71
Mn2 = 0.638
// but Mn2 from equation 1 is 0.638
```

```




// Result
printf("\n Freestream mach number is %.1f", M1)
```

Scilab code Exa 4.10 Drag coefficient

```
clc
// Example 4.10.py
// Consider an infinitely this flat plate at 5 degrees angle of attack in a Mach
// 2.6 free stream. Calculate the lift and drag coefficients.

//

// Variable declaration
alpha = 5.0     // angle of attack in degrees (in degrees)
M1 = 2.6        // freestream mach number
gamma1 = 1.4     // ratio of specific heats

// Calculations

// from table A5 for M1 = 2.6
v1 = 41.41       // (in degrees)
v2 = v1 + alpha // (in degrees)
// from table A5 for v2 = 46.41 deg
M2 = 2.85
// from A1 for M1 = 2.6
po1_by_p1 = 19.95
// from A1 for M2 = 2.85
po2_by_p2 = 29.29
```

```
p2_by_p1 = 1/po2_by_p2 * po1_by_p1 // p2/p1 = p2/po2
    * po2/po1 * po1/p1 and po2 = po1

// from theta-beta1-M diagram for M1 = 2.6
theta = 5.0                // deflection (in degrees)
beta1 = 26.5                // shock angle (in
   degrees)
Mn1 = M1*sin(beta1*%pi/180) // mach number normal to
    the shock

// from table A2 for Mn1 = 1.16
p3_by_p1 = 1.403           // p3/p1

cl = 2.0/(gamma1*M1*M1)*(p3_by_p1 - p2_by_p1)*cos(
   alpha*%pi/180) // coefficient of lift
cd1 = 2.0/(gamma1*M1*M1)*(p3_by_p1 - p2_by_p1)*sin(
   alpha*%pi/180) // coefficient of drag


// Results
printf("\n Lift coefficient : %.3f",(cl))

printf("\n Drag coefficient : %.4f",(cd1))
```

Chapter 5

quasi one dimensional flow

Scilab code Exa 5.1 Converging section

```
clc
// Example 5.1.py
// Consider the subsonic-supersonic flow through a
   convergent-divergent nozzle. The
// reservoir pressure and temperature are 10 atm and
    300 K, repectively. There are
// two locations in the nozzle where A/Astar = 6,
   one in the convergent section and
// the other in the divergent section. At each
   location calculate M, p, T, u.

// Variable declaration
po = 10.0          // reservoir pressure (in atm)
To = 300.0         // reservoir temperature (in K)
A_by_Astar = 6.0   // area ratio
gamma1 = 1.4         // ratio of specific heat
R = 287.0          // gas constant (in J/ Kg K)

// Calculations

// from table A1 for subsonic flow with A/Astar =
```

```
6.0
Msub = 0.097                 // mach number in converging section
po_by_p = 1.006              // po/p in converging section
To_by_T = 1.002              // To/T in converging section

psub = 1 / po_by_p * po         // pressure (in atm) in converging section
Tsub = 1 / To_by_T * To         // temperature (in K) in converging section
asub = (gamma1*R*Tsub** 0.5) // speed of sound (in m/s) in converging section
usub = Msub*asub                // velocity (in m/s) in converging section

// from table A1 for supersonic flow with A/Astar = 6.0
Msup = 3.368                 // mach number in diverging section
po_by_p = 63.13              // po/p in diverging section
To_by_T = 3.269              // To/T in diverging section

psup = 1 / po_by_p * po         // pressure (in atm) in diverging section
Tsup = 1 / To_by_T * To         // temperature (in K) in diverging section
asup = (gamma1*R*Tsup** 0.5) // speed of sound (in m/s) in diverging section
usup = Msup*asup                // velocity (in m/s) in diverging section


// Results
printf("\n Converging section")
```

```
printf("\n M = %.3f", Msub)

printf("\n p = %.2f atm", psub)

printf("\n T = %.1f K", Tsub)

printf("\n u = %.2f m/s", usub)


printf("\n Divering section")
printf("\n M = %.3f", Msup)

printf("\n p = %.4f atm", psup)

printf("\n T = %.2f K", Tsup)

printf("\n u = %.2f m/s", usup)
```

Scilab code Exa 5.2 Reservoir temperature required

```
clc
// Example 5.2.py
// A supersonic wind tunnel is designed to produce
   Mach 2.5 flow in the test section
// with standard sea level conditions. Calculate the
    exit area ratio and reservoir
// conditions necessary to achieve these design
   conditions.

// Variable declaration
Me = 2.5            // exit mach number
pe = 1.0            // sea level pressure (in atm)
Te = 288.0          // sea level temperature (in K)
// Calculations

```

```
// from table A1 for Me = 2.5
Ae_by_Astar = 2.637          // Ae/Astar
po_by_pe = 17.09             // po/p
To_by_Te = 2.25              // To/T

po = po_by_pe * pe           // reservoir pressure (in atm)
To = To_by_Te * Te           // reservoir temperature (in K)

// Results
printf("\n Area ratio required %.3f", Ae_by_Astar)

printf("\n Reservoir pressure required %.2f atm", po)

printf("\n Reservoir temperature required %.1f K", To)
```

Scilab code Exa 5.3 Area of the exit

```
clc
// Example 5.3.py
// Consider a rocket engine burning hydrogen and oxygen combustion chamber temper-
// ature and pressure are 3571 K and 25 atm, respectively. The molecular weight of
// the chemically reacting gas in the combustion chamber is 16.0 and gamma1 = 1.22.
// The pressure at the exit of the convergent-divergent rocket nozzle is 1.174*10^-2
// atm. The area of the throat is 0.4 m^2. Assuming a calorifically perfect gas,
// calculate (a) the exit mach number (b) the exit velocity (c) the mass through the
```

```
// nozzle and (d) the area of the exit.

// Variable declaration
po = 25.0             // combustion chamber pressure (in atm)
To = 3571.0           // combustion chamber temperature (in K)
pe = 1.174e-2         // pressure at the exit of the nozzle (in atm)
Astar = 0.4           // throat area (in m^2)
gamma1 = 1.22          // ratio of specific heats
mol_wt = 16.0         // molecular weight (in gms)

// Calculations

// part (a)
Me = (2/(gamma1-1) *((po/pe**(gamma1-1)/gamma1) - 1)** 0.5) // Exit mach number

// part (b)
Te_by_To = (pe/po** (gamma1-1)/gamma1) // Te/To
Te = Te_by_To * To                     // exit temperature (in K)

R = 8314.0/mol_wt                      // gas constant (in J/Kg K)
ae = (gamma1*R*Te** 0.5)              // speed of sound at exit (in m/s)
ve = Me * ae                           // velocity at exit (in m/s)

// part (c)
rhoo = po*101325/R/To                            // density at reservoir (in Kg/m^3)
rhostar_by_rhoo = (2.0/(gamma1+1)**1/(gamma1-1)) // rhostar/rhoo
rhostar = rhostar_by_rhoo * rhoo                 // rhostar, throat density (in Kg/m^3)
```

```

Tstar_by_To = 2.0/(gamma1+1)                    // Tstar/To
Tstar = Tstar_by_To * To                        // Tstar, throat temperature (in K)
astar = (gamma1*R*Tstar** 0.5)                  // speed of sound at throat (in m/s)
mass = rhostar*Astar*astar                      // mass flow rate at throat (in Kg/s)

// part (d)
rhoe = pe*101325/R/Te   // density at exit (in Kg/m^3)
Ae = mass/rhoe/ve       // exit area (in m^2)

// Results

printf("\n Exit mach number %.2f", Me)

printf("\n Exit velocity %.2f m/s", ve)

printf("\n Mass flow rate %.2f Kg/s", mass)

printf("\n Area of the exit %.2f m^2", Ae)
```

Scilab code Exa 5.4 Mach number at throat

```
clc
// Example 5.4.py
// Consider the flow through a convergent-divergent duct with an exit to throat area
// ratio of 2. The reservoir pressure is 1 atm, and the exit pressure is 0.95 atm.
// Calculate the mach numbers at the throat and at the exit.
```

```
// Variable declaration
po = 1.0            // reservoir pressure (in atm)
pe = 0.95           // pressure at the exit (in atm)
Ae_by_At = 2.0  // ratio of exit to throat area

// Calculations
// from table A1 for po/pe = 1.053
Me = 0.28             // mach number at exit
Ae_by_Astar = 2.17  // nearest entry

At_by_Astar = 1 / Ae_by_At * Ae_by_Astar // At/Astar
    = At/Ae * Ae/Astar

// from table A1 for At/A* = 1.085
Mt = 0.72             // mach number at throat


// Results
printf("\n Mach number at exit %.2f", Me)

printf("\n Mach number at throat %.2f", Mt)
```

Scilab code Exa 5.5 Exit to reservoir required pressure ratio

```
clc
// Example 5.5.py
// Consider a convergent divergent duct with an exit
    to throat area ratio of 1.6.
// Calculate the exit to reservoir pressure ratio
   required to achieve sonic flow
// at the throat, but subsonic flow everywhere else.

// Variable declaration
Ae_by_At = 1.6  // ratio of exit to throat area
```

```

// Calculations

// since M = 1 at the throat Mt = Astar
// Ae/At = Ae/Astar = 1.6

// from table A1 for Ae/Astar = 1.6
po_by_pe = 1.1117        // po/pe
pe_by_po = 1/po_by_pe    // pe/po


// Results
printf("\n Exit to reservoir required pressure ratio is %.1f", pe_by_po)
```

Scilab code Exa 5.6 Exit to reservoir pressure ratio

```
clc
// Example 5.6.py
// Consider a convergent divergent nozzle with an exit to throat area ratio of 3.
// A normal shock wave is inside the divergent portion at a location where the local
// area ratio is A/At = 2.0. Calculate the exit to reservoir pressure ratio.

// Variable declaration
Ae_by_At = 3.0 // ratio of exit to throat area

// Calculations

// from table A1 for A/At = 2.0
M1 = 2.2                 // mach number in front the shock

```

```
// from table A2 for M1 = 2.2
M2 = 0.5471            // mach number behind the shock
po2_by_po1 = 0.6281   // stagnation pressure ratio accross the shock

// from table A1 for M2 = 0.5471
A2_by_A2star = 1.27   // A2/A2star
At_by_A2 = 1/2.0      // At/A2
Ae_by_A2star = Ae_by_At * At_by_A2 * A2_by_A2star // Ae/A2star = Ae/At * At/A2 * A2/A2star

// from table A1 for Ae/A2star = 1.905
Me = 0.32             // exit mach number
poe_by_pe = 1.074     // poe/pe

// po = po1 and poe = po2
pe_by_po = 1 / poe_by_pe * po2_by_po1 // pe/po = pe/poe * poe/po2 * po2/po1 * po1/po

// Results
printf("\n Exit to reservoir pressure ratio is %.3f", pe_by_po)
```

Scilab code Exa 5.7 Ratio of total pressure at the diffuser exit to the reservoir

```
clc
// Example 5.7.py
// Consider the wind tunnel described in example 5.2. Estimate the ratio of diffuser
// throat area to nozzle throat area required to allow the tunnel to start. Also,
// assuming that the diffuser efficiency is 1.2 after the tunnel has started, calculate
// the pressure ratio across the tunnel necessary for running i.e. calculate the ratio
```

```
// of total pressure at the diffuser exit to the reservoir pressure.

// Variable declaration

M = 2.5        // mach number before the shock
eta_d = 1.2    // diffuser efficiency

// Calculations

// from table for M = 2.5
po2_by_po1 = 0.499                  // po2/po1
At2_by_At1 = 1 / po2_by_po1         // At2/At1 = po1/po2

Pdo_by_po = eta_d * po2_by_po1      // pdo/po

// Results
printf("\n Ratio of diffuser throat area to nozzle throat area %.2f", At2_by_At1)

printf("\n Ratio of total pressure at the diffuser exit to the reservoir pressure, %.3f",(Pdo_by_po))
```