

Scilab Textbook Companion for
Fluid Mechanics And Thermodynamics Of
Turbo Machinery
by S. L. Dixon¹

Created by
Jagadeeshwar Goshika
B.Tech.
Others
IIT Bombay
College Teacher
None
Cross-Checked by
Bhavani Jalkrish

April 30, 2015

¹Funded 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: Fluid Mechanics And Thermodynamics Of Turbo Machinery

Author: S. L. Dixon

Publisher: Butterworth-Heinemann

Edition: 5

Year: 2005

ISBN: 0-7506-7870-4

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	4
2 Basic Thermodynamics and Fluid Mechanics	6
3 Two dimensional Cascades	10
4 Axial flow Turbines Two dimensional Theory	14
5 Axial flow Compressors and Fans	20
6 Three dimensional Flows in Axial Turbomachines	24
7 Centrifugal Pumps Fans and Compressors	28
8 Radial Flow Gas Turbines	34
9 Hydraulic Turbines	43
10 Wind Turbines	51

List of Scilab Codes

Exa 2.1	Ex 1	6
Exa 2.2	Ex 2	7
Exa 2.3	Ex 3	7
Exa 2.4	Ex 4	8
Exa 2.5	Ex 5	9
Exa 3.1	Ex 1	10
Exa 3.2	Ex 2	11
Exa 3.3	Ex 3	12
Exa 4.1	Ex 1	14
Exa 4.2	Ex 2	15
Exa 4.3	Ex 3	16
Exa 4.4	Ex 4	17
Exa 5.1	Ex 1	20
Exa 5.2	Ex 2	21
Exa 6.1	Ex 1	24
Exa 6.2	Ex 2	25
Exa 7.1	Ex 1	28
Exa 7.2	Ex 2	29
Exa 7.3	Ex 3	31
Exa 7.4	Ex 4	32
Exa 8.1	Ex 1	34
Exa 8.2	Ex 2	35
Exa 8.3	Ex 3	36
Exa 8.4	Ex 4	37
Exa 8.5	Ex 5	38
Exa 8.6	Ex 6	40
Exa 9.1	Ex 1	43
Exa 9.2	Ex 2	43

Exa 9.3	Ex 3	44
Exa 9.4	Ex 4	46
Exa 9.5	Ex 5	48
Exa 9.6	Ex 6	49
Exa 10.2	Ex 2	51
Exa 10.3	Ex 3	51
Exa 10.4	Ex 4	52
Exa 10.5	Ex 5	53
Exa 10.6	Ex 6	54
Exa 10.7	Ex 7	56
Exa 10.8	Ex 8	57
Exa 10.9	Ex 9	58
Exa 10.10	Ex 10	61

Chapter 2

Basic Thermodynamics and Fluid Mechanics

Scilab code Exa 2.1 Ex 1

```
1 clear all;
2 clc;
3 funcprot(0);
4
5 //given data
6 gamma = 1.4;
7 pi = 8; //pressure ratio
8 T01 = 300; //inlet temperature in K
9 T02 = 586.4; //outlet temperature in K
10
11 //Calculations
12 //Calculation of Overall Total to Total efficiency
13 Tot_eff = ((pi^((gamma-1)/gamma))-1)/((T02/T01)-1);
14
15 //Calculation of polytropic efficiency
16 Poly_eff = ((gamma-1)/gamma)*((log(pi))/log(T02/T01)
17 );
18 //Results
```

```

19 printf('The Overall total-to-total efficiency is %.2
    f.\n',Tot_eff);
20 printf('The polytropic efficiency is %.4f.',Poly_eff
    );

```

Scilab code Exa 2.2 Ex 2

```

1 clear all;
2 clc;
3 funcprot(0);
4
5 //given data
6 T01 = 1200; //Stagnation temperature at which gas
    enters in K
7 p01 = 4; //Stagnation pressure at which gas enters in
    bar
8 c2 = 572; //exit velocity in m/s
9 p2 = 2.36; //exit pressure in bar
10 Cp = 1.160*1000; //in J/kgK
11 gamma = 1.33
12
13 //calculations
14 T2 = T01 - 0.5*(c2^2)/Cp; //Calculation of exit
    temperature in K
15 Noz_eff = ((1-(T2/T01))/(1-(p2/p01)^((gamma-1)/gamma
    ))); //Nozzle efficiency
16
17 //Results
18 printf('Nozzle efficiency is %.4f.',Noz_eff);

```

Scilab code Exa 2.3 Ex 3

```

1 clear all;

```



```

2  clc;
3  funcprot(0);
4
5  //given data
6  cp = 0.6; //coefficient of pressure
7  AR = 2.13; //Area ratio
8  N_R1 = 4.66;
9
10 //calculations
11 cpi = 1 - (1/(AR^2));
12 Diff_eff = cp/cpi; //diffuser efficiency
13 theta = 2*(180/%pi)*atan((AR^0.5 - 1)/(N_R1)); //
    included cone angle
14
15 //Results
16 printf('cpi = %.2f.\n',cpi);
17 printf('The included cone angle can be found = %.2f
    deg. ',theta);

```

Scilab code Exa 2.4 Ex 4

```

1  clear all;
2  clc;
3  funcprot(0);
4
5  //given data
6  AR = 1.8; //Area ratio
7  cp = 0.6; //coefficient of pressure
8  N_R1 = 7.85;
9
10 //calculations
11 Theta = 2*(180/%pi)*atan((AR^0.5 - 1)/(N_R1)); //
    included cone angle
12 cpi = 1-(1/(AR^2));
13 Diff_eff = cp/cpi; //diffuser efficeincy

```

```
14
15 //Results
16 printf('The included cone angle can be found = %.1f
    deg.\n',Theta);
17 printf('cpi = %.2f.\n',cpi);
18 printf('Diffuser efficiency = %.2f.',Diff_eff);
```

Scilab code Exa 2.5 Ex 5

```
1 clear all;
2 clc;
3 funcprot(0);
4
5 //given data
6 AR = 2.0;//Area ratio
7 alpha1 = 1.059;
8 B1 = 0.109;
9 alpha2 = 1.543;
10 B2 = 0.364;
11 cp = 0.577;//coefficient of pressure
12
13 //calculations
14 cp = (alpha1 - (alpha2/(AR^2))) - 0.09;
15 Diff_eff = cp/(1-(1/(AR^2)));//Diffuser efficiency
16
17 //Results
18 printf('The diffuser efficiency = %.4f.',Diff_eff);
```

Chapter 3

Two dimensional Cascades

Scilab code Exa 3.1 Ex 1

```
1 clear all;
2 clc;
3 funcprot(0);
4
5 //function to calculate m and delta
6 function [m,delta] = func(a_1,alpha2,theta)
7     m = 0.23*(2*a_1)^2 + alpha2/500;
8     delta = m*theta;
9 endfunction
10
11 //given data
12 alpha1_ = 50; // in deg
13 alpha2_ = 20; // in deg
14 a_1 = 0.5; //percentage
15 s_1 = 1.0;
16 eps = 21; //in deg
17
18 //Calculations
19 theta = alpha1_ - alpha2_;
20 alpha21 = 20; //in deg
21 [m1,delta1] = func(a_1,alpha21,theta);
```

```

22 alpha22 = 28.1; //in deg
23 [m2,delta2] = func(a_1,alpha22,theta);
24 alpha23 = 28.6; //in deg
25 [m3,delta3] = func(a_1,alpha23,theta);
26 alpha1 = eps + alpha23;
27 i = alpha1 - alpha1_;
28 alphas = (180/%pi)*atan(0.5*(tan(alpha1*%pi/180) +
    tan(alpha23*%pi/180)));
29 CL = 2*(s_1)*cos(alphas*%pi/180)*(tan(alpha1*%pi
    /180) - tan(alpha23*%pi/180));
30
31 //Results
32 printf('The fluid deflection = %d deg.',eps);
33 printf('\n The fluid deviation = %.1f deg.',i);
34 printf('\n The ideal lift coefficient at the design
    point = %.2f',CL);

```

Scilab code Exa 3.2 Ex 2

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 s_1 = 1.0;
7 alpha1_ = 50; //in deg
8 alpha2_ = 20; //in deg
9 eps_ = 21; //in deg
10 i_ = -0.4; //in deg
11 i = 3.8; //in deg
12 CD = 0.017;
13 eps = 1.15*eps_;
14
15 //Calculations
16 alpha1 = alpha1_+i;

```

```

17 alpha2 = alpha1-eps;
18 alphas = (180/%pi)*atan(0.5*(tan(alpha1*%pi/180) +
    tan(alpha2*%pi/180)));
19 zeta = CD/((s_1)*(cos(alphas*%pi/180))^3);
20 Cf = 2*(tan(alpha1*%pi/180) - tan(alpha2*%pi/180));
21 eff_D = 1 - zeta/(Cf*tan(alphas*%pi/180));
22
23 // Results
24 printf('The tangential lift force coefficient = %.3f
    ',Cf);
25 printf('\n The diffuser efficiency = %d percentage.'
    ,eff_D*100);

```

Scilab code Exa 3.3 Ex 3

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 alpha1 = 58;//in deg
7 alpha2 = 44;//in deg
8 AVR = 1.0;
9
10 //Calculations
11 alphas = (180/%pi)*atan(0.5*(tan(alpha1*%pi/180) +
    tan(alpha2*%pi/180)));
12 zeta = (180/%pi)*atan(tan(alphas*%pi/180) - 0.213);
13 Cpi = 1-(cos(alpha1*%pi/180)/cos(alpha2*%pi/180))^2;
14 s_1 = 9*(0.567-Cpi);
15 theta = ((zeta-alpha2+1.1*(s_1)^(1/3))/(0.5-0.31*(
    s_1)^(1/3)));
16 delta = alpha2-zeta-0.5*theta;
17
18 // Results

```

```
19 printf('The suitable space chord ratio = %.4f',s_1
);
20 printf('\n The suitable blade camber = %.2f deg.',
theta);
```

Chapter 4

Axial flow Turbines Two dimensional Theory

Scilab code Exa 4.1 Ex 1

```
1 clear all;
2 clc;
3 funcprot(0);
4
5 //given data
6 phi = 0.4;
7 epsilon = 28.6;//in deg
8
9 //calculations
10 alpha2 = (180/%pi)*atan(1/phi);//in deg
11 zeta = 0.04*(1+ 1.5*(alpha2/100)^2);
12 eta = 1 + (phi^2)*(zeta*((1/cos(%pi*alpha2/180))^2)
    +0.5);
13
14 //results
15 printf('The efficiency = %.3f.\n',1/eta);
16 printf('This value appears to be the same as the
    peak value of efficiency curve.\n');
```

Scilab code Exa 4.2 Ex 2

```
1 clear all;
2 clc;
3 funcprot(0);
4
5 //given data
6 alpha2 = 70;//in deg
7 p01 = 311;//in kPa
8 T01 = 850;//in degC
9 p3 = 100;//in kPa
10 eff_tot_stat = 0.87;
11 U = 500;//in m/s
12 Cp = 1.148;//in kJ/(kgC)
13 gamma = 1.33;
14
15 //Calculations
16 delW = eff_tot_stat*Cp*(T01+273.15)*(1-(p3/p01)^((
    gamma-1)/gamma));//specific work
17 cy2 = delW*1000/U;//in m/s
18 c2 = cy2/sin(%pi*alpha2/180);//in m/s
19 T2 = (T01+273.15) - 0.5*(c2^2)/(Cp*1000);//Nozzle
    exit temperature in K
20 M2 = c2/sqrt(gamma*287*T2);//Nozzle exit mach number
21 cx = c2*cos(%pi*alpha2/180);//axial velocity in m/s
22 eff_tot_tot = 1/((1/eff_tot_stat)-((cx^2)/(2*1000*
    delW)));//Total to total efficiency
23 R = 1 - 0.5*(cx/U)*tan(%pi*alpha2/180);//stage
    reaction
24
25 //results
26 printf('(i) The specific work done = %d kJ/kg.\n',
    delW);
27 printf('(ii) The Mach number leaving the nozzle = %
```



```

        .2f.\n',M2);
28 printf('(iii) The axial velocity = %d m/s.\n',cx);
29 printf('(iv) The total-to-total efficiency = %.2f.\n
    ',eff_tot_tot);
30 printf('(v) The stage reaction = %.3f.\n',R);
31
32
33 //there are small errors in the answers given in the
    book

```

Scilab code Exa 4.3 Ex 3

```

1 clear all;
2 clc;
3 funcprot(0);
4
5 //given data
6 H_b = 5.0; //average blade aspect ratio for the stage
7 t_c = 0.2; //max. blade thickness to chord ratio
8 Re = 1*10^5; //average Reynolds number
9 cx = 200; //in m/s
10 cy2 = 552; //in m/s
11 U = 500; //in m/s
12 c2 = 588; //in m/s
13 delW = 276; //in kJ
14 c3 = 200; //in m/s
15 Cp = 1.148; //in kJ/(kgC)
16 T2 = 973; //in K
17 T01 = 1123; //in K
18 alpha1 = 0; //in deg
19 alpha2 = 70; //in deg
20
21 //calculations
22 eps = alpha1 + alpha2; //in deg
23 zetaN = 0.04*(1 + 1.5*(eps/100)^2);

```

```

24 zetaN1 = (1+zetaN)*(0.993 + 0.021/H_b) - 1;
25 beta2 = (180/%pi)*atan((cy2-U)/cx);
26 beta3 = (180/%pi)*atan(U/cx);
27 epsR = beta2 + beta3;
28 zetaR = 0.04*(1 + 1.5*(epsR/100)^2);
29 zetaR1 = (1+zetaR)*(0.975 + 0.075/H_b) - 1;
30 w3_U = sqrt(1+(cx/U)^2);
31 eff_ts = 1/(1 + (zetaR1*w3_U + zetaN1*((c2/U)^2) + (
      cx/U)^2)/(2*cy2/U));
32 T3 = T01 - (delW*1000 + 0.5*c3^2)/(Cp*1000);
33 eff_ts1 = 1/(1 + (zetaR1*(w3_U)^2 + (T3/T2)*zetaN1
      *((c2/U)^2) + (cx/U)^2)/(2*cy2/U));
34
35 //Results
36 printf('The total-to static efficiency = %.3f.',
      eff_ts);
37 printf('\n The result is very close to the value
      assumed in first example.')
38 printf('\n The total-to-static efficiency after
      including the temperature ratio in the equation =
      %.3f.',eff_ts1);
39
40 //there are small errors in the answers given in the
      book

```

Scilab code Exa 4.4 Ex 4

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 T02 = 1200;//in K
7 p01 = 4.0;//in bar
8 dt = 0.75;//tip diameter in m

```

```

9 hb = 0.12; //blade height in m
10 v = 10500; //shaft speed in rev/min
11 R = 0.5; //degree of reaction at mean radius
12 phi = 0.7; //flow coefficient
13 psi = 2.5; //stage loading coefficient
14 eff_noz = 0.96; //Nozzle efficiency
15 Cp = 1160; //in kJ/(kgC)
16 gamma = 1.33;
17 Rg = 287.8; //specific gas constant
18 A2 = 0.2375; //in m^2
19 K = 2/3; //stress taper factor
20 rho = 8000; //in kg/m^3
21
22 //calculations
23 beta3 = (180/%pi)*atan((0.5*psi + R)/phi);
24 beta2 = (180/%pi)*atan((0.5*psi - R)/phi);
25 alpha2 = beta3;
26 alpha3 = beta2;
27 rm = (dt-hb)/2;
28 Um = (v/30)*%pi*rm;
29 cx = phi*Um;
30 c2 = cx/(cos(alpha2*%pi/180));
31 T2 = T02 - 0.5*(c2^2)/Cp;
32 p2 = p01*((1-((1-(T2/T02))/eff_noz))^(gamma/(gamma
-1))));
33 mdot = ((p2*10^5)/(Rg*T2))*A2*cx;
34 Ut = (v/30)*%pi*0.5*dt;
35 sig_rho = K*0.5*(Ut^2)*(1-((dt-2*hb)/dt)^2);
36 sig = rho*sig_rho;
37 Tb = T2 + 0.85*((cx/cos(beta2*%pi/180))^2)/(2*Cp);
38
39 //Results
40 printf('(i)The relative and absolute angles for the
flow: \n beta3 = %.1f deg, and beta2 = %.2f deg.'
,beta3,beta2);
41 printf('\n alpha2 = %.1f deg, and alpha3 = %.2f deg.
',alpha2,alpha3);
42 printf('\n (ii) The velocity at nozzle exit = %.2f m

```

```

/s',c2);
43 printf('\n (iii)The static temperature and pressure
    at nozzle exit assuming a nozzle efficiency of %
    .2f: \n T2 = %.1f K\n p2 = %.3f bar',eff_noz,T2,
    p2);
44 printf('\n and mass flow = %.1f kg/s',mdot);
45 printf('\n (iv)The rotor blade root stress assuming
    the blade is tapered with a stress taper factor K
    of 2/3 and \n the blade material density is %d
    kg/m2 = %.1f MPa',rho,sig/(10^6));
46 printf('\n (v) The approximate average mean blade
    temperature is Tb = %.1f K',Tb);
47 printf('\n (vi)Inspection of the data for Inconel
    713 cast alloy suggests that it might be a better
    choice \n of blade material as the
    temperature stress point of the above
    calculation is to the \n left of the line marked
    creep strain of 0.2 percentage in 1000 hr.')
48
49
50 //there are very small errors in the answers given
    in textbook

```

Chapter 5

Axial flow Compressors and Fans

Scilab code Exa 5.1 Ex 1

```
1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 T01 = 293; //in K
7 pi = 5; //pressure ratio
8 R = 0.5; //stage reaction
9 Um = 275; //in m/s
10 phi = 0.5; //flow coefficient
11 psi = 0.3; //stage loading factor
12 eff_stage = 0.888; //stage efficiency
13 Cp = 1005; //J/(kgC)
14 gamma = 1.4;
15
16 //Calculations
17 beta1 = (180/%pi)*atan((R + 0.5*psi)/phi);
18 beta2 = (180/%pi)*atan((R - 0.5*psi)/phi);
19 alpha2 = beta1;
```

```

20 alpha1 = beta2;
21 delT0 = psi*(Um^2)/Cp;
22 N = (T01/delT0)*((pi^((gamma-1)/(eff_stage*gamma)))
    - 1);
23 N = ceil(N);
24 eff_ov = ((pi^((gamma-1)/gamma)) - 1)/((pi^((gamma
    -1)/(eff_stage*gamma))) - 1);
25 printf('The flow angles are: beta1 = alpha2 = %.2f
    deg and beta2 = alpha1 = %d deg.',beta1,ceil(
    beta2));
26 printf('\n The number of stages required = %d',N);
27 printf('\n The overall efficiency = %.1f percentage'
    ,eff_ov*100);
28
29 //there is a small error in the answer given in
    textbook

```

Scilab code Exa 5.2 Ex 2

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 R = 0.5; //stage reaction
7 s_c = 0.9; //space-chord ratio
8 beta1_ = 44.5; //in deg
9 beta2_ = -0.5; //in deg
10 h_c = 2.0; //height-chord ratio
11 lamda = 0.86; //work done factor
12 i = 0.4; //mean radius relative incidence
13 rho = 3.5; //density in kg/m^3
14 Um = 242; //in m/s
15 eps_max = 37.5; //in deg
16 eps = 37.5; //in deg

```

```

17 delp0 = 0.032; //the profile total pressure loss
    coefficient
18 //Calculations
19 theta = beta1_ - beta2_;
20 deltaN = (0.229*theta*(s_c^0.5))/(1 - (theta*(s_c
    ^0.5)/500));
21 beta2N = deltaN + beta2_;
22 eps_ = 0.8*eps_max;
23 i_ = beta2N + eps_ - beta1_;
24 i = 0.4*eps_ + i_;
25 beta1 = beta1_ + i;
26 beta2 = beta1 - eps;
27 alpha2 = beta1;
28 alpha1 = beta2;
29 phi = 1/(tan(alpha1*pi/180) + tan(beta1*pi/180));
30 psi = lamda*phi*(tan(alpha2*pi/180) - tan(alpha1*
    pi/180));
31 betam = (180/pi)*atan(0.5*(tan(beta1*pi/180) + tan
    (beta2*pi/180)));
32 CL = 2*s_c*cos(betam*pi/180)*(tan(beta1*pi/180) -
    tan(beta2*pi/180));
33 CDp = s_c*(delp0)*((cos(betam*pi/180))^3)/((cos(
    beta1*pi/180))^2);
34 CDa = 0.02*s_c/h_c;
35 CDx = 0.018*CL^2;
36 CD = CDp + CDa + CDx;
37 eff_tt = 1 - (CD*phi^2)/(psi*s_c*((cos(betam*pi
    /180))^3));
38 delp = eff_tt*psi*rho*Um^2;
39
40 //Results
41 printf('(i)The nominal deflection= %.1f deg.\n the
    nominal incidence = %.1f deg.',eps_,i_);
42 printf('\n (ii)The inlet flow angle, beta1 = alpha2
    = %.1f deg\n outlet flow angle beta2 = alpha1 = %
    .1f deg.',beta1,beta2);
43 printf('\n (iii)The flow coefficient = %.3f\n The
    stage loading factor = %.3f',phi,psi);

```

```
44 printf('\n (iv) The rotor lift coefficient = %.2f.',
    CL);
45 printf('\n (v) The overall drag coefficient of each
    row = %.3f.',CD);
46 printf('\n (vi) The total-to-total stage efficiency
    = %.3f.\n The pressure rise across the stage = %d
    kPa',eff_tt,delp/1000);
47
48
49 //there are small errors in the answers given in
    textbook
```

Chapter 6

Three dimensional Flows in Axial Turbomachines

Scilab code Exa 6.1 Ex 1

```
1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 dt = 1.0; //tip diameter in m
7 dh = 0.9; //hub diameter in m
8 alpha1 = 30; //in deg
9 beta1 = 60; //in deg
10 alpha2 = 60; //in deg
11 beta2 = 30; //in deg
12 N = 6000; //rotational speed in rev/min
13 rhog = 1.5; //gas density in kg/m^3
14 Rt = 0.5; //degree of reaction at the tip
15
16 //Calculations
17 omega = 2*%pi*N/60;
18 Ut = omega*0.5*dt;
19 Uh = omega*0.5*dh;
```

```

20 cx = Ut/(tan(alpha1*pi/180) + tan(beta1*pi/180));
21 mdot = %pi*((0.5*dt)^2 - (0.5*dh)^2)*rhog*cx;
22 Wcdot = mdot*Ut*cx*(tan(alpha2*pi/180) - tan(alpha1*
    %pi/180));
23 ctheta1t = cx*tan(alpha1*pi/180);
24 ctheta1h = ctheta1t*(dt/dh);
25 ctheta2t = cx*tan(alpha2*pi/180);
26 ctheta2h = ctheta2t*(dt/dh);
27 alpha1_ = (180/%pi)*atan(ctheta1h/cx);
28 beta1_ = (180/%pi)*atan((Uh/cx) - tan(alpha1_*%pi
    /180));
29 alpha2_ = (180/%pi)*atan(ctheta2h/cx);
30 beta2_ = (180/%pi)*atan((Uh/cx) - tan(alpha2_*%pi
    /180));
31 k = Rt*(0.5*dt)^2;
32 Rh = 1 - (k/(0.5*dh)^2);
33
34 // Results
35 printf('(i)The axial velocity , cx = %d m/s',cx);
36 printf('\n (ii)The mass flow rate = %.1f kg/s',mdot)
    ;
37 printf('\n (iii)The power absorbed by the stage = %
    .1f MW',Wcdot/(10^6));
38 printf('\n (iv)The flow angles at the hub are:\n
    alpha1 = %.2f deg,\n beta1 = %.2f deg,\n alpha2 =
    %.1f deg, and\n beta2 = %.2f deg.',alpha1_,
    beta1_,alpha2_,beta2_);
39 printf('\n (v)The reaction ratio of the stage at the
    hub, R = %.3f.',Rh);
40
41
42 //there are small errors in the answers given in
    textbook

```

Scilab code Exa 6.2 Ex 2

```

1  clear;
2  clc;
3  funcprot(0);
4
5  //given data
6
7  R = 0.5; //degree of reaction
8  Cp = 1005; //kJ/(kgC)
9  cx1_Ut_rt = 0.4;
10 delT0 = 16.1; //temperature rise
11 Ut = 300; //in m/s
12
13 //calculations
14 A1 = cx1_Ut_rt^2 +(0.5-0.18*log(1));
15 c1 = 2*(1-R);
16 c2 = Cp*delT0/(2*Ut^2 *(1-R));
17 A2 = 0.56;
18 k = 0.4:0.01:1.0;
19 n = (1.0-0.4)/0.01 + 1;
20 i = 1;
21 for i = 1:n
22     cx1_Ut(i) = sqrt(A1 - (c1^2)*(0.5*k(i)^2 - c2*
23         log(k(i))));
24     cx2_Ut(i) = sqrt(A2 - (c1^2)*(0.5*k(i)^2 + c2*
25         log(k(i))));
26     R_(i) = 0.778+log(k(i));
27     Rn(i) = 0.5;
28 end
29 //Results
30 plot(k,cx1_Ut,'bo-');
31 plot(k,cx2_Ut,'<>r-');
32 title("Solution of exit axial-velocity profile for a
33     first power stage","fontsize",3) ; //title of the
34     plot
35 xlabel("Radius ratio , r/rt","fontsize",3) ; //x label
36 ylabel("cx/Ut","fontsize",3) ; //y label
37 legend(["(cx2/Ut)"; "(cx1/Ut)"] , opt=2); //legend

```

```
    box
35 a=gca();
36 b = newaxes();
37 b.y_location = "right";
38 b.filled = "off";
39 b.axes_visible = ["off","on","on"];
40 b.axes_bounds = a.axes_bounds;
41 b.font_size = a.font_size;
42 plot(k,R_,"g");
43 plot(k,Rn,);
44 ylabel("Reaction","fontsize",3) ;//y label
45 legend(["True Reaction";"Nominal Reaction"] , opt=1)
    ; //legend box
```

Chapter 7

Centrifugal Pumps Fans and Compressors

Scilab code Exa 7.1 Ex 1

```
1 clear;
2 clc;
3 funcprot(0);
4
5 //function to calculate blade cavitation coefficient
6 function [res] = fun(sigmab,k,omega_ss)
7     res = (sigmab^2)*(1 + sigmab) - (((3.42*k)^2)/(
8         omega_ss^4));
9 endfunction
10
11 //given data
12 Q = 25; //flow rate in dm^3/s
13 omega = 1450; //rotational speed in rev/min
14 omega_ss = 3; //max. suction specific speed in rad/
15     sec
16 r = 0.3; //inlet eye radius ratio
17 g = 9.81; //in m/s^2
18 //Calculations
```

```

18 k = 1-(r^2);
19 sigmab = 0.3; //initial guess
20 res = fun(sigmab,k,omega_ss); //initial value
21 i = 0;
22 while (abs(res)>0.0001)
23     if res>0.0 then
24         sigmab = sigmab - 0.0001;
25     elseif res<0.0
26         sigmab = sigmab + 0.0001;
27     end
28     res = fun(sigmab,k,omega_ss);
29 end
30 phi = (sigmab/(2*(1+sigmab)))^0.5;
31 rs1 = ((Q*10^-3)/(%pi*k*(omega*%pi/30)*phi))^(1/3);
32 ds1 = 2*rs1;
33 cx1 = phi*(omega*%pi/30)*rs1;
34 Hs = (0.75*sigmab*cx1^2)/(g*phi^2);
35
36 //Results
37 printf('(i)The blade cavitation coefficient = %.3f',
        sigmab);
38 printf('\n (ii)The shroud radius at the eye = %.5f m
        \n The required diameter of the eye = %.1f mm',
        rs1,ds1*10^3);
39 printf('\n (iii)The eye axial velocity = %.3f m/s',
        cx1);
40 printf('\n (iv)The NPSH = %.3f m',Hs);

```

Scilab code Exa 7.2 Ex 2

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data

```

```

6 alpha1 = 30; //prewhirl in deg
7 hs = 0.4; //inlet hub-shrub radius ratio
8 Mmax = 0.9; //max Mach number
9 Q = 1; //air mass flow in kg/s
10 p01 = 101.3; //stagnation pressure in kPa
11 T01 = 288; //stagnation temperature in K
12 gamma = 1.4;
13 Rg = 287; //in J/(kgK)
14
15 //Calculations
16 beta1 = 49.4; //in deg
17 f = 0.4307;
18 a01 = sqrt(gamma*Rg*T01);
19 rho01 = p01*1000/(Rg*T01);
20 k = 1-(hs^2);
21 omega = (%pi*f*k*rho01*a01^3)^0.5;
22 N = (omega*60/(2*%pi));
23 rho1 = rho01/(1 + 0.2*(Mmax*cos(beta1*%pi/180))^2)
    ^2.5;
24 cx = ((omega^2)/(%pi*k*rho1*(tan(beta1*%pi/180) +
    tan(alpha1*%pi/180))^2))^(1/3);
25 rs1 = (1/(%pi*rho1*cx*k))^0.5;
26
27 ds1 = 2*rs1;
28 U = omega*rs1;
29
30 //Results
31 printf('(i)The rotational speed of the impeller = %
    .1f rad/s and N = %d rev/min.',omega,N);
32 printf('\n (ii)The inlet static density downstream
    of the guide vanes at the shroud = %.5f kg/m^3.\n
    The axial velocity = %.2f m/s.',rho1,cx);
33 printf('\n (iii)The inducer tip diameter = %.3f cm\n
    U = %.1f m/s.',ds1*100,U);
34
35 //there are small errors in the answers given in
    textbook

```

Scilab code Exa 7.3 Ex 3

```
1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 Q = 0.1; //in m^3/s
7 N = 1200; //rotational speed in rev/min
8 beta2_ = 50; //in deg
9 D = 0.4; //impeller external diameter in m
10 d = 0.2; //impeller internal diameter in m
11 b2 = 31.7; //axial width in mm
12 eff = 0.515; //diffuser efficiency
13 H = 0.1; //head losses
14 De = 0.15; //diffuser exit diameter
15 A = 0.77;
16 B = 1;
17 g = 9.81;
18
19 //Calculations
20 U2 = %pi*N*D/60;
21 cr2 = Q/(%pi*D*b2/1000);
22 sigmaB = (A - H*tan(beta2_*%pi/180))/(B - H*tan(
    beta2_*%pi/180));
23 ctheta2 = sigmaB*U2*(1-H*tan(beta2_*%pi/180));
24 Hi = U2*ctheta2/g;
25 c2 = sqrt(cr2^2 + ctheta2^2);
26 c3 = 4*Q/(%pi*De^2);
27 HL = 0.1*Hi + 0.485*((c2^2)-(c3^2))/(2*g) + (c3^2)
    /(2*g);
28 H = Hi - HL;
29 eff_hyd = H/Hi;
30
```



```

31 //Results
32 printf('The slip factor = %.3f.',sigmaB);
33 printf('\n The manometric head = %.1f m.',H);
34 printf('\n The hydraulic efficiency = %.1f
    percentage.',eff_hyd*100);
35
36 //there is a very small error in the answer given in
    textbook

```

Scilab code Exa 7.4 Ex 4

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 T01 = 22;//stagnation temperature in degC
7 Z = 17;//number of vanes
8 N = 15000;//rotational speed in rev/min
9 r = 4.2;//stagnation pressure ratio between diffuser
    and impeller
10 eff_ov = 0.83;//overall efficiency
11 mdot = 2;//mass flow rate in kg/s
12 eff_m = 0.97;//mechanical efficiency
13 rho2 = 2;//air density at impeller outle in kg/m^3
14 gamma = 1.4;
15 R = 0.287;//in kJ/(kg.K)
16 b2 = 11;//axial width at the entrance to the
    diffuser in mm
17
18 //Calculations
19 Cp = gamma*R*1000/(gamma-1);
20 sigmaS = 1 - 2/Z;
21 U2 = sqrt(Cp*(T01+273)*((r)^((gamma-1)/gamma) -1)/(
    sigmaS*eff_ov));

```

```
22 omega = N*pi/30;
23 rt = U2/omega;
24 Wdot_act = mdot*sigmaS*(U2^2)/(eff_m);
25 cr2 = mdot/(rho2*2*pi*rt*b2/1000);
26 ctheta2 = sigmaS*U2;
27 c2 = sqrt(ctheta2^2 + cr2^2);
28 delW = sigmaS*U2^2;
29 T2 = T01+273+(delW - 0.5*c2^2)/Cp;
30 M2 = c2/sqrt(gamma*R*1000*T2);
31
32 //Results
33 printf('Absolute mach number, M2 = %.2 f.',M2);
```

Chapter 8

Radial Flow Gas Turbines

Scilab code Exa 8.1 Ex 1

```
1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 D2 = 23.76; //diameter of rotor in cm
7 N = 38140; //rotational speed in rev/min
8 alpha2 = 72; //absolute flow angle in deg
9 d = 0.5*D2; //rotor mean exit diameter
10
11 //Calculations
12 U2 = %pi*N*D2/(100*60);
13 w2 = U2/tan(alpha2*%pi/180);
14 c2 = U2*sin(alpha2*%pi/180);
15 w3 = 2*w2;
16 U3 = 0.5*U2;
17 c3 = sqrt(w3^2 - U3^2);
18 delW = 0.5*((U2^2 - U3^2)+(w3^2 - w2^2)+(c2^2 - c3
    ^2));
19 inp_U2 = 0.5*(U2^2 - U3^2)/delW;
20 inp_w2 = 0.5*(w3^2 - w2^2)/delW;
```

```

21 inp_c2 = 0.5*(c2^2 - c3^2)/delW;
22
23 //Results
24 printf('The fractional inputs from the three terms
        are, for the U^2 terms, %.3f; \n for the w^2
        terms, %.3f; for the c^2 terms, %.3f.',inp_U2,
        inp_w2,inp_c2);
25
26 //there are errors in the answers given in textbook

```

Scilab code Exa 8.2 Ex 2

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 r = 1.5; //operating pressure ratio
7 K1 = 1.44*10^-5;
8 K2 = 2410;
9 K3 = 4.59*10^-6;
10 T01 = 400; //in K
11 D2 = 72.5; //rotor inlet diamete in mm
12 D3_av = 34.4; //rotor mean outlet diameter in mm
13 b = 20.1; //rotor outlet annulus width in mm
14 zetaN = 0.065; //enthalpy loss coefficient
15 alpha2 = 71; //in deg
16 beta3_av = 53; //in deg
17 Cp = 1005; //inJ/(kg.K)
18 gamma = 1.4;
19
20 //Calculations
21 N = K2*sqrt(T01);
22 U2 = %pi*N*D2/(60*1000)
23 delW = U2^2;

```

```

24 delh = Cp*T01*(1-(1/r)^((gamma-1)/gamma));
25 eff_ts = delW/(delh);
26 delW_act = K3*K2*pi*T01/(30*K1);
27 eff_ov = delW_act/delh;
28 zetaR = (2*((1/eff_ts)-1) - (zetaN/sin(alpha2*pi
    /180)))*((D2/D3_av)^2)*(sin(beta3_av*pi/180))^2
    - (cos(beta3_av*pi/180))^2;
29 r3 = 0.5*(D3_av-b)*10^-3;
30 w3_w2av_min = (D3_av/D2)*tan(alpha2*pi/180)*((2*r3/
    D3_av)^2 + (1/tan(beta3_av*pi/180))^2)^0.5;
31 w3_w2av = (D3_av/D2)*tan(alpha2*pi/180)*(1+((1/tan(
    beta3_av*pi/180))^2))^0.5;
32
33 //Results
34 printf('The total-to-static efficiency = %.2f
    percentage.',eff_ts*100);
35 printf('\n The overall efficiency = %.2f percentage.
    ',eff_ov*100);
36 printf('\n The rotor enthalpy loss coefficient = %.3
    f',zetaR);
37 printf('\n The rotor relative velocity ratio = %.2f'
    ,w3_w2av);
38
39
40 //there are small errors in the answers given in
    textbook

```

Scilab code Exa 8.3 Ex 3

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 Z = 12;//number of vanes

```

```

7 delW = 230; //in kW
8 T01 = 1050; //stagnation temperature in K
9 mdot = 1; //flow rate in kg/s
10 eff_ts = 0.81; //total-to-static efficiency
11 Cp = 1.1502; //in kJ/(kg.K)
12 gamma = 1.333;
13 R = 287; //gas constant
14
15 // Calculations
16 S = delW/(Cp*T01);
17 alpha2 = (180/%pi)*acos(sqrt(1/Z));
18 beta2 = 2*(90-alpha2);
19 p3_p01 = (1-(S/eff_ts))^(gamma/(1-gamma));
20 M02 = sqrt((S/(gamma-1))*((2*cos(beta2*%pi/180))/(1+
    cos(beta2*%pi/180))));
21 M2 = sqrt((M02^2)/(1-0.5*(gamma-1)*(M02^2)));
22 U2 = sqrt((gamma*R*T01)*(1/cos(beta2*%pi/180))*(S/(
    gamma-1)));
23
24 // Results
25 printf('(i) The absolut and relative flow angles:\n
    alpha2 = %.2f deg\n beta2 = %.2f deg',alpha2,
    beta2);
26 printf('\n (ii) The overall pressure ratio = %.3f',
    p3_p01);
27 printf('\n (iii) The rotor rip speed = %.1f m/s\n
    The inlet absolute Mach number = %.3f',U2,M2);
28
29
30 //there are small errors in the answers given in
    textbook

```

Scilab code Exa 8.4 Ex 4

```
1 clear;
```

```

2  clc;
3  funcprot(0);
4
5  //given data
6  cm3_U2 = 0.25;
7  nu = 0.4;
8  r3s_r2 = 0.7;
9  w3av_w2 = 2.0;
10
11 //Calculations
12 r3av_r3s = 0.5*(1+nu);
13 r3av_r2 = r3av_r3s*r3s_r2;
14 beta3_av = (180/%pi)*atan(r3av_r2/cm3_U2);
15 beta3s = (180/%pi)*atan(r3s_r2/cm3_U2);
16 w3s_w2 = 2*cos(beta3_av*pi/180)/cos(beta3s*pi/180)
    ;
17
18 //Results
19 printf('The relative velocity ratio = %.3f.',w3s_w2)
    ;

```

Scilab code Exa 8.5 Ex 5

```

1  clear;
2  clc;
3  funcprot(0);
4
5  //given data
6  Z = 12; //number of vanes
7  delW = 230; //in kW
8  T01 = 1050; //stagnation temperature in K
9  mdot = 1; //flow rate in kg/s
10 eff_ts = 0.81; //total-to-static efficiency
11 Cp = 1.1502; //in kJ/(kg.K)
12 gamma = 1.333;

```

```

13 R = 287; //gas constant
14 cm3_U2 = 0.25;
15 nu = 0.4;
16 r3s_r2 = 0.7;
17 w3av_w2 = 2.0;
18 p3 = 100; //static pressure at rotor exit in kPa
19 zetaN = 0.06; //nozzle enthalpy loss coefficient
20 U2 = 538.1; //in m/s
21 p01 = 3.109*10^5; //in Pa
22
23 //Calculations
24 S = delW/(Cp*T01);
25 T03 = T01*(1-S);
26 T3 = T03 - (cm3_U2^2)*(U2^2)/(2*Cp*1000);
27 r2 = sqrt(mdot/((p3*1000/(R*T3))*(cm3_U2)*U2*pi*(
    r3s_r2^2)*(1-nu^2)));
28 D2 = 2*r2;
29 omega = U2/r2;
30 N = omega*30/pi;
31 ctheta2 = S*Cp*1000*T01/U2;
32 alpha2 = (180/pi)*acos(sqrt(1/Z));
33 cm2 = ctheta2/tan(alpha2*pi/180);
34 c2 = ctheta2/sin(alpha2*pi/180);
35 T2 = T01 - (c2^2)/(2*Cp*1000);
36 p2 = p01*(1-(((c2^2)*(1+zetaN))/(2*Cp*1000*T01)))^(
    gamma/(gamma-1));
37 b2_D2 = (0.25/pi)*(R*T2/p2)*(mdot/(cm2*r2^2));
38
39 //Results
40 printf('(i) The diameter of the rotor = %.4f m\n
    its speed of rotation = %.1f rad/s (N = %d rev/
    min)',D2,omega,N);
41 printf('\n(ii) The vane width to diameter ratio at
    rotor inlet = %.4f',b2_D2);
42
43 //there are some errors in the answers given in
    textbook

```


Scilab code Exa 8.6 Ex 6

```
1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 Z = 12; //number of vanes
7 delW = 230; //in kW
8 T01 = 1050; //stagnation temperature in K
9 mdot = 1; //flow rate in kg/s
10 eff_ts = 0.81; //total-to-static efficiency
11 Cp = 1.1502; //in kJ/(kg.K)
12 gamma = 1.333;
13 R = 287; //gas constant
14 cm3_U2 = 0.25;
15 nu = 0.4;
16 r3s_r2 = 0.7;
17 w3av_w2 = 2.0;
18 p3 = 100; //static pressure at rotor exit in kPa
19 zetaN = 0.06; //nozzle enthalpy loss coefficient
20 U2 = 538.1; //in m/s
21 p01 = 3.109*10^5; //in Pa
22
23 //results of Example 8.4 and Example 8.5
24 r3av_r3s = 0.5*(1+nu);
25 r3av_r2 = r3av_r3s*r3s_r2;
26 alpha2 = (180/%pi)*acos(sqrt(1/Z));
27 beta2 = 2*(90-alpha2);
28 beta3_av = (180/%pi)*atan(r3av_r2/cm3_U2);
29 beta3s = (180/%pi)*atan(r3s_r2/cm3_U2);
30 w3s_w2 = 2*cos(beta3_av*%pi/180)/cos(beta3s*%pi/180)
    ;
31 S = delW/(Cp*T01);
```

```

32 T03 = T01*(1-S);
33 T3 = T03 - (cm3_U2^2)*(U2^2)/(2*Cp*1000);
34 r2 = sqrt(mdot/((p3*1000/(R*T3))*(cm3_U2)*U2*pi*(
    r3s_r2^2)*(1-nu^2)));
35 D2 = 2*r2;
36 omega = U2/r2;
37 N = omega*30/pi;
38 ctheta2 = S*Cp*1000*T01/U2;
39 alpha2 = (180/pi)*acos(sqrt(1/Z));
40 cm2 = ctheta2/tan(alpha2*pi/180);
41 c2 = ctheta2/sin(alpha2*pi/180);
42 T2 = T01 - (c2^2)/(2*Cp*1000);
43 p2 = p01*(1-(((c2^2)*(1+zetaN))/(2*Cp*1000*T01)))^(
    gamma/(gamma-1));
44 b2_D2 = (0.25/pi)*(R*T2/p2)*(mdot/(cm2*r2^2));
45
46 // Calculations
47 c3 = cm3_U2*U2;
48 cm3 = c3;
49 w3_av = 2*cm3/(cos(beta2*pi/180));
50 w2 = w3_av/2;
51 c0 = sqrt(2*delW*1000/eff_ts);
52 zetaR = (c0^2*(1-eff_ts)-(c3^2)-zetaN*(c2^2))/(
    w3_av^2);
53 i = beta2;
54 n = 1.75;
55 eff_ts_new = 1-(((c3^2)+zetaN*(c2^2)+zetaR*(w3_av^2)
    +(1-(cos(i*pi/180))^n)*(w2^2))/(c0^2));
56
57 // Results
58 printf('(a)The rotor enthalpy loss coefficient = %.4
    f',zetaR);
59 printf('\n(b) The total-to-static efficiency of the
    turbine = %.3f',eff_ts_new);
60
61
62 //there are some errors in the answers given in
    textbook

```


Chapter 9

Hydraulic Turbines

Scilab code Exa 9.1 Ex 1

```
1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 Q = 2.272; //water volume flow rate in m^3/s
7 l = 300; //length in m
8 Hf = 20; //head loss in m
9 f = 0.01; //friction factor
10 g = 9.81; //acceleration due to gravity in m/s^2
11
12 //Calculations
13 d = (32*f*l*((Q/%pi)^2)/(g*Hf))^(1/5);
14
15 //Results
16 printf('The diameter of the pipe = %.4f m',d);
```

Scilab code Exa 9.2 Ex 2

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 P = 4.0; //in MW
7 N = 375; //in rev/min
8 H_eps = 200; //in m
9 KN = 0.98; //nozzle velocity coefficient
10 d = 1.5; //in m
11 k = 0.15; //decrease in relative flow velocity across
    the buckets
12 alpha = 165; //in deg
13 g = 9.81; //in m/s^2
14 rho = 1000; //in kg/m^3
15
16 //Calculations
17 U = N*pi*d*0.5/30;
18 c1 = KN*sqrt(2*g*H_eps);
19 nu = U/c1;
20 eff = 2*nu*(1-nu)*(1-(1-k)*cos(alpha*pi/180));
21 Q = (P*10^6 /eff)/(rho*g*H_eps);
22 Aj = Q/(2*c1);
23 dj = sqrt(4*Aj/pi);
24 omega_sp = (N*pi/30)*sqrt((P*10^6)/rho)/((g*H_eps)
    ^ (5/4));
25
26 //Results
27 printf('(i)The runner efficiency = %.4f',eff);
28 printf('\n (ii)The diameter of each jet = %.4f m',dj
    );
29 printf('\n (iii)The power specific speed = %.3f rad'
    ,omega_sp);

```

Scilab code Exa 9.3 Ex 3

```

1  clear;
2  clc;
3  funcprot(0);
4
5  //given data
6  H_eps = 150; //in m
7  z = 2; //in m
8  U2 = 35; //runner tip speed in m/s
9  c3 = 10.5; //meridional velocity of water in m/s
10 c4 = 3.5; //velocity at exit in m/s
11 delHN = 6.0; //in m
12 delHR = 10.0; //in m
13 delHDT = 1.0; //in m
14 g = 9.81; //in m/s^2
15 Q = 20; //in m^3/s
16 omega_sp = 0.8; //specific speed of turbine in rad
17 c2 = 38.73; //in m/s
18
19 //Calculations
20 H3 = ((c4^2 - c3^2)/(2*g)) + delHDT - z;
21 H2 = H_eps - delHN - (c2^2)/(2*g);
22 delW = g*(H_eps - delHN - delHR - z) - 0.5*c3^2 - g*H3;
23 ctheta2 = delW/U2;
24 alpha2 = (180/pi)*atan(ctheta2/c3);
25 beta2 = (180/pi)*atan((ctheta2-U2)/c3);
26 eff_H = delW/(g*H_eps);
27 omega = (omega_sp*(g*H_eps)^(5/4))/sqrt(Q*delW);
28 N = omega*30/pi;
29 D2 = 2*U2/omega;
30
31 //Results
32 printf('(i)The pressure head H3 relative to the
    trailrace = %.1f m\n The pressure head H2 at exit
    from the runner = %.2f m',H3,H2);
33 printf('\n(ii)The flow angles at runner inlet and at
    guide vane exit:\n alpha2 = %.1f deg\n beta2 = %
    .2f deg',alpha2,beta2);
34 printf('\n(iii)The hydraulic efficiency of the

```

```

    turbine = %.4f',eff_H);
35 printf('\n The speed of rotation , N = %d rev/min',N)
    ;
36 printf('\n The runner diameter is , D2 = %.3f m',D2);
37
38
39 //there are small errors in the answers given in
    textbook

```

Scilab code Exa 9.4 Ex 4

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //function to calculate flow angles
6 function [alpha2,beta2,beta3] =fun(r,N,cx2,ctheta2)
7     alpha2 = (180/%pi)*atan(ctheta2/cx2);
8     beta2 = (180/%pi)*atan((U2)*(r)/cx2 - tan(alpha2
9         *%pi/180));
10    beta3 = (180/%pi)*atan((U2)*r/cx2) ;
11 endfunction
12
13 //given data
14 P = 8;//output power in MW
15 HE = 13.4;//available head at entry in m
16 N = 200;//in rev/min
17 L = 1.6;//length of inlet guide vanes
18 d1 = 3.1;//diameter of trailing edge in m
19 D2t = 2.9;//runner diameter in m
20 nu = 0.4;//hub-tip ratio
21 eff = 0.92;//hydraulic efficiency
22 rho = 1000;//density in kg/m^3
23 g = 9.81;//acceleration due to gravity in m/s^2

```

```

24 // Calculations
25 Q = P*10^6 / (eff*rho*g*HE);
26 cr1 = Q / (2*pi*0.5*d1*L);
27 cx2 = 4*Q / (pi*D2t^2 * (1-nu^2));
28 U2 = N*(pi/30)*D2t/2;
29 ctheta2 = eff*g*HE/U2;
30 ctheta1 = ctheta2*(D2t/d1);
31 alpha1 = (180/pi)*atan(ctheta1/cr1);
32
33 // calculating flow angle for different radii
34 [alpha21, beta21, beta31] = fun(1.0, U2, cx2, ctheta2);
35 [alpha22, beta22, beta32] = fun(0.7, U2, cx2, ctheta2
    /0.7);
36 [alpha23, beta23, beta33] = fun(0.4, U2, cx2, ctheta2
    /0.4);
37
38 // Results
39 printf('Calculated values of flow angles:\n
    Parameter                               Ratio of r
    /ri                                     ');
40 printf('\n
    _____');
41 printf('\n
    1.0   0.4   0.7');
42 printf('\n
    _____');
43 printf('\n ctheta2 (in m/s)           %.3f           %.3f
    %.3f', ctheta2/0.4, ctheta2/0.7, ctheta2
    /1.0);
44 printf('\n tan(alpha2)             %.3f           %.4f
    %.3f', tan(alpha23*pi/180), tan(alpha22
    *pi/180), tan(alpha21*pi/180));
45 printf('\n alpha2 (deg)            %.2f           %.2f
    %.2f', alpha23, alpha22, alpha21);
46 printf('\n U/cx2                   %.3f           %.4f
    %.3f', (U2/cx2)*0.4, (U2/cx2)*0.7, (U2/
    cx2)*1.0);

```



```

47 printf( '\n beta2(deg)           %.2 f           %.2 f
           %.2 f ', beta23 , beta22 , beta21 );
48 printf( '\n beta3(deg)           %.2 f           %.2 f
           %.2 f ', beta33 , beta32 , beta31 );
49 printf( '\n
_____
');
_____

```

Scilab code Exa 9.5 Ex 5

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 k = 1/5; //scale ratio
7 Pm = 3; //in kW
8 Hm = 1.8; //in m
9 Nm = 360; //in rev/min
10 Qm = 0.215; //in m^3/s
11 Hp = 60; //in m
12 n = 0.25;
13 rho = 1000; //in kg/m^3
14 g = 9.81; //in m/s^2
15
16 //Calculations
17 Np = Nm*k*(Hp/Hm)^0.5;
18 Qp = Qm*(Nm/Np)*(1/k)^3;
19 Pp = Pm*((Np/Nm)^3)*(1/k)^5;
20 eff_m = Pm*1000/(rho*Qm*g*Hm);
21 eff_p = 1 - (1-eff_m)*0.2^n;
22 Pp_corrected = Pp*eff_p/eff_m;
23
24 //Results
25 printf( 'The speed = %.1 f rev/min. ', Np);

```

```

26 printf('\n The flow rate = %.2f m3/s.', Qp);
27 printf('\n Power of the full-scale = %.2f MW.', Pp
    /1000);
28 printf('\n The efficiency of the model turbine = %.2
    f.', eff_m);
29 printf('\n The efficiency of the prototype = %.4f.',
    eff_p);
30 printf('\n The power of the full-size turbine = %.1f
    MW.', Pp_corrected/1000);

```

Scilab code Exa 9.6 Ex 6

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 //data from EXAMPLE 9.3
7 H_eps = 150; //in m
8 z = 2; //in m
9 U2 = 35; //runner tip speed in m/s
10 c3 = 10.5; //meridonal velocity of water in m/s
11 c4 = 3.5; //velocity at exit in m/s
12 delHN = 6.0; //in m
13 delHR = 10.0; //in m
14 delHDT = 1.0; //in m
15 g = 9.81; //in m/s2
16 Q = 20; //in m3/s
17 omega_sp = 0.8; //specific speed of turbine in rad
18 c2 = 38.73; //in m/s
19
20 //data from this example
21 Pa = 1.013; //atmospheric pressure in bar
22 Tw = 25; //temperature of water in degC
23 Pv = 0.03166; //vapor pressure of water at Tw

```

```

24 rho = 1000; //density of wate in kg/m^3
25 g = 9.81; //acceleration due to gravity in m/s^2
26
27 H3 = ((c4^2 - c3^2)/(2*g)) + delHDT - z;
28 H2 = H_eps-delHN-(c2^2)/(2*g);
29 delW = g*(H_eps-delHN-delHR-z)-0.5*c3^2 -g*H3;
30 ctheta2 = delW/U2;
31 alpha2 = (180/%pi)*atan(ctheta2/c3);
32 beta2 = (180/%pi)*atan((ctheta2-U2)/c3);
33 eff_H = delW/(g*H_eps);
34 omega = (omega_sp*(g*H_eps)^(5/4))/sqrt(Q*delW);
35
36 Hs = (Pa-Pv)*(10^5)/(rho*g) - z;
37 sigma = Hs/H_eps;
38 omega_ss = omega*(Q^0.5)/(g*Hs)^(3/4);
39
40 //Results
41 printf('The NSPH for the turbine = %.3f m.',Hs);
42 if omega_ss>4.0 then
43     printf('\n Since the suction specific speed (= %
         .4f.) is greater than 4.0(rad), the cavitation
         is likely to occur.',omega_ss);
44 end
45
46 //there is small error in the answer given in
    textbook

```

Chapter 10

Wind Turbines

Scilab code Exa 10.2 Ex 2

```
1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 a_ = 1/3;
7
8 //Calculations
9 R2_R1 = 1/(1-a_)^0.5;
10 R3_R1 = 1/(1-2*a_)^0.5;
11 R3_R2 = ((1-a_)/(1-2*a_))^0.5;
12
13 //Results
14 printf('R2/R1 = %.3 f\n R3/R1 = %.3 f\n R3/R2 = %.3 f ',
        R2_R1 , R3_R1 , R3_R2);
```

Scilab code Exa 10.3 Ex 3

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 d = 30; //tip diameter in m
7 cx1 = 7.5; //in m/s
8 cx2 = 10; //in m/s
9 rho = 1.2; //in kg/m^3
10 a_ = 1/3;
11
12 //Calculations
13 P1 = 2*a_*rho*(%pi*0.25*d^2)*(cx1^3)*(1-a_)^2;
14 P2 = 2*a_*rho*(%pi*0.25*d^2)*(cx2^3)*(1-a_)^2;
15
16
17 //Results
18 printf('(i)With cx1 = %.1f m/s , P = %d kW. ',cx1,P1
19 /1000);
20 printf('\n(ii)With cx1 = %d m/s , P = %.1f kW. ',cx2,
21 P2/1000);

```

Scilab code Exa 10.4 Ex 4

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 P = 20; //power required in kW
7 cx1 = 7.5; //steady wind speed in m/s
8 rho = 1.2; //density in kg/m^3
9 Cp = 0.35;
10 eta_g = 0.75; //output electrical power
11 eff_d = 0.85; //electrical generation efficiency

```

```

12
13 // Calculations
14 A2 = 2*P*1000/(rho*Cp*eta_g*eff_d*cx1^3);
15 D2 = sqrt(4*A2/%pi);
16
17 // Results
18 printf('The diameter = %.1f m.',D2);

```

Scilab code Exa 10.5 Ex 5

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 Z = 3; //number of blades
7 D = 30; //rotor diameter in m
8 J = 5.0; //tip-speed ratio
9 l = 1.0; //blade chord in m
10 r_R = 0.9; //ratio
11 beta = 2; //pitch angle in deg
12
13 // Calculations
14 //iterating to get values of induction factors
15 a = 0.0001; //inital guess
16 a_ = 0.0001; //inital guess
17 a_new = 0.0002; //inital guess
18 i = 0;
19 while (a_~=a_new)
20     phi = (180/%pi)*atan((1/(r_R*J))*((1-a)/(1-a_)));
21     alpha = phi-beta;
22     CL = 0.1*alpha;
23     lamda = (Z*l*CL)/(8*%pi*0.5*r_R*D);
24     a = 1/(1+(1/lamda)*sin(phi*%pi/180)*tan(phi*%pi
        /180));

```

```

25     a_new = 1/((1/lamda)*cos(phi*pi/180) -1);
26     if a_ < a_new
27         a_ = a_ + 0.0001;
28     elseif a_ > a_new
29         a_ = a_ - 0.0001;
30     end
31     if (abs((a_-a_new)/a_new) < 0.1) then
32         break;
33     end
34     i = i+1;
35 end
36
37 //Results
38 printf('Axial induction factor , a = %.4f',a);
39 printf('\n Tangential induction factor = %.5f',a_new
40 );
41 printf('\n phi = %.3f deg.',phi);
42 printf('\n Lift coefficient = %.3f.',CL);
43 //The answers given in textbook are wrong

```

Scilab code Exa 10.6 Ex 6

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 D = 30; //tip diameter in m
7 CL = 0.8; //lift coefficient
8 J = 5.0;
9 l = 1.0; //chord length in m
10 Z = 3; //number of blades
11 r_R = [0.2 0.3 0.4 0.6 0.8 0.9 0.95 1.0];
12 n = 8;

```

```

13 //Calculations
14 //iterating to get values of induction factors
15 a = 0.1;//inital guess
16 anew = 0;
17 a_ = 0.006;//inital guess
18 a_new = 0.0;//inital guess
19 for i = 1:n
20     while (a_~=a_new)
21         lamda = (Z*l*CL)/(8*pi*0.5*r_R(i)*D);
22         phi = (180/pi)*atan((1/(r_R(i)*J))*((1-a)
                /(1-a_)));
23         a = 1/(1+(1/lamda)*sin(phi*pi/180)*tan(phi*
                pi/180));
24         a_new = 1/((1/lamda)*cos(phi*pi/180) -1);
25         alpha = CL/0.1;
26         beta = phi-alpha;
27         if a_ < a_new
28             a_ = a_ + 0.0001;
29         elseif a_ > a_new
30             a_ = a_ - 0.0001;
31         end
32         if (abs((a_-a_new)/a_new) < 0.01) then
33             break;
34         end
35     end
36     p(i) = phi;b(i) = beta;a1(i) = a;a2(i) = a_new;
37 end
38
39 //Results
40 printf('Summary of results of iterations (N.B. CL =
        0.8 along the span)');
41 printf('\n
        ');
42 printf('\n r/R      %.1f      %.1f      %.1f
        %.1f      %.1f      %.1f      %.2f      %.1f '
        ,r_R(1),r_R(2),r_R(3),r_R(4),r_R(5),r_R(6),r_R(7)
        ,r_R(8));

```



```

43 printf( '\n
    ');
44 printf( '\n phi    %.2 f    %.2 f    %.2 f    %.2 f
    %.2 f    %.2 f    %.2 f    %.3 f ', p(1), p(2), p
    (3), p(4), p(5), p(6), p(7), p(8));
45 printf( '\n beta  %.2 f    %.2 f    %.2 f    %.2 f
    %.2 f    %.2 f    %.2 f    %.2 f ', b(1), b(2)
    , b(3), b(4), b(5), b(6), b(7), b(8));
46 printf( '\n a    %.4 f    %.5 f    %.5 f    %.4 f    %.4 f
    %.4 f    %.4 f    %.4 f ', a1(1), a1(2), a1(3), a1(4)
    , a1(5), a1(6), a1(7), a1(8));
47 printf( '\n a'   %.5 f    %.5 f    %.5 f    %.5 f    %.5 f
    %.5 f    %.5 f    %.5 f ', a2(1), a2(2), a2(3), a2(4), a2
    (5), a2(6), a2(7), a2(8));
48 printf( '\n
    ');
49
50 //there are some errors in the answers given in
    textbook

```

Scilab code Exa 10.7 Ex 7

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 //data from Exempla 10.5
7 Z = 3; //number of blades
8 D = 30; //rotor diameter in m
9 J = 5.0; //tip-speed ratio
10 l = 1.0; //blade chord in m
11 beta = 2; //pitch angle in deg

```

```

12 omega = 2.5; //in rad/s
13
14 rho = 1.2; //density in kg/m^3
15 cx1 = 7.5; //in m/s
16 sum_var1 = 6.9682; //from Table 10.3
17 sum_var2 = 47.509*10^-3; //from Table 10.4
18
19 //Calculations
20 X = sum_var1*0.5*rho*Z*1*0.5*D*cx1^2;
21 tau = sum_var2*0.5*rho*Z*1*(omega^2)*(0.5*D)^4;
22 P = tau*omega;
23 A2 = 0.25*pi*D^2;
24 P0 = 0.5*rho*A2*cx1^3;
25 Cp = P/P0;
26 zeta = (27/16)*Cp;
27
28 //Results
29 printf('The total axial force = %d N. ',X);
30 printf('\n The torque = %.3f *10^3 Nm. ',tau/1000);
31 printf('\n The power developed = %.3f kW. ',P/1000);
32 printf('\n The power coefficient = %.3f',Cp);
33 printf('\n The relative power coefficient = %.3f',
      zeta);

```

Scilab code Exa 10.8 Ex 8

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 X = 10583; //in N
7 D = 30; //rotor diameter in m
8 Cx = X/23856;
9 rho = 1.2; //density in kg/m^3

```

```

10 cx1 = 7.5; //in m/s
11
12 //solving quadratic equation
13 a = 0; //initial guess
14 res = 1;
15 i = 0;
16 while (res~=0)
17     res = a*(1-a) - Cx/4;
18     if (res>0) then
19         a = a-0.001;
20     elseif (res<0)
21         a = a+0.001;
22     end
23     if abs(res)<0.0001
24         break;
25     end
26 end
27 A2 = 0.25*%pi*D^2
28 P = 2*rho*A2*(cx1^3)*a*(1-a)^2;
29
30 //Results
31 printf('P = %.3f kW. ',P/1000);
32
33 //there is small error in the answer given in
    textbook

```

Scilab code Exa 10.9 Ex 9

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //given data
6 //data from Exempla 10.5
7 Z = 3; //number of blades

```

```

8 D = 30; //rotor diameter in m
9 J = 5.0; //tip-speed ratio
10 l = 1.0; //blade chord in m
11 beta = 1.59; //pitch angle in deg
12 omega = 2.5; //in rad/s
13 rho = 1.2; //density in kg/m^3
14 cx1 = 7.5; //in m/s
15 c1 = 1518.8; //from Ex 10.6
16 c2 = 0.5695*10^6;
17 P0 = 178.96; //Power developed in kW from Ex 10.7
18 X1 = 10582; //Total axial force in N from Ex 10.7
19 Cp1 = 0.378; //Power coefficient from Ex 10.7
20 zeta1 = 0.638; //relative power coefficient from Ex
    10.7
21
22 //Calculations
23 r_R = 0.25:0.1:0.95;
24 b = [28.4;19.49;13.80;9.90;7.017;4.900;3.00;1.59];
25 //b =
    [27.2985;17.8137;11.8231;7.8176;4.9972;3.0511;1.6476;1.59];
26 for j = 1:8
27     i = 1;
28     atemp = 0; a_temp = 0;
29     while i>0,          i = i+1;
30         f = (2/%pi)*acos(exp(-0.5*Z*(1-r_R(j))*(1+J
            ^2)^0.5));
31         phi = (180/%pi)*atan((1/(J*r_R(j)))*((1-
            atemp)/(1+a_temp)));
32         CL = (phi-b(j))/10;
33         lamda = f/(63.32/CL);
34         anew = (lamda*cos(phi*%pi/180)/(lamda*cos(
            phi*%pi/180)+f*(sin(phi*%pi/180))^2));
35         if atemp<anew then
36             atemp = atemp+0.0001;
37         elseif atemp>anew
38             atemp = atemp-0.0001;
39         end

```

```

40         if (abs((atemp-aneu)/aneu) < 0.001) then
41             break;
42         end
43     end
44     F(j) = f;
45     ph(j) = phi;
46     cl(j) = CL;
47     a(j) = aneu;
48     Var1(j) = ((1-aneu)/sin(phi*%pi/180))^2 *cos(phi
                *%pi/180)*CL*0.1;
49 //     a_(j) = lamda/(F*cos(phi*%pi/180)-lamda);
50 // printf('r_R = %.2f, F = %.4f, a = %.4f, phi = %.4f
            \n',r_R(j),F(j),a(j),ph(j));
51 end
52
53 for k = 1:8
54     lam(k) = F(k)*cl(k)/63.32;
55     a_new(k) = lam(k)/(F(k)*cos(ph(k)*%pi/180)-lam(k)
                );
56     Var2(k) = ((1+a_new(k))/cos(phi*%pi/180))^2 *(
                r_R(k))^3 *cl(k)*sin(ph(k)*%pi/180)*0.1;
57 end
58 X = c1*sum(Var1(1:8));
59 sum_Var2 = 40.707*10^-3;
60 tau = c2*sum(Var2(1:8));
61 P = tau*omega;
62 Cp = P/(P0*1000);
63 zeta = (26/17)*Cp;
64
65 //Results
66 printf('                Summary of Results:');
67 printf('\n

```

```

        ');
68 printf('\n                Axial force ,
        kN                Power, kW                Cp
        zeta ');
69 printf('\n

```

```

    ');
70 printf('\n Without tip correction           %.3 f      %
           %.2 f           %.3 f           %
           .3 f ',X1/1000,P0*Cp1,Cp1,zeta1);
71 printf('\n With tip correction           %.3 f
           %.2 f           %.3 f
           %.3 f ',X/1000,P/1000,Cp,zeta);
72 printf('\n
    ');
73
74 //There are errors in the answers given in textbook

```

Scilab code Exa 10.10 Ex 10

```

1 clear;
2 clc;
3 funcprot(0);
4
5 //function to calculate values of blade chord and
   radius (optimum conditions)
6 function [j,lamda,r,l] = fun(phi)
7     lamda = 1-cos(phi*%pi/180);
8     j = sin(phi*%pi/180)*(2*cos(phi*%pi/180)-1)
       /(1+2*cos(phi*%pi/180))/(lamda);
9     r = 3*j;
10    l = 8*%pi*j*lamda;
11 endfunction
12
13 //given data
14 D = 30;//tip diameter in m
15 J = 5.0;//tip-speed ratio
16 Z = 3;//in m
17 CL = 1.0;

```

```

18
19 // Calculations
20 phi1 = 30; //in deg
21 phi2 = 20; //in deg
22 phi3 = 15; //in deg
23 phi4 = 10; //in deg
24 phi5 = 7.556; //in deg
25 //Values of blade chord and radius (optimum
    conditions)
26 [j1,lamda1,r1,l1] = fun(phi1);
27 [j2,lamda2,r2,l2] = fun(phi2);
28 [j3,lamda3,r3,l3] = fun(phi3);
29 [j4,lamda4,r4,l4] = fun(phi4);
30 [j5,lamda5,r5,l5] = fun(phi5);
31
32 printf('Values of blade chord and radius(optimum
    conditions): ');
33 printf('\n
    ');
34 printf('\n phi(deg)      j      4flamda
    r(m)      l(m) ');
35 printf('\n
    ');
36 printf('\n %d      %.2 f      %.3 f      %
    .1 f      %.3 f ', phi1, j1, 4*j1*lamda1, r1, l1)
    ;
37 printf('\n %d      %.2 f      %.3 f      %
    .2 f      %.3 f ', phi2, j2, 4*j2*lamda2, r2, l2);
38 printf('\n %d      %.2 f      %.3 f      %
    .2 f      %.3 f ', phi3, j3, 4*j3*lamda3, r3, l3);
39 printf('\n %d      %.3 f      %.4 f      %.1
    f      %.3 f ', phi4, j4, 4*j4*lamda4, r4, l4);
40 printf('\n %.3 f      %d      %.4 f
    %d      %.3 f ', phi5, ceil(j5), 4*j5*lamda5,
    ceil(r5), l5);
41 printf('\n

```

```
    ');
42
43 l_R = [l1,l2,l3,l4,l5]/(0.5*D);
44 r_R = [r1,r2,r3,r4,r5]/(0.5*D);
45 plot(r_R,l_R);
46 xlabel("r/R", 'fontsize',3);
47 ylabel("l/R", 'fontsize',3);
48 title("Optimal variation of chord length with radius
      ", 'fontsize',3);
49
50 //there are very small errors in the ansers given in
      textbook
```
