Scilab Textbook Companion for Irrigation and Water Power Engineering by B. C. Punmia[1]

Created by
Ahmed Belal
B.Tech
Civil Engineering
Aligarh Muslim University
College Teacher
Dr Javed Alam
Cross-Checked by
K. V. P. Pradeep

May 26, 2016

[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: Irrigation and Water Power Engineering

Author: B. C. Punmia

Publisher: Laxmi Publications Ltd., New Delhi

Edition: 16

Year: 2009

ISBN: 978-81-318-0763-7

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

List of Figures

Chapter 2

METHODS OF IRRIGATION

Scilab code Exa 2.1 EX2 1

```


//example 2.1
//calculate time required to cover 0.1 hectare area
   by tubewell
clc;
//Given
Q=0.0108,   //discharge through well
y=0.075,   //average depth of flow
I=0.05,   //average infiltration rate
A=0.1,   //area to cover
t=(60*2.303*y*log10(Q/(Q-I*A)))/I,
t=round(t);
mprintf("Time required to cover given area=%f
   minutes.",t);
```

Scilab code Exa 2.2 EX2 2

```
//example 2.2
//calculate maximum area that can be irrigated
clc;
//Given
Q=0.0108,   //discharge through well
y=0.075,   //average depth of flow
I=0.05,   //average infiltration rate
A=0.1,   //area to cover
Amax=Q/I;
mprintf("Maximum area that can be irrigated =%f
   hectare.",Amax);
```

Scilab code Exa 2.3 EX2 3

```
//example 2.3

//calculate
//time of water application
//optimum length of each border strip
//dischrge for each border strip

clc;
//Given
d=0.05;//depth of root zone
I=1.25D-5;//average infiltration rate
s=0.0035//slope of border strip
t=d/(I*3600);
t=round(t*1000)/1000;
mprintf("Time of water application=%f hours.",t);

//Part (a)
q=2D-3;//discharge entering water source
qdash=q*100^2*60;
```

```
n=0.55425-(0.0001386*qdash);
yo=(n*q/(s^0.5))^0.6;
y=0.665*yo;
L=(q/I)*(1-%e^(-d/y));
L=round(10*L)/10;
mprintf("\nPart (a):");
mprintf("\nOptimum length of each border strip=%f m.",L);

//Part (b)
Lgiven=150//given value of length
//First Trial
q=3D-3;
qdash=q*100^2*60;
n=0.55425-(0.0001386*qdash);
yo=(n*q/(s^0.5))^0.6;
y=0.665*yo;
L=(q/I)*(1-%e^(-d/y));
//second trial
q=3.15D-3;
qdash=q*100^2*60;
n=0.55425-(0.0001386*qdash);
yo=(n*q/(s^0.5))^0.6;
y=0.665*yo;
L=(q/I)*(1-%e^(-d/y));
q=9*Lgiven*q*1000/L;
q=round(q*10)/10;
mprintf("\nPart (b):");
mprintf("\nDischarge for each border strip=%f lps.",q);
```

Scilab code Exa 2.4 EX2 4

```
//example 2.4
//calculate
//deep percolation loss
//water application efficiency and time of
   irrigation.

clc;
//Given
B=12;//breadth of basin
L=36//length of basin
d=70//depth of irrigation
Ic=70//cumulative infiltration
kdash=9;
ndash=0.42;
//Part (a)
a=5;
b=0.6;
q=1.5;//stream size
Q=(q*B)/1000;
tl=(L/a)^(1/b);
td=(Ic/kdash)^(1/ndash);
T=tl+td;
p=(1-(td/T)^(ndash))*100;
eita=(1-p/100)*100;
Tdash=(d*L*B)/(10*eita*Q*60);
p=round(p*100)/100;
eita=round(eita*100)/100;
Tdash=round(Tdash*10)/10;
mprintf("Part (a):")
mprintf("\nDeep percolation loss= %f percent.",p);
mprintf("\nWater application efficiency= %f percent.
   ",eita);
mprintf("\nTime of irrigation= %f minutes.",Tdash);
//part (b)
a=8;
b=0.6;
q=3;
Q=(q*B)/1000;
```

```
tl=(L/a)^(1/b);
td=(Ic/kdash)^(1/ndash);
T=tl+td;
p=(1-(td/T)^(ndash))*100;
eita=(1-p/100)*100;
Tdash=(d*L*B)/(10*eita*Q*60);
p=round(p*100)/100;
eita=round(eita*100)/100;
Tdash=round(Tdash*10)/10;
mprintf("\nPart (b):")
mprintf("\nDeep percolation loss= %f percent.",p);
mprintf("\nWater application efficiency= %f percent.",eita);
mprintf("\nTime of irrigation= %f minutes.",Tdash);
```

Scilab code Exa 2.5 EX2 5

```
//example 2.5
//calculate
//size of cut-back stream.
//time required for putting 37.5 mm depth of water
//average depth of water applied

clc;
//given
d=37.5//crop water requirement
W=1//furrow spacing
L=120//length of furrow
n=-0.49;
k=38;
Ttotal=143;//Total time of irrigation
A=[0 23 52 88 127]//given values of time of advance

for i=1:5//loop to find respective values of time of ponding
```

```
    B(i)=143-A(i);
end


for j=1:5//loop to find respective furrow infiltration
    C(j)=B(j)^(n)*k;
end


for K=1:4//loop to find respective average infiltration

   D(K)=(C(K)+C(K+1))/2;
end

E(1)=D(1);
for l=2:4//loop to determine cumulative infiltration
    E(l)=D(l)+E(l-1);
end
I=E(4);

T=(30*d*W*(n+1)/k)^(1/(n+1));
dav=((24.5*Ttotal)+(I*(T-Ttotal)))/L;
q=((120*37.5)-(24.5*143))/62;
T=round(T);
dav=round(dav*10)/10;
q=round(q*100)/100;
I=round(I*100)/100;
mprintf("Maximum size of cut-back stream=%f lpm.",I);
mprintf("\nMinimum size of cut-back stream=%f lpm.",q);
mprintf("\nTime required for putting 37.5mm depth of water=%f minutes.",T);
mprintf("\nAverage depth of water required=%f mm.",dav);
```

Scilab code Exa 2.6 EX2 6

```

//example 2.6
//calculate Average depth of water applied
clc;
//Given
L=100;//length of furrow
W=1;//furrow spacing
s=0.3//longitudnal slope of furrow
t1=80//initial time flow of  stream
t2=35//final time flow of stream
qm=0.6/s;
q=qm*0.4;
dav=((q*t2*60)+(2*t1*60))/100;
mprintf("Average depth of water applied=%f mm.",dav)
    ;
```

Scilab code Exa 2.7 EX2 7

```


//example 2.7
//calculate
//time required to irrigate
//maximum area that can be irrigated
clc;
//Given
Q=0.0072;//discharge through well
y=0.1;//average depth of flow
I=0.05//infiltration capacity of soil
A=0.04//area of land
```

```
t=(2.303*y*60/I)*log10(Q/(Q-I*A));
Amax=Q/I;
t=round(t*100)/100;
mprintf("Time required to irrigate=%f minutes.",t);
mprintf("\nMaximum area that can be irrigated=%f ha.
  ",Amax);
```

Chapter 3

WATER REQUIREMENTS OF CROPS

Scilab code Exa 3.1 EX3 1

```
//example 3.1
//classify the irrigation water
clc;
//Given
Na=24;                                          //
   concentration of sodium ion
Ca=3.6;                                         //
   concentration of calcium ion
Mg=2;                                          //
   concentration of magnesium ion
EC=180;                                        //
   electrical conductivity
SAR=Na/(((Ca+Mg)/2)^(0.5));                    //
   Sodium absorption ratio
SAR=round(SAR*100)/100;
mprintf("SAR=%f.",SAR);
mprintf("\nWater falls under S2 class.");    //from
```

```
    table 3.2
mprintf("\nFor EC=180,");
mprintf("\nwater falls under C1 class.");     //from
    table 3.1
mprintf("\nWater is medium sodium and low saline
    water.");
```

Scilab code Exa 3.2 EX3 2

```


//example 3.2
//calculate
//Depth of moisture in root zone at field capacity
//Depth of moisture in root zone at permanent
    wilting point
//Depth of moisture available in root zone
clc;
//Given
gammad=15;          //dry weigth of soil
gammaw=9.81;        //unit weigth of water
Fc=0.3;             //field capacity
pwp=0.08;           //permanent wilting point
d=0.8;              //root zone depth
d1=gammad*Fc*1000/gammaw;
d2=gammad*pwp*1000/gammaw;
d3=gammad*d*(Fc-pwp)*1000/gammaw;
d1=round(d1);
d2=round(d2);
d3=round(d3);
mprintf("Depth of moisture in root zone at field
    capacity=%f mm/m.",d1);
mprintf("\nDepth of moisture in root zone at
    permanent wilting point=%f mm/m.",d2);
mprintf("\nDepth of moisture available in root zone=
```

```
%f mm/m.",d3);
```

Scilab code Exa 3.3 EX3 3

```


//example 3.3
//calculate
//depth upto which soil profile is wetted
clc;
//Given
gammad=15.3;        //dry weigth of soil
gammaw=9.81;        //unit weigth of water
Fc=0.15;            //field capacity
Mc=0.08;            //moisture content before
   irrigation
D=60;               //Depth of water applied
d=(gammaw*D)/(gammad*(Fc-Mc));
d=round(d);
mprintf("Depth upto which soil profile is wetted=%f
   mm.",d);
```

Scilab code Exa 3.4 EX3 4

```


//example 3.4
//calculate
//Depth of water required to irrigate the soil
clc;
//Given
Sg=1.6;      //Apparent specific gravity
Fc=0.2;          //Field capacity
```

```
M1=150;           //mass of sample soil
M2=136;           //mass of sample after drying
d=0.9;            //depth of soil to be irrigated
Mc=(M1-M2)/M2;
D=Sg*d*1000*(Fc-Mc);
D=round(D);
mprintf("Depth of water required to irrigate the
   soil=%f mm.",D);
```

Scilab code Exa 3.5 EX3 5

```


//example 3.5
//calculate Field Capacity
clc;
//Given
d=2;               //root zone depth
Wc=0.05;           //existing water content
gammad=15;         //dry density of soil
gammaw=9.81;       //unit weigth of water
Vw=500             //water applied to the soil
Wl=0.1;            //water loss
A=1000;            //area of plot
Vu=Vw*0.9;         //volume of water used in soil
Wu=Vu*gammaw;      //weigth of water used in soil
Ws=A*d*gammad;     //total dry weigth of soil
Wa=Wu*100/Ws;      //percent water added
Fc=Wc*100+Wa;
Fc=round(Fc*100)/100;
mprintf("The Field Capacity of soil is=%f percent.",
   Fc);
```

Scilab code Exa 3.6 EX3 6

```
//example 3.6
//calculate
//storage capcity of soil
//Water depth required to be applied
clc;
//Given
Fc=0.22;                                          //Field Capacity
wc=0.1;  //wilting coefficient
gammad=15;  //dry unit weigth of soil
gammaw=9.81;  //unit wiegth of water
d=0.7;  //root zone depth
w=0.14;  //falled moisture content
E=0.75;                                           //water application efficiency
SC=gammad*d*(Fc-wc)*100/gammaw;
D=gammad*d*(Fc-w)*1000/gammaw;
FIR=D/E;                                          //Field irrigation requirement
SC=round(SC*10)/10;
D=round(D);
FIR=round(FIR)+1;
mprintf("Maximum storage capacity of soil=%f cm.",SC
  );
mprintf("\nWater depth required to be applied=%f mm"
  ,D);
mprintf("\nField Irrigation Requirement=%f mm",FIR);
```

Scilab code Exa 3.7 EX3 7

```
//example 3.7
//calculate watering frequency
clc;
//Given
Fc=0.27;            //Field capacity
pwp=0.14;          //permanent wilting point
gammad=15;         //dry density of soil
gammaw=9.81;       //unit weigth of water
d=0.75;           //effective depth of root zone
Du=11;           //daily consumptive use of water
Am=Fc-pwp;       //Available moisture
//let readily available moisture be 80 percent of
   available moisture
RAm=0.8*Am;
Mo=Fc-RAm;
D=gammad*d*(Fc-Mo)*100/gammaw;
WF=D*10/Du;
mprintf("Watering Frequency=%i days.",WF);
```

Scilab code Exa 3.8 EX3 8

```
//example 3.8
//calculate
//net depth of irrigation water reqiured
//time required to irrigate field
clc;
```

```
//given
Fc=0.22;      //Field capacity
Sg=1.56;      //Apparent specific gravity
d=0.6;        //root zone depth
//irrigation is started when 70 percent of moisture is used
l=250;         //length of field
b=40;          //width of field
q=20;          //Discharge


m=(1-0.7)*Fc;
D=Sg*d*(Fc-m)*1000;
A=l*b;
t=A*D/(q*3600);
D=round(D);
t=round(t);
mprintf("Net depth of irrigation water required=%f mm.",D);
mprintf("\nTime required to irrigate field=%f hours.",t);
```

Scilab code Exa 3.9 EX3 9

```


//example 3.9
//calculate delta for crop
clc;
//Given
B=110;      //Base period
D=1400;     //Duty of water

delta=8.64*B*100/D;
delta=round(delta);
```

```
mprintf("Delta for crop is=%f cm.",delta);
```

Scilab code Exa 3.10 EX3 10

```


//example 3.10
//calculate Duty of water
clc;
//Given
B=120;      //Base period
delta=92;   //total depth requirement of crop

D=8.64*B*100/delta;
D=round(D);
mprintf("Duty of water=%f hectares/cumec.",D);
```

Scilab code Exa 3.11 EX3 11

```


//example 3.11
//calculate Dischage required at head of canal
clc;
//Given
Cr=2;            //crop ratio
A=80000;         //Area of field
CI=85;           //percent field culturable
   irrigable
IK=30;           //irrigation intensity during
   kharif season
IR=60;           //irrigation intensity for rabi
   season
```

```
DuK=800;            //Duty of water for kharif season
DuR=1700;           //Duty of water for rabi season

CIA=A*CI/100;     //Culturable irrigable area
AK=CIA*IK/100;    //Area under kharif season
AR=CIA*IR/100;    //Area under rabi season
DK=AK/DuK;
DR=AR/DuR;
mprintf("Dischage required at head of canal during
   Kharif season=%f cumecs.",DK);
mprintf("\nDischage required at head of canal during
    Rabi season=%f cumecs.",DR);
mprintf("\nWater requirement during kharif is
   greater than during rabi season");
mprintf("\nHence,canal should be designed to carry
   discharge of %f cumecs.",DK);
```

Scilab code Exa 3.12 EX3 12

```


//example 3.12
//calculate Dischage required at head of canal
clc;
//Given
CA=2600;        //culturable area
IS=20;          //irrigation intensity for sugarcane
IR=40;          //irrigation intensity for rice
DuS=750;          //Duty of water for sugarcane
DuR=1800;         //Duty of water for rice
PK=1.2;           //Peak demand

AS=CA*IS/100;      //Area under sugarcane
AR=CA*IR/100;      //Area under rice
DS=AS/DuS;
```

```
DR=AR/DuR;
DT=DS+DR;
DD=PK*DT-0.005333+0.01;
DR=round(DR*1000)/1000;
DT=round(DT*1000)/1000;
mprintf("Water required for Rice=%f cumecs.",DR);
mprintf("\n Sugarcane is a perennial crop.");
mprintf("\nHence,Water required for Sugarcane=%f
    cumecs.",DT);
mprintf("\nDesign dischage to meet the peak demand=
    %f cumecs.",DD);
```

Scilab code Exa 3.13 EX3 13

```


//example 3.13
//compare the efficiency
clc;funcprot(0);
//given
ql=20;            //discharge in left branch
Al=20000;         //culturable area in left branch
Bl=120;           //Base period in left branch
Il=0.8;           //intensity of rabi in left branch
qr=8;             //discharge in rigth branch
Ar=12000;         //culturable area in rigth branch
Br=120;           //Base period in rigth branch
Ir=0.5;           //intensity of rabi in rigth branch

//for left canal
ARl=Al*Il;
Dl=ARl/ql;
mprintf("Duty for left canal is=%i hectares/cumecs."
    ,Dl);

```

```
//for rigth canal
ARr=Ar*Ir;
Dr=ARr/qr;
mprintf("\nDuty for left canal is=%i hectares/cumecs
    .",Dr);
mprintf("\nSince,left canal has higher duty,it is
    more efficient");
```

Scilab code Exa 3.14 EX3 14

```


//example 3.14
//calculate Discharge for water course
clc;
//Given
CA=1200;          //culturable area
IA=0.4;            //intensity of irrigation of crop A
IB=0.35;           //intensity of irrigation of crop B
bA=20;             //kor period of crop A
bB=15;             //kor period of crop B
deltaA=0.1;            //kor depth of crop A
deltaB=0.16;            //kor depth of crop B

//crop A
A=CA*IA;
Du=8.64*bA/deltaA;
qA=A/Du;
qA=round(qA*1000)/1000;
mprintf("Discharge required for crop A=%f cumec.",qA
    );

//crop B
A=CA*IB;
Du=8.64*bB/deltaB;
```

```
qB=A/Du;
qB=round(qB*1000)/1000;
mprintf("\nDischarge required for crop B=%f cumec.",
   qB);
D=qA+qB;
D=round(D*10)/10;
mprintf("\nDesign discharge of water course=%f cumec
   .",D);
```

Scilab code Exa 3.15 EX3 15

```


//example 3.15
//calculate
//duty of irrigation water
//discharge required
clc;
//Given
B=12;           //transplantaion period
D=0.5;          //total depth of water required by the
    crop
R=0.1;          //rain falling on field
L=0.2;          //loss of water
A=600;          //irrigated area
I=0.6;          //intensity of irrigation
delta=D-R;
Dui=8.64*B/delta;
//since water loss is 20 percent
Du=(1-L)*Dui;
mprintf("Duty of water required=%f hectares/cumec.",
   Du);

TA=I*A;
q=TA/Du;
```

```
q=round(q*100)/100;
mprintf("\nDischarge at head of water course=%f cumecs.",q);
```

Scilab code Exa 3.16 EX3 16

```


//example 3.16
//calculate
//discharge required at the head
//design discharge
clc;
//Given
CF=0.8;                          //Capacity factory
Tf=13/20;                        //time factor
A=[850 120 600 500 360];        //given values of area
B=[320 90 120 120 120];         //given values of Base period
D=[580 580 1600 2000 600];      //given values of duty at head canal

DS=A(1)/D(1);                   //discharge for sugarcane
DOS=A(2)/D(2);                  //discharge for overlap sugarcane
DW=A(3)/D(3);                   //discharge for wheat
DB=A(4)/D(4);                   //discharge for bajri
DV=A(5)/D(5);                   //discharge for vegetables
DR=DS+DW;
DM=DS+DB;
DH=DS+DOS+DV;
mprintf("Maximum demand is in hot weather");
q=DH/Tf;
```

```
D=q/CF;
q=round(1000*q)/1000;
D=round(100*D)/100;
mprintf("\nFull supply discharge at head=%f cumecs",
   q);
mprintf("\nDesign discharge=%f cumecs.",D);
```

Scilab code Exa 3.17 EX3 17

```


//example 3.17
//calculate resvior capacity
clc;
//Given
CL=0.2;                    //Canal loss
RL=0.12;                   //Reservior loss
A=[4800 5600 2400 3200 1400];     //given values of
   area under crop
D=[1800 800 1400 900 700];     //given values of
   duty at field
B=[120 360 200 120 120];      //given values of
   base period

//(a) Wheat
d=A(1)/D(1);
V1=d*B(1);
//(b) Sugarcane
d=A(2)/D(2);
V2=d*B(2);
//(c) Cotton
d=A(3)/D(3);
V3=round(d*B(3));
//(d) Rice
d=A(4)/D(4);
```

```
V4=round(d*B(4));
//(e) vegetables
d=A(5)/D(5);
V5=d*B(5);

Vd=(V1+V2+V3+V4+V5)*8.64;
SC=Vd/((1-CL)*(1-RL));
mprintf("Reservior capacity=%f hectare-metres.",SC);

//Alternative method

for i=1:5
    delta(i)=8.64*B(i)/D(i);
end

for j=1:5
    V(j)=A(j)*delta(j);
end
s=0;
for k=1:5
    s=s+V(k);
end
SC=s/((1-CL)*(1-RL));
SC=round(SC);
mprintf("\n By Alternative method.\nStorage capacity
   =%f hectare-metres.",SC);
```

Scilab code Exa 3.18 EX3 18

```


//example 3.18
//Calculate
//consumptive use
//consumptive irrigatin requirement
```

```
//field irrigatio reqiurement
clc;
//Given
eita=0.7;         //water application efficiency
k=0.75;           //crop factor
T=[19 16 12.5 13];    //given values of temperature
p=[7.19 7.15 7.30 7.03];   //daytime hours of the
    year
RD=1.2;               //rainfall in december
RJ=0.8;               //rainfall in january
for i=1:4
    f(i)=p(i)*(1.8*T(i)+32)/40;
end
s=0;
for i=1:4
    s=s+f(i);
end
C=k*s;
R=RD+RJ;
CIR=C-R;
FIR=CIR/eita;
C=round(10*C)/10;
CIR=round(CIR*10)/10;
FIR=round(FIR*10)/10;
mprintf("Consumptive use=%f cm.",C);
mprintf("\nconsumptive irrigatin requirement=%f cm."
    ,CIR);
mprintf("\nfield irrigatio reqiurement=%f cm.",FIR);
```

Scilab code Exa 3.19 EX3 19

```


//example 3.19
//calculate
```

```
//consumptive use of rice using penman's formula in january
clc;
//Given
L=20;              //latitude of place(degree North)
T=15;              //mean monthly temperature(degree celcius)
RH=0.5;             //relative humidity
E=250;             //elevation of area
V=25;              //wind velocity at 2 m heigth
//from table 3.10
VP=12.79;          //saturation vapour pressure
s=0.8;             //slope of curve between vapur pressure and temperature
//from table 3.11
R=10.8;
//from table 3.12
N=11.1;
//from table 3.9
n=7.74;
p=n/N;
e=VP*RH;
Ea=0.002187*(160+V)*(VP-e);
r=0.2;
alpha=0.49;
sigma=2.01D-9;
Ta=293;
H=R*(1-r)*(0.29*cos(%pi/9)+0.55*p)-sigma*Ta^4*(0.56-0.092*e^0.5)*(0.10+0.9*p);
Et=(s*H+alpha*Ea)*31/(s+alpha);
Et=round(Et*10)/10;
mprintf("consumptive use of rice in january=%f mm of water.",Et);
```

Scilab code Exa 3.20 EX3 20

```
//example 3.20
//calculate
//maximum storage capacity;depth of irrigation water
//field irrigation requirement;water required at
    canal outlet
clc;
//Given
Fc=0.27;                    //Field capacity
pwp=0.13;                   //permanent wilting point
d=80;                      //depth of soil(cm)
gammad=1.5;                //dry unit weigth of soil(g/cc
    )
gammaw=1;                  //unit weigth of water(g/cc)
M=0.18;                    //avearge soil moisture
eita=0.8;                  //field efficiency
FC=0.15;                   //field channel
SC=gammad*d*(Fc-pwp)/gammaw;
D=gammad*d*(Fc-M)/gammaw;
FIR=D/eita;
W=FIR/(1-FC);
W=round(W*10)/10;
mprintf("maximum storage capacity=%f cm",SC);
mprintf("\ndepth of irrigation water=%f cm",D);
mprintf("\nfield irrigation requirement=%f cm",FIR);
mprintf("\nwater required at canal outlet=%f cm",W);
```

Scilab code Exa 3.21 EX3 21

```
//example 3.21
//calculate reservior capacity
clc;
```

```
//given
W=0.4;                          //amount of water
   available from precipitation
Cl=0.15;                        //Channel loss
RL=0.1;                         //reservior loss
B=[120 320 120 200 100];        //Base period
D=[1800 800 900 1400 1200];     //Duty at field
A=[500 600 300 1200 500];       //Area under crop

for i=1:5
    delta(i)=8.64*B(i)/D(i);
end

for i=1:5
    V(i)=delta(i)*A(i);

end
s=0;
for i=1:5
    s=s+V(i);
end
C=s*(1-W)/((1-Cl)*(1-RL));

mprintf("Reservior capacity=%i ha-m.",C);
```

Scilab code Exa 3.22 EX3 22

```


//example 3.22
//calculate
//discharge required at head of distributory
clc;
//given
GCA=10000;                         //gross commanded area
```

```
CCA=0.75*GCA;                       //Culturable commanded area
IR=0.6;                             //intensity of irrigation during rabi season
IK=0.3;                             //intensity of irrigation during kharif season
DuR=2500;                           //duty during rabi season
DuK=1000;                           //duty during kharif season
AR=IR*CCA;                          //area under irrigation in rabi season
AK=IK*CCA;                          //area under irrigation in kharif season
DR=AR/DuR;
DK=AK/DuK;
mprintf("discharge required at head of distributory=%f cumecs.",DK);
```

Scilab code Exa 3.23 EX3 23

```


//example 3.23
//calculate irrigation schedule
clc;
//given
Fc=0.18;            //field capacity
wc=0.07;            //wilting coefficient
Sg=1.35;            //bulk density of soil
d=1.2;              //root zone depth
m=Fc-wc;
mo=wc+m/3;
dw=100*Sg*d*(Fc-mo);
mprintf("Depth of water required=%f cm",dw);
```

```
ev1=1.1;           //average evapotranspiration rates in 1 NOV-30 NOV
ev2=1.7;           //average evapotranspiration rates in 1 DEC-31 DEC
ev3=2.4;           //average evapotranspiration rates in 1 JAN-31 JAN
ev4=1.5;           //average evapotranspiration rates in 1 FEB-28 FEB
ev5=3.5;           //average evapotranspiration rates in 1 MAR-25 MAR
//irrigation requirement from 1 NOV to 3 JAn
dev=(ev1*30+ev2*31+ev3*3)/10;
mprintf("\n\nWater consumed by evapotranspiration=%f cm.",dev);
mprintf("\nNo water is required during 1 NOV-3 JAN");

//irrigation requirement after 3rd JAN
ws=(ev3-1.5)*16/10;                //water consumed from soil from 4 JAN-19 JAN
ts=ws+dev;                  //water withdrawn from soil from 1 NOV-19 JAN
s=(dw-ts)*10;
day=s/ev3;
depth=ts+(4*ev3)/10+(2*ev3)/10;
mprintf("\n\ndepth of water required in first irrigation=%f cm.",depth);
///irrigation requirement from 26 JAn to 25 MAR
w1=ev3*6;
w2=ev4*28;
w3=ev5*25;
W=w1+w2+w3;
x=(dw*10-(14.4+42))/ev5;
mprintf("\n\nHence second irrigation is required after %f days i.e on 18th March.",x);
depth1=(W-(dw*10))/10;
mprintf("\nrequired water depth=%f cm",depth1);
mprintf("\n\nFirst Watering on 29 JAn and 30 JAN=%f
```

```
    cm.\nSecond watering required on 18th March=%f cm
    .",depth,depth1);
```

Scilab code Exa 3.24 EX3 24

```


//example 3.24
//calculate daily consumptive
//discharge in canal
clc;
//given
Fc=0.26;                    //Field capacity of soil
A=3000;                     //Area of field
OM=0.12;                    //optimum moisture
pwp=0.1;                    //permanent wilting point
d=80;                      //depth of root zone
RD=1.4;                     //relative density of soil
f=10;                       //frequency of irrigation
eita=0.23;                  //overall efficiency
D=RD*d*(Fc-OM);
U=D*10/f;
Wr=A*D*100;
q=Wr/(f*24*3600);
q=round(q*100)/100;
mprintf("daily consumptive=%f mm.",U);
mprintf("\ndischarge in canal=%f q cumecs.",q);
```

Scilab code Exa 3.25 EX3 25

```


//example 3.25
```

```
//calculate total water to be delivered
clc;
//given
C1=0.2;                //consumptive requirement of
    crop for 1 to 15 days
C2=0.3;                //consumptive requirement of
    crop for 16 to 40 days
C3=0.5;                //consumptive requirement of
    crop for 41 to 50 days
C4=0.1;                //consumptive requirement of
    crop for 51 to 55 days
A=50;                  //area of land
wr=5;                  //presowing water requirement
R=3.5;                 //rainfall during 36th and 45th
     day
w1=15*C1*100;
w2=25*C2*100;
w3=10*C3*100;
w4=5*C4*100;
w5=5*100;
W=w1+w2+w3+w4+w5;
ER=3.5*100;
q=(W-ER)*A;
mprintf("total water to be delivered=%i cubic metre.
    ",q);
```

Scilab code Exa 3.26 EX3 26

```


//example 3.26
//calculate watering interval
clc;
//given
Fc=0.3;            //field capacity
```

```
pwp=0.11;          //permanent wilting percent
gammad=1300;       //density of soil
gammaw=1000;       //density of water
d=700;             //root zone depth
CW=12;             //daily consumptive use of water
WHC=Fc-pwp;
mo=Fc-(0.75*WHC);
D=gammad*d*(Fc-mo)/gammaw;
I=D/CW;
mprintf(" watering interval=%i days",I);
```

Scilab code Exa 3.27 EX3 27

```


//example 3.27
//calculate Duty of water
//discharge required in water course
clc;
//given
A=1000;                 //total area
AI=0.7*A;               //area under irrigation
B=15;                   //Base period
d=500;                  //depth of water required during transplantation
R=120;                  //useful rain falling
Wl=0.2;                  //water loss
delta=d-R;
Du=8.64*B*1000/delta;
DuH=Du*(1-Wl);
q=AI/DuH;
q=round(q*100)/100;
mprintf("Duty of water=%i hec/cumec.",Du);
mprintf("\ndischarge required in water course=%f cumecs.",q);
```

Scilab code Exa 3.28 EX3 28

```


//example 3.28
//calculate reservior capacity
clc;
//given
Ar=4000;                //culturable commanded area
CL=0.25;               //canal loss
RL=0.15;                //reservior loss
B=[120 360 180 120 120];  //base period
D=[1800 1700 1400 800 700];//duty of water
I=[20 20 10 15 15];        //intensity of irrigation
for i=1:5
    A(i)=Ar*I(i)/10;      //area under crop
end
for i=1:5
    Q(i)=A(i)/D(i);          //discharge required
end
for i=1:5
    V(i)=8.64D4*Q(i)*B(i);      //quantity of water
end
s=0;
for i=1:5
    s=s+V(i);
end
SC=round(s/((1-CL)*(1-RL)*1000000));
mprintf("Storage capacity=%iD+06 cubic metre.",SC);
```

Chapter 4

HYDROLOGY

Scilab code Exa 4.1 EX4 1

```
//example 4.1
//calculate mean rainfall;additional guages needed
clc;funcprot(0);
//given
p=[78.8 90.2 98.6 102.4 70.4];    //rain guage
    readings at respective stations
s=0;
for i=1:5
    s=s+p(i);
end
pavg=s/5;
u=0;
for i=1:5
    u=u+(p(i)-pavg)^2;
end
sx=(u/4)^0.5;
Cv=sx*100/pavg;
N=(Cv/6)^2;
N=round(N*100)/100;
```

```
mprintf("mean rainfall=%f cm.",pavg);
mprintf("\ntotal stations needed=%f.",N);
//taking N=7
N=7;
n=N-5;
mprintf("\nadditional guages needed=%i.",n);
```

Scilab code Exa 4.2 EX4 2

```


//example 4.2
//calculate precipitation at A using arithmatic mean
    method
clc;funcprot(0);
//given
pB=48;          //precipitation at B
pC=51;          //precpitation at C
pD=45;          //precipitation at D

pA=(pB+pC+pD)/3;
mprintf("precipitation at A=%i mm.",pA);
```

Scilab code Exa 4.3 EX4 3

```


//example 4.3
//calculate precipitation at x
clc;funcprot(0);
//given
pA=6.6;          //precipitation at A
pB=4.8;          //precpitation at B
```

```
pC=3.7;           //precipitation at C
nA=72.6;          //normal precipitation at A
nB=51.8;          //normal precipitation at B
nC=38.2;          //normal precipitation at C
nX=65.6;          //normal precipitation at X

pX=(nX*pA/nA+nX*pB/nB+nX*pC/nC)/3;
pX=round(pX*100)/100;
mprintf("precipitation at x=%f cm.",pX);
```

Scilab code Exa 4.4 EX4 4

```


//example 4.4
//calculate precipitation at A by inverse distance
   method
clc;funcprot(0);
//given
pB=74;            //precipitation at B
pC=88;            //precpitation at C
pD=71;            //precipitation at D
pE=80;            //precipitation at E
Bx=9;By=6;
Cx=12;Cy=-9;
Dx=-11;Dy=-6;
Ex=-7;Ey=7;
Ax=0;Ay=0;
Db=(Bx^2+By^2);
Dc=(Cx^2+Cy^2);
Dd=(Dx^2+Dy^2);
De=(Ex^2+Ey^2);
Wb=1/Db;
Wc=1/Dc;
Wd=1/Dd;
```

```
We=1/De;
s=pB*Wb+pC*Wc+pD*Wd+pE*We;
pA=s/(Wb+Wc+Wd+We);
pA=round(pA*10)/10;
mprintf("precipitation at A=%f mm.",pA);
```

Scilab code Exa 4.5 EX4 5

```


//example 4.5
//calculate average rainfall using
//arithmatic average method
//isohytel method
//thiesson polygon method
clc;funcprot(0);
//given
p=[58 61 69 56 84 86 69 79 71];    //values of
    precipitation
s=0;
for i=1:9
    s=s+p(i);
end
ar=s/9;
ar=round(ar*10)/10;
mprintf("using arithmatic average method:")
mprintf("\nAverage rainfall=%f cm.",ar);

I=[86 85 80 75 70 65 60 55 50];          //isphytes
A=[0.43 5.20 4.0 5.04 5.85 4.53 4.09 1.27];  //area
    between isohytes
for i=1:8
    a(i)=(I(i)+I(i+1))/2;
end
for i=1:8
```

```
    P(i)=A(i)*a(i);
end
s=0;
for i=1:8
    s=s+P(i);
end
t=0;
for i=1:8
    t=t+A(i);
end
ar=s/t;
ar=round(ar*10)/10;
mprintf("\n\nisohytel method:")
mprintf("\nAverage rainfall=%f cm.",ar);

A=[3.26 0.39 1.61 2.04 2.46 0.84 3.91 5.09 0.41 3.94
    2.06 4.40];   //thiessen area
p=[58 63 71 69 86 81 84 56 53 69 61 79];
   //observed precipitation
for i=1:12
    P(i)=A(i)*p(i);
end
s=0;
for i=1:12
    s=s+P(i);
end
disp(s);
t=0;
for i=1:12
    t=t+A(i);
end
ar=s/t;
ar=round(ar*10)/10;
mprintf("\n\nthiesson polygon method:")
mprintf("\nAverage rainfall=%f cm.",ar);
//mean rainfall obtained by thiesson polygon method
   is different from book as product(A*P) is round
   offed in book.
```

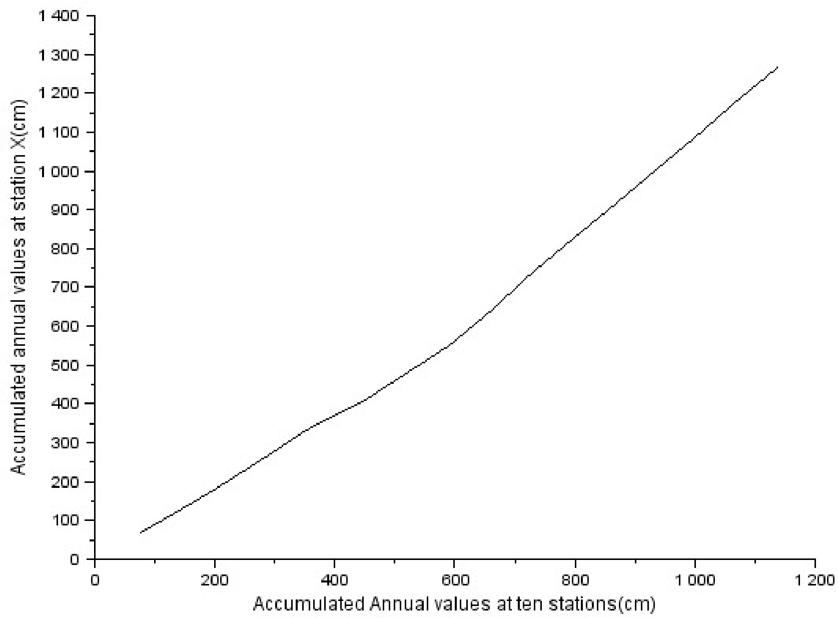

Figure 4.1: EX4 6

Scilab code Exa 4.6 EX4 6

```
//example 4.6
//ceck whether data at station X is consistence
//year in which regime is indicated
//compute the adjusted rainfall atX
clc;funcprot(0);
//given
```

```
X=[69 55 62 67 87 70 65 75 90 100 90 95 85 90 75 95];    //annual rainfall at X
Y=[77 62 67 68 86 90 65 75 70 70 70 75 65 70 55 75];        //average rainfall at 10 base stations
cx(1)=69;                    //accumulated annual values at station X
for i=2:16
    cx(i)=cx(i-1)+X(i);
end
cy(1)=77;
for i=2:16
    cy(i)=cy(i-1)+Y(i);   //accumulated annual values at ten stations

end

//since curve is not having unform slope
mprintf("Record at X is not consistent.");
mprintf("\nFrom the curve regime is observed in the year 1978.")

Q=[1970 1971 1972 1973 1974 1975 1976 1977];
O=[95 75 90 85 95 90 100 90];
for i=1:8
    A(i)=0.7051*O(i);
end
mprintf("\n\nYear            Observed rainfall            Adjusted rainfall");
for i=1:8
    mprintf("\n%i                %i                %i",Q(i),O(i),A(i));
end
//graph is plotted between cx and cy
```

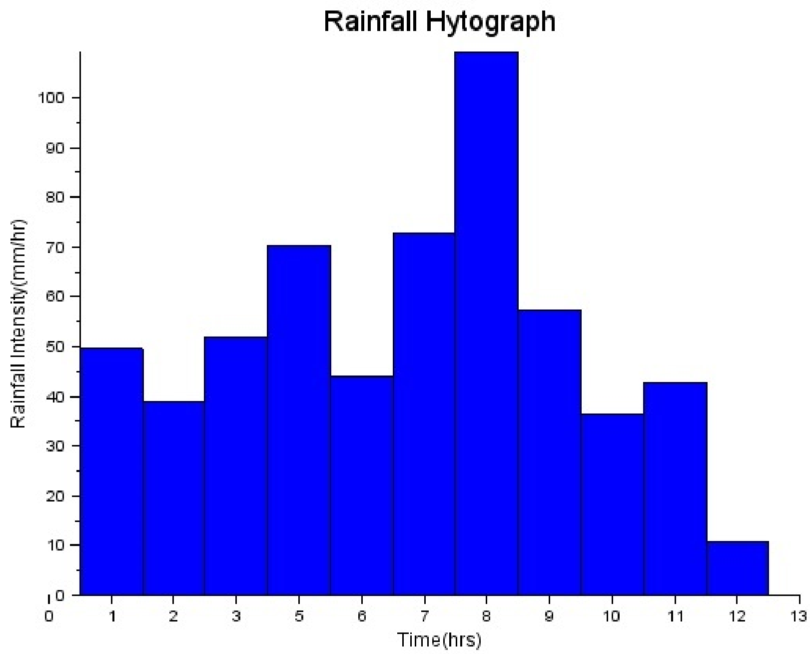

Figure 4.2: EX4 7

Scilab code Exa 4.7 EX4 7

```


//example 4.7
//construct hyetograph
clc;funcprot(0);
//given
c=[0 12.4 22.1 35.1 52.7 63.7 81.9 109.2 123.5 132.6
    143.3 146 146];  //cumulative rainfall
T=[0:1:13];          //Time
t=15/60;            //time interval
r(1)=0;
mprintf("Rainfall intensity:");
I(1)=0;
for i=2:13
  r(i)=c(i)-c(i-1);
   I(i)=r(i)/t;          //Rainfall intensity
mprintf("\n%f",I(i));
end

//graph is plotted between I and T
```

Scilab code Exa 4.8 EX4 8

```


//example 4.8
//compute maximum rainfall intensities for
   15,30,45,60,90,120 minutes
//plot intensity duration graph
clc;funcprot(0);
//given
```

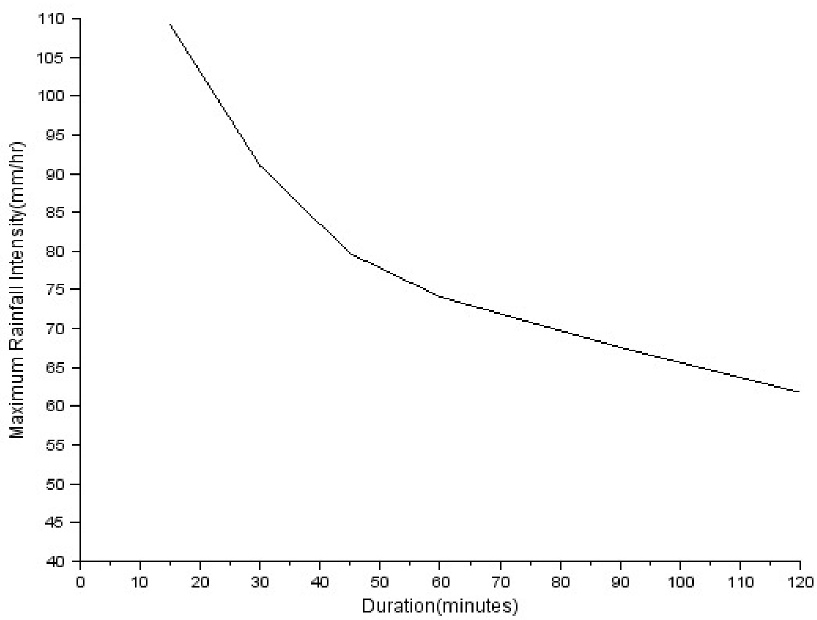

Figure 4.3: EX4 8

```
CR=[0 12.4 22.1 35.1 52.7 63.7 81.9 109.2 123.5 132.6 143.3 146.0 146.0];  //cumulative rainfall

c15(2)=12.4;
c30(3)=22.1;
c45(4)=35.1;
c60(5)=52.7;
c90(7)=81.9;
c120(9)=123.5;
for i=3:13
    c15(i)=CR(i)-CR(i-1);
end
for i=4:13
    c30(i)=CR(i)-CR(i-2);
end
for i=5:13
    c45(i)=CR(i)-CR(i-3);
end
for i=6:13
    c60(i)=CR(i)-CR(i-4);
end
for i=8:13
    c90(i)=CR(i)-CR(i-6);
end
for i=10:13
    c120(i)=CR(i)-CR(i-8);
end
mprintf("15min                30min                45 min                60min          90min          120min");
for i=1:13
    mprintf("\n%f          %f          %f          %f          %f          %f",c15(i),c30(i),c45(i),c60(i),c90(i),c120(i));
end
I=[109.2 91 79.7 74.1 67.6 61.75];    //maximum intensity at respective durations
D=[15 30 45 60 90 120];               //durations
```

```
//greph is plotted between I and D
```

Scilab code Exa 4.9 EX4 9

```


//example 4.9
//calculate preciptation value which has recurrence
   period of 6 years
clc;funcprot(0);
//given
p=[475 377 731 1066 361 305 926 628 409 236 337
   853];   //precipitation value
N=12;                          //total number of years
T=6;                           //recurrence interval
m=N/T;
mprintf("Ranking of storm=%i.",m);
//hence pick 2nd severest storm
mprintf("\npreciptation value which has recurrence
   period of 6 years=%i mm.",p(7));
```

Scilab code Exa 4.10 EX4 10

```


//example 4.10
//calculate average depth of precipitation using
   depth area curve
clc;funcprot(0);
//given
I=[25:-1:16];          //isohytes
```

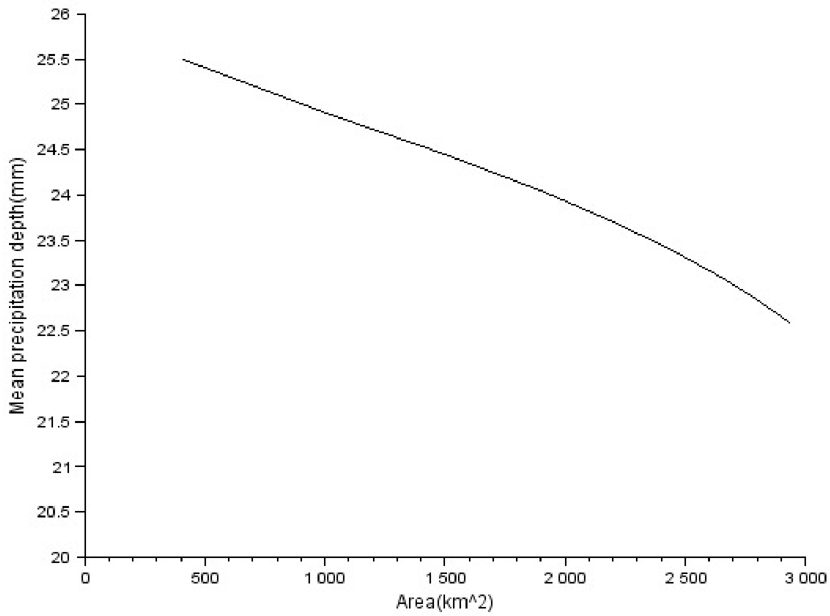

Figure 4.4: EX4 10

```
a=[407 1008 1522 1909 2216 2460 2651 2782 2910
    2936];  //enclosed area
ia(1)=407;
for i=2:10
    ia(i)=a(i)-a(i-1);
end
r=[25.5:-1:16.5]
for i=1:10
    rv(i)=r(i)*ia(i);
end
cv(1)=10378;
for i=2:10
    cv(i)=cv(i-1)+rv(i);
end
for i=1:10
    eud(i)=cv(i)/a(i);                    //mean
       precipitation
end

mprintf("From depth area curve we obtain average
    depth of precipitation=24.1 mm for an \narea of
    1800 sq. km.");
//graph is plotted between eud and a.
```

Scilab code Exa 4.11 EX4 11

```


//example 8.11
//calculate
//24h max. rainfall with return period of 8,15 and
    25.
//24h max rainfall with 40%,24% and 8% probability.
```

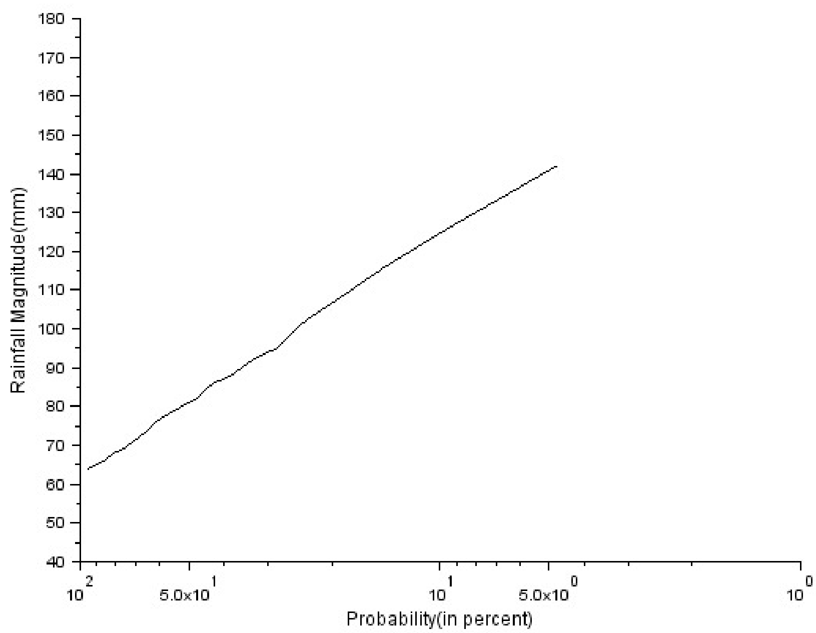

Figure 4.5: EX4 11

```
//probabilty of rainfall of magnitude equal to or exceeding 100 mm.
clc;funcprot(0);
//given
N=20;
r=[142 126 116 108 102 95 92 88 86 82 80 78 76 73 71 69 68 66 65 64];    //rainfall in respective years
m=[1:1:20];              //ranking of storm
for i=1:20
    p(i)=m(i)*100/(N+1);          //probability(percent)
    T(i)=100/p(i);                //recurrence interval
end
//from frequency curve obtained we get
//Part (a)
T1=[8 15 25];
r1=[119 134 149];
mprintf("T(years)            Rainfall(mm)");
for i=1:3
    mprintf("\n%i                        %i",T1(i),r1(i));
end

//Part (b)
p1=[40 24 8];
r2=[87 101 130];
mprintf("\n\nprobability(percent)          Rainfall(mm)");
for i=1:3
    mprintf("\n%i                                   %i",p1(i),r2(i));
end
//graph is plotted on semi-log graph between r and p

mprintf("\n\nFor rainfall=100 m.\nT=4 years.\nProbability=25 percent.");
```

Scilab code Exa 4.12 EX4 12

```


//example 4.12
//plot IDF curve for return period of 10,2 and 1
    years using california formula
clc;funcprot(0);
//given
t=[5 10 20 30 60 90 120];        //duration
//value of P for respective return period is
p10=[10.6 14.7 19.3 20.8 25.5 29 34.7];  //rainfall
    for T=10 years
p2=[8.2 10.3 13.2 14.2 16.6 19.4 21.4];  //rainfall
    for T=2 years
p1=[3.5 6.2 8.9 10 13.2 15 16.5];        //rainfall
    for T=1 year
for i=1:7
    i1(i)=p10(i)*60/t(i);                  //intensity
        of rainfall with return period of 10 years
    i2(i)=p2(i)*60/t(i);                   //intensity
        of rainfall with return period of 2 years
    i3(i)=p1(i)*60/t(i);                   //intensity
        of rainfall with return period of 1 year
end
//graph is plotted between
//t and i1
//t and i2
//t and i3
```

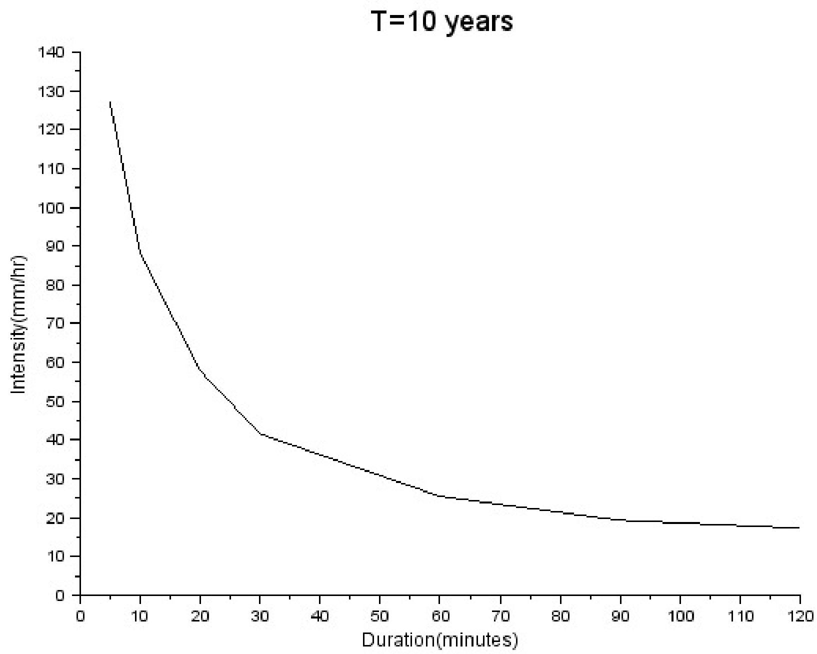

Figure 4.6: EX4 12

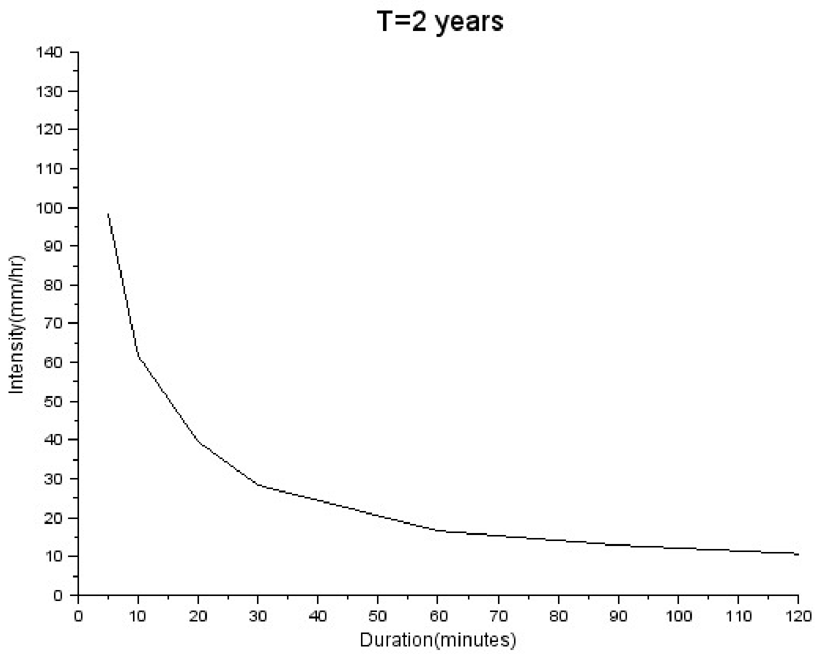

Figure 4.7: EX4 12

Scilab code Exa 4.13 EX4 13

```
//example 4.13
//calculate maximum and minimum rainfall
clc;funcprot(0);
//given
N=20;
m=[1:1:20];                    //rank number
rd=[82 78 75 72 70 68 65 63 61 58 56 54 52 50 46 40 36 34 32 30];    //rainfall in decresing order
for i=1:20
    ri(i)=rd(21-i);
end
for i=1:20
    T(i)=N/(m(i)-0.5);
end
//from the curves
mprintf("maximum rainfall=79cm for T=15 years.");
mprintf("\nminimum rainfall =31 cm for T=15 years.");
//graph is plotted between rd and T;ri and T
```

Scilab code Exa 4.14 EX4 14

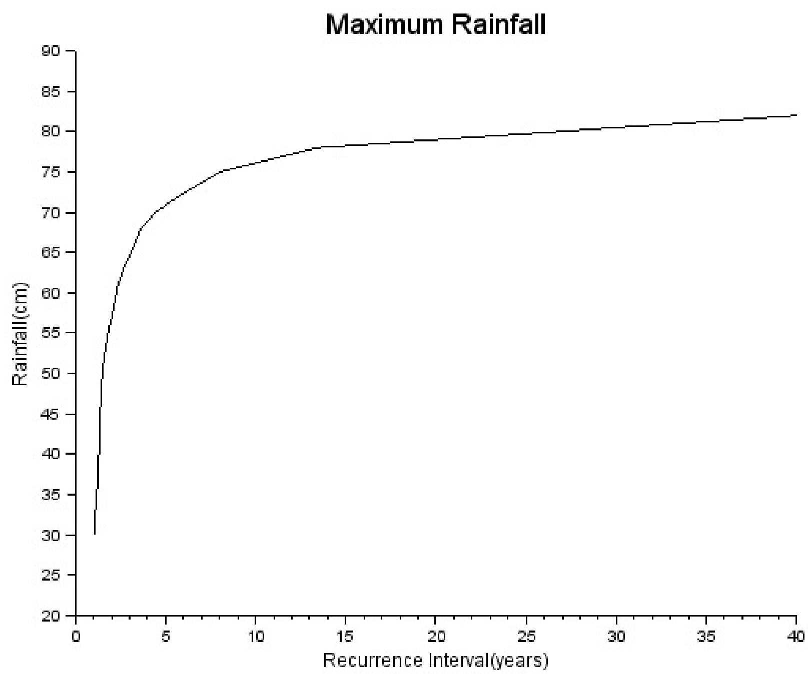

Figure 4.8: EX4 13

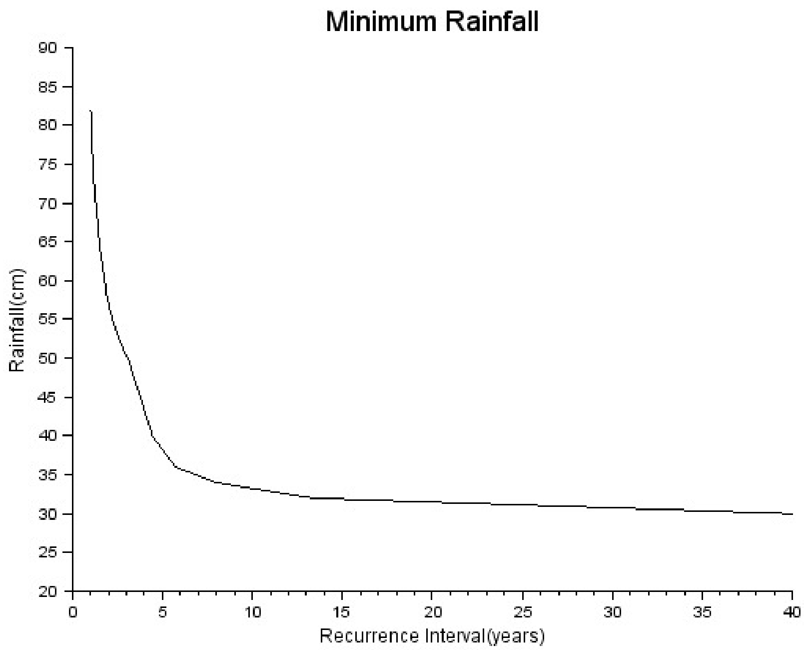

Figure 4.9: EX4 13

```
//example 4.14
//calculate daily lake evaporation
//average evaporation for one week
clc;funcprot(0);
//given
w=[12 5 2 -3 1 6 11];      //water added or taken out
r=[0 6 8 12 9 5 0];        //rainfall
for i=1:7
    pan(i)=w(i)+r(i);      //Pan evaporation
    le(i)=0.8*pan(i);      //lake evaporation
end

s=0;
for i=1:7
    s=s+le(i);
end
mprintf("daily lake evaporation(mm):");
for i=1:7
    mprintf("\n%f",le(i));
end
av=s/7;
av=round(av*100)/100;
mprintf("\n\naverage evaporation for one week=%f mm.
   ",av);
```

Scilab code Exa 4.15 EX4 15

```


//example 4.15
//calculate average evaporation loss from reservior
//total depth and volume of evaporation loss
clc;funcprot(0);
//given
Rh=0.4;          //relative humidity
```

```
A=4.8;            //average surface spread of
   reservior
v3=18;            //wind velocity at 3m above ground
es=31.81;         //saturated vapour pressure
Km=0.36;          //for large deep waters

//using Meyer's formula
ea=es*Rh;
v9=v3*(9/3)^(1/7);
E=Km*(es-ea)*(1+v9/16);
d=7*E;
v=d*A*100/1000;
E=round(E*10)/10;
d=round(d*10)/10;
v=round(v*100)/100;
mprintf("using Meyers formula:");
mprintf("\naverage evaporation loss from reservior=
   %f mm/day.",E);
mprintf("\ntotal depth=%f mm",d);
mprintf("\ntotal volume=%f hectare-m.",v);

//using Rohwer's formula
Pa=760;
vdash=(0.6/2)^(1/7)*18;
E=0.771*(1.465-0.000732*Pa)*(0.44+0.0733*vdash)*(es-
   ea);
d=7*E;
v=d*A*100/1000;
E=round(E*10)/10;
d=round(d*10)/10;
v=round(v*10)/10;
mprintf("\n\nusing Rohwers formula:");
mprintf("\naverage evaporation loss from reservior=
   %f mm/day.",E);
mprintf("\ntotal depth=%f mm",d);
mprintf("\ntotal volume=%f hectare-m.",v);
```

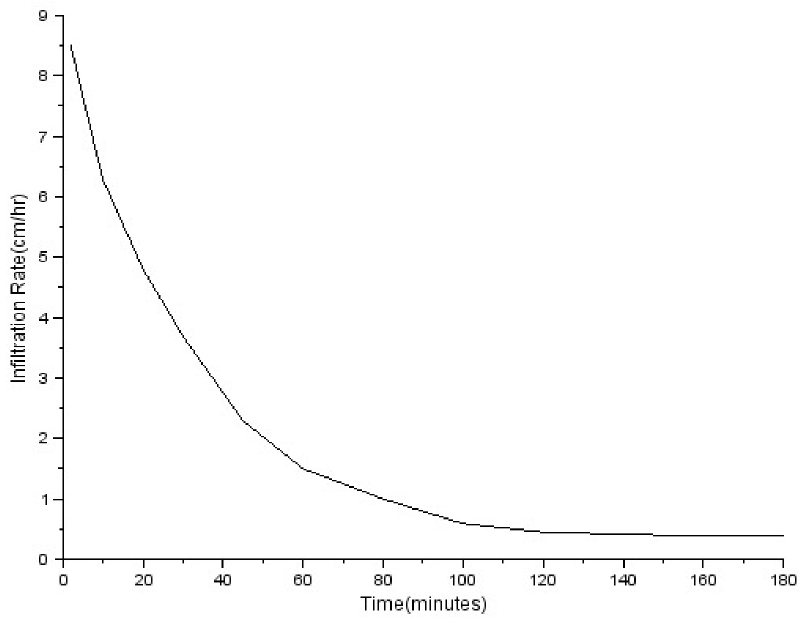

Figure 4.10: EX4 16

Scilab code Exa 4.16 EX4 16

```


//example 4.16
//plot infiltration capacity curve
//calculate constant rate of infiltration
clc;funcprot(0);
//given
D=30;            //diameter of inside ring of
```

```
    infiltrometer
A=%pi*D^2/4;
V=[0 200 470 840 1405 1840 2245 2510 2745 2885 2990
    3130 3270];  //cumulative volume;
t=[0 2 5 10 20 30 45 60 80 100 120 150 180];
                              //Time(minutes)
dt(1)=0;
for i=2:13
    dt(i)=(t(i)-t(i-1))/60;
end
for i=1:13
    F(i)=V(i)/A;
end
Fd(1)=F(1);
for i=2:13
    Fd(i)=F(i)-F(i-1);
end
for i=2:13
    ft(i)=Fd(i)/dt(i);      //infirltration rate
end
//from the graph
mprintf("constant rate of infiltration=0.40 cm/hr.")
    ;
avg10=F(4)*60/10;
avg30=F(6)*60/30;
avg10=round(avg10*100)/100;
avg30=round(avg30*100)/100;
mprintf("\naverage rate of infiltration for first 10
    min=%f cm/hr.",avg10);
mprintf("\naverage rate of infiltration for first 30
    min=%f cm/hr.",avg30);
//graph is plotted between ft and t
```

Scilab code Exa 4.17 EX4 17

```
//example 4.17
//calculate
//drainage desity
//form factor
//channel slope
//average overland flow length
clc;funcprot(0);
//given
A=82;           //area of watershed
d=12.6;         //distance between outlet and farther
   most point
l=440;          //total length of channel
e=656;          //elevation differnce between outlet
   and further most point

Dd=l/A;
ff=A/d^2;
cs=e/(d*1000);
lo=1000/(2*Dd);
Dd=round(Dd*100)/100;
ff=round(ff*1000)/1000;
mprintf("drainage desity=%f km/square.km.",Dd);
mprintf("\nform factor=%f.",ff);
mprintf("\nchannel slope=%f.",cs);
mprintf("\naverage overland flow length=%i m.",lo);
```

Scilab code Exa 4.18 EX4 18

```
//example 4.18
//compute fi and W index
clc;funcprot(0);
```

```
//given
R=3.6;          //surface runoff
r=[0 1.3 2.8 4.1 3.9 2.8 2.0 1.8 0.9];   //rainfall at respective time
t=4;                 //total time
s=0;
for i=3:8
    s=s+r(i)
end
fi=(s-R*2)/6;
//since fi >1.3 and <1.8
mprintf("fi index=%f cm.",fi);
mprintf("\ncomputations are correct.");

s=0;
for i=1:9
    s=s+r(i);
end
P=s/2;
Sr=0;
W=(P-R-Sr)/t;
mprintf("\nW index=%f cm/hr.",W);
```

Scilab code Exa 4.19 EX4 19

```
//example4.19
//calculate fi index and time of rainfall excess
clc;funcprot(0);
//given
T=[1:1:9];        //time from start
r=[0.7 1.4 2.4 3.7 2.9 2.6 1.7 0.8 0.5];   //increamental rainfall
R=9.3;          //total run-off
```

```
s=0;
for i=1:9
    s=s+r(i);
end
ti=s-R;

//first trial
tr=9;   //assumed
fi1=ti/tr;
//this makes 1st,8th and 9th hour ineffective

//second trial
tr=6;
ti=s-R-r(1)-r(8)-r(9);
fi=ti/tr;
for i=1:9
    P(i)=r(i)-fi;
    if (P(i)<0) then
        P(i)=0;
    end
end
mprintf("Time(h)          rainfall excess.");
for i=1:9
    mprintf("\n%f          %f",T(i),P(i));
end
mprintf("\n\nfi index=%f cm/hr.",fi);
mprintf("\n\ntime of rainfall excess=%i hours..",tr);
```

Scilab code Exa 4.20 EX4 20

```


//example 4.20
//calculate relation between R and P
```

```
clc;funcprot(0);
//given
P=[72.2 70.1 73.3 42.5 81.3 50.6 52.9 59.4 60.3 64.3 68.8 56.7 77.2 40.5 44.1 65.5]; //Precipitation
R=[24.1 22.7 25.6 11.3 28.4 12.7 13.4 15.7 16.2 17.7 19.2 14.9 25.4 10.6 11.7 17.9]; //runoff
for i=1:16
    Ps(i)=P(i)^2;
    Rs(i)=R(i)^2;
    PR(i)=P(i)*R(i);
end

s=0;t=0;u=0;q=0;w=0;
for i=1:16
    s=s+Ps(i);
    t=t+Rs(i);
    u=u+PR(i);
    q=q+P(i);
    w=w+R(i);
end
N=16;
a=(N*u-q*w)/(N*s-q^2);
b=(w-a*q)/N;
b=round(b*1000)/1000;
a=round(a*10000)/10000;
mprintf("Equation is:\n%fP%f.",a,b);
```

Scilab code Exa 4.21 EX4 21

```


//example4.21
//calculateeffective rainfall hyetograph and volume
```

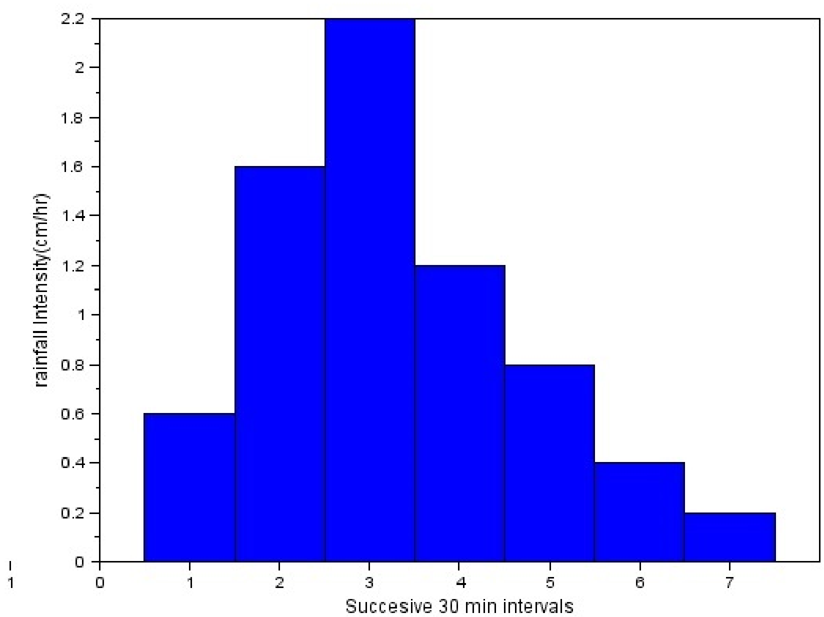

Figure 4.11: EX4 21

```
of diret run-off
clc;funcprot(0);
//given
A=8.6;                                          //catchment area
T=[0:0.5:4];                                    //time
r=[0 0.4 1.1 2.3 3.8 4.8 5.6 6.2 6.7];          //accumulated rainfall
fi=0.4;                                         //fi index
dt=0.5;                                         //time interval
d(1)=0;
for i=2:9
    d(i)=r(i)-r(i-1);                           //accumulated rainfall
end
mprintf("Intensity of effective Rainfall:");
I(1)=0;
s=0;
for i=2:9
    p(i)=d(i)-fi;                       //effective rainfall
    I(i)=p(i)/dt;                       //Intensity of effective Rainfall
    s=s+I(i);
    mprintf("\n%f",I(i));
end
//graph is plotted between I and T
run=s*dt;
V=run*A*10000;
mprintf("\nVolume of direct run-off=%f cubic metre.",V);
```

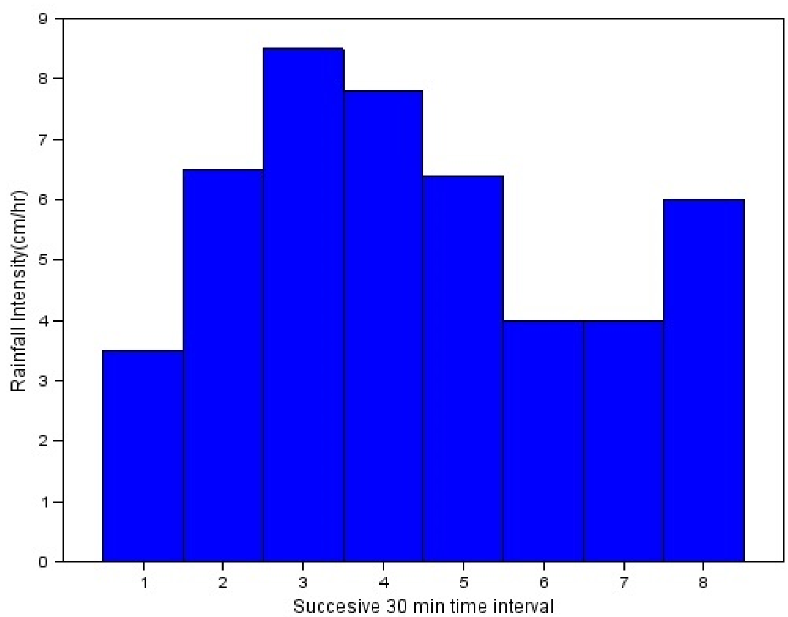

Figure 4.12: EX4 22

Scilab code Exa 4.22 EX4 22

```
//example 4.22
//calculate
//total rainfall
//total rainfall excess
//W index
clc;funcprot(0);
//given
r=[3.5 6.5 8.5 7.8 6.4 4 4 6];        //rainfall
   intensity
T=[0:30:240];                          //time
dt=30;                                //time interval
//graph is plotted between r and T
s=0;
for i=1:8
    s=s+r(i);
end
P=s*dt/60;
Pe=((6.5-4.5)+(8.5-4.5)+(7.8-4.5)+(6.4-4.5)+(6-4.5))
   *dt/60;    //area of graph above r=4.5.
w=(P-Pe)/4;
mprintf("total rainfall=%f cm.",P);
mprintf("\ntotal rainfall excess=%f cm.",Pe);
mprintf("\nW index=%f cm/hr.",w);
```

Scilab code Exa 4.23 EX4 23

```
//example 4.23
//calculate fi index
clc;funcprot(0);
```

```
//given
r=[0 8 22 74 92 105 114 120];        //raccumulated rainfall
T=[0 2 4 6 8 10 12 14];              //time for start of rainfall
V=2D6;                               //volume of run-off
A=40;                                //catchment area
tr=14;                               //duration of rainfall

d=V*1000/(40*1000000);

l=r(8)-d;
W=l/tr;
for i=2:8
    I(i)=r(i)-r(i-1);             //incremental rainfall
end

//rainfall excess is available in 4 time intervals of 2 hrs
tre=8;
fi=(l-I(2)-I(7)-I(8))/tre;
fi=round(fi*100)/100;
mprintf("fi index=%f mm/hr.",fi);
```

Scilab code Exa 4.24 EX4 24

```


//example 4.24
//calculate average infiltration index
clc;funcprot(0);
//given
```

```
r=[2.0 2.5 7.6 3.8 10.6 5.0 7.0 10.0 6.4 3.8 1.4
    1.4];      //rainfall depths
R=25.5;
    //total rum-off
s=0;
for i=1:12
    s=s+r(i);
end
tf=s-R;
af=tf/12;
//rainfall is less than average infiltration in1st,2
    nd,11th and 12th hours

f=(tf-r(1)-r(2)-r(11)-r(12))/8;
f=round(f*10)/10;
mprintf("average infiltration index=%f cm/hour.",f);
```

Scilab code Exa 4.25 EX4 25

```


//example 4.25
//calculate average depth of hourly rainfall excess
clc;funcprot(0);
//given
r=[2.0 2.5 7.6 3.8 10.6 5.0 7.0 10.0 6.4 3.8 1.4
    1.4];      //rainfall depths
A1=20;
A2=40;
A3=60;
A=A1+A2+A3;
fi1=7.6;
fi2=3.8;
fi3=1.0;
```

```
for i=1:12
    R1(i)=r(i)-fi1;                    //rainfall excess
    R2(i)=r(i)-fi2;
    R3(i)=r(i)-fi3;
    if (R1(i)<0)  then
    R1(i)=0;
end
if (R2(i)<0)  then
    R2(i)=0;
end
if (R3(i)<0)  then
    R3(i)=0;
end
end
mprintf("average depth of hourly rainfall excess(cm/
   hr)");
for i=1:12
    a1(i)=R1(i)*A1/A;                          //average
       rainfall excess
    a2(i)=R2(i)*A2/A;
a3(i)=R3(i)*A3/A;
T(i)=a1(i)+a2(i)+a3(i);                    //total hourly
    rainfall excess
T(i)=round(T(i)*100)/100;
mprintf("\n%f",T(i));
end
```

Scilab code Exa 4.26 EX4 26

```


//example4.26
//derive the unit hydrograph
clc;funcprot(0);
//given
```

```
A=92;                         //area of drainage basin
t=[6 8 10 12 14 16 18 20 22 24 2 4 6 8 10 12 14 16];    //time
r=[10.6 9.7 107.8 175.6 193.9 150.3 126.2 106.9 90 72.8 58.2 48 36.2 28.4 20.2 14 10.2 10.4];   //total run-off
B=[10.6 9.7 9.73 9.77 9.8 9.83 9.87 9.9 9.93 9.97 10 10.03 10.07 10.10 10.13 10.16 10.20 10.40];   //base flow
s=0;
for i=1:18
    d(i)=r(i)-B(i);                   //direct run-off ordinate
s=s+d(i);
end
n=0.36*s*2/A;
mprintf("ordinates of unit hydrograph:");
for i=1:18
    u(i)=d(i)/n;                      //ordinates of unit hydrograph
    u(i)=round(u(i)*100)/100;
     mprintf("\n%f",u(i));
end
mprintf("\nHydograph is 4-hr unit hydrograph");
//graph is plotted between:
//r and t
//u and t
```

Scilab code Exa 4.27 EX4 27

```

```

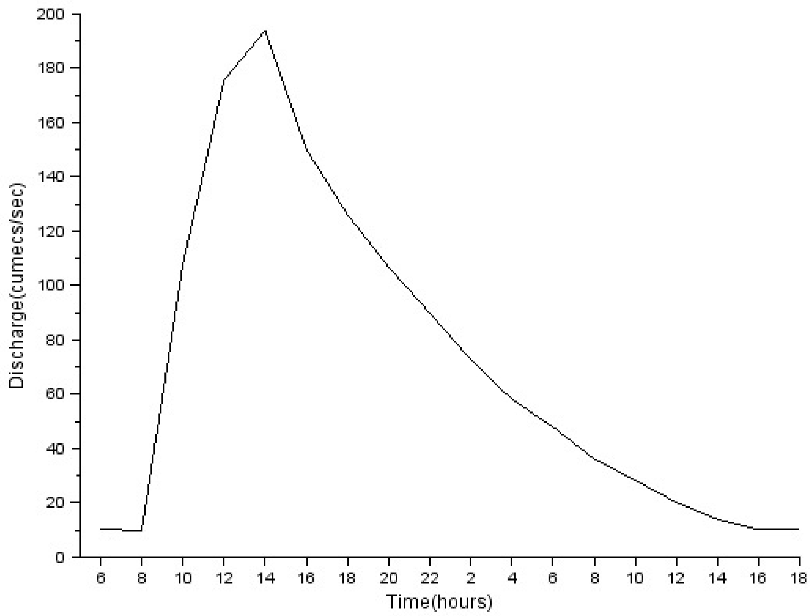

Figure 4.13: EX4 26

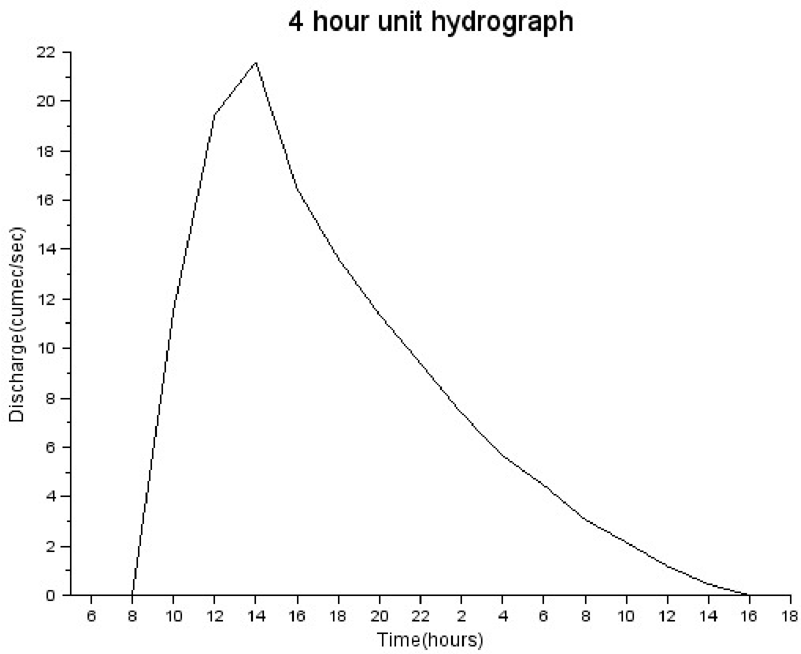

Figure 4.14: EX4 26

```
//example 4.27
//calculate rainfall excess
clc;funcprot(0);
//given
A=316;          //drainage area
B=17;           //base flow
t=6;
O=[17.0 113.2 254.5 198.0 150.0 113.2 87.7 67.9 53.8
    42.5 31.1 22.6 17.0];     //ordinates of storm
   hydrograph
for i=1:13
    Or(i)=O(i)-B;                 //ordinates of
       direct run-off
    Oh(i)=Or(i)/6.477;            //ordinates of
       unit hydrograph
end
s=0;
for i=1:13
    s=s+Or(i);
end
re=s*60*60*t/(A*10000);
re=round(re*1000)/1000;
mprintf("rainfall excess=%f cm.",re);
```

Scilab code Exa 4.28 EX4 28

```


//example 4.28
//calculate ordinates of storm hydrograph
clc;funcprot(0);
//given
fi=2.5;          //infiltration index
B=10;            //Base flow
```

```
O=[0 110 365 500 390 310 250 235 175 130 95 65 40 22
    10 0 0 0];    //ordinates of unit hydrograph
R1=2;R2=6.75;R3=3.75;
r1=(R1*10-(fi*3)-5)/10;                    //rainfall
   excess in first three hour
r2=(R2*10-(fi*3))/10;                      //rainfall
   excess in second three hour
r3=(R3*10-(fi*3))/10;                      //rainfall
   excess in third three hour

for i=1:18
    s1(i)=r1*O(i);
end
for i=2:18
    s2(i)=r2*O(i-1);
end
for i=3:18
    s3(i)=r3*O(i-2);
end                                        //surface run-
   off from rainfall excess during succesive unit
   periods
mprintf("ordinates of storm hydrograph");
for i=1:18
    T(i)=s1(i)+s2(i)+s3(i);
    t(i)=T(i)+B;
    t(i)=round(t(i)*10)/10;
    mprintf("\n%f",t(i));
end
```

Scilab code Exa 4.29 EX4 29

```


//example4.29
//derive and plot 6 hr unit hydrograph
```

```
clc;funcprot(0);
//given
A=103.4;        //area of basin
t=[0:3:36];     //time
q=[0 21 80 82 189 123 184 87 55.5 25.25 9 6 0];   //flow
mprintf("ordinates of unit hydrograph are:");
u(1)=0;
u(2)=q(2)/2;
u(3)=(q(3)-4*u(1))/2;
u(4)=(q(4)-4*u(2))/2;
for i=5:9
    u(i)=(q(i)-3*u(i-4)-4*u(i-2))/2;        //ordinates of unit hydrograph
end
for i=1:9
    mprintf("\n%f",u(i));
end
mprintf("\n\nThe succesive unit hydrograph will have same ordinates but will be shifted\nlaterally by 6 hrs.");
//graph is plotted between u and t.
```

Scilab code Exa 4.30 EX4 30

```


//example 4.30
//derive ordinates of 6 hrs unit hydrograph
clc;funcprot(0);
//given
```

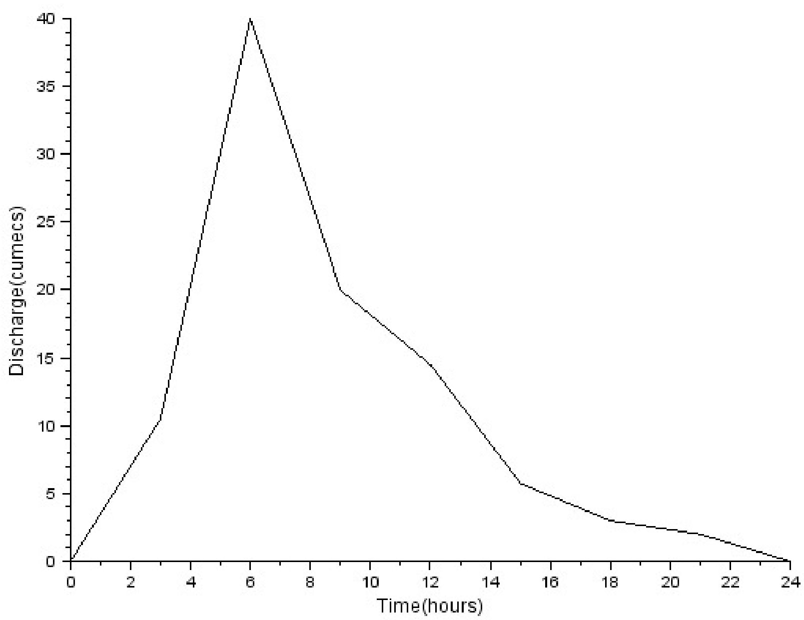

Figure 4.15: EX4 29

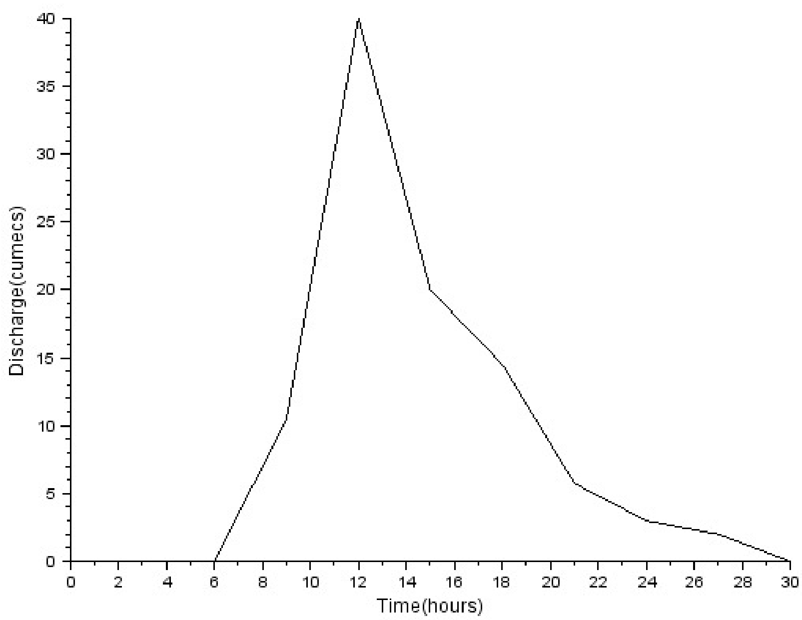

Figure 4.16: EX4 29

```
R=[0 1 2.7 5 8 9.8 9 7.5 6.3 5 4 2.9 2.1 1.3 0.5 0 0 0 0 0]; //2hrs unit hydrograph
mprintf("ordinates of 6 hrs unit hydrograph");
for i=1:18
    O1(i+2)=R(i);


end
for i=1:16
    O2(i+4)=R(i);
end                                    //offset unit hydrograph
for i=1:20
    S(i)=O1(i)+O2(i)+R(i);             //sum
    f(i)=S(i)/3;                       //ordinates of 6 hrs unit hydrograph
    f(i)=round(f(i)*10)/10;
    mprintf("\n%f",f(i));
end
```

Scilab code Exa 4.31 EX4 31

```


//example 4.31
//calculate ordinate of 9 hr unit hydrograph
clc;funcprot(0);
//given
t=[0:3:45];     //time
O=[0 9 20 35 49 43 35 28 22 17 12 9 6 3 0 0];       //ordinate of 2 hr unit hydrograph
of(1)=0;
of(2)=0;
for i=3:16
    of(i)=O(i-2)+of(i-2);                    //offset
```

```
ordinate
end
for i=1:16
   s(i)=O(i)+of(i);                              //ordinate of s-curve
end
of1(3)=0;
for i=4:16
    of1(i)=s(i-3);                               //offset of s-curve
end
mprintf("ordinates of 9 hrs unit hydrograph:");
for i=1:16
    y(i)=s(i)-of1(i);
    u(i)=2*y(i)/3;                                   // ordinate of 9 hrs unit hydrograph
    u(i)=round(u(i)*10)/10;
    mprintf("\n%f",u(i));
end
```

Scilab code Exa 4.32 EX4 32

```


//example 4.32
//calculate recurrence interval of flood using
//california method
//Hazens method
//gumbels method
clc;funcprot(0);
//given
q=[9200 7800 6600 5800 5260 4980 4525 3810 3630 3250 3110 3090 2380 2390 1723]; //Discharge arranged in decreasing order
N=15;
```

```
C=0.3;
m=[1:1:15];
C=[0.3 0.44 0.52 0.57 0.61 0.66 0.7 0.74 0.78 0.82
   0.86 0.88 0.94 0.96 1];    //from table 4.25
mprintf("California           Hazen            Gumbel")
   ;
for i=1:15
    Ca(i)=N/m(i);
    H(i)=2*N/(2*m(i)-1);
    G(i)=N/(m(i)+C(i)-1);
    Ca(i)=round(Ca(i)*100)/100;
    G(i)=round(G(i)*100)/100;
    H(i)=round(H(i)*100)/100;
    mprintf("\n%f            %f            %f",Ca(i),H(i
       ),G(i));
end
```

Scilab code Exa 4.33 EX4 33

```


//example 4.33
//calculate flood magnitude with return period of
   240 years
clc;funcprot(0);
//given
T1=40;T2=80;              //Return period
F1=27000;F2=31000;        //Peak flood
y80=-(2.303*log10(2.303*log10(T2/(T2-1))));
y40=-(2.303*log10(2.303*log10(T1/(T1-1))));
y=(F2-F1)/(y80-y40);
T=240;
y240=-(2.303*log10(2.303*log10(T/(T-1))));
x240=F2+(y240-y80)*y;
mprintf("flood magnitude with return period of 240
```

```
years=%i cumec.",x240);
```

Scilab code Exa 4.34 EX4 34

```
//example 4.34
//calculate flood discharge with recurrence period of 100 years and 200 years
clc;funcprot(0);
//given
N=40;
Sn=1.1413;yn=0.5436;          //from table 4.21 (a) and(b)
q=[1330 1095 1030 980 975 950 945 940 925 855 853 840 835 825 810 795 756 710 708 705 700 670 625 620 610 605 595 585 570 550 530 505 500 495 485 465 460 420 390 380];  //discharge
s=0;
for i=1:40
    s=s+q(i);
    end
xavg=s/N;
w=0;
for i=1:40
    t(i)=(q(i)-xavg)^2;
    w=w+t(i);
end
sigma=(w/(N-1))^0.5;
N=10;
y10=-(2.303*log10(2.303*log10(N/(N-1))));
K10=(y10-yn)/Sn;
x10=xavg+K10*sigma;
N=20;
y20=-(2.303*log10(2.303*log10(N/(N-1))));
```

```
K20=(y20-yn)/Sn;
x20=xavg+K20*sigma;
N=5;
y5=-(2.303*log10(2.303*log10(N/(N-1))));
K5=(y5-yn)/Sn;
x5=xavg+K5*sigma;

T=100;
y100=-(2.303*log10(2.303*log10(T/(T-1))));
K100=(y100-yn)/Sn;
x100=xavg+K100*sigma;

T=200;
y200=-(2.303*log10(2.303*log10(T/(T-1))));
K200=(y200-yn)/Sn;
x200=xavg+K200*sigma;
x100=round(x100);
mprintf("For T=100 years:\nflood discharge=%f cumecs
    .\n\nFor T=200 years:\nflood discharge=%i cumecs.
    ",x100,x200);
```

Scilab code Exa 4.35 EX4 35

```


//example 4.35
//calculate 200 year flood for stream using gumbel's
    method
clc;funcprot(0);
//given
sigma=1.1413;          //standard deviation
yn=0.5436;
T=50;
y50=-2.303*log10(2.303*log10(T/(T-1)));
K50=(y50-yn)/sigma;
```

```
T=100;
y100=-2.303*log10(2.303*log10(T/(T-1)));
K100=(y100-yn)/sigma;
x50=878;  x100=970;      //given peak flood
A=[K50 1;K100 1];
B=[x50;x100];
C=A\B;
xavg=C(2);
sigmad=C(1);
T=200;
y200=-2.303*log10(2.303*log10(T/(T-1)));
K200=(y200-yn)/sigma;
x200=xavg+K200*sigmad;
x200=round(x200);
mprintf("200 year flood for stream=%f cumecs.",x200)
    ;
```

Scilab code Exa 4.36 EX4 36

```


//example 4.36
//calculate
//risk of failure of cofferdam
//return period
clc;funcprot(0);
//given
T=30;               //deign for period
n=6;                //period of construction
R=(1-(1-(1/T))^n)*100;
R1=0.1;             //reduced risk
T1=1/(1-(1-R1)^(1/6));
R=round(R*10)/10;
T1=round(T1*100)/100;
mprintf("risk of failure of cofferdam=%f percent.",R
```

```
);
mprintf("\nreturn period=%f years.",T1);
```

Scilab code Exa 4.37 EX4 37

```


//example 4.37
//calculate
//probability of excedence
//probability of flood magnitude occuring at:
//at least once in 10 years
//two times in 10 succesive years
//once in 10 succesive years
clc;funcprot(0);
//given
T=40;              //return period
P=1/T;
n=10;
Rsk=1-(1-P)^n;
s=1;t=1;
for i=1:n
    s=s*i;
end
for i=1:(n-2)
    t=t*i;
end
P2n=s*P^2*(1-P)^8/(t*2);
P1n=n*P*(1-P)^(n-1);
Rsk=round(Rsk*1000)/1000;
P2n=round(P2n*10000)/10000;
P1n=round(P1n*1000)/1000;
mprintf("probability of excedence=%f.",P);
mprintf("\nprobability of flood magnitude occuring at least once in 10 years=%f",Rsk);
```

```
mprintf("\nprobability of flood magnitude occuring
   at two times in 10 succesive years=%f",P2n);
mprintf("\nprobability of flood magnitude occuring
   at once in 10 succesive years=%f",P1n);
```

Scilab code Exa 4.38 EX4 38

```


//example 4.38
//calculate peak rate of run off
clc;funcprot(0);
//given
C1=0.22;C2=0.12;C3=0.32;      //run-off coefficient
A1=3.2;A2=4.8;A3=1.8;         //calculated area
L=2.4;                        //length of water course
H=30;                         //fall
T=30;                         //frequency
t=60*0.000323*(L*1000)^0.77*(H/(L*1000))^(-0.385);
i=78*T^0.22/(t+12)^0.45;
q=2.778*i*(C1*A1+C2*A2+C3*A3);
q=round(q*10)/10;
mprintf("peak rate of run off=%f cumecs.",q);
```

Scilab code Exa 4.39 EX4 39

```


//example 4.39
//calculate peak flow rate
clc;funcprot(0);
//given
T=30;               //return period
```

```
A=2.4;              //area of watershed
s=1/200;            //slope oof catchment
L=1.8;              //length of travel of water
C=0.25;             //average run-off coefficient
r=[2.5 3.8 4.8 5.9 6.7 7.4 8.4 8.7 9.2];    //
   rsinfall depth
t=60*0.000323*(L*1000)^0.77*(s)^(-0.385);
rmax=r(7)+(r(8)-r(7))*7.84/10;
i=rmax*60/t;
q=2.778*C*A*i;
q=round(q*100)/100;
mprintf("peak flow rate=%f cumecs.",q);
```

Scilab code Exa 4.40 EX4 40

```


//example 4.40
//calculate precipitation at x
clc;funcprot(0);
//given
pA=75;            //precipitation at A
pB=58;            //precpitation at B
pC=47;            //precipitation at C
nA=826;           //normal precipitation at A
nB=618;           //normal precipitation at B
nC=482;           //normal precipitation at C
nX=757;           //normal precipitation at X

pX=(nX*pA/nA+nX*pB/nB+nX*pC/nC)/3;
pX=round(pX*10)/10;
mprintf("precipitation at x=%f cm.",pX);
```

Scilab code Exa 4.41 EX4 41

```


//example 4.1
//calculate mean rainfall;additional guages needed
clc;funcprot(0);
//given
p=[41 51 32 55 50 68];    //rain guage readings at
   respective stations
s=0;
for i=1:6
    s=s+p(i);
end
pavg=s/6;
u=0;
for i=1:6
    u=u+(p(i)-pavg)^2;
end
sx=(u/5)^0.5;
Cv=sx*100/pavg;
N=(Cv/8)^2;
N=round(N*100)/100;
mprintf("mean rainfall=%f cm.",pavg);
mprintf("\ntotal stations needed=%f.",N);
```

Scilab code Exa 4.42 EX4 42

```


//example 4.42
//calculate mean precipitaion using thiesson polygon
    method
clc;funcprot(0);
//given
```

```
a=4;                    //dimension of plot sides
P1=4.8;P2=13;P3=8;P4=5.4;P5=3.2;P6=9.4;      //
   precipitaion at respective stations
A1=a^2/8+a^2/(4*1.73);
A2=a^2/8;
A3=A2;A4=A1;
A5=a^2/(4*1.73);
A6=a^2/2;
A=A1+A2+A3+A4+A5+A6;
Pavg=(P1*A1+P2*A2+P3*A3+P4*A4+P5*A5+P6*A6)/A;
mprintf("Mean precipitaion=%f cm.",Pavg);
```

Scilab code Exa 4.43 EX4 43

```


//example 4.43
//calculate average depth of precipitation
clc;funcprot(0);
//given
A=[90 140 125 140 85 40 20];   //area of isohytes
I=[13:-2:1];                    //average isohytel
   interval
s=0;t=0;
for i=1:7
    s=s+A(i)*I(i);
    t=t+A(i);
end
Pavg=s/t;
Pavg=round(Pavg*10)/10;
mprintf(" average depth of precipitation=%f cm.",
   Pavg);
```

Scilab code Exa 4.44 EX4 44

```
//example 4.44
//calculate mean rainfall;additional guages needed
clc;funcprot(0);
//given
p=[120 95 96 60 65 70 45 21];    //rain guage
    readings at respective stations
s=0;
for i=1:8
    s=s+p(i);
end
pavg=s/8;
u=0;
for i=1:8
    u=u+(p(i)-pavg)^2;
end
sx=(u/7)^0.5;
Cv=sx*100/pavg;
N=(Cv/13.99)^2;
N=round(N*100)/100;
mprintf("mean rainfall=%f cm.",pavg);
mprintf("\ntotal stations needed=%f.",N);
//taking N=10
N=10;
n=N-8;
mprintf("\nadditional guages needed=%i.",n);
```

Scilab code Exa 4.45 EX4 45

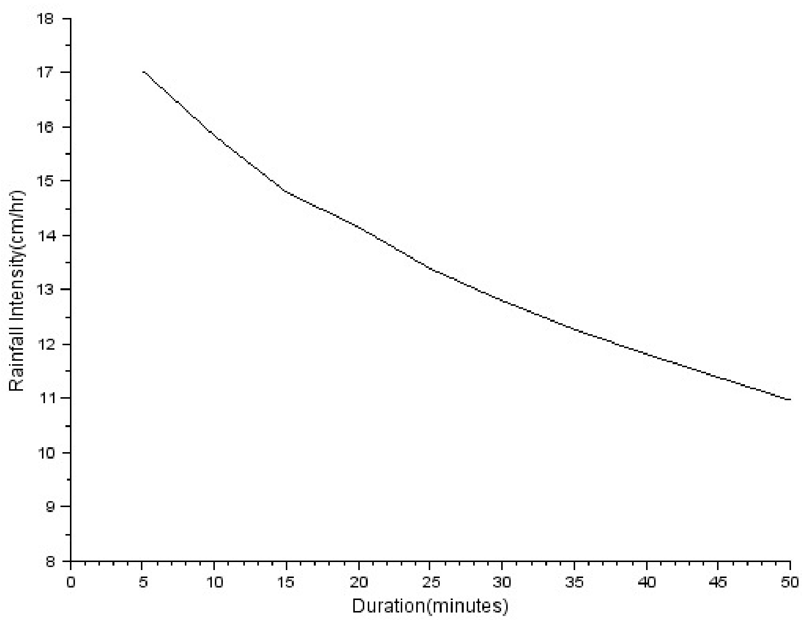

Figure 4.17: EX4 45

```
//example 4.45
//compute maximum rainfall intensities for
    5,10,15,20,25,30,35,40,45,50 minutes
//plot intensity duration graph
clc;funcprot(0);
//given
CR=[0 1.02 2.08 3.30 4.72 5.58 6.40 7.16 7.88 8.54
    9.14];  //cumulative rainfall

c5(2)=CR(2);
c10(3)=CR(3);
c15(4)=CR(4);
c20(5)=CR(5);
c25(6)=CR(6);
c30(7)=CR(7);
c35(8)=CR(8);
c40(9)=CR(9);
c45(10)=CR(10);
c50(11)=CR(11);
for i=3:11
    c5(i)=CR(i)-CR(i-1);
end
for i=4:11
    c10(i)=CR(i)-CR(i-2);
end
for i=5:11
    c15(i)=CR(i)-CR(i-3);
end
for i=6:11
    c20(i)=CR(i)-CR(i-4);
end
for i=7:11
    c25(i)=CR(i)-CR(i-5);
end
for i=8:11
    c30(i)=CR(i)-CR(i-6);
end
```

```
for i=9:11
    c35(i)=CR(i)-CR(i-7);
end
for i=10:11
    c40(i)=CR(i)-CR(i-8);
end
for i=11:11
    c45(i)=CR(i)-CR(i-9);
end                                    //rainfall in any possible time interval

mprintf("5min                 10min                 15min                 20min                 25min                 30min                 35min                 40min                 45min                 50min");
for i=1:11
    mprintf("\n%f            %f            %f            %f            %f            %f            %f            %f            %f            %f",c5(i),c10(i),c15(i),c20(i),c25(i),c30(i),c35(i),c40(i),c45(i),c50(i));
end
I=[17.04 15.84 14.80 14.16 13.39 12.80 12.27 11.82 11.39 10.97];     //maximum intensity at respective durations
D=[5:5:50];                    //durations
//graph is plotted between I and D
```

Scilab code Exa 4.46 EX4 46

```


//example 4.46
//draw storm hyetograph and intensity duration curve
clc;funcprot(0);
```

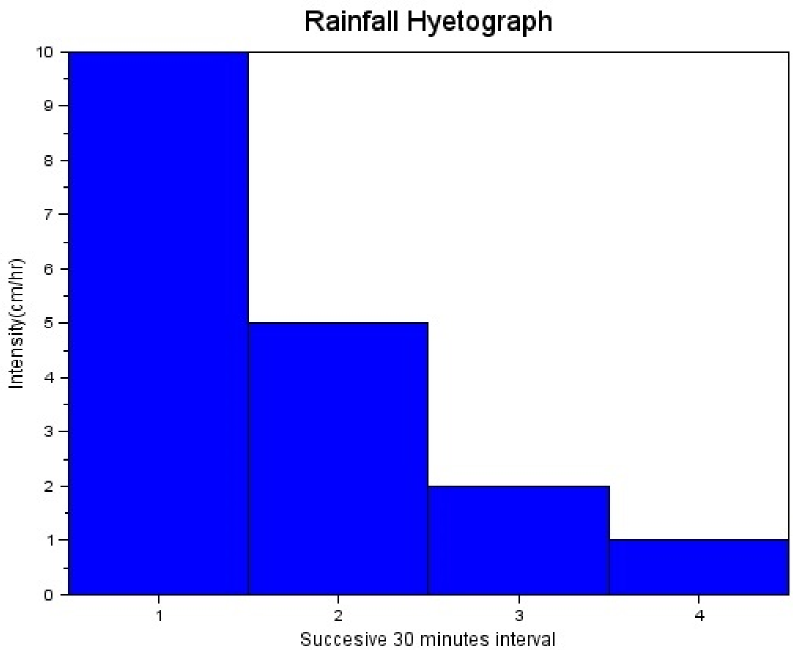

Figure 4.18: EX4 46

```
//given
p=[0 5 7.5 8.5 9];        //accumulated precipitation
t=[0 30 60 90 120];       //time
r(1)=0;
mprintf("Rainfall intensity:");
for i=2:5
    r(i)=p(i)-p(i-1);             //rainfall in succesive 30 min interval
    I(i)=r(i)*60/30;              //rainfall intensity
    mprintf("\n%f",I(i));
end
//graph is plotted between I and t.
```

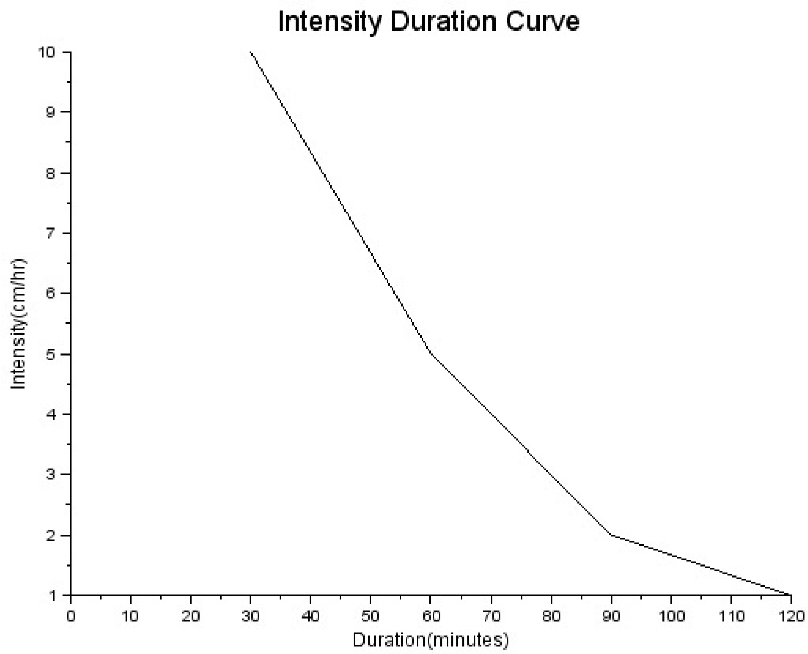

Figure 4.19: EX4 46

Scilab code Exa 4.47 EX4 47

```
//example 4.47
//calculate average depth of precipitation using
   depth area curve
clc;funcprot(0);
```

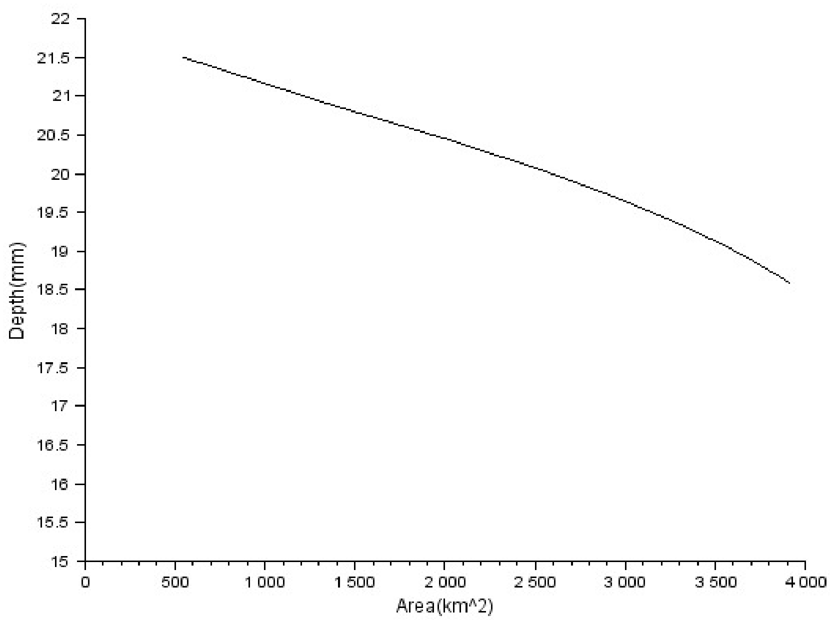

Figure 4.20: EX4 47

```
//given
I=[21:-1:12];            //isohytes
a=[543 1345 2030 2545 2955 3280 3535 3710 3880 3915];  //enclosed area
ia(1)=543;
for i=2:10
    ia(i)=a(i)-a(i-1);               //net incremental area between isohytes
end
r=[21.5:-1:12.5]
for i=1:10
    rv(i)=r(i)*ia(i);                //rainfall volume
end
cv(1)=11675;
for i=2:10
    cv(i)=cv(i-1)+rv(i);             //cumulative volume
end
for i=1:10
    eud(i)=cv(i)/a(i);                //depth(mm)
end

mprintf("From depth area curve we obtain average depth of precipitation=20.15 mm for an\narea of 2400 sq. km.");
//graph is plotted between eud and a
```

Scilab code Exa 4.48 EX4 48

```


//example 4.48
//calculate evaporation from reservior surface during the week
```

```
clc;funcprot(0);
//given
h1=7.75;          //initial depth of water
r=3.80;           //rainfall during the week
hr=2.50;          //depth of water removed
C=0.7;            //pan coefficient
ha=r-hr;
hl=ha+h1;
h2=8.32;
ev=hl-h2;
evs=ev*C;
evs=round(evs*100)/100;
mprintf("evaporation from reservior surface during
    the week=%f cm.",evs);
```

Scilab code Exa 4.49 EX4 49

```


//example4.49
//calculate fi index and time of rainfall excess
clc;funcprot(0);
//given
T=[1:1:12];        //time from start
r=[1.8 2.6 7.8 3.9 10.6 5.4 7.8 9.2 6.5 4.4 1.8
    1.6];  //increamental rainfall
R=24.4;           //total run-off
s=0;
for i=1:12
    s=s+r(i);
end
ti=s-R;

//first trial
tr=7;    //assumed
```

```
ti=s-R-r(1)-r(2)-r(4)-r(11)-r(12);
fi=ti/tr;
for i=1:12
    P(i)=r(i)-fi;
    if (P(i)<0) then
        P(i)=0;
    end
end
mprintf("Time(h)           rainfall excess.");
for i=1:12
    mprintf("\n%f           %f",T(i),P(i));
end
mprintf("\n\nfi index=%f cm/hr.",fi);
```

Scilab code Exa 4.50 EX4 50

```


//example 4.50
//calculate fi index
clc;funcprot(0);
//given
r=[0.6 1.35 2.25 3.45 2.7 2.4 1.5 0.75];       //
    incremental rainfall
T=[1:1:8];              //time from start of rainfal
t=8;
P=15;                   //total rainfall
R=8.7;                  //direct run-off
W=(P-R)/t;
//since fi wil be more than W
tre=6;
fi=((P-R)-r(1)-r(8))/tre;
mprintf("fi index=%f cm/hr.",fi);
```

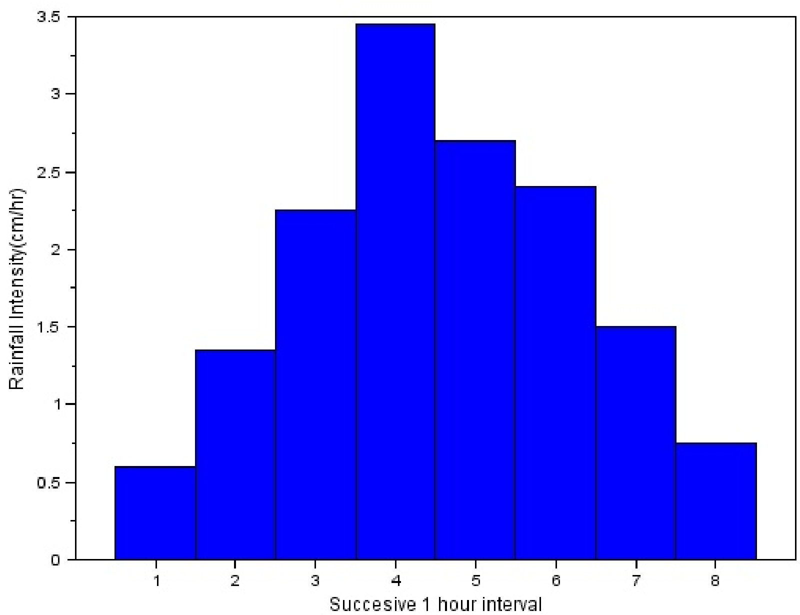

Figure 4.21: EX4 50

Scilab code Exa 4.51 EX4 51

```
//example 4.51
//calculate total infiltration depth lasting 6 hrs
clc;funcprot(0);
//given
I=10;             //total infiltration rate
fI=5;             //final infiltration rate
k=0.95;           //rate of decay of difference
   between final and initial infiltration rate

q=integrate('fI+(I-fI)*%e^(-k*t)','t',0,6);
q=round(q*100)/100;
mprintf("total infiltration depth=%f mm.",q);
```

Scilab code Exa 4.52 EX4 52

```
//example 4.52
//find the equation of infiltration capacity
clc;funcprot(0);
//given
fc=1;               //constant infiltration rate
ft=[10.4 5.6 3.2 2.1 1.5 1.2 1.1 1 1];    //
   infiltration capacity
f=ft(1)-fc;
t=[0:0.25:2];

for i=1:9
    r(i)=ft(i)-fc;
```

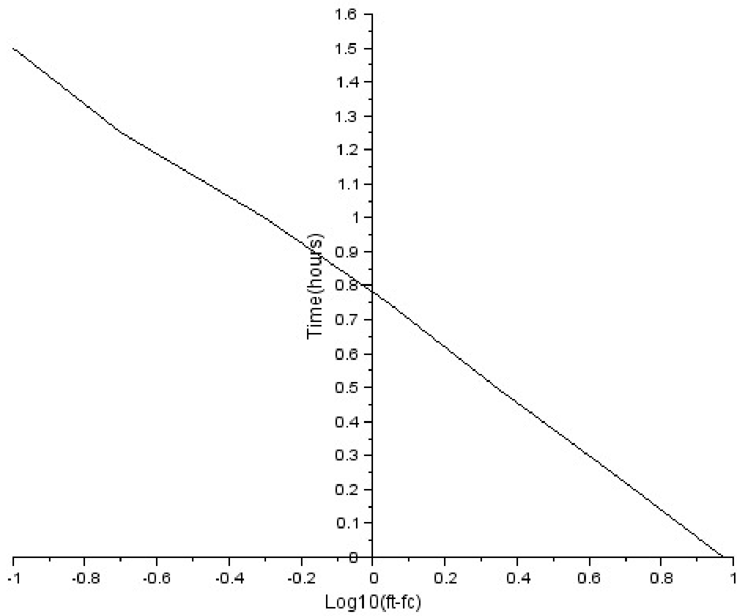

Figure 4.22: EX4 52

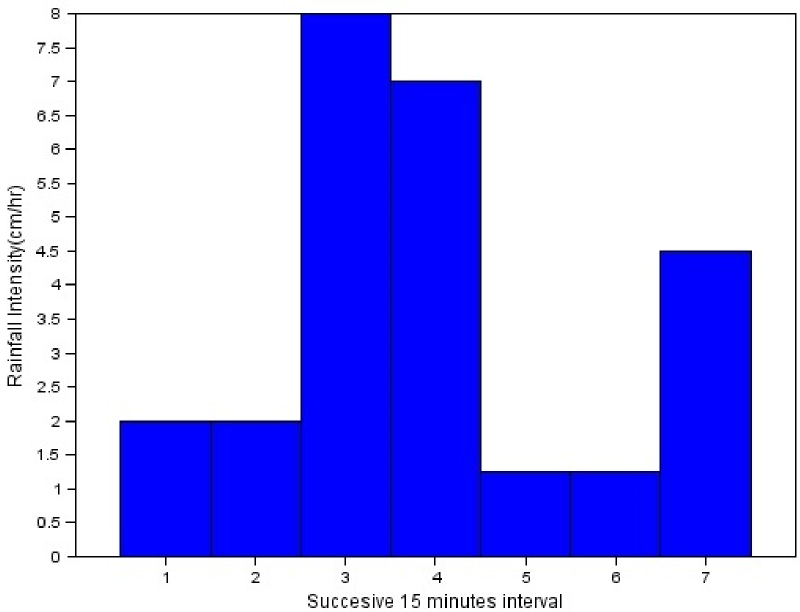

Figure 4.23: EX4 53

```
end
for i=1:7
    h(i)=log10(r(i));
end
s=0.775;            //from graph
k=1/(log10(%e)*s);
k=round(k*100)/100;
mprintf("Equation is:\nft=fc+%fe^(-%ft)",f,k);
```

Scilab code Exa 4.53 EX4 53

```
//example 4.53
//calculate
//total rainfall
//net run-off
//W index
clc;funcprot(0);
//given
r=[2 2 8 7 1.25 1.25 4.5];          //rainfall
    intensity
T=[15 30 45 60 70 90 105];                        //
    time
dt=15;                                  //time interval
fi=3;                                   //fi index
//graph is plotted between r and T
s=0;
for i=1:7
    s=s+r(i);
end
P=s*dt/60;
Pe=((8-3)+(7-3)+(4.5-3))*dt/60;     //area of graph
    above r=3.0.
w=(P-Pe)/(105/60);
w=round(w*1000)/1000;
mprintf("total rainfall=%f cm.",P);
mprintf("\nnet run-off=%f cm.",Pe);
mprintf("\nW index=%f cm/hr.",w);
```

Scilab code Exa 4.54 EX4 54

```
//example 4.54
//calculate Total rainfall in catchment
```

```
//run-off by rainfall of 3.3cm in 3hrs
clc;funcprot(0);
//given
A=[36 18 66];                //area of catchment
fi=[0.9 1.1 0.5];            //fi index
r1=[0.6 0.9 1.0];            //rainfall in first hour
r2=[2.4 2.1 2.0];           //rainfall in second hour
r3=[1.3 1.5 0.9];           //rainfall in third hour

t36=r1(1)+r2(1)+r3(1);
t18=r1(2)+r2(2)+r3(2);
t66=r1(3)+r2(3)+r3(3);

p=(t36*A(1)+t18*A(2)+t66*A(3))/(A(1)+A(2)+A(3));
mprintf("Total rainfall in catchment=%f cm.",p);

ro1=[0 0 0.5];ro2=[1.5 1.0 1.5];ro3=[0.4 0.4 0.4];
      //rainfall-fi
t1=ro1(1)+ro2(1)+ro3(1);
t2=ro1(2)+ro2(2)+ro3(2);
t3=ro1(3)+ro2(3)+ro3(3);
run=(A(1)*t1+A(2)*t2+A(3)*t3)/(A(1)+A(2)+A(3));
         //run-off from entire catchment
mprintf("\nrun-off by rainfall of 3.3cm in 3hrs=%f
   cm.",run);
fia=(fi(1)*A(1)+fi(2)*A(2)+fi(3)*A(3))/(A(1)+A(2)+A
   (3));
tr=(1.1-fia)*3;
mprintf("\nTotal run-off=%f cm.",tr);
```

Scilab code Exa 4.55 EX4 55

```


//example 4.55
```

```
//calculate relation between R and P
clc;funcprot(0);
//given
P=[4 22 28 15 12 8 4 15 10 5];  //Precipitation
R=[0.2 7.1 10.9 4.0 3.0 1.3 0.4 4.1 2.0 0.3];  //runoff
for i=1:10
    Ps(i)=P(i)^2;
    Rs(i)=R(i)^2;
    PR(i)=P(i)*R(i);
end

s=0;t=0;u=0;q=0;w=0;
for i=1:10
    s=s+Ps(i);
    t=t+Rs(i);
    u=u+PR(i);
    q=q+P(i);
    w=w+R(i);
end
N=10;
a=(N*u-q*w)/(N*s-q^2);
b=(w-a*q)/N;
a=round(a*10000)/10000;
b=round(b*10000)/10000;
mprintf("Equation is:\n%fP%f.",a,b);
```

Scilab code Exa 4.56 EX4 56

```


//example 4.56
//calculate peak discharge of 6 hrs unit hydrograph
clc;funcprot(0);
//given
```

```
Q=470;                     //peak discharge of flood
   hydrograph
B=15;                      //base flow
l=0.25;                    //infiltration loss
Qr=Q-B;
d=8;                       //average depth of rainfall
re=d-l*6;                 //rainfall excess
q=Qr/re;
mprintf("peak discharge of 6 hrs unit hydrograph=%i
   cumecs.",q);
```

Scilab code Exa 4.57 EX4 57

```


//example 4.57
//calculate ordinates of storm hydrograph
clc;funcprot(0);
//given
fi=0.25;            //infiltration index
B=20;             //Base flow
O=[0 20 60 150 120 90 70 50 30 20 10 0 0 0];    //
   ordinates of unit hydrograph
R1=5;R2=0.8;R3=3;
r1=R1-(fi*4);                                   //
   rainfall excess in first four hour
r2=R2-(fi*4);                                   //
   rainfall excess in second four hour
r3=R3-(fi*4);                                   //
   rainfall excess in third four hour
if r2<0
    r2=0;
    end

for i=1:14
```

```
    s1(i)=r1*O(i);
end
for i=2:14
    s2(i)=r2*O(i-1);
end
for i=3:14
    s3(i)=r3*O(i-2);
end                                         //surface run-off from rainfall excess during succesive unit periods
mprintf("ordinates of storm hydrograph");
for i=1:14
    T(i)=s1(i)+s2(i)+s3(i);                 //sub-total
    t(i)=T(i)+B;                            //ordinate of flood hydrograph
    mprintf("\n%i",t(i));
end
```

Scilab code Exa 4.58 EX4 58

```


//example 4.58
//calculate ordinates of discharge hydrograph and peak discharge
clc;funcprot(0);
//given
fi=2.5;             //fi index
t=24;
A=200;              //area of catchment
R1=7.5;R2=2.0;R3=5; //rainfall
r1=R1-fi;r2=R2-fi;r3=R3-fi;
r2=0;
r=[5 0 2.5];        //excess rainfall
D=[5 15 40 25 10 5 0 0 0];  //distribution
```

```
for i=1:9
    d1(i)=D(i)*r(1)/100;
end
for i=1:8
     d2(i+1)=D(i)*r(2)/100;
end
for i=1:7
     d3(i+2)=D(i)*r(3)/100;
end                                       //distribution run-off for rainfall excess

for i=1:9
    tr1(i)=d1(i)+d2(i)+d3(i);          //total run-off as depth
    tr2(i)=23.148*tr1(i);              //total run-off as discharge
    tr2(i)=round(tr2(i)*1000)/1000;
end
s=0;
for i=1:9
    s=s+tr2(i);
end
mprintf("Total run-off:");
mprintf("\nas depth           as discharge");
for i=1:9
    mprintf("\n%f           %f",tr1(i),tr2(i));
end
r=0.36*s*t/A;
r=round(r*10)/10;
mprintf("\ntotal run-off=%f cm.",r);
```

Scilab code Exa 4.59 EX4 59

```
//example 4.59
//calculate ordinate of 6 hr unit hydrograph
clc;funcprot(0);
//given
O=[10 30 90 220 280 220 166 126 92 62 40 20 10];      //ordinates of 6 hr flood hydrograph
B=10;          //Base flow
for i=1:13
    r(i)=O(i)-B;           //ordinates of direct run-off
end
mprintf("Ordinates of 6 hr unit hydrograph");
u(1)=0;
for i=2:13
    u(i)=r(i)-u(i-1);          //ordinates of 6 hrs unit hydrograph
end
for i=1:13
    mprintf("\n%i",u(i));
end
```

Scilab code Exa 4.60 EX4 60

```


//example 4.60
//determine the ordinates of 1 cm-6 hour hydrograph
clc;funcprot(0);
//given
t=6;
A=450;        //catchment area
O=[5 15 40 80 60 50 25 15 5];    //ordinates of flood hydrograph
B=5;         //base flow assumed
s=0;
```

```
for i=1:9
    r(i)=O(i)-B;          //ordinates of direct run-off
s=s+r(i);
end
n=s*0.36*12/A;
mprintf("ordinates of unit hydrograph");
for i=1:9
    u(i)=r(i)/n;
    u(i)=round(u(i)*100)/100;
    mprintf("\n%f",u(i));
end
```

Scilab code Exa 4.61 EX4 61

```


//example 4.61
//obtain ordinates 24 hr unit hydrograph
clc;funcprot(0);
//given
O=[0 5.5 13.5 26.5 45 82 162 240 231 165 112 79 57 42 31 22 14 9.5 6.6 4 2 1 0 0 0 0 0];  //ordinates of 1st 8 hrs unit hydrograph
for i=1:25
    o1(i+2)=O(i);                       //ordinates of 2nd 8 hrs unit hydrograph
    o2(i+4)=O(i);                       //ordinates of 3rd 8 hrs unit hydrograph
end
mprintf("ordinates 24 hr unit hydrograph:");
for i=1:27
    o3(i)=o1(i)+o2(i)+O(i);         //total 24 hr hydrograph of 3 cm run-off
    t(i)=o3(i)/3;
```

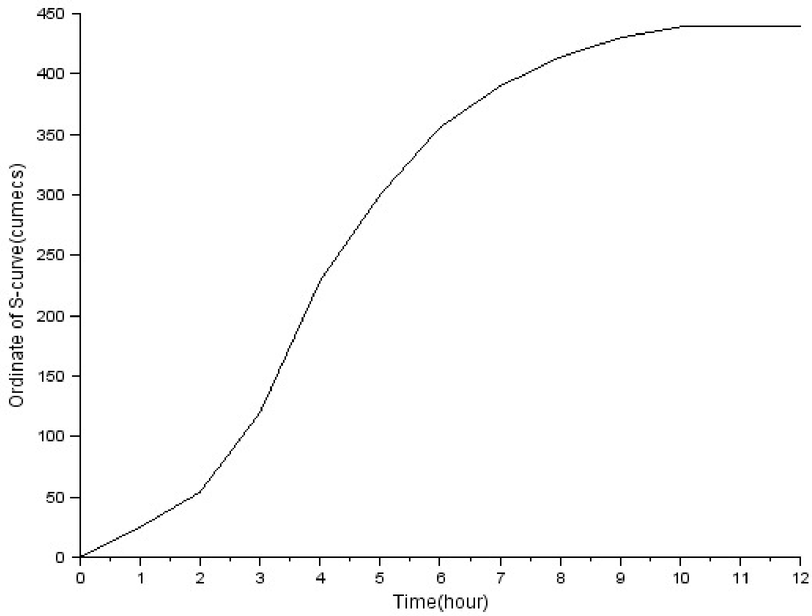

Figure 4.24: EX4 62

```
    t(i)=round(t(i)*10)/10;
    mprintf("\n%f",t(i));
end
```

Scilab code Exa 4.62 EX4 62

```
//example 4.62
//ordinates of 1 hr unit hydrograph
clc;funcprot(0);
//given
t=[0:1:12];        //time
```

```
O=[0 0 54 0 175 0 127 0 58 0 25 0 0 0];              //ordinate of 2 hr unit hydrograph
of(1)=0;
of(2)=0;
for i=3:13
    if modulo(i,2)==0;
        of(i)=0;

else
    of(i)=O(i-2)+of(i-2);
end
end
s=[0 25 54 120 229 300 356 390 414 430 439 439 439];       //Ordinates of S-curve
for i=2:13
    of1(i)=s(i-1);
end
mprintf("ordinates of 1 hr unit hydrograph:");
for i=1:13
    y(i)=s(i)-of1(i);
u(i)=y(i)*2;
mprintf("\n%i",u(i));
end
//graph is plotted between u and t
```

Scilab code Exa 4.63 EX4 63

```


//example 4.63
//calculate  design disharge
clc;funcprot(0);
//given
xavg=1200;          //sample mean
n=50;               //assurance year
```

```
A=0.95;            //assurance percent
Rsk=1-A;
sigma=650;         //standard deviation
yn=0.53622;        //mean of reduced variate
sigma30=1.11238;   //standard deviation of reduced
    variate

T=1/(1-(1-Rsk)^(1/n));
yt=-2.303*log10(2.303*log10(T/(T-1)));
K=(yt-yn)/sigma30;
xt=xavg+K*sigma;
mprintf(" design disharge=%i cumecs.",xt);
```

Scilab code Exa 4.64 EX4 64

```


//example 4.64
//calculate flood magnitude with return period of
    500 years
clc;funcprot(0);
//given
T1=50;T2=100;              //Return period
F1=20600;F2=22150;         //Peak flood
y100=-(2.303*log10(2.303*log10(T2/(T2-1))));
y50=-(2.303*log10(2.303*log10(T1/(T1-1))));
y=(F2-F1)/(y100-y50);
T=500;
y500=-(2.303*log10(2.303*log10(T/(T-1))));
x500=F2+(y500-y100)*y;
x500=round(x500);
mprintf("flood magnitude with return period of 240
    years=%f cumec.",x500);
```

Scilab code Exa 4.65 EX4 65

```
//example 4.65
//calculate recurrence interval of 10 minutes storm
   using Gumbel's method
clc;funcprot(0);
//given
xavg=1.65;        //mean of data
sigma=0.45;       //standard deviation
x=3;
y=1.2825*(x-xavg)/sigma+0.577;
l=%e^(%e^(-y));
T=l/(l-1);
T=round(T*10)/10;
mprintf("recurrence interval of 10 minutes storm=%f
   years.",T);
```

Scilab code Exa 4.66 EX4 66

```
//example 4.66
//calculate flood magnitude with return period of
   200 years
clc;funcprot(0);
//given
T1=50;T2=100;              //Return period
F1=30800;F2=36300;         //Peak flood
y100=-(2.303*log10(2.303*log10(T2/(T2-1))));
y50=-(2.303*log10(2.303*log10(T1/(T1-1))));
```

```
y=(F2-F1)/(y100-y50);
T=200;
y200=-(2.303*log10(2.303*log10(T/(T-1))));
x200=F2+(y200-y100)*y;
x200=round(x200);
mprintf("flood magnitude with return period of 240
    years=%f cumecs.",x200);
```

Scilab code Exa 4.67 EX4 67

```


//example 4.67
//calculate return period of flood of 9950 cumec/s
clc;funcprot(0);
//given
xavg=4200;       //mean
sigma=1705;      //standard deviation
xt=9550;         //flood value
K=(xt-xavg)/sigma;
yt=1.2825*K+0.577;
l=%e^(%e^(-yt));
T=l/(l-1);
T=round(T*100)/100;
mprintf("Return period of flood of 9950 cumec/s=%f
    years.",T);
```

Scilab code Exa 4.68 EX4 68

```


//example 4.68
```

```
//calculate flood magnitude with return period of 1000 years
clc;funcprot(0);
//given
T1=100;T2=50;              //Return period
F1=485;F2=445;         //Peak flood
y50=-(2.303*log10(2.303*log10(T2/(T2-1))));
y100=-(2.303*log10(2.303*log10(T1/(T1-1))));
y=(F2-F1)/(y50-y100);
T=1000;
y1000=-(2.303*log10(2.303*log10(T/(T-1))));
x1000=F2+(y1000-y50)*y;
x1000=round(x1000*10)/10;
mprintf("flood magnitude with return period of 240 years=%f cumecs.",x1000);
```

Scilab code Exa 4.69 EX4 69

```


//example 4.69
//calculate
//probability of exceedence
//probability of occurence in next 12 years
clc;funcprot(0);
//given
T=25;          //return period
n=12;
P=1/T;
Rsk=1-(1-P)^n;
P=round(P*100)/100;
Rsk=round(Rsk*10000)/10000;
mprintf("probability of exceedence=%f.",P);
mprintf("\nprobability of occurence in next 12 years=%f.",Rsk);
```

Chapter 5

GROUND WATER WELL IRRIGATION

Scilab code Exa 5.1 EX5 1

```


//example 5.1
//design an open wellin fine sand
clc;
//given
Q=0.003;          //required discharge
H=2.5;            //depression head
A=Q*3600/(0.5*H);
d=(4*A/%pi)^0.5;
d=round(d*100)/100
mprintf("Well diameter=%f m.",d);

//Alternative solution
C=7.5D-5;     //permeability constant from table 5.2
A=Q/(C*H);
d=(16*3/%pi)^0.5;
d=round(d*10)/10;
mprintf("\nBy alternative solution:")
```

```
mprintf("\nWell diameter=%f m",d);
```

Scilab code Exa 5.2 EX5 2

```


//example 5.2
//calculate
//yield from well
//diameter of well
clc;
//given
h1=2.5;                    //initial pumping depression
h=1.8;                     //heigth after recuperation
t=80;                      //time
h2=h1-h;
KbyA=2.303*60*log10(h1/h2)/t;


//Part (a)
d=4;              //diameter of well
H=3;              //depression head
A=%pi*d^2/4;
Q=(KbyA)*A*H/3.6;
mprintf("Part (a)");
Q=round(Q);
mprintf("\nYield from well=%f lit/sec.",Q);

//Part (b)
Q=8;             //yield(lit/sec)
H=2;
A=Q*3.6/(H*(KbyA));
d=(4*A/%pi)^0.5;
d=round(d*10)/10;
```

```
mprintf("\nPart (b)");
mprintf("\nDaimeter of well=%f m",d);
```

Scilab code Exa 5.3 EX5 3

```


//example 5.3
//calculate yield from well
clc;
//given
d=30;           //well diameter
L=15;           //strainer length
P=50;           //coefficient of permeability
s=0.2;          //effective size of sand
b=3;            //drawdown
r=150;          //radius of drawdown

Q=2.72*L*P*b/(log10(r*2*100/d)*24*3.6);
Q=round(Q*10)/10;
mprintf(" yield from well=%f lit/sec.",Q);
```

Scilab code Exa 5.4 EX5 4

```


//example 5.4
//calculate discharge from tubewell
clc;
//given
d=30;              //diameter of well
s=2;               //drawdown
L=10;              //length of stainer
```

```
k=0.05;          //coefficient of permeability
r=300;           //radius of zero drawdown
Q=2.72*k*s*(L+s/2)/(100*log10(2*100*r/d));
Q=round(Q*10000)/10000;
mprintf(" discharge from tubewell=%f cumec.",Q);
```

Scilab code Exa 5.5 EX5 5

```


//example 5.5
//design tube well
clc;
//given
Q=0.08;               //yield required
b=30;                 //thickness of acquifer
R=300;                //Radius of circle of influence
k=60;                 //permeability coefficient
s=5;                  //Drawdown
r=R/(10^(2.72*b*s*k/(3600*24*Q)));
r=round(r*10000)/10000;
mprintf("Radius of well=%f m",r);
```

Scilab code Exa 5.6 EX5 6

```


//example 5.6
//calculate yield from well
clc;
//given
b=30;                     //thickness of acquifer
s=4;                      //drawdown
```

```
r=0.1;                    //well radius
k=36;                     //permeability coefficient
R=3000*s*(k/(24*3600))^0.5;

Q=2.72*b*k*s/(log10(R/r)*24*3.6);
Q=round(Q*10)/10;
mprintf("yield from well=%f lit/sec.",Q);
```

Scilab code Exa 5.7 EX5 7

```


//example 5.7
//calculate discharge and percent increase in discharge
clc;
//given
k=0.005;              //coefficient of permeability
r=0.1;                //well radius
s=4;                  //drawdown
b=10;                 //thickness
R=300;                //radius of circle of influence
//Part(a)
Q1=2.72*b*k*s/log10(R/r);
Q1=round(Q1*10000)/10000;
mprintf("Discharge=%f cumec",Q1);

//Part (b)
r=0.2;
Q2=2.72*b*k*s/log10(R/r);
I=(Q2-Q1)*100/Q1;
I=round(I*10)/10;
mprintf("\npercent increase in discharge=%f percent.",I);
```

Scilab code Exa 5.8 EX5 8

```
//example 5.8
//calculate coefficient of permeability
//percentage error
//actual radius of influence
clc;
//given
d=0.2;              //diameter of well
Q=240;              //discharge
RL1=240.5;          //reduce level of original water surface
RL2=235.6;          //reduced level of water at pumping
RL3=210;            //reduced level of impervious layer
RL4=239.8;          //reduced level of water in well
D=50;               //radial distance of well from tube well
//Part(a)
h1=RL2-RL3;
h2=RL4-RL3;
k1=Q*24*log10(D*2/d)/(1.36*(h2^2-h1^2));
k1=round(k1*100)/100;
mprintf("Part(a)");
mprintf("\ncoefficient of permeability=%f m/day.",k1);
//Part (b)
R=300;                  //radius of influence
H=RL1-RL3;
h=RL2-RL3;
k2=Q*24*log10(R*2/d)/(1.36*(H^2-h^2));
PE=(k2-k1)*100/k1;
mprintf("\nPart(b)");
```

```
mprintf("\npercentage error=%i percent.",PE);
//Part (b)
R=(d/2)*10^(1.36*k1*(H^2-h^2)/(24*Q));
mprintf("\nPart(c)");
mprintf("\nActual radius of influence=%i m.",R);
```

Scilab code Exa 5.9 EX5 9

```


//example 5.9
//calculate input h.p of pump
clc;
//given
A=20;              //area of field
H=129;             //level to the highest land
h1=120.2;          //water level in well during
   discharge
Du=800;            //duty for rise;
eita=0.6;          //efficiency of the pump
Q=A/Du;
w=Q*1000;
lift=H-h1;
//design lift is taken as 9m
wd=w*9;
o=wd/75;
i=o/eita;
mprintf("Input h.p of pump=%i h.p",i);
```

Scilab code Exa 5.10 EX5 10

```
//example 5.10
//calculate culturable area
clc;
//given
Q=150;              //discharge from tubewell
t=4000;             //working period of tubewell
I=0.45;             //intensity of irrigation
d=0.38;             //average depth of rabi and kharif crop
V=Q*t;
A=V/d;
CA=A/(I*10000);
CA=round(CA);
mprintf("culturable area=%f hectares.",CA);
```

Scilab code Exa 5.11 EX5 11

```
//example 5.11
//calculate discharge if one well discharges
//percent decrease when two well discharges
clc;
//given
d=0.2;                //diameter of well
r=d/2;
B=100;                 //distance between wells
b=12;                 //thickness of acquifer
k=60;                 //coefficient of permeability
s=3;                  //dispersion head
R=250;                //radius of influence
Q=2.72*b*k*s/(24*log10(R/r));
mprintf("discharge if one well discharges=%i cubic metre/hour.",Q);
//when both well are discharging
```

```
Q1=2.72*k*b*s/(24*log10(R^2/(r*B)));
Q1=round(Q1*10)/10;
mprintf("\ndischarge if both wells discharges=%f
    cubic metre/hour.",Q1);
PE=(Q-Q1)*100/Q;
PE=round(PE*100)/100;
mprintf("\npercentage decrease in discharge=%f
    percent.",PE);
```

Scilab code Exa 5.12 EX5 12

```


//example 5.12
//calculate radius of zero drawdown
//coefficient of permeability
//drawdown in well
//specific capacity
//maximum rate at which water can be pumped
clc;
//given
d=0.6;              //diameter of well;
rw=d/2;
H=40;               //depth of water in well before
    pumping
Q=2000;             //discharge from well
s1=4;               //drawdown in well
B1=10;              //distance between well
s2=2;
B2=20;
//Part (a)
h1=H-s1;
h2=H-s2;
t=(H^2-h2^2)/(H^2-h1^2);
R=(B2/(B1^t))^(1/(1-t));
```

```
R=round(R*100)/100;
mprintf(" radius of zero drawdown=%f m",R);
//Part (b)
r=10;
k=Q*log10(R/r)*60*24/(1.36*(H^2-h1^2)*1000);
k=round(k*100)/100;
mprintf("\ncoefficient of permeability=%f m/day.",k);

//part (c)
Ho=(H^2-(Q*log10(R/rw)*24*60/(1000*1.36*k)))^0.5;
D=H-Ho;
D=round(D*100)/100;
mprintf("\ndrawdown in well=%f m.",D);

//part (d)
C=Q/(1000*R);
//for R=1 m;Q=Sc
//hence on putting the values in discharge equation we get
//Sc*log10(61.2*Sc)=0.3223.
//on solving this by trial and error method we get Sc=0.266 m^2/min.
mprintf("\nSpecific capacity=0.266 cubic metre/minutes/metre.");

//part (e)
//this is obtained when Q=H
//hence from equation of discharge,we get
//Q*log10(69.2*Q)=6.528.
//solving it by trial and error method we get Q=2.85 m^3/min.
mprintf("\nmaximum rate at which water can be pumped =2.85 cubic metre/min");
```

Scilab code Exa 5.13 EX5 13

```


//example 5.13
//calculate formation constant of acquifer using
   theis method
clc;funcprot(0);
//given
Q=2500;      //discharge(l/min)
r=60;        //distance of observation well from
   acquifer
tmin=[1 1.5 2 2.5 3 4 5 6 8 10 12 14 18 24 30 40 50
   60 80 100 120 150 180 210 240];  //time in
   minutes
s=[0.2 0.26 0.3 0.33 0.36 0.41 0.45 0.48 0.53 0.56
   0.59 0.62 0.66 0.71 0.75 0.80 0.83 0.86 0.91 0.95
    0.98 1.03 1.05 1.08 1.10];  //drawdown
u=[1:1:9];
Wu=[0.2194 0.04891 0.01315 0.003779 0.001148
   0.000360 0.000116 0.0000377 0.0000125];
for i=1:25
    tday(i)=tmin(i)/(60*24);
end

for i=1:25
    rt(i)=r^2/tday(i);
end
//graph is plotted between s and r^2/t and W(u) and
   u and they are superimposed.
//from which we get
s1=0.52;
Wu1=2.96;
rt1=700000; u1=0.03;
Q=3600;                  //discharge in cumec/day
T=Q*Wu1/(4*%pi*s1);
S=4*u1*T/rt1;
T=round(T);
```

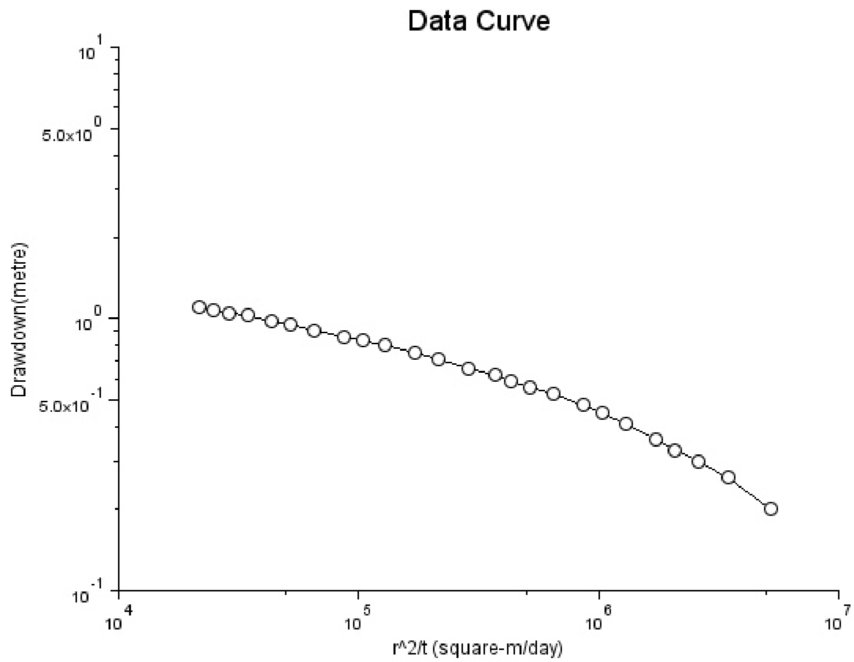

Figure 5.1: EX5 13

```
mprintf("formation constant of acquifer:");
mprintf("\nT=%f cubic metre/day/m.\nS=%f.",T,S);
```

Scilab code Exa 5.14 EX5 14

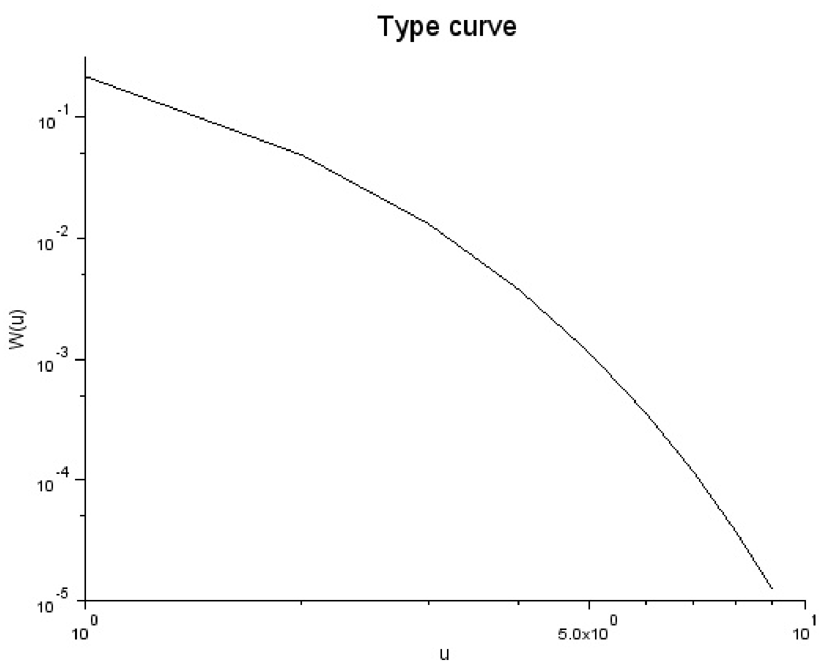

Figure 5.2: EX5 13

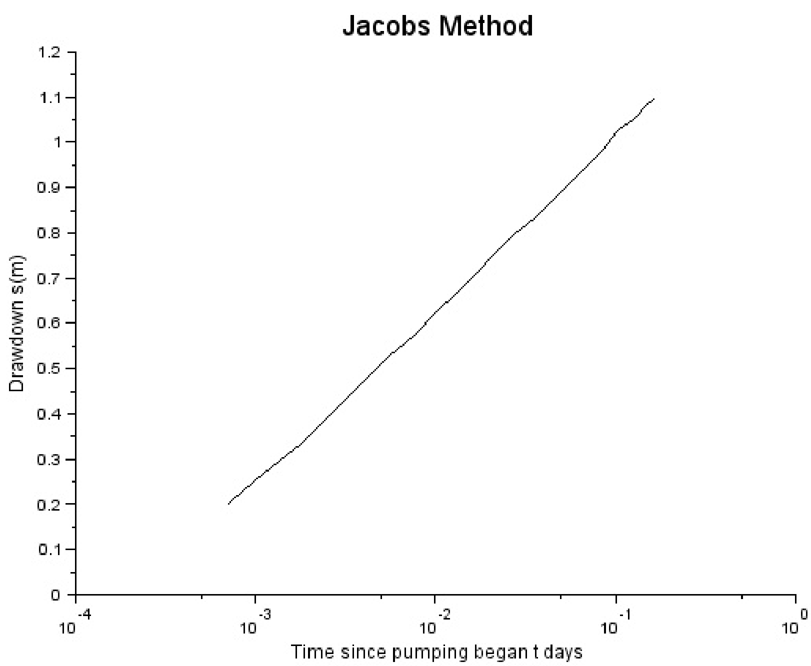

Figure 5.3: EX5 14

```
//example 5.14
//calculate formation constant of acquifer using Jacob's method
clc;funcprot(0);
//given
Q=2500;        //discharge(l/min)
r=60;          //distance of observation well from acquifer
tmin=[1 1.5 2 2.5 3 4 5 6 8 10 12 14 18 24 30 40 50 60 80 100 120 150 180 210 240];  //time in minutes
s=[0.2 0.26 0.3 0.33 0.36 0.41 0.45 0.48 0.53 0.56 0.59 0.62 0.66 0.71 0.75 0.80 0.83 0.86 0.91 0.95 0.98 1.03 1.05 1.08 1.10];  //drawdown
for i=1:25
    tday(i)=tmin(i)/(60*24);
end
//from the graph between s and t we get
ds=0.38;
Q=3600;        //discharge in cumec/day
T=2.303*Q/(4*%pi*ds);
//extending the straight line we get
to=0.00024;
S=2.25*T*to/r^2;
mprintf("formation constant of acquifer:");
mprintf("\nT=%i cubic metre/day/m.\nS=%f.",T,S);
```

Scilab code Exa 5.15 EX5 15

```
//example 5.15
//calculate formation constant of acquifer using Chow's method
clc;funcprot(0);
```

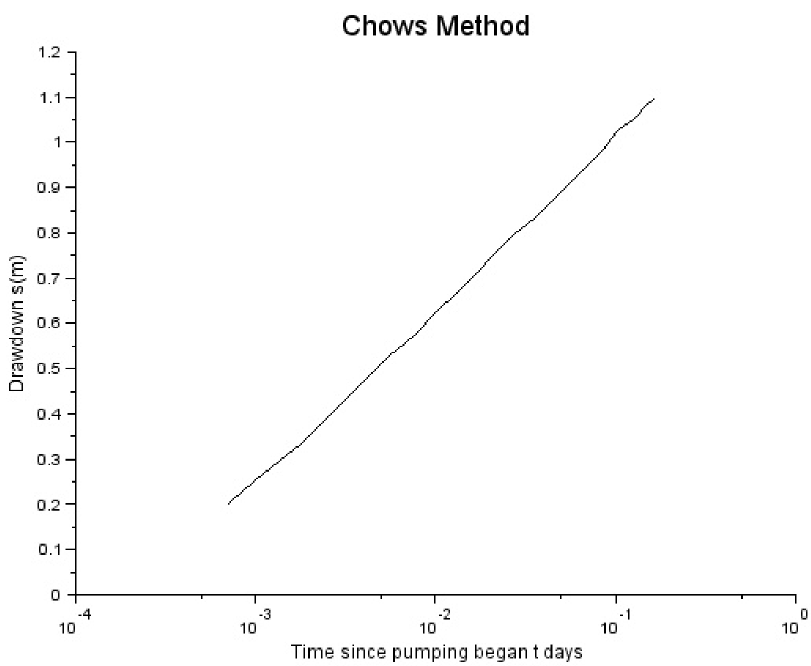

Figure 5.4: EX5 15

```
//given
Q=2500;        //discharge(l/min)
r=60;          //distance of observation well from acquifer
tmin=[1 1.5 2 2.5 3 4 5 6 8 10 12 14 18 24 30 40 50 60 80 100 120 150 180 210 240];  //time in minutes
s=[0.2 0.26 0.3 0.33 0.36 0.41 0.45 0.48 0.53 0.56 0.59 0.62 0.66 0.71 0.75 0.80 0.83 0.86 0.91 0.95 0.98 1.03 1.05 1.08 1.10];  //drawdown
for i=1:25
    tday(i)=tmin(i)/(60*24);
end
//graph is plotted between s and t
//point P is choosen on it whose ordinate is:
s1=0.45;
t=0.00347;
ds=0.38;                //for one log cycle of time
Fu=s1/ds;
//from fig 5.43
//or using relation
Wu=2.303*Fu;
u=0.035;     //from table 5.2
Q=3600;          //discharge in cumec/day
T=Q*Wu/(4*%pi*s1);
S=4*u*t*T/r^2;
mprintf("formation constant of acquifer:");
mprintf("\nT=%i cubic metre/day/m.\nS=%f.",T,S);
```

Scilab code Exa 5.16 EX5 16

```


//example 5.16
//calculate transmissibility of acquifer
```

```
//draw daown in main well
clc;
//given
H=25;           //static water level
rw=0.15;        //radius of well
Q=5400;         //discharge(litre/min)
t=24;           //time of discharge
r1=30;           //distance of first well
s1=1.11;        //drawdown
h1=H-s1;
r2=90;          //distance of second well
s2=0.53;        //drawdown
h2=H-s2;
k=(Q*2.303*log10(r2/r1))/(%pi*(h2^2-h1^2)*60000);
T=k*H;
T=round(T*10000)/10000;
mprintf("transmissibility of acquifer=%f cumec/sec."
   ,T);
hw=(h2^2-(Q*2.303*log10(r2/rw))/(%pi*k*60000))^0.5;
sw=H-hw;
sw=round(sw*100)/100;
printf("\ndraw daown in main well=%f m.",sw);
```

Scilab code Exa 5.17 EX5 17

```


//example 5.17
//calculate discharge at 18m drawdown
clc;
//given
Q=250;            //discharge(lit/min)
H=100;            //water level in acquifer
s1=12;            //drawdown
h1=H-s1;
```

```
//let t=ln(R/r)/(pi*k)
t=(H^2-h1^2)/Q;
h2=H-18;
Q1=(H^2-h2^2)/t;
mprintf("discharge at 18m drawdown=%i lpm",Q1);
```

Scilab code Exa 5.18 EX5 18

```


//example 5.18
//calculate effective well diameter
clc;
//given
b=10;                //thickness of acquifer
k=48;                //permeability coefficient
R=500;               //radius of influence
s=12;                //drawdown
Q=5000;              //discharge(cumec/day)
r=R/%e^(2*%pi*b*k*s/Q);
D=2*r;
D=round(D*100)/100;
mprintf("effective well diameter=%f m.",D);
```

Scilab code Exa 5.19 EX5 19

```


//example 5.19
//calculate drawdown at 40m
clc;
//given
Q=1500;              //discharge(lit/min)
```

```
S=0.004;                //storage coefficient
k=35;                   //permeability
t=20;                   //time of pumping
b=30;                   //thickness of acquifer
r=40;                   //distance of observation well
T=k*b;
s=Q*(2.303*log10(4*T*t*3600/(60*60*24*r^2*S))
   -0.5772)*60*60*24/(4*%pi*T*60000);      //Jacob's
   equation
s=round(s*100)/100;
mprintf("drawdown at 40m=%f m.",s);
```

Scilab code Exa 5.20 EX5 20

```


//example 5.20
////calculate
//yield from well
clc;
//given
h1=2.5;                       //initial pumping
   depression
h=1.8;                        //heigth after recuperation
t=80;                         //time
h2=h1-h;
KbyA=2.303*60*log10(h1/h2)/t;
d=4;             //diameter of well
H=3;             //depression head
A=%pi*d^2/4;
Q=(KbyA)*A*H/3.6;
Q=round(Q);
mprintf("\nYield from well=%f lit/sec.",Q);
```

Scilab code Exa 5.21 EX5 21

```


//example 5.21
//calculate transmissibility
//drawdown at pumping well
clc;
//given
rw=0.15;            //radius of well
b=40;               //depth of acquifer
Q=1500;             //discharge(lpm)
s1=3.5;             //drawdown of first well
s2=2;               //drawdown of second well
H=40;
r1=25;              //distance of first well
r2=75;              //distance of second well
h1=H-s1;
h2=H-s2;
k=Q*2.303*log10(r2/r1)/(%pi*1000*60*(h2^2-h1^2));
T=b*k*1000;
mprintf("transmissibility=%fD-3 square metre/sec",T);

hw=(h2^2-(Q*2.303*log10(r2/rw)/(%pi*k*60000)))^0.5;
sw=H-hw;
sw=round(sw*100)/100;
mprintf("\ndrawdown at pumping well=%f m.",sw);
```

Scilab code Exa 5.22 EX5 22

```

```

```

//example 5.22
//calculate coefficient of permeability
//drawdown in test well
clc;
//given
r=0.25;        //radius of test well
r1=10;         //distance of first well
r2=60;         //distance of second well
Q=0.1;         //discharge(cumec/sec)
s1=4;          //drawdown of first well
s2=3;          //drawdown of second well
b=20;          //thickness of well
k=1000*Q*log10(r2/r1)/(2.72*b*(s1-s2));
mprintf("coefficient of permeability=%fD-3 m/sec",k);
s=s2+Q*log10(r2/r)/(2.72*b*k);
s=round(s*100)/100;
mprintf("\ndrawdown in test well=%f m.",s);
```

Scilab code Exa 5.23 EX5 23

```


//example 5.23
//calculate
//diameter of well
clc;
//given
h1=2.1;                  //initial pumping depression
h=1.6;                   //heigth after recuperation
t=90;                    //time
h2=h1-h;
KbyA=2.303*60*log10(h1/h2)/t;
```

```
Q=10;           //yield(lit/sec)
H=2;
A=Q*3.6/(H*(KbyA));
d=(4*A/%pi)^0.5;
d=round(d*10)/10;
mprintf("\nDaimeter of well=%f m",d);
```

Scilab code Exa 5.24 EX5 24

```


//example 5.24
//calculate yield from well
clc;
//given
h1=2.5;                    //initial pumping depression
h=1;                      //heigth after recuperation
t=60;                      //time
h2=h1-h;
KbyA=2.303*60*log10(h1/h2)/t;
d=2;             //diameter of well
H=3;             //depression head
A=%pi*d^2/4;
Q=(KbyA)*A*H;
Q=round(Q*1000)/1000;
mprintf("\nYield from well=%f cubic metre/hour.",Q);
```

Chapter 6

RESERVIOR PLANNING

Scilab code Exa 6.1 EX6 1

```



//exampple 6.1
//determine maximum reservior level
//maximum discharge over spillway
//plot inflow and routed hydrograph and find peak
   flow and peak lag
clc;funcprot(0);
//given
e=[100 100.3 100.6 100.9 101.2 101.5 101.8 102.1
   102.4 102.7];     //elevation(km)
A=[405 412 420 425 428 436 445 453 460 469];
            //area
o=[0 14.9 42.2 77.3 119 169 217 272 334 405];
           //outflow
c(1)=0;
for i=2:10
    dh(i)=e(i)-e(i-1);
    s(i)=dh(i)/3*(A(i-1)+A(i)+(A(i-1)*A(i))^0.5);
             //storage between contours
    c(i)=c(i-1)+s(i);                        //
```

```
        cumulative storage
    h(i)=c(i)/1.08;                                  //2s/t
    h1(i)=h(i)-o(i);                                 //2s/t-o
    h2(i)=h(i)+o(i);                                 //2s/t+o
end
T=[0:6:102];
I=[42 45 57 88 147 210 272 340 350 338 314 288 263 240 198 170 143 120];  //inflow
h4=[0 0 60 122 185 266 362 455 545 605 623 620 600 575 550 515 470 430];    //2s/t-0 obtained from curve a
O=[0 10 24 42 74 130 194 260 316 334 328 312 286 264 236 204 177 150];      //outflow read from curve a
re=[100.2 100.39 100.58 100.86 101.26 101.65 102.03 102.31 102.4 102.37 102.3 102.18 102.06 101.9 101.72 101.56 102.4];    //reservior elevation read from curve b
for i=2:17
    t(i)=I(i-1)+I(i);                                //I1+I2
    h3(i)=t(i)+h4(i);                                //2s/t+O
end
pt=T(10)-T(9);
d=I(9)-O(10);
//results
mprintf(" maximum reservior level=%f m.",re(10));
mprintf("\nmaximum discharge over spillway=%f cumecs.",O(10));
mprintf("\nreduction in peak discharge=%f cumecs.",d);
mprintf("\npeak lag=%f hours.",pt);
```

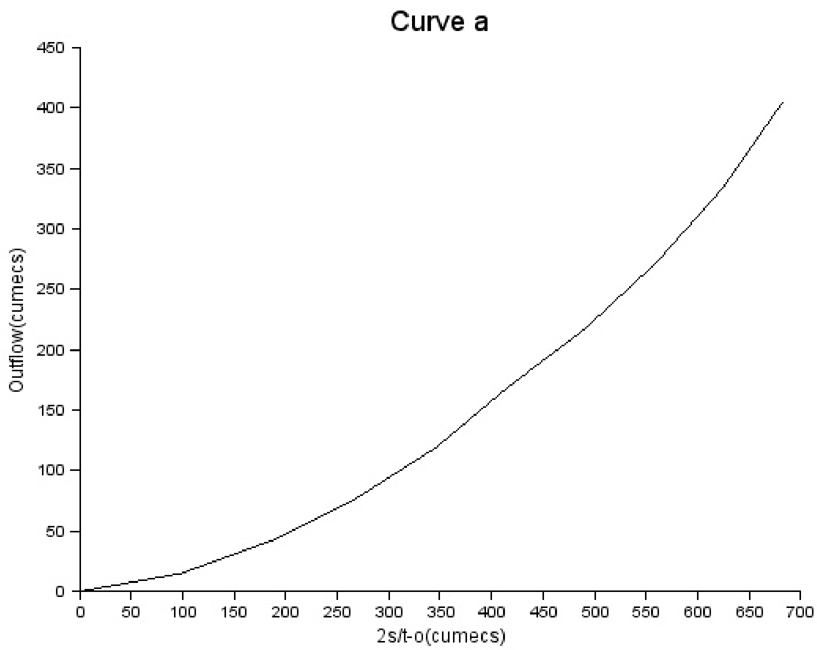

Figure 6.1: EX6 1

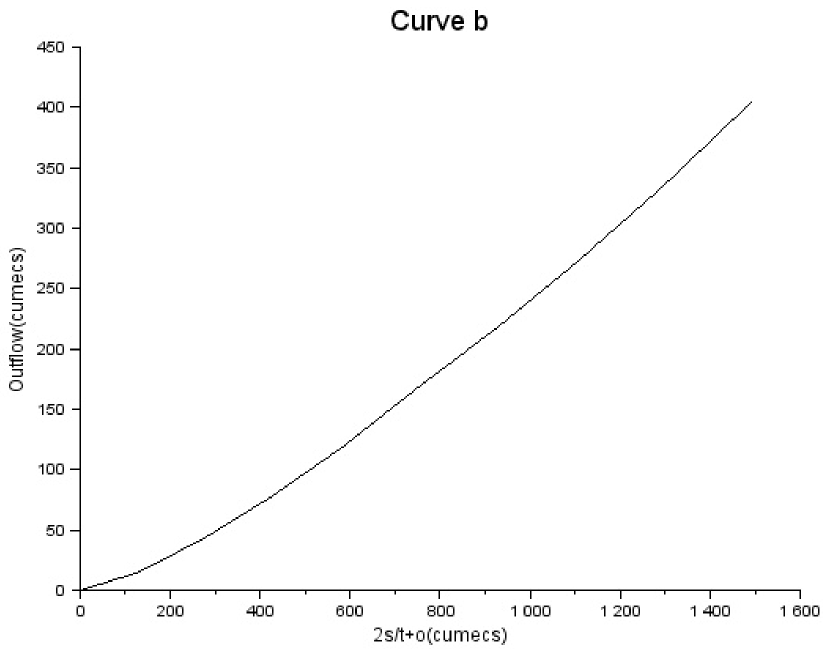

Figure 6.2: EX6 1

Scilab code Exa 6.2 EX6 2

```


//example6.2
//calculate required useful storage
clc;funcprot(0);
//given
in=[8.6 2.2 1.8 0 0 13.5 280.6 510.2 136 52.5 20.6
    12.3];        //inflow(ha-m)
```

```
pan=[2.2 2.3 3.1 8.6 12.8 15.6 12.3 10.6 10 8.2 5.8
    3];        //pan evaporation
p=[0.8 1.2 0 0 0 4.8 12.2 18.6 8.6 1.5 0 0]
                    //precipitation
D=[14.5 15.8 16.2 16.8 17.5 18 18 17 16.5 16 15.8
    15];        //Demand
s=0;
for i=1:12
    if in(i)<10 then
        r(i)=in(i);                                   //D/S requirement
    else
        r(i)=10;
    end
    E(i)=3.6*pan(i);                                  //Evaporation over reservior area
    P(i)=3.5*p(i);                                    //Precipitation
    I(i)=in(i)-r(i)-E(i)+P(i);                        //Adjusted inflow
    S(i)=D(i)-I(i);                                   //Water required from storage
    if S(i)<0 then
        S(i)=0;
    end
    s=s+S(i);
end
mprintf("required useful storage=%f ha-m.",s);
```

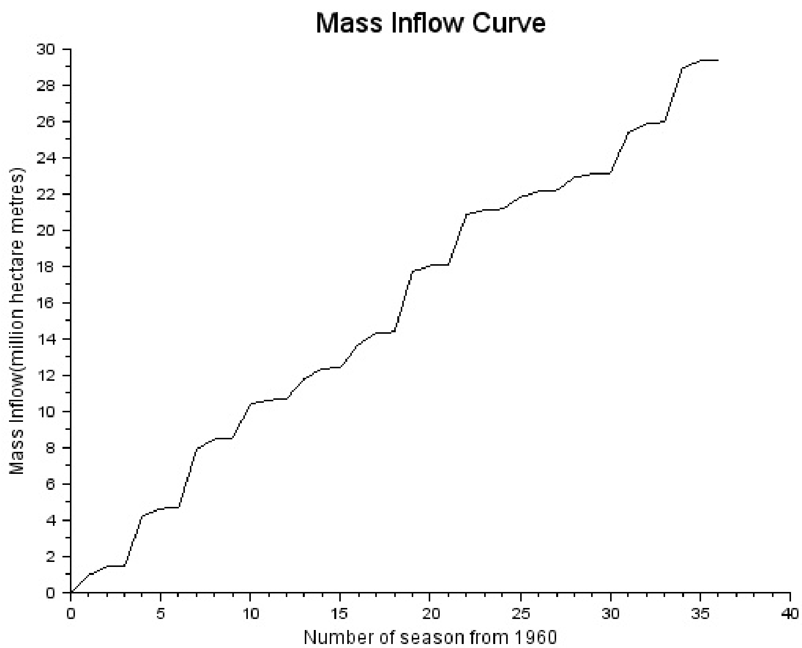

Figure 6.3: EX6 3

Scilab code Exa 6.3 EX6 3

```


//example 6.3
//calculate storage capacity of reservior
clc;funcprot(0);
//given
V=475;              //flow required to be maintained throughout the year
Y=V*365*8.64;       //yearly demand
//yearly demand gives the slope of demand curve
t=[0:1:36];              //number of season startin from 1960;each year is diveded into 3 seasons.
q=[0 1050 300 50 3000 250 40 3500 370 90 2000 150 120 1200 350 65 1400 400 100 3600 200 80 3000 200 80 3000 150 120 700 210 50 800 120 80 2400 320 120 3200 280 80];     //average discharge
v=[0 0.9707 0.4717 0.0328 2.7734 0.3981 0.0263 3.2357 0.5818 0.0591 1.8490 0.2356 0.0788 1.1094 0.5504 0.0427 1.2943 0.6290 0.0657 3.3281 0.3145 0.0525 2.7734 0.2359 0.0788 0.6441 0.3302 0.028 0.7396 0.1887 0.0525 2.2188 0.5032 0.0788 2.9583 0.4403 0.0525];          //voloume
cv(1)=v(1);
for i=2:37
    cv(i)=cv(i-1)+v(i);
end
//each year is divided into three seasons(monsoon, winter and summer).and readings are taken for 12 years
//mass inflow curve is plotted and tangent are drawn at the apexes and parellel to demand curve slope;
//the respectiv ordinates represent the deficiency during dry period
//maximum of these ordinates gives the desired reservior capacity
```

```
mprintf("storage capacity of reservior=1.6 million ha-m.");
```

Scilab code Exa 6.4 EX6 4

```


//example 6.4
//calculate probable life of reservior
clc;funcprot(0);
//given
asi=3.6;                                    //annual sediment inflow(x10^6)
gamma_s=12;                                 //specific weigth of sediment
vs=asi/12;
ir=30;                                      //initial reservior capacity
fr=60;                                      //final reservior capacity
r=ir/fr;                                    //initial capacity/inflow ratio
//r=0.5; hence we start capacity/inflow ratio from 0.5
c=[0.5:-0.1:0.1];                           //capacity inflow ratio
e=[0.96 0.955 0.95 0.93 0.87];              //trap efficiency
for i=1:4
    ae(i)=(e(i)+e(i+1))/2;                  //average efficiency for interval
end
as=[0.2872 0.2857 0.2820 0.2700];           //annual sediment trapped
s=0;
```

```
for i=1:4
    y(i)=6/as(i);                    //year to fill
    s=s+y(i);
end
mprintf(" probable life of reservior=%i years.",s);
```

Scilab code Exa 6.5 EX6 5

```


//example 6.5
//calculate maximum outflow discharge over spillway
//corresponding maximum level of water above
    spillway crest
clc;funcprot(0);
//given
I=[60 480 900 470 270 160 110 80 60];    //inflow
//for the first time interval 0 hours to 3 hours
I1=I(1);
I2=I(2);
t=3*3600;
ti=(I1+I2)*t/2;                    //total inflow
//outflow=1.62*h1^1.5;
//storage change=(30+3h1)h1
//from the basic equation i.e total inflow=total
    outflow+change in storage
//on solving we get
//h1^2+0.54h1^1.5+10h1-0.972=0;
//solving it by trial and error method;we get
h1=0.0954;
//for the second time interval 3 hours to 6 hours
I1=I(2);
I2=I(3);
t=3*3600;
ti=(I1+I2)*t/2;                    //total inflow
```

```
//outflow=0.0477+1.62*h2^1.5;
//storage change=(30+3h2)h2
//from the basic equation i.e total inflow=total
   outflow+change in storage
//on solving we get
//h2^2+0.54h2^1.5+10h2-3.4312=0;
//solving it by trial and error method;we get
h2=0.323;
//for the third time interval 6 hours to 9 hours
I1=I(3);
I2=I(4);
t=3*3600;
ti=(I1+I2)*t/2;                   //total inflow
//outflow=0.2974+1.62*h3^1.5;
//storage change=(30+3h3)h3
//from the basic equation i.e total inflow=total
   outflow+change in storage
//on solving we get
//h3^2+0.54h3^1.5+10h3-5.7012=0;
//solving it by trial and error method;we get
h3=0.522;
//for the fourth time interval 9 hours to 12 hours
I1=I(4);
I2=I(5);
t=3*3600;
ti=(I1+I2)*t/2;                   //total inflow
//outflow=0.611+1.62*h4^1.5;
//storage change=(30+3h4)h4
//from the basic equation i.e total inflow=total
   outflow+change in storage
//on solving we get
//h4^2+0.54h4^1.5+10h4-6.6208=0;
//solving it by trial and error method;we get
h4=0.601;
//for the fifth time interval 12 hours to 15 hours
I1=I(5);
I2=I(6);
t=3*3600;
```

```
ti=(I1+I2)*t/2;                    //total inflow
//outflow=0.7548+1.62*h5^1.5;
//storage change=(30+3h5)h5
//from the basic equation i.e total inflow=total
    outflow+change in storage
//on solving we get
//h5^2+0.54h5^1.5+10h5-6.8936=0;
//solving it by trial and error method;we get
h5=0.624;
//for the sixth time interval 12 hours to 15 hours
I1=I(6);
I2=I(7);
t=3*3600;
ti=(I1+I2)*t/2;                    //total inflow
//outflow=0.7985.62*h6^1.5;
//storage change=(30+3h6)h6
//from the basic equation i.e total inflow=total
    outflow+change in storage
//on solving we get
//h6^2+0.54h6^1.5+10h6-6.8492=0;
//solving it by trial and error method;we get
h6=0.620;
hmax=h5;
q=300*(h5)^1.5;          //equation given
q=round(q*100)/100;
mprintf("maximum outflow discharge over spillway=%f
    cumecs.",q);
mprintf("\nmaximum level of water above spillway
    crest=%f m.",h5);
```

Scilab code Exa 6.6 EX6 6

```


//example 6.6
```

```
//calculate the allocations to each project purpose
clc;funcprot(0);
//given
t=240;              //total cost of project(million rupees)
s=[32 88 72];       //separable cost
eb=[40 138 112];    //estimated benifit
sp=[47 104 101];    //alternate single purpose cost
//using remaining benifit method
ts=s(1)+s(2)+s(3); //total separable cost
tj=t-ts;            //total joint cost
w=0;
for i=1:3
    if eb(i)<sp(i) then
        b(i)=eb(i);            //benifit limited by alternate cost
    else
        b(i)=sp(i);
    end
   rb(i)=b(i)-s(i);            //remaining benifit
   w=w+rb(i);

end
y=0;
for i=1:3
     aj(i)=tj*rb(i)/w;         //allocated joint cost
     ta(i)=s(i)+aj(i);         //total allocations
     y=y+ta(i);
end
mprintf("Using remaining benifit method.");
mprintf("\n\nallocations to each project purpose(percent):");
for i=1:3
    per(i)=ta(i)*100/y;             //total allocation percent
    mprintf("\n%f",per(i));
end
```

```
//using alternate justifiable method
w=0;
for i=1:3
    ac(i)=sp(i)-s(i);              //alternate cost
       less separable cost
    w=w+ac(i);

end
y=0;
for i=1:3
    ajc(i)=tj*ac(i)/w;           //allocated joint
       cost
    ta(i)=s(i)+ajc(i);           //total allocation
   y=y+ta(i);
end
mprintf("\n\nUsing alternate justifiable expenditure
    method method.");
mprintf("\n\nallocations to each project purpose(
   percent):");
for i=1:3
    pr(i)=ta(i)*100/y;           //total allocation
       percent
mprintf("\n%f",pr(i));
end
```

Scilab code Exa 6.8 EX6 8

```


//example 6.8
//calculate outflow hydrograph
clc;funcprot(0);
//given
```

```
I=[35 55 92 130 160 140];    //inflow(cumec/sec)
x=0.28;K=1.6;                //studied value
t=6;
K=K*24;                      //in hours
co=(-K*x+0.5*t)/(K-K*x+0.5*t);
c1=(K*x+0.5*t)/(K-K*x+0.5*t);
c2=(K-K*x-0.5*t)/(K-K*x+0.5*t);
c=co+c1+c2;
//c=1; which implies (OK)
//from Muskingum equation
O(1)=35;
mprintf("outflow hydrograph:\n%f",O(1));
for i=2:6
    p1(i)=co*I(i);
    p2(i)=c1*I(i-1);
    p3(i)=c2*O(i-1);
    O(i)=p1(i)+p2(i)+p3(i);
    O(i)=round(O(i)*100)/100;
   mprintf("\n%f",O(i));
end
```

Scilab code Exa 6.9 EX6 9

```


//example 6.9
//calculate minimum storage to meet the demand
clc;funcprot(0);
//given
md=[50 75 80 85 130 120 25 25 40 45 50 60];        //
   monthly demand
e=[6 8 13 17 22 22 14 11 13 12 7 5];               //
   evaporation
r=[1 0 0 0 0 19 43 39 22 6 2 1];                   //
   rainfall
```

```
in=[50 40 30 25 20 30 200 225 150 90 70 60];        //
    monthly inflow
A=30;                                                //
    area of reservior
Cr=0.4;                                              //
    run-off coefficient
for i=1:12
    er(i)=0.4*r(i);                                  //
        effective rainfall
    ni(i)=er(i)-e(i);                                //
        net inflow
    niv(i)=ni(i)*0.01*A;                             //
        net inflow volume
    nd(i)=md(i)-niv(i);                              //
        net demand
end
cnd(1)=nd(1);                                       //
    cumulative demand
ci(1)=in(1);                                        //
    cumulative inflow
for i=2:12
    cnd(i)=cnd(i-1)+nd(i);
    ci(i)=ci(i-1)+in(i);
end
mprintf("Excess demand:");
for i=1:12
    ed(i)=cnd(i)-ci(i);                             //
        excess demand
    if ed(i)<0 then
        es(i)=ed(i);                                //
            excess supply
        ed(i)=0;
    end
    mprintf("\n%f",ed(i));
end
mprintf("\nminimum storage required=Maximum of
    excess demand=%f Mm^3.",ed(6));
```

Scilab code Exa 6.10 EX6 10

```


//example 6.10
//calculate
//minimum capacity of reservior
//the initial storage storage required to maintain uniform demand
clc;funcprot(0);
//given
in=[2.83 4.25 5.66 18.4 22.64 22.64 19.81 8.49 7.1 7.1 5.66 5.66];      //inflow(x10^5)
s=0;
for i=1:12
    s=s+in(i);
end
avd=s/12;                                           //average demand(x10^5)
s=0;t=0;
for i=1:12
    e(i)=avd-in(i);
    if e(i)<0 then
        S(i)=-e(i);                                  //surplus(x10^5)
        s=s+S(i);
    else
        D(i)=e(i);                                   //Deficit(x10^5)
        t=t+D(i);
    end
```

```
end

d=(s-(t-D(1)-D(2)-D(3)));
s=s;

mprintf("minimum capacity of reservior=%fD+5 cumec."
   ,s);
mprintf("\nstorage required to maintain uniform
   demand=%fD+5 cumec",d);
```

Chapter 8

GRAVITY DAMS

Scilab code Exa 8.1 EX8 1

```


//example 8.1
//calculate forces induced due to earthquake
clc;funcprot(0);
//given
H=100;                //heigth of dam
wb=70;                //width of base of dam
wt=7;                 //width of top of dam
l=1;                  //length of dam
hw=98;                //heigth of water in dam
hsu=90;               //heigth of slope on downstream
    side
s=1/0.7;              //slope on downstream side
gammad=24;            //unit weigth of dam
gammaw=9.81;          //unit weigth of water
E=2.05D7;             //modulus of elasticity

//(a) inertial forces and moments
alpha0=0.05;          //from table 8.1
alphah=2*alpha0;
```

```
//at 10m from top
F10=integrate('25.2-0.25*y','y',0,10);
M10=integrate('25.2*(1-0.01*y)*(10-y)','y',0,10);
//at 100m below top
F100=F10+integrate('0.15*(1-0.01*y)*16.8*y','y'
    ,10,100);
M100=M10+90*F10+integrate('0.15*(1-0.01*y)*16.8*y
    *(100-y)','y',10,100);
mprintf("Inertial forces:\nAt 10m from top: F=%f kn;
    M=%ikn-m\nAt 100m from top: F=%f kn;M=%ikn-m.",
    F10,M10,F100,M100);

//(b) hydrodynamic pressure and moment
//at 10m from top
y=8;
W10=1680;
alphah=F10/W10;
Cm=0.735;
Cy=(Cm/2)*((y*(2-y/hw)/hw)+(y*(2-y/hw)/hw)^0.5);
p=Cy*alphah*gammaw*hw;
P10=0.726*p*y;
Mp10=0.299*p*y^2;
P10=round(P10*100)/100;
Mp10=round(Mp10*100)/100;
//at 100m from top
y=98;
W100=84840;
alphah=F100/W100;
Cm=0.735;
Cy=(Cm/2)*(y*(2-y/hw)/hw+(y*(2-y/hw)/hw)^0.5);
p=Cy*alphah*gammaw*hw;
P100=0.726*p*y;
Mp100=0.299*p*y^2;
mprintf("\nHydrodynamic forces:\nAt 10m from top: F=
    %f kn;M=%fkn-m\nAt 100m from top: F=%i kn;M=%ikn-
    m.",P10,Mp10,P100,Mp100);
```

Scilab code Exa 8.2 EX8 2

```


//example 8.2
//calculate forces induced due to earthquake by
   responce spectrum method
clc;funcprot(0);
//given
H=100;                  //heigth of dam
wb=70;                  //width of base of dam
wt=7;                   //width of top of dam
l=1;                    //length of dam
hw=98;                  //heigth of water in dam
hsu=90;                 //heigth of slope on downstream
   side
s=1/0.7;                //slope on downstream side
gammad=24;              //unit weigth of dam
gammaw=9.81;            //unit weigth of water
E=2.05D7;               //modulus of elasticity
beta=1;
I=2;
Fo=0.25;               //from table 8.2
                      //t=Sa/g;
t=0.19;               //from fig. 8.4
alphah=beta*I*Fo*t;
T=5.55*H^2/wb*(gammad/(gammaw*E))^0.5;
//(a) Base shear
W=l*gammad*(wt*H+((hsu/s)*hsu)/2);
Fb=0.6*W*alphah;
mprintf("Base shear=%f KN.",Fb);

//(b) Base moment
hbar=((wt*H^2/2)+((hsu/s)*hsu^2/6))/((wt*H)+(hsu/s)*
```

```
   hsu/2);
Mb=0.9*W*hbar*alphah;
mprintf("\nBase moment=%f KN-m.",Mb);

//(c) shear at 10m from top
Cv=0.08;
F10=Cv*Fb;
F10=round(F10);
mprintf("\nshear at 10m from top=%f KN.",F10);

//(d) Moment at 10m from top
Cm=0.02;
M10=Cm*Mb;
M10=round(M10);
mprintf("\nmoment at 10m from top=%f KN.",M10);
//(e) Hydrodynamic pressure
//at 10m from top
y=8;
W10=1680;
Cm=0.735;
Cy=(Cm/2)*((y*(2-y/hw)/hw)+(y*(2-y/hw)/hw)^0.5);
p=Cy*alphah*gammaw*hw;
P10=0.726*p*y;
Mp10=0.299*p*y^2;
P10=round(P10*100)/100;
Mp10=round(Mp10*100)/100;
//at 100m from top
y=98;
W100=84840;
Cm=0.735;
Cy=(Cm/2)*(y*(2-y/hw)/hw+(y*(2-y/hw)/hw)^0.5);
p=Cy*alphah*gammaw*hw;
P100=0.726*p*y;
Mp100=0.299*p*y^2;
mprintf("\nHydrodynamic forces:\nAt 10m from top: F=
   %f kn;M=%fkn-m\nAt 100m from top: F=%i kn;M=%ikn-
   m.",P10,Mp10,P100,Mp100);
```

Scilab code Exa 8.3 EX8 3

```


//example 8.3
//calculate forces induced due to earthquake
clc;funcprot(0);
//given
H=100;                  //heigth of dam
wb=70;                  //width of base of dam
wt=7;                   //width of top of dam
l=1;                    //length of dam
hw=98;                  //heigth of water in dam
hsu=90;                 //heigth of slope on downstream
    side
s=1/0.7;                //slope on downstream side
gammad=24;              //unit weigth of dam
gammaw=9.81;            //unit weigth of water
E=2.05D7;               //modulus of elasticity
//(a) Seismic coefficient method
alpha0=0.05;            //from table 8.1
alphah=2*alpha0;
alphav=0.75*alphah;
//at 10m from top
F10=integrate('alphav*168*(1-0.01*y)','y',0,10);
//at 100m below top
F100=F10+integrate('alphav*(1-0.01*y)*16.8*y','y'
    ,10,100);
mprintf("Part(a):\nAt 10m from top: F=%f kn\nAt 100m
     from top: F=%f kn.",F10,F100);

//(b)Response spectrum method
beta=1;
I=2;
```

```
Fo=0.25;             //from table 8.2
                     //t=Sa/g;
t=0.19;              //from fig. 8.4
alphah=beta*I*Fo*t;
alphav=0.75*alphah;
//at 10m from top
F10=integrate('alphav*168*(1-0.01*y)','y',0,10);
//at 100m below top
F100=F10+integrate('alphav*(1-0.01*y)*16.8*y','y'
    ,10,100);
F100=round(F100*100)/100;
mprintf("\nPart(b):\nAt 10m from top: F=%f kn\nAt
    100m from top: F=%f kn.",F10,F100);
```

Scilab code Exa 8.4 EX8 4

```


//example 8.4
//calculate hydrodynamic pressure on10m,40m and 100m
     from top
clc;funcprot(0);
//given
H=100;                //heigth of dam
wb=73;                //width of base of dam
wt=7;                 //width of top of dam
l=1;                  //length of dam
hw=98;                //heigth of water in dam
hsu=90;               //heigth of slope on downstream
    side
s=1/0.7;              //slope on downstream side
gammad=24;            //unit weigth of dam
gammaw=9.81;          //unit weigth of water
E=2.05D7;             //modulus of elasticity

```

```
//at 10m from top
y=8;
alphah=0.1;
Cm=0.72;
Cy=(Cm/2)*((y*(2-y/hw)/hw)+(y*(2-y/hw)/hw)^0.5);
p10=Cy*alphah*gammaw*hw;
F10=0.726*p10*y;
Mp10=0.299*p10*y^2;

//at 40m from top
y=38;
alphah=0.1;
Cm=0.72;
Cy=(Cm/2)*((y*(2-y/hw)/hw)+(y*(2-y/hw)/hw)^0.5);
p40=Cy*alphah*gammaw*hw;
F40=0.726*p40*y;
Mp40=0.299*p40*y^2;

//at 100m from top
y=98;
alphah=0.1;
Cm=0.72;
Cy=(Cm/2)*((y*(2-y/hw)/hw)+(y*(2-y/hw)/hw)^0.5);
p100=Cy*alphah*gammaw*hw;
F100=0.726*p100*y;
Mp100=0.299*p100*y^2;
p10=round(p10*1000)/1000;
F10=round(F10*1000)/1000;
Mp10=round(Mp10*10)/10;
p40=round(p40*1000)/1000;
F40=round(F40*1000)/1000;
Mp40=round(Mp40*10)/10;
p100=round(p100*100)/100;
F100=round(F100*1000)/1000;
Mp100=round(Mp100*10)/10;
mprintf("\nHydrodynamic Forces:\nAt 10m from top: P=
    %f KN/square m;F=%f KN;M=%f KN-m.\nAt 40m from
    top: P=%f KN/square m.;F=%f KN;M=%f KN-m.\nAt 100
```

```
m from top: P=%f KN/square m;F=%f KN;M=%f KN-m.",p10,F10,Mp10,p40,F40,Mp40,p100,F100,Mp100);

//vertical component of reservior water on horizontal section
s1=3/60;
Wh=(F100-F40)*s1;
Wh=round(Wh*100)/100;
mprintf("\n\nvertical component of reservior water on horizontal section=%f kN/m.",Wh);
```

Scilab code Exa 8.8 EX8 8

```


//example 8.8
//calculate Heigth of dam when
//no tension is permissible
//factor of safety against slidingis 1.5
clc;funcprot(0);
//given

wb=3;           //width of dam;
miu=0.5;        //coefficient of friction
Sg=2.4;         //specific gravity of masonary
gamma_w=9.81;   //unit weigth of water
c=1;

//when uplift is considered
//when no tension is permissible then e=wb/6;

p1=wb*Sg*gamma_w;
p2=c*wb*gamma_w/2;
p3=p1-p2;
p4=p1*wb/2-p2*2;
```

```
p5=gamma_w/6;
d1=p4/p3; d2=p5/p3;
d3=1.5-d1;
H=((0.5-d3)/d2)^0.5;
H=round(H*100)/100;
mprintf("when uplift is considered:")
mprintf("\nHeigth of dam when no tension is
    permissible=%f m.",H);
H=p3*0.5/(1.5*p5*3);
mprintf("\nHeigth of dam when factor of safety
    against sliding is 1.5=%f m.",H);

//when uplift is not considered
p1=wb*Sg*gamma_w;
p4=p1*wb/2;
p5=gamma_w/6;
d1=p4/p1;
d2=p5/p1;
H=(0.5/d2)^0.5;
H=round(H*100)/100;
mprintf("\n\nwhen uplift is not considered:")
mprintf("\nHeigth of dam when no tension is
    permissible=%f m.",H);
H=p1*0.5/(1.5*p5*3);
mprintf("\nHeigth of dam when factor of safety
    against sliding is 1.5=%f m.",H);
```

Scilab code Exa 8.9 EX8 9

```


//example 8.9
//calculate streeses at heel and toe of dam
clc;funcprot(0);
//given
```

```
c=1;
hw=6;              //heigth of water in reservior
Bt=1.5;               //width of top of dam
H=6;              //heigth of the dam
wb=4.5;              //width of base of dam
Sg=2.4;              //specific gravity of masonary
gamma_w=9.81;      //weigth density of water

W1=Bt*gamma_w*Sg*H;
W2=gamma_w*Sg*H*(wb-Bt)/2;
L1=(wb-Bt)+(Bt/2);
L2=(2*(wb-Bt))/3,
M1=W1*L1,M2=W2*L2,

//Reaervior empty
SumW=W1+W2;
SumM=M1+M2;
x=SumM/SumW;
e=wb/2-x;
pnt=(SumW/wb)*(1+(6*e/wb));
pnh=(SumW/wb)*(1-(6*e/wb));
pnt=round(pnt*10)/10;
pnh=round(pnh*10)/10;
mprintf("Reservior empty:");
mprintf(" \nNormal stress at toe=%f kN/square.m.",
   pnt);
mprintf("\nNormal stress at heel=%f kN/square.m.",
   pnh);

//Reservior full
W3=gamma_w*H^2/2;
U=gamma_w*H*c*wb/2;
SumV=SumW-U;
L3=hw/3;
L4=2*wb/3;                //lever arm
M3=W3*L3;
M4=U*L4;                  //moment about toe
SumM1=SumM-M4-M3;
```

```
x=SumM1/SumV;
e=wb/2-x;
pnt=(SumV/wb)*(1+(6*e/wb));
pnh=(SumV/wb)*(1-(6*e/wb));
pnt=round(pnt*10)/10;
pnh=round(pnh*10)/10;
mprintf("\n\nReservior full:");
mprintf(" \nNormal stress at toe=%f kN/square.m.",
    pnt);
mprintf("\nNormal stress at heel=%f kN/square.m.",
    pnh);
```

Scilab code Exa 8.10 EX8 10

```


//example 8.10
//calculate width of base if no tension is to
    develop
//check the stability
clc;funcprot(0);
//given
c=1;
hw=6;                 //heigth of water in reservior
Bt=1.5;               //width of top of dam
H=6;                  //heigth of the dam
gamma_m=20;           //unit weigth of masonary
gamma_w=9.81;         //weigth density of water
f=1800;               //compressive strength
miu=0.6;              //coefficient of friction

//to develop no tension e=b/6;x=b/3.
//hence on solving the relations we get

P=poly([-39.074 2.944 1],'b','c');    //equation is
```

```
    written wrong in book
wb=roots(P);                                //sign of coefficient is 2.944 is not taken correctly in book


//roots are 4.94 and -7.89
//since negative value cannot be taken

wb=4.94;
mprintf("Neglecting the negative value.\nWidth of base is=4.94 m.");
W1=Bt*gamma_m*H;
W2=gamma_m*H*(wb-Bt)/2;
L1=(wb-Bt)+(Bt/2);
L2=(2*(wb-Bt))/3;
M1=W1*L1,
M2=W2*L2;
U=gamma_w*H*c*wb/2;
L4=2*wb/3;
M4=U*L4;
W3=gamma_w*H^2/2;
L3=hw/3;
M3=W3*L3;
SumW=W1+W2-U;
SumM=M1+M2-M4-M3;
pn=2*SumW/wb;
pn=round(pn*10)/10;
mprintf("\nMaximum stress=%f kN/square.m.",pn);
mprintf("\nDam is safe against compression");
FOS=miu*SumW/W3;
FOS=round(FOS*100)/100;
mprintf("\nFactor of safety against sliding=%f. <1",FOS);
mprintf("\nDam is unsafe against sliding.");
```

Scilab code Exa 8.11 EX8 11

```


//example 8.11
//calculate width of base if no tension is to
   develop
//check the stability if uplift is neglected
clc;funcprot(0);
//given
c=1;
hw=6;                  //heigth of water in reservior
Bt=1.5;                //width of top of dam
H=6;                  //heigth of the dam
gamma_m=20;           //unit weigth of masonary
gamma_w=9.81;        //weigth density of water
f=1800;              //compressive strength
miu=0.6;             //coefficient of friction

//to develop no tension e=b/6;x=b/3.
//hence on solving the relations we get

P=poly([-19.908 1.5 1],'b','c')
wb=roots(P);

//roots are 3.774 and -5.27
//since negative value cannot be taken

wb=3.77;
mprintf("Neglecting the negative value.\nWidth of
   base is=3.77 m.");

W1=Bt*gamma_m*H;
W2=gamma_m*H*(wb-Bt)/2;
```

```
L1=(wb-Bt)+(Bt/2);
L2=(2*(wb-Bt))/3;
M1=W1*L1,
M2=W2*L2;
W3=gamma_w*H^2/2;
L3=hw/3;
M3=W3*L3;
SumW=W1+W2;
SumM=M1+M2-M3;
pn=2*SumW/wb;
pn=round(pn*10)/10;
mprintf("\nMaximum stress=%f kN/square.m.",pn);
mprintf("\nDam is safe against compression");

FOS=miu*SumW/W3;
FOS=round(FOS*1000)/1000;
mprintf("\nFactor of safety against sliding=%f. > 1"
    ,FOS);
mprintf("\nDam is safe against sliding.");
```

Scilab code Exa 8.12 EX8 12

```


//example8.12
// calculate maximum permissible heigth of shutter
    so that no tension develops
clc;funcprot(0);
//given
Bt=3;               //width of top of dam
H=12;             //heigth of the dam
wb=9;             //width of base of dam
gamma_m=21;             //unit weigth of masonary
gamma_w=9.81;     //weigth density of water

```

```
W1=Bt*gamma_m*H;
W2=gamma_m*H*(wb-Bt)/2;

//taking moment about a point on base at 3m from toe
L1=3+Bt/2;
L2=(2*(wb-Bt)/3)-3;
M1=W1*L1,
M2=W2*L2;
M=M1+M2;

//net moment about this point should be zero for
   equilibrium
s=(M*6/gamma_w)^(1/3)-12;
s=round(s*100)/100;
mprintf("maximum permissible heigth of shutter=%f m.
   ",s);
```

Scilab code Exa 8.13 EX8 13

```


//example 8.13
//calculate hydrodynamic earthquake pressure
//moment at 50m below water surface
clc;funcprot(0);
//given
c=1;
H=100;            //heigth of dam
hw=100;           //heigth of water in reservior
FB=1;             //free board
s=0.15;           //slope of upstream face
gamma_w=9.81;     //unit weigth of water
alphah=0.1;

theta=atan(s);
```

```
y=50;
Cm=0.735*(1-(theta*2/%pi));
Cy=(Cm/2)*((y*(2-y/hw)/hw)+(y*(2-y/hw)/hw)^0.5);
pe=Cy*alphah*gamma_w*hw;
F=0.726*pe*y;
M=0.299*pe*y^2;
pe=round(pe*1000)/1000;
F=round(F*10)/10;
M=round(M*10)/10;
mprintf("hydrodynamic earthquake pressure=%f kN/
    square.m\nshear=%f kN/m.\nMoment=%f kN-m/m.",pe,F
    ,M);
```

Scilab code Exa 8.14 EX8 14

```


//example 8.14
//check stability
//calculate stresses at  toe and heel
clc;funcprot(0);
//given
c=1;
H=10;                  //heigth of dam
hw=10;                 //heigth of water in reservior
wb=8.25;               //bottom width
Bt=1;                  //top width
Hs1=0.1;               //slope on upstream side
gamma_w=9.81;          //unit weigth of water
gamma_m=22.4;          //unit weigth of masonary
f=1400;                //permissible shear stress at
    joint
miu=0.75;              //coefficient of friction
fi=atan(0.625);
theta=atan(0.1);
```

```
W1=Bt*H*gamma_m;
W2=H*H*Hs1*gamma_m/2;
W3=H*6.25*gamma_m/2;
W4=hw*gamma_w*H*Hs1/2;
P=gamma_w*hw^2/2;
U=wb*gamma_w*hw*c/2;
SumV=W1+W2+W3+W4-U;
L3=2*(wb-(Hs1*H)-Bt)/3;
L1=(wb-(Hs1*H)-Bt)+Bt/2;
L2=(wb-(Hs1*H)-Bt)+Bt+(Hs1*H/3);
L4=(wb-(Hs1*H)-Bt)+Bt+(2*Hs1*H/3);
L5=2*wb/3;L6=hw/3;
M1=W1*L1;M2=W2*L2;M3=W3*L3;M4=W4*L4;
M5=U*L5;M6=P*L6;
SumM=M1+M2+M3+M4-M5-M6;
Mplus=M1+M2+M3+M4;
Mminus=M5+M6;
FOS=miu*SumV/P;
SFF=(miu*SumV+wb*1400)/P;
FOO=Mplus/Mminus;
FOS=round(FOS*100)/100;
SFF=round(SFF*10)/10;
FOO=round(FOO*100)/100;
mprintf("Factor of safety against sliding=%f. >1 ",
    FOS);
mprintf("\nShear friction factor=%f.",SFF);
mprintf("\nFactor of safety against overturning=%f.
    <1.5",FOO);
mprintf("\nDam is unsafe against overturning");

x=SumM/SumV;
e=wb/2-x;
p=hw*gamma_w;
pnt=(SumV/wb)*(1+(6*e/wb));        //calculation is
    done wrong in book;value of b is not taken
    correctly
pnh=(SumV/wb)*(1-(6*e/wb));
```

```
sigmat=pnt*sec(fi)^2;
sigmah=pnh*sec(theta)^2-p*tan(theta)^2;
taut=pnt*tan(fi);
tauh=-(pnh-p)*tan(theta);
pnt=round(pnt*10)/10;
pnh=round(pnh*10)/10;
sigmat=round(sigmat*10)/10;
sigmah=round(sigmah*10)/10;
taut=round(taut*10)/10;
tauh=round(tauh*10)/10;
mprintf("\n\nNormal stress at toe=%f kN/square.m.",
   pnt);
mprintf("\nNormal stress at heel=%f kN/square.m.",
   pnh);
mprintf("\nPrincipal stress at toe=%f kN/square.m.",
   sigmat);
mprintf("\nPrincipal stress at heel=%f kN/square.m."
   ,sigmah);
mprintf("\nShear stress at toe=%f kN/square.m.",taut
   );
mprintf("\nShear stress at heel=%f kN/square.m.",
   tauh);
```

Scilab code Exa 8.15 EX8 15

```


//example 8.15
//Check the stability and determine sliding factor
   and shear factor
clc; funcprot(0);
//Given
c=1;
miu=0.75;          //coefficient of friction
H=90;              //heigth of dam
```

```
wb=73.1;              //width of base
Bt=7;                 //width of top of dam
hw=89;                //heigth of water in reservior
Hs1=28;               //heigth of slope on upstream side
Hs2=83;               //heigth of slope on downstream side
Cm=0.735;
alphah=0.1;
gamma_m=23.5;          //unit weigth of concrete
gamma_w=9.81;          //unit weigth of water
theta=atan(8/28);
fi=atan(0.7);
//self weigth of dam
W1=(Hs1*8*gamma_m)/2,
W2=(Bt*H*gamma_m),
W3=(Hs2^2*0.7*gamma_m)/2,
//weigth of superimposed water
W4=(Hs1*8*gamma_w)/2,
W5=(hw-Hs1)*8*gamma_w,
U=hw*wb*2*gamma_w/6;                    //uplift force
wp=hw^2*gamma_w/2;                      //water pressure
hp=0.726*Cm*alphah*gamma_w*hw^2;        //hydrodynamic pressure
Mhp=0.299*Cm*alphah*gamma_w*hw^3;       //moment due to hydrodynamic pressure
//inertial load due to horizontal acceleration
I1=W2/10;
I2=W3/10;
I3=W1/10;
SumV=W1+W2+W3+W4+W5-U;
SumH=wp+hp+I1+I2+I3;
L1=(wb-8)+8/3,
L2=(0.7*Hs2)+(Bt/2),
L3=(2*Hs2*0.7)/3,
L4=(wb-8)+(2*8)/3,
L5=(wb-8)+(8/2),
```

```
L6=hw/3;
L7=2*wb/3;
M1=W1*L1,M2=W2*L2,M3=W3*L3,M4=W4*L4;
M5=W5*L5;
M6=wp*L6;
M7=U*L7;
M8=I1*45;
M9=I2*83/3;
M10=I3*28/3;
Mplus=M1+M2+M3+M4+M5;
Mminus=M6+M7+M8+M9+M10+Mhp;
SumM=Mplus-Mminus;
x=SumM/SumV;
e=wb/2-x;
pnt=(SumV/wb)*(1+(6*e/wb));
pnh=(SumV/wb)*(1-(6*e/wb));
sigmat=pnt*sec(fi)^2;
p=hw*gamma_w;
pe=Cm*alphah*gamma_w*hw;
sigmah=pnh*sec(theta)^2-(p+pe)*tan(theta)^2;
taut=pnt*tan(fi);
tauh=-(-pnh-(p+pe))*tan(theta);
mprintf("Normal stress at toe=%i kN/square.m.",pnt);
mprintf("\nNormal stress at heel=%i kN/square.m.",
   pnh);
mprintf("\nPrincipal stress at toe=%i kN/square.m.",
   sigmat);
mprintf("\nPrincipal stress at heel=%i kN/square.m."
   ,sigmah);
mprintf("\nShear stress at toe=%i kN/square.m.",taut
   );
mprintf("\nShear stress at heel=%i kN/square.m.",
   tauh);

FOS=miu*SumV/SumH;
SFF=(miu*SumV+wb*1400)/SumH;
FOO=Mplus/Mminus;
Ffi=1.2;Fc=2.4;
```

```
F=(miu*SumV/Ffi+1400*wb/Fc)/SumH;
FOS=round(FOS*100)/100;
F=round(F*100)/100;
SFF=round(SFF*100)/100;
FOO=round(FOO*100)/100;
mprintf("\n\nFactor of safety against sliding as per IS:6512-1972=%f. <1.5",FOS);
mprintf("\nFactor of safety against sliding as per IS:6512-1984=%f. >1",F);
mprintf("\nShear friction factor=%f. <6",SFF);
mprintf("\nFactor of safety against overturning=%f. <1.5",FOO);
mprintf("\n\nDam is unsafe for given loading conditions");
```

Scilab code Exa 8.16 EX8 16

```


//example 8.16
//Check the stability and determine principal and shear stress at toe and heel
clc; funcprot(0);
//Given
c=1;
miu=0.7;           //coefficient of friction
H=70;               //heigth of dam
ht=0;               //heigth of tail water
Lf=6.5;            //location of foundation gallery from heel
wb=52.5;            //width of base
Bt=7;               //width of top of dam
hw=70;              //heigth of water in reservior
Hs1=35;             //heigth of slope on upstream side
```

```
Hs2=60;             //heigth of slope on downstream
   side
gamma_m=24;         //unit weigth of concrete
gamma_w=9.81;         //unit weigth of water
theta=atan(0.1);
fi=atan(0.7);
//self weigth of dam
W1=(Hs1*3.5*gamma_m)/2,
W2=(Bt*H*gamma_m),
W3=(Hs2^2*0.7*gamma_m)/2,
//weigth of superimposed water
W4=(Hs1*3.5*gamma_w)/2,
W5=(hw-Hs1)*3.5*gamma_w,
wp=hw^2*gamma_w/2;                   //water
   pressure
Pt=gamma_w*ht,
Ph=gamma_w*hw,
Pg=(ht+(hw-ht)/3)*gamma_w,
U=(Pt*(wb-Lf))+(Pg*Lf)+((Ph-Pg)*Lf/2)+((Pg-Pt)*(wb-
   Lf)/2)*c,
l1=(wb-Lf)/2,l2=(2*(wb-Lf))/3,l3=(wb-Lf)+(Lf/2),l4=(
   wb-Lf)+((2*Lf)/3),
L7=(((Pt*(wb-Lf))*l1)+((Pg-Pt)*(wb-Lf)*l2/2)+((Pg*Lf
   )*l3)+((Ph-Pg)*Lf*l4/2))/U,
L1=(wb-3.5)+3.5/3,
L2=(0.7*Hs2)+(Bt/2),
L3=(2*Hs2*0.7)/3,
L4=(wb-3.5)+(2*3.5)/3,
L5=(wb-3.5)+(3.5/2),
L6=hw/3;
M1=W1*L1,M2=W2*L2,M3=W3*L3,M4=W4*L4;
M5=W5*L5;
M6=wp*L6;
M7=U*L7;
SumV1=W1+W2+W3;
SumM1=M1+M2+M3;
SumV2=SumV1+W4+W5;
SumM2=SumM1+M4+M5-M6;
```

```
SumV3=SumV2-U;
SumM3=SumM2-M7;
Mplus=1547377;
Mminus=870421;
SumH=wp;

//case 1. Reservior empty
x=SumM1/SumV1;
e=wb/2-x;
pnt=(SumV1/wb)*(1+(6*e/wb));
pnh=(SumV1/wb)*(1-(6*e/wb));
sigmat=pnt*sec(fi)^2;
sigmah=pnh*sec(theta)^2;
taut=pnt*tan(fi);
tauh=pnh*tan(theta);
pnt=round(pnt*10)/10;
pnh=round(pnh*10)/10;
sigmat=round(sigmat*10)/10;
sigmah=round(sigmah*10)/10;
taut=round(taut*10)/10;
tauh=round(tauh*10)/10;
mprintf("case 1. Reservior empty:");
mprintf("\nNormal stress at toe=%f kN/square.m.",pnt
   );
mprintf("\nNormal stress at heel=%f kN/square.m.",
   pnh);
mprintf("\nPrincipal stress at toe=%f kN/square.m.",
   sigmat);
mprintf("\nPrincipal stress at heel=%f kN/square.m."
   ,sigmah);
mprintf("\nShear stress at toe=%f kN/square.m.",taut
   );
mprintf("\nShear stress at heel=%f kN/square.m.",
   tauh);

//case2. reservior full without uplift
x=SumM2/SumV2;
e=wb/2-x;
```

```
p=hw*gamma_w;
pnt=(SumV2/wb)*(1+(6*e/wb));
pnh=(SumV2/wb)*(1-(6*e/wb));
sigmat=pnt*sec(fi)^2;
sigmah=pnh*sec(theta)^2-p*tan(theta)^2;
taut=pnt*tan(fi);
tauh=-(pnh-p)*tan(theta);
pnt=round(pnt*10)/10;
pnh=round(pnh*10)/10;
sigmat=round(sigmat*10)/10;
sigmah=round(sigmah*10)/10;
taut=round(taut*10)/10;
tauh=round(tauh*10)/10;
mprintf("\n\ncase 2. reservior full without uplift:");
mprintf("\nNormal stress at toe=%f kN/square.m.",pnt);
mprintf("\nNormal stress at heel=%f kN/square.m.",pnh);
mprintf("\nPrincipal stress at toe=%f kN/square.m.",sigmat);
mprintf("\nPrincipal stress at heel=%f kN/square.m.",sigmah);
mprintf("\nShear stress at toe=%f kN/square.m.",taut);
mprintf("\nShear stress at heel=%f kN/square.m.",tauh);

//case3. reservior full with uplift
x=SumM3/SumV3;
e=wb/2-x;
p=hw*gamma_w;
pnt=(SumV3/wb)*(1+(6*e/wb));
pnh=(SumV3/wb)*(1-(6*e/wb));
sigmat=pnt*sec(fi)^2;
sigmah=pnh*sec(theta)^2-p*tan(theta)^2;
taut=pnt*tan(fi);
tauh=-(pnh-p)*tan(theta);
```

```
pnt=round(pnt);
pnh=round(pnh);
sigmat=round(sigmat);
sigmah=round(sigmah);
taut=round(taut);
tauh=round(tauh);
mprintf("\n\ncase 3. reservior full with uplift:");
mprintf("\nNormal stress at toe=%f kN/square.m.",pnt);
mprintf("\nNormal stress at heel=%f kN/square.m.",pnh);
mprintf("\nPrincipal stress at toe=%f kN/square.m.",sigmat);
mprintf("\nPrincipal stress at heel=%f kN/square.m.",sigmah);
mprintf("\nShear stress at toe=%f kN/square.m.",taut);
mprintf("\nShear stress at heel=%f kN/square.m.",tauh);

FOS=miu*SumV3/SumH;
SFF=(miu*SumV3+wb*1400)/SumH;
FOO=Mplus/Mminus;
Ffi=1.5;Fc=3.6;
F=(miu*SumV3/Ffi+1400*wb/Fc)/SumH;
FOS=round(FOS*1000)/1000;
SFF=round(SFF*100)/100;
FOO=round(FOO*100)/100;
F=round(F*1000)/1000;
mprintf("\n\nFactor of safety against sliding=%f.",FOS);
mprintf("\nShear friction factor=%f.",SFF);
mprintf("\nFactor of safety against overturning=%f.",FOO);
mprintf("\nFactor of safety for load combination B=%f. > 1",F);
mprintf("\nDam is safe ");
```

```
//Case4.considering seismic forces
Cm=0.712;
alphah=0.1;
alphav=0.08;
hp=0.726*Cm*alphah*gamma_w*hw^2;      //hydrodynamic pressure
Mhp=0.299*Cm*alphah*gamma_w*hw^3;     //moment due to hydrodynamic pressure
//inertial load due to horizontal acceleration
I1=W2/10;
I2=W3/10;
I3=W1/10;
v=SumV1*alphav;
Mv=116444;
SumV4=SumV3-v;
SumH1=SumH+I1+I2+I3+hp;
M8=I1*35;
M9=I2*20;
M10=I3*35/3;
Mminus1=1161849;
SumM4=SumM3-M8-M9-M10-Mhp-Mv;

x=SumM4/SumV4;
e=wb/2-x;
p=hw*gamma_w;
pe=Cm*alphah*gamma_w*hw;
pnt=(SumV4/wb)*(1+(6*e/wb));
pnh=(SumV4/wb)*(1-(6*e/wb));
sigmat=pnt*sec(fi)^2;
sigmah=pnh*sec(theta)^2-(p+pe)*tan(theta)^2;
taut=pnt*tan(fi);
tauh=(-pnh+(p+pe))*tan(theta);
pnt=round(pnt);
pnh=round(pnh);
sigmat=round(sigmat);
sigmah=round(sigmah);
taut=round(taut);
tauh=round(tauh);
```

```
mprintf("\n\ncase 4.considering seismic forces");
mprintf("\nNormal stress at toe=%f kN/square.m.",pnt);
mprintf("\nNormal stress at heel=%f kN/square.m.",pnh);
mprintf("\nPrincipal stress at toe=%f kN/square.m.",sigmat);
mprintf("\nPrincipal stress at heel=%f kN/square.m.",sigmah);
mprintf("\nShear stress at toe=%f kN/square.m.",taut);
mprintf("\nShear stress at heel=%f kN/square.m.",tauh);   //answer is wrong in book

FOS=miu*SumV4/SumH1;
SFF=(miu*SumV4+wb*1400)/SumH1;
FOO=Mplus/Mminus1;
Ffi=1.2;Fc=2.7;
F=(miu*SumV4/Ffi+1400*wb/Fc)/SumH1;
FOS=round(FOS*1000)/1000;
SFF=round(SFF*100)/100;
FOO=round(FOO*100)/100;
F=round(F*100)/100;
mprintf("\n\nFactor of safety against sliding=%f.",FOS);
mprintf("\nShear friction factor=%f.",SFF);
mprintf("\nFactor of safety against overturning=%f.",FOO);
mprintf("\nFactor of safety for load combination E=%f. > 1",F);
mprintf("\nDam is safe ");
```

Scilab code Exa 8.17 EX8 17

```
//example 8.17
//design practical profile of gravity dam
clc;funcprot(0);
//given
c=1;
rlb=1450;            //R.L of base of dam
rlw=1480.5;          //R.L of water level
Sg=2.4;              //specific gravity of masonary
gamma_w=9.81;        //unit weigth of water
w=1;                 //heigth of waves
f=1200;              //safe compressive stress for masonary
FB=1.5*w;
rlt=FB+rlw;          //R.L of top of dam
H=rlt-rlb;           //heigth of dam
LH=f/(gamma_w*(Sg+1))
LH=round(LH*100)/100;
mprintf("Heigth of dam=%f m.",H);
mprintf("\nlimiting heigth of dam=%f m.",LH);
mprintf("\nDam is low gravity dam");
hw=rlw-rlb;
//keep top width,a=4.5.
a=4.5;
P=hw/(Sg^0.5);
P=round(P*10)/10;
mprintf("\nBase width of elementary profile=%f m.",P);
uo=a/16;
wb=uo+P;
wb=round(wb);
mprintf("\nBase width=%f m.",wb);
D=2*a*(Sg^0.5);
D=round(D);
mprintf("\nDistance upto which u/s slope is vertical from water level=%f m.",D);
```

Scilab code Exa 8.18 EX8 18

```
//example 8.18
//determine if dam is safe against sliding
clc;funcprot(0);
//given
hw=97;              //heigth of water in reservior
Bt=7;               //width of top of dam
H=100;              //heigth of the dam
Hs2=90;             //heigth of slope on downstream
   side
wb=75;              //width of base of dam
miu=0.75;           //coefficient of friction
gamma_d=2.4;        //weigth density of concrete
gamma_w=1000;       //weigth density of water

P=gamma_w*hw^2/(2*1000);
W1=Bt*gamma_d*H;
W2=(wb-Bt)*Hs2*gamma_d/2;
W=W1+W2;
FOS=miu*W/P;
FOS=round(FOS*1000)/1000;
mprintf("Factor of safety against sliding=%f.",FOS);
mprintf("\nDam is safe against sliding");
```

Scilab code Exa 8.19 EX8 19

```
//example 8.19
```

```
//calculate
//Factor of safety against overturning
//Factor of safety against sliding
//Shear friction factor
clc;funcprot(0);
//given
c=1;
H=10;                  //heigth of dam
hw=10;                 //heigth of water in reservior
wb=8.25;               //bottom width
Bt=1;                  //top width
Hs1=0.1;               //slope on upstream side
gamma_w=9.81;          //unit weigth of water
gamma_m=22.4;          //unit weigth of masonary
f=1400;                //permissible shear stress at joint
miu=0.75;              //coefficient of friction
fi=atan(0.625);
theta=atan(0.1);

W1=Bt*H*gamma_m;
W2=H*H*Hs1*gamma_m/2;
W3=H*6.25*gamma_m/2;
W4=hw*gamma_w*H*Hs1/2;
P=gamma_w*hw^2/2;
U=wb*gamma_w*hw*c/2;
SumV=W1+W2+W3+W4-U;
L3=2*(wb-(Hs1*H)-Bt)/3;
L1=(wb-(Hs1*H)-Bt)+Bt/2;
L2=(wb-(Hs1*H)-Bt)+Bt+(Hs1*H/3);
L4=(wb-(Hs1*H)-Bt)+Bt+(2*Hs1*H/3);
L5=2*wb/3;L6=hw/3;
M1=W1*L1;M2=W2*L2;M3=W3*L3;M4=W4*L4;
M5=U*L5;M6=P*L6;
SumM=M1+M2+M3+M4-M5-M6;
Mplus=M1+M2+M3+M4;
Mminus=M5+M6;
FOS=miu*SumV/P;
```

```
SFF=(miu*SumV+wb*1400)/P;
FOO=Mplus/Mminus;
FOS=round(FOS*100)/100;
SFF=round(SFF*10)/10;
FOO=round(FOO*100)/100;
mprintf("Factor of safety against sliding=%f.",FOS);
mprintf("\nShear friction factor=%f.",SFF);
mprintf("\nFactor of safety against overturning=%f."
    ,FOO);
mprintf("\nDam is unsafe against overturning");
```

Scilab code Exa 8.20 EX8 20

```


//example 8.20
//calculate streeses at heel and toe of dam
clc;funcprot(0);
//given
c=1;
hw=80;              //heigth of water in reservior
Bt=6;               //width of top of dam
H=84;              //heigth of the dam
Hs2=75;             //heigth of slope on downstream
    side
wb=56;              //width of base of dam
Lf=8;                //distance of foundation gallery
    from heel
gamma_d=23.5;        //weigth density of concrete
gamma_w=9.81;       //weigth density of water
ht=6;               //heigth of tail water

W1=Bt*gamma_d*H;
W2=gamma_d*Hs2*(wb-Bt)/2;
W3=gamma_w*ht*4/2;
```

```
W4=gamma_w*hw^2/2;
W5=gamma_w*ht^2/2;
Pt=gamma_w*ht,
Ph=gamma_w*hw,
Pg=(ht+(hw-ht)/3)*gamma_w,
U=(Pt*(wb-Lf))+(Pg*Lf)+((Ph-Pg)*Lf/2)+((Pg-Pt)*(wb-Lf)/2)*c,
l1=(wb-Lf)/2,l2=(2*(wb-Lf))/3,l3=(wb-Lf)+(Lf/2),l4=(wb-Lf)+((2*Lf)/3),
L6=(((Pt*(wb-Lf))*l1)+((Pg-Pt)*(wb-Lf)*l2/2)+((Pg*Lf)*l3)+((Ph-Pg)*Lf*l4/2))/U,
L1=(wb-Bt)+(Bt/2),
L2=(2*(wb-Bt))/3,
L3=4/3;
L4=hw/3;
L5=ht/3;
M1=W1*L1,M2=W2*L2,M3=W3*L3,M4=W4*L4,M5=W5*L5,M6=U*L6;
SumV=W1+W2+W3-U;
SumH=W4-W5;
SumM=M1+M2+M3-M4+M5-M6;
x=SumM/SumV;
e=wb/2-x;
pnt=(SumV/wb)*(1+(6*e/wb));
pnh=(SumV/wb)*(1-(6*e/wb));
pnt=round(pnt*10)/10;
pnh=round(pnh*10)/10;
mprintf("Maximum Normal stress at toe=%f kN/square.m.",pnt);
mprintf("\nMaximum Normal stress at heel=%f kN/square.m.",pnh);
```

Chapter 10

EARTH AND ROCKFILL DAM

Scilab code Exa 10.1 EX10 1

```


//example 10.1
//calculate seepage flow per unit length of dam
clc;funcprot(0);
//given
K=5D-4;             //coefficient of permeability of
    soil
Bt=6;               //width of top of dam
wb=146;             //width of base of dam
H=20;               //heigth of dam
hw=2;               //heigth of water in reservior
hs1=4;              //slope on upstream side
hs2=3;              //slope on downstream side
df=30;              //length of drainage filter

x=wb-df-72+72*0.3;
y=18;
s=(x^2+y^2)^0.5-x;
```

```

x=[0 10 20 30 40 50 60 65.6];
for i=1:8
    y(i)=(4.849*x(i)+5.879)^0.5;
    y(i)=round(y(i)*1000)/1000;
end

mprintf("\nx                              y");
for i=1:8
    mprintf("\n%f             %f",x(i),y(i));
end

sf=K*s*10000;
sf=round(sf*1000)/1000;
mprintf("\nSeepage flow per unit length of dam=%fD-6
    cumecs/metre length of dam.",sf);
```

Scilab code Exa 10.2 EX10 2

```


//example 10.2
//calculate discharge per meter length of dam
clc;funcprot(0);
//given
K=3D-3;              //coefficient of permeability
nd=25;                //number of potential drops
nf=4;                //number of flow channels
lf=40;               //filter length
H=52;                //heigth of dam
fb=2;                //free board

q=K*(H-fb)*nf/(nd*100);
mprintf("Discharge per meter length of dam=%f cumec/
   metre length.",q);
```

Scilab code Exa 10.3 EX10 3

```
//example10.3
//calculate factor of safety for slope
clc;funcprot(0);
//given
x=4;                //given scale
An=14.4;            //area of N rectangle
At=6.4;             //area of T rectangle
Au=4.9;             //area of U rectangle
L=12.6;             //length of arc;
gamma_m=19;         //unit weigth of soil
gamma_w=9.81;       //unit weigth of water
fi=26;              //effective angle(degree)
co=19.5;            //cohesion value

//consider 1m length of dam
SumN=An*x^2*gamma_m;
SumT=At*x^2*gamma_m;
SumU=Au*x^2*gamma_w;
Le=x*L;
F=((Le*co)+(SumN-SumU)*tand(fi))/SumT;
F=round(F*100)/100;
mprintf("Factor of safety for slope=%f.",F);
```

Scilab code Exa 10.4 EX10 4

```
//example 10.4
//check section for:
//Stability of d/s slope against steady seepage
//Sloughing of u/s slope against sudden drawdown
//Stability of the foundation against shear
//Seepage through body of dam
clc;funcprot(0);
//given
//Dimensions
H=20;              //Heigth of dam
Bt=6;              //top width of dam
s1=4;              //u/s slope
s2=3;              //d/s slope
fb=2;              //free board
//Properties of materials of dam
gamma_d=17.27;       //dry density
wc=0.15;             //optimum water content
gamma_s=21.19;       //saturated density
gamma_w=9.81;        //unit weigth of water
wavg=19.62;          //average unit weigth under seepage
theta=26;            //average angle of internal friction(degree)
co=19.13;            //average cohesion
K=5D-4;              //coefficient of permeability
//properties of foundation materials
gamma_f=17.27;      //average unit weigth
cof=47.87;          //average cohesion
fi=8;               //average angle internal friction
t=6;                //thickness of clay
FOSp=1.5;           //permissible factor of safety of slope
PS=8D-6;            //permissible seepage


//(a) Stability of d/s slope against steady seepage
An=302.4;            //area of N diagram
```

```
At=91.2;               //area  of T diagram
Au=98.4;               //area of U diagram
Le=60;                 //length of arc
SumN=An*gamma_s;
SumT=At*gamma_s;
SumU=Au*gamma_w;
F=((Le*co)+(SumN-SumU)*tand(theta))/SumT;
F=round(F*100)/100;
mprintf("Part(a):")
mprintf("\nFactor of safety for slope=%f.",F);
mprintf("\nSafe");

//(b) Sloughing of u/s slope against sudden drawdown
h1=15;
b=80;
P=gamma_s*H^2*tand(45-(theta/2))^2/2+gamma_w*h1^2/2;
sav=P/b;
smax=2*sav;
Ne=(gamma_s-gamma_w)*b*H/2;
R=Ne*tand(theta)+co*b;
fs=R/P;
fs=round(fs*100)/100;
mprintf("\n\nPart(b):")
mprintf("\nFactor of safety w.r.t average shear=%f."
    ,fs);
mprintf("\nSafe");
sr=0.6*H*(gamma_s-gamma_w)*tand(theta)+co;
FS=sr/smax;
FS=round(FS*100)/100;
mprintf("\n\nFactor of safety w.r.t maximum shear=%f
    .",FS);
mprintf("\nSafe");

//(c) Stability of the foundation against shear
h1=26;
h2=6;
gamma_m=(wavg*(h1-h2)+gamma_f*h2)/h1;
l=(gamma_m*h1*tand(fi)+cof)/(gamma_m*h1);
```

```
fi1=atand(l);
P=(h1^2-h2^2)/2*gamma_m*tand(45-(fi1/2))^2;
sav=P/b;
smax=2*sav;
s1=cof+gamma_f*h2*tand(fi);
s2=cof+gamma_m*h1*tand(fi);
as=(s1+s2)/2;
fs=as/sav;
fs=round(fs*100)/100;
mprintf("\n\nPart(c):")
mprintf("\nFactor of safety w.r.t overall shear=%f."
    ,fs);
mprintf("\nSafe");

gamma_av=(wavg*0.6*H+gamma_f*h2)/((0.6*H)+h2);
s=cof+gamma_av*0.6*H*tand(fi);
fs=s/smax;
fs=round(fs*100)/100;
mprintf("\n\nFactor of safety w.r.t overall shear=%f
    .",fs);
mprintf("\nUnsafe");

//(d) Seepage through body of dam
s=2;      //measured
q=K*s*100000/100;
mprintf("\n\nPart(d):")
mprintf("\n Seepage through body of dam=%fD-5 cumecs
    /m length of dam",q);
```

Scilab code Exa 10.5 EX10 5

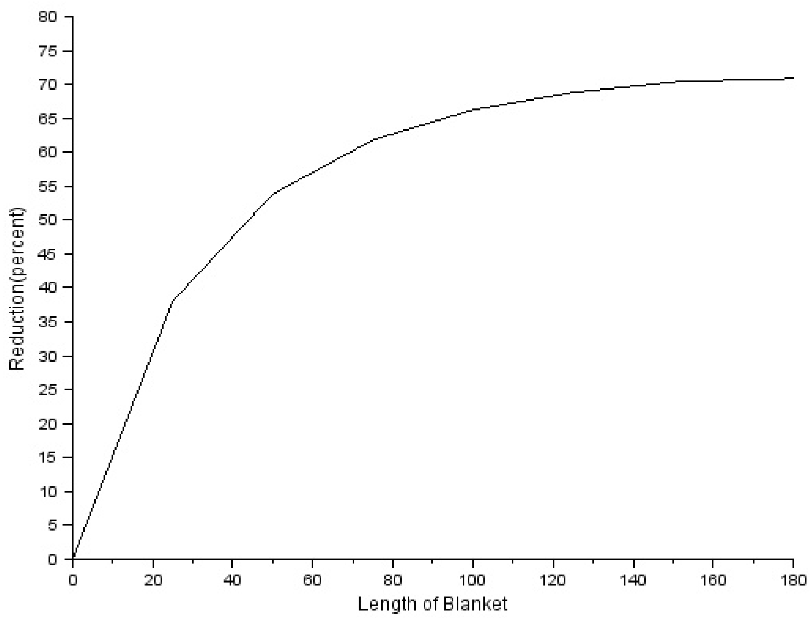

Figure 10.1: EX10 5

```
//example 10.5
//design upstream impervious blanket
clc;funcprot(0);
//given
Zb=1.2;        //thickness of blanket
Zf=8;          //distance of blanket from foundation
kb=0.06;       //coefficient of permeability of blanket material
kf=72;         //coefficient of permeability of foundation soil
Hw=10;          //heigth of water in reservior
Xd=40;

a=(kb/(kf*Zb*Zf))^0.5;
Xo=1.414/a;

//we vary value of x
x=[0 25 50 75 100 125 151.8 300]
for i=1:8
    e=exp(2*a*x(i));
    Xr(i)=(e-1)/(a*(e+1));
    ho(i)=Xr(i)*Hw/(Xr(i)+Xd);
    r(i)=Xr(i)*100/(Xr(i)+Xd);
end
mprintf("\nx                    Xr              ho           reduction q(percent)");
for i=1:8
    mprintf("\n%f         %f         %f         %f",x(i),Xr(i),ho(i),r(i));
end
//graph is plotted between r and x.
//after around 130m length there is only slight increase in head dissipated(ho)
L=130;
mprintf("\nThickness of blanket=%f m",Zb);
mprintf("\nLength of blanket=%i m.",L);
```

Chapter 11

SPILLWAYS

Scilab code Exa 11.1 EX11 1

```
//example 11.1
//calculate compute the dynamic force on curved
    section
clc;funcprot(0);
//given
h=1.2;           //head of water
Cd=2.2;          //coefficient of discharge
rho=1;           //density of water
gamma_w=9.81;    //unit weigth of water

q=Cd*h^1.5;

//applying bernaulli's equation at u/s water surface
     at section A and B
//solving it by error and trial method we get
v1=13.7;v2=14.7;
d1=0.212;d2=0.197;

F1=gamma_w*d1^2*cosd(60)/2;
```

```
F2=gamma_w*d2^2/2;
W=gamma_w*60*2*%pi*3*((d1+d2)/2)/360;
Fx=rho*q*(v2-v1*cosd(60))-F1/2+F2;
Fy=rho*q*(v1*sind(60))+F1*sind(60)+W;
F=(Fx^2+Fy^2)^0.5;
F=round(F*100)/100;
mprintf("Resultant force=%f kN/m.",F);
```

Scilab code Exa 11.2 EX11 2

```


//example 11.2
//calculate discharge over oggy weir
clc;funcprot(0);
//given
C=2.4;          //coefficient of discharge
H=2;            //head
L=100;          //length of spillway
wc=8;           //heigth of weir crest above bottom
g=9.81;         //acceleration due to gravity
h=H+wc;
Q1=C*L*H^(1.5); //neglecting approach velocity and
   end contractions
va=Q1/(h*L);
ha=va^2/(2*g);
Ha=ha+H;
Q=C*L*Ha^1.5;
Q=round(Q*10)/10;
mprintf("discharge over oggy weir=%f cumecs.",Q);
```

Scilab code Exa 11.3 EX11 3

```
//example 11.3
//calculate
//capacity of siphon
//head required in oggy spillway
//length of oggy weir required
clc;funcprot(0);
//given
t=6;            //tail water elevation
h=1;            //heigth of siphon spillway
w=4;            //width of siphon spillway
hw=1.5;         //head water elevation
C=0.6;          //coefficient of discharge
Co=2.25;        //coefficient of discharge of oggy
    spillway
lo=4;           //length of oggy spillway
hc=1.5;         //head on weir crest
g=9.81;         //acceleration due to gravity

//part (a)
Q=C*h*w*(2*g*(t+hw))^0.5;
Q=round(Q*10)/10;
mprintf("capacity of siphon=%f cumecs.",Q);

//part (b)
h1=(Q/(Co*lo))^(2/3);
h1=round(h1*100)/100;
mprintf("\nhead required in oggy spillway=%f m",h1);

//part (c)
L=Q/(Co*(hc)^1.5);
L=round(L*100)/100;
mprintf("\nlength of oggy weir required=%f m.",L);
```

Scilab code Exa 11.4 EX11 4

```


//example 11.4
//calculate number of siphons units required
clc;funcprot(0);
//given
rl=435;           //full reservior level
cl=429.6;         //level of centre of siphon
hfl=435.85;       //high flood level
hfd=600;          //high flood discharge
w=4;              //width of throat
h=2;              //heigth of throat
C=0.65;           //coefficient of discharge
g=9.81;         //acceleration due to gravity

H=hfl-cl;
Q=C*w*h*(2*g*H)^0.5;
n=hfd/Q;
n=round(n*100)/100;
mprintf(" number of siphons units required=%f.\nhence provide 11 siphons units.",n);
```

Scilab code Exa 11.5 EX11 5

```


//example 11.5
//design oggy spillway for concrete gravity dam
clc;funcprot(0);
//given
rbl=250;          //avarage river bed level
rlc=350;          //R.L of spillway crest
s=0.75;           //slope on downstream side
```

```
Q=6500;          //discharge
L=5*9;           //length of spillway
Cd=2.2;          //coefficient of discharge
t=2;             //thickness of each pier

//step 1. computation of design head
H=(Q/(Cd*L))^(2/3);
P=rlc-rbl;

//P/H=6.15,which is <1.33;it is a high overflow spillway

//H+P/H=7.15>1.7; hence discharge coefficient is not affected by downstream apron interface

Kp=0.01;Ka=0.1;N=4;
He=17.5;                    //assumed
Le=L-2*(N*Kp+Ka)*He;
He1=(Q/(Cd*Le))^(2/3);
He1=round(He1*100)/100;
//He1 is almost equal to He
mprintf("crest profile will be designed for Hd=%f m.",He1);

//step 2. determination of d/s profile

//equating the slope of d/s side and derivative of profile equation suggested by WES
x=27.03;
y=0.04372*x^1.85;
mprintf("\n\ndownstream profile:");
x=[1:1:26]
for i=1:26
    y(i)=0.04372*x(i)^1.85;
    y(i)=round(y(i)*1000)/1000;
end
mprintf("\nx          y");
for i=1:26
```

```
    mprintf("\n%i           %f",x(i),y(i));
end
mprintf("\n27.03           19.48");


//step 3. determination of u/s profile
// cosidering equation for vertical u/s face and Hd
   =17.58

mprintf("\n\nupstream profile:");
x=[-0.5 -0.1 -1.5 -2.0 -3.0 -4.0 -4.75];
for i=1:7
    y(i)=0.0633*(x(i)+4.7466)^1.85+2.2151-1.2643*(x(
       i)+4.7466)^0.625;
    y(i)=round(y(i)*1000)/1000;
end
mprintf("\nx                       y");
for i=1:7
    mprintf("\n%f           %f",x(i),y(i));
end

//step 4.design of d/s bucket

R=P/4;
mprintf("\n\nradius of bucket=%i m.",R);
mprintf("\nbucket will subtend angle of 60 degree at
    the centre.");
```

Scilab code Exa 11.6 EX11 6

```


//example 11.6
//design length and depth of stilling basin
clc;funcprot(0);
```

```
//given
q=1;              //discharge of spillway
Cd=0.7;           //coefficient of discharge
h1=10;             //heigth of crest above downstream
   silting basin
g=9.81;             //acceleration due to gravity
Cv=0.9;            //coefficient of velocity

h=(3*q/(2*Cd*(2*g)^0.5))^(2/3);
H=h1+h/2;
vt=(2*g*H)^0.5;
v1=Cv*vt;
y1=q/v1;
F1=v1/(g*y1)^0.5;
//F>1, flow is super-critical
y2=1;
v2=q/y2;
F2=v2/(g*y2)^0.5;             //<1
y2=(y1/2)*((1+8*F1^2)^0.5-1);
de=y2-1;
le=5*(y2-y1);
de=round(de*1000)/1000;
le=round(le*10)/10;
mprintf("stilling basin should be depressed by %f m.
   ",de);
mprintf("\nlength of stilling basin=%f m.",le);
```

Scilab code Exa 11.7 EX11 7

```


//example 11.7
//calculate leading dimension of hydraulic jump
   stilling basin
clc;funcprot(0);
```

```
//given
q=7.83;          //discharge through spillway
w=12.5;          //width of fall
d=2;             //depth of water in downstream
g=9.8;

y1=0.5;
v1=q/y1;
F1=v1/(g*y1)^0.5;

//F>1,flow is super-critical
v2=q/d;
F2=v2/(g*d)^0.5;
y2=(y1/2)*((1+8*F1^2)^0.5-1);
de=y2-d;
le=5*(y2-y1);
de=round(de*100)/100;
le=round(le*10)/10;
mprintf("stilling basin should be depressed by %f m.
   ",de);
mprintf("\nlength of stilling basin=%f m.",le);
```

Scilab code Exa 11.8 EX11 8

```


//example 11.8
//calculate force to be exerted to lift the gate
clc;funcprot(0);
//given
Ag=5*2.5;            //area of gate
miu=0.25;           //coefficient of friction
w=0.5;              //weigth of gate
h=2;                //head of water over crest
g=9.81;             //acceleration due to gravity
```

```
gamma_w=1000;      //unit weigth of water

m=w*g*1000;
F=gamma_w*Ag*h*g/10;
ff=miu*F;
tf=(m+ff)/1000;
mprintf("force to be exerted to lift the gate=%f kN.
   ",tf);
```

Scilab code Exa 11.9 EX11 9

```


//example 11.9
//calculate depth of flow at both end of jumps
clc;funcprot(0);
//given
q=19;          //dischrge through spillway
E=1;           //energy loss

//from energy loss equation;E=(y2-y1)^3/4y2y1; and
   solving it we get
//x=0.5*(-1+(1+294.39*(x-1)^9/64*x^3))
//by trial and error method x=2.806
x=2.806;
y1=4*x/(x-1)^3;
y2=x*y1;
y1=round(y1*1000)/1000;
y2=round(y2*1000)/1000;
mprintf("depth of flow at both end of jumps=%f m and
    %f m. respectively.",y1,y2);
```

Chapter 12

DIVERSION HEADWORKS

Scilab code Exa 12.1 EX12 1

```
//example 12.1
//calculate average hydraulic gradient
//uplift presuures and thickness of floor at 6m, 12m and 18m from u/s
clc;funcprot(0);
//given
rho=2.24;               //relative density of material
gamma_w=9.81;           //unit weigth of water
L=22;                   //total length
lc=(2*6)+L+(2*8);       //length of creep
hg=4/lc;                //hydraulic gradient
mprintf("avearge hydraulic gradient=%f.",hg);
//at 6 m from u/s
x=6;
lg=(6*2)+x;
h1=4*(1-lg/50);         //unbalanced head
up=gamma_w*h1;
t=4*h1/(3*(rho-1));
up=round(up*100)/100;
```

```
t=round(t*100)/100;
mprintf("\n\nuplift at 6 m from u/s=%f kN/square
   metre.",up);
mprintf("\nthickness at 6 m from u/s=%f m.",t);

//at 12 m from u/s
x=12;
lg=(6*2)+x;
h1=4*(1-lg/50);        //unbalanced head
up=gamma_w*h1;
t=4*h1/(3*(rho-1));
up=round(up*100)/100;
t=round(t*100)/100;
mprintf("\n\nuplift at 12 m from u/s=%f kN/square
   metre.",up);
mprintf("\nthickness at 12 m from u/s=%f m.",t);

//at 18m from u/s
x=18;
lg=(6*2)+x;
h1=4*(1-lg/50);        //unbalanced head
up=gamma_w*h1;
t=4*h1/(3*(rho-1));
up=round(up*10)/10;
t=round(t*100)/100;
mprintf("\n\nuplift at 18 m from u/s=%f kN/square
   metre.",up);
mprintf("\nthickness at 18 m from u/s=%f m.",t);
```

Scilab code Exa 12.2 EX12 2

```


//example 12.2
//calculate uplift pressure and exit gradient
```

```
//check whether section is safe against overturning
   and piping
clc;funcprot(0);
//given
b=54;              //width of section
D1D2=16;           //distance between points D1 and D2
D2D3=37;           //distance between points D2 and D3

//first pipe line
//taking data from figure
d=105-97;
b1=0.5;
alpha=b/d;
//from the curves we get
fic1=0.665;
fid1=0.76;
fie1=1;
t=105-104;                          //floor thickness
corec=(fid1-fic1)*100*t/d;               //correction for
    floor thickness
//for pile no. 2
D=104-97;
d=104-97;
bdash=16;
C=19*(D/bdash)^0.5*(d+D)/b;     //correction for pile
    no. 2
fic1=fic1*100+corec+C;          //corrected pressures

//intermedite pipe line
d=105-97;
b1=16.5;
alpha=b/d;
r=b1/b;                  //ratio b1/b
//from the curves we get
fic2=0.52;
fie2=0.725;
fid2=0.615;
corec_c1=(fid2-fic2)*100*t/d;
```

```
corec_e1=(fie2-fid2)*100/d;

//for pile no. 1
C1=C;
d=104-97;
bdash=37;
D=104-95;
C2=19*(D/bdash)^0.5*(d+D)/b;
//correction due to slope
corec_e2=3.3;                    //from table 12.4
//correction is negative due to upwrd slope
l=4;                //horizontal length of slope
corec_c2=corec_e2*l/bdash;

fie2=fie2*100-corec_e1-corec_e2;
fic2=fic2*100+corec_c1+C2-corec_c2;

//pile no. 3 at d/s end
d=103.5-95;
alpha_=d/b;
//for curves
fie3=0.35;fid3=0.242;
corec_t=(fie3-fid3)*100*(103.5-102)/d;

//correction for interference at pile no. 2
d=102-95;
D=102-97;
C3=19*(D/bdash)^0.5*(d+D)/b;
fie3=fie3*100-corec_t-C3;

point=['C1' 'C2' 'E2' 'E3'];              //Point
P=[fic1 fic2 fie2 fie3];                   //pressure
   percent
P_=[3.55 2.78 3.39 1.58];                  //pressure
   head
mprintf("Points          Pressure percent
   Pressure head");
for i=1:4
```

```
    P(i)=round(P(i)*10)/10;
    mprintf("\n%s                    %f                    %f",point(i),P(i),P_(i));
end

//check for floor thickness
Pa=P_(2)-((P_(2)-P_(4))*6.5/37);
Pb=P_(2)-((P_(2)-P_(4))*24/37);
Pc=P_(2)-((P_(2)-P_(4))*30/37);
rho=2.24;                                   //specific gravity of concrete
ta=Pa/(rho-1);
tb=Pb/(rho-1);
tc=Pc/(rho-1);
ta=round(ta*100)/100;
tb=round(tb*100)/100;
tc=round(tc*100)/100;
mprintf("\n\nThickness required at A=%f m.",ta);
mprintf("\nThickness required at B=%f m.",tb);
mprintf("\nThickness required at C=%f m.",tc);
t=103.5-102;
mprintf("\nThickness provided=%f m.",t);
mprintf("\nFloor thickness at B and C are adequate");

//exit gradient
H=108.5-103.5;                  //seepage head
d=103.5-95;                     //depth cut-off
//from exit gradient curve
alpha=6.35;
lambda=(1+(1+alpha^2)^0.5)/2;
Ge=H/(d*%pi*lambda^0.5);
mprintf("\n\nexit gradient=%f.",Ge);
mprintf("\n it is less than permissible exit gradient  < 1/6\nHence safe..");
```

Scilab code Exa 12.3 EX12 3

```


//example 12.3
//design a vertical drop weir on Bligh's theory
//test floor by Khosla's theory
clc;funcprot(0);
//given
Q=2800;                          //maximum flood discharge
hfl=285;                         //H.F.L before construction
hw=278;                          //minimum water level
fsl=284;                         //F.S.L of canal
c=12;                            //coefficient of creep
flux=1;                          //allowable afflux
Ge=1/6;                          //permissible exit gradient
rho=2.24;                        //specific gravity of concrete

//Hydraulic calculation
L=4.75*Q^0.5;
q=Q/L;
q=round(q*10)/10;
mprintf("Hydraulic calculation:");
mprintf("\ndischarge per unit width of river=%f cumecs.",q);
f=1;
R=1.35*(q^2/f)^(1/3);
R=round(R*100)/100;
mprintf("\nregime scour depth=%f m.",R);
V=q/R;                         //regime velocity
vh=V^2/(2*9.81);             //velocity head
```

```
l_down=hfl+vh;
l_up=l_down+flux;
hfl_up=l_up-vh;
hfl_down=hfl-0.5;
hfl_down=round(hfl_down*100)/100;
mprintf("\nactual d/s H.F.L allowing 0.5 m for retrogation=%f m.",hfl_down);
K=(q/1.7)^(2/3);
cl=l_up-K;                    //crest level
cl=round(cl*100)/100;
mprintf("\ncrest level=%f m.",cl);
pl=fsl+0.5;                   //pond level
s=hfl_down-cl;                     //heigth of shutter
mprintf("\nheigth of shutter=%f m.",s);
rl_up_pile=hfl_up-1.5*R;     //R.L of bottom u/s pile
d_up_cut=hw-276;             //depth of upstream cut-off
mprintf("\ndepth of upstream cut-off=%f m.",d_up_cut);
mprintf("\n provide concrete cut off 2 m depth.");
rl_bot_ds=hfl_down-2*R;
Hs=hfl_down-hw;              //seepage head
Hc=cl-hw;                    //heigth of crest
mprintf("\nR.L of gates crest=%f m.",Hs);
mprintf("\nHeigth of crest=%f m.",Hc);

//design of weir wall
d=hfl_up-cl;
a=d/(rho)^0.5;
a=3*d/(2*rho);               //from sliding consideration
a=s+1;                       //from practical consieration
a=a+1;
mprintf("\n\ndesign of weir wall:")
mprintf("\nprovide top width of %i m.",a);
Mo=9.81*Hs^3/6;                   //overtirning moment
//equating the moment of resistance to overturning
```

```
moment and putting the values we get
y=poly([-1.084,0.020,0.039],'x','c');
b=roots(y);
//we get b= - 5.5347261 and 5.0219056
//taking
b=5;
//when weir is submerged
C=0.58;
d=(q^2/((2*C/3)^2*2*9.81))^(1/3);
Mo=9.81*d*Hc^2/2;
//from equation of moment of resistence we get
y=poly([-77.55,3,1],'x','c');
b=roots(y);
//we get b= - 10.433085 and 7.4330846
//taking
b=8;
mprintf("\nbottom width=%i m.",b);

//design of impervious and pervious aprons
C=12;
L=C*Hs;
mprintf("\n\ndesign of impervious and pervious aprons:");
mprintf("\ntotal creep length=%i m.",L);
l1=2.21*C*(Hs/13)^0.5;
l1_=l1+1;
mprintf("\nlength of downstream impervious apron=%i m.",l1_);
d1=hw-276;
d2=hw-271;
l2=L-l1-(b+2*d1+2*d2);
mprintf("\nlength of upstream impervious apron=%i m.",l2);
l3=18*C*(Hs*q/975)^0.5;
mprintf("\ntotal length of d/s apron=%i m.",l3);   //calculation is wrong in book
l=l3-l1;
le=l/2;
```

```
le=round(le*100)/100;
mprintf('\nprovide filter of length %f m. and
   launching apron of length %f m.',le,le);
t=d2*10^0.5/le;
mprintf("\nthickness of launching apron in
   horizontal position=%f m.",t);
mprintf("\nprovide launching apron of thickness 1.5
   m.");
T=2*d1;
V=d1*10^0.5;
ta=V/T;
ta=round(ta*10)/10;
mprintf("\nthickness of apron in horizontal position
   =%f m.",ta);
Hr=Hs-Hs*(4+33+8)/L;
t=4*Hr/(3*(rho-1));
t=round(t*10)/10;
mprintf('\nprovide thickness of %f m from d/s of
   weir wall to point 6 m from it.',t);
Hr=Hs-Hs*(4+33+8+6)/L;
t=4*Hr/(3*(rho-1));
t=round(t*10)/10;
mprintf("\nprovide thickness of %f m from 6 m to 12
   m from d/s end of weir wall.",t);
Hr=Hs-Hs*(4+33+8+12)/L;
t=4*Hr/(3*(rho-1));
t=round(t*10)/10;
mprintf("\nprovide thickness of %f m for rest of
   length of weir floor.",t);

//check by khosla's theory
b=33+8+19;              //total horizontal length of
   impervious floor
d=7;                    //depth of downstream pile
alpha=b/d;
n=0.14;                  //n=1/%pi*(lambda)^0.5;
Ge=Hs*n/d;
mprintf("\n\ncheck by Khosla theory:");
```

```
mprintf("\nexit gradient=%f. < 1/6\n hence safe",Ge);
alpha_=d/b;
fic1=0.83;fid1=0.88;
corec_c1=(fid1-fic1)*100/2;
bdash=b;
d=2;D=7;
C1=19*(D/bdash)^0.5*(d+D)/b;
fic1=fic1*100+corec_c1+C1;
Pc=Hs*fic1/100;                          //pressure head at C
alpha_=d/b;
fie2=0.31;fid2=0.21;
corec_e1=(fie2-fid2)*1.7*100/7;
bdash=b;
d=7;D=2;
C1=19*(D/bdash)^0.5*(d+D)/b;
fie2=fie2*100-corec_e1-C1;               //in book 3.53 value is wrong
Pe=Hs*fie2/100;                           //pressue head at E
//assuming linear variation of pressure for intermediate points
Pa=Pc-(Pc-Pe)*(33+8)/b;
t=Pa/1.24;
Pa=round(Pa*100)/100;
t=round(t*100)/100;
mprintf("\npressure at d/s of weir wall=%f m.",Pa);
mprintf("\nthickness at d/s of weir wall=%f m. < thickness by Bligh theory;\nhence safe.",t);
Pb=Pc-(Pc-Pe)*(33+8+6)/b;
t=Pb/1.24;
Pa=round(Pa*100)/100;
t=round(t*100)/100;
mprintf("\npressure at 6 m from d/s of weir wall=%f m.",Pb);
mprintf("\nthickness at 6m from d/s of weir wall=%f m. < thickness by Bligh theory;\nhence safe.",t);
```

```
Pc=Pc-(Pc-Pe)*(33+8+12)/b;
t=Pc/1.24;
Pa=round(Pa*100)/100;
t=round(t*100)/100;
mprintf("\npressure at 12 m from d/s of weir wall=%f
    m.",Pc);
mprintf("\nthickness at 12m from d/s of weir wall=%f
    m. > thickness by Bligh theory;\nhence unsafe.",
    t);
mprintf("\nhence increase th ethickness to 1.9 m for
    a length of 7 m of impervious floor.");
```

Scilab code Exa 12.4 EX12 4

```


//example 12.4
//design a slopeing glacis
clc;funcprot(0);
//given
q=10;                         //maximum discharge
   intensity on weir crest
hfl=255;                      //H.F.L before
   construction of weir
rb=249.5;                     //R.L of river bed
pl=254;                       //pond level
s=1;                          //heigth of crest shutter
dhw=251.5;                    //anticipated downstream
   water level in river when water is dischrging
   with pond level upstream
br=0.5;                       //bed retrogression
f=0.9;                        //Laecey silt factor
Ge=1/7;                       //permissible exit
   gradient
flux=1;                       //permissible afflux
```

```
cl=pl-s;                    //crest level
mprintf("crest level=%f m.",cl);
K=(q/1.7)^(2/3);
tel_up=cl+K;
tel_up=round(tel_up*100)/100;
mprintf("\nelevation of u/s T.E.L=%f m.",tel_up);
R=1.35*(q^2/f)^(1/3);
R=round(R*10)/10;
mprintf("\nregime scour depth=%f m.",R);
V=q/R;                      //regime velocity
vh=V^2/(2*9.81);            //velocity head
hfl_up=tel_up-vh;
tel_down=hfl+vh;
flux=hfl_up-hfl;
flux=round(flux*100)/100;
mprintf("\nafflux=%f. which is near to permissible",flux);
hfl_down=hfl-br;            //downstream H.F.L after retrogression
tel_down=tel_down-br;       //downstream T.F.L after retrogression
Hl=tel_up-tel_down;         //loss of head in flood
Hl=round(Hl*100)/100;
mprintf("\nloss of head in at high flood=%f m.",Hl);
K=pl-cl;               //head over crest
q_=1.7*(K)^1.5;
Hl_=pl-dhw;             //loss of head
mprintf("\nloss of head=%f m.",Hl_);
Ef2=4.3;
Ef2_=1.7;               //from Blench curve
jump=tel_down-Ef2;
jump_=251.5-Ef2_;       //level at which jump will form
Ef1=Ef2+Hl;
Ef1_=Ef2_+Hl_;
D1=1.03;
D1_=0.15;                //calculated from Ef1 and
```

```
    Ef1_ respectively
D2=3.96;D2_=1.68;         //calculated from Ef2 and Ef2_ respectively
hj=D2-D1;
hj_=D2_-D1_;              //heigth of jump
concrete=5*hj;
concrete_=5*hj_;           //length of concrete floor
mprintf("\n\nHydraulic jump calculation:");
mprintf("\nheigth of jump for high flood condition=%f m.",hj);
mprintf("\nlength of concrete floor for high flood condition=%f m.",concrete);
mprintf("\nheigth of jump for pond level condition=%f m.",hj_);
mprintf("\nlength of concrete floor for high pond level condition=%f m.",concrete_);

cw=2;                      //crets width
us=2;                      //upstream slope
ds=3;                      //downstream slope
l=15;
mprintf("\n\n upstream slope of glacis=%i:1.",us);
mprintf("\ndownstream slope of glacis=%i:1.",ds);
mprintf("\nhorizontal length of floor beyond the toe=%i m..",l);

R=6.5;
sh_up=hfl_up-1.5*R;
sh_down=hfl_down-2*R;
sh_up=round(sh_up*100)/100;
mprintf("\nR.L of bottom of upstream sheet pile=%f m.",sh_up);
mprintf("\nR.L of downstream sheet pile=%f m.",sh_down);
mprintf("\nprovide intermediate sheet pile at d/s toe of glacis.");
Hs=pl-249.6;                           //maximum percolation head
```

```
d=249.6-sh_down;                              //depth of d/s
    cut-off
n=Ge*d/Hs;                                     //n=1/(%pi*
    lambda^0.5);
//from khosla exit gradient curve
alpha=1.5;
b=alpha*d;
mprintf("\n\nlength of impervious floor=%f m.",b);
fl=(2*(253-249.5))+2+(3*(253-249.6))+15;
us=36-fl;
mprintf("\nlength of floor already provide=%f m.",fl
    );
mprintf("\nwhich is more than required from
    permissible exit gradient.\nno upstream floor is
    required.");
mprintf("\nprovide %f m upstream floor so that total
     length becomes 36 m.",us);
alpha_1=0.089;
alpha_2=0.225;                //alpha_=1/alpha
b1=21;
alpha=4.44;
mprintf("\n\nPressure percent at points:");
point=['C1' 'D1' 'C2' 'E2' 'D2' 'D3' 'E3'];
bc=[72 82 31.5 45.5 58.5 29 44];
crt=[3.1 0 3.5 0 -3.2 0 0 -3.6];
crs=[0 0 0 0 2.3 0 0 0];
cri=[3.7 0 6.4 0 -2.4 0 -6.4];
mprintf("\nPoints        Before correction
              After correction");
for i=1:7
    after(i)=bc(i)+crt(i)+crs(i)+cri(i);
    mprintf("\n%s                     %i
                                   %f",point(i),bc(i),
        after(i));
end
Hs=254-249.6;                  //no flow condition
Hs_=256.13-254.5;              //high flood condition
Hs__=254-251.5;                //flow at pond level
```

```
mprintf("\n\nelevation of subsoil H.G above datum:");
mprintf("\nno flow condition:");
fie1=1*Hs;
fid1=0.82*Hs;
fic1=0.788*Hs;
fie2=0.552*Hs;
fid2=0.455*Hs;
fic2=0.414*Hs;
fie3=0.34*Hs;
fid3=0.29*Hs;
fic3=0;
fie1=round(fie1*100)/100;fid1=round(fid1*100)/100;fic1=round(fic1*100)/100;
fie2=round(fie2*100)/100;fid2=round(fid2*100)/100;fic2=round(fic2*100)/100;
fie3=round(fie3*100)/100;fid3=round(fid3*100)/100;fic3=round(fic3*100)/100;
mprintf("\nfie1=%f.;fid1=%f.;fic1=%f.\nfie2=%f.;fid2=%f.;fic2=%f.\nfie3=%f.;fid3=%f.;fic3=%f.",fie1,fid1,fic1,fie2,fid2,fic2,fie3,fid3,fic3);
mprintf("\nhigh flood condition:");
fie1=1*Hs_;
fid1=0.82*Hs_;
fic1=0.788*Hs_;
fie2=0.552*Hs_;
fid2=0.455*Hs_;
fic2=0.414*Hs_;
fie3=0.34*Hs_;
fid3=0.29*Hs_;
fic3=0;
fie1=round(fie1*100)/100;fid1=round(fid1*100)/100;fic1=round(fic1*100)/100;
fie2=round(fie2*100)/100;fid2=round(fid2*100)/100;fic2=round(fic2*100)/100;
fie3=round(fie3*100)/100;fid3=round(fid3*100)/100;fic3=round(fic3*100)/100;
mprintf("\nfie1=%f.;fid1=%f.;fic1=%f.\nfie2=%f.;fid2
```

```
    =%f.; fic2=%f.\nfie3=%f.; fid3=%f.; fic3=%f.",fie1,
    fid1,fic1,fie2,fid2,fic2,fie3,fid3,fic3);
mprintf("\nflow at pond level:");
fie1=1*Hs__;
fid1=0.82*Hs__;
fic1=0.788*Hs__;
fie2=0.552*Hs__;
fid2=0.455*Hs__;
fic2=0.414*Hs__;
fie3=0.34*Hs__;
fid3=0.29*Hs__;
fic3=0;
fie1=round(fie1*100)/100;fid1=round(fid1*100)/100;
    fic1=round(fic1*100)/100;
fie2=round(fie2*100)/100;fid2=round(fid2*100)/100;
    fic2=round(fic2*100)/100;
fie3=round(fie3*100)/100;fid3=round(fid3*100)/100;
    fic3=round(fic3*100)/100;
mprintf("\nfie1=%f.; fid1=%f.; fic1=%f.\nfie2=%f.; fid2
    =%f.; fic2=%f.\nfie3=%f.; fid3=%f.; fic3=%f.",fie1,
    fid1,fic1,fie2,fid2,fic2,fie3,fid3,fic3);

mprintf("\n\nPrejump profile:");
mprintf("\nhigh flood condition:");
dist=[3 6 8.4];                     //distance
glacis=[252 251 250.32];            //R.L of glacis
D1=[1.3 1.15 1.03];
mprintf("\nEf1                  D1");
for i=1:3
    Ef1(i)=256.25-glacis(i);
    mprintf("\n%f          %f",Ef1(i),D1(i));
end
mprintf("\npond level flow:");
dist=[3 6 9 9.6];                   //distance
glacis=[252 251 250 249.9];         //R.Lof glacis
D1=[0.31 0.23 0.16 0.15];
mprintf("\nEf1                  D1");
for i=1:4
```

```
    Ef1(i)=254-glacis(i);
    mprintf("\n%f          %f",Ef1(i),D1(i));
end


rho=2.24;
Uf=4;                                 //unbalanced head
   for high flood condtion
Us=2.56;                              //unbalanced static
   head
Hf=2*Uf/3;
t=Hf/(rho-1);
t=round(t*10)/10;
mprintf("\n\nfloor thickness at the point of
   formation of hydraulic jump=%f m.",t);
Uf=2.9;                                 //unbalanced head
   for high flood condtion
Us=2.2;                                //unbalanced static
   head
Hf=2*Uf/3;
t=Us/(rho-1);
t=round(t*10)/10;
mprintf("\nfloor thickness at the point of formation
    of hydraulic jump at the pond level condition=%f
    m.",t);
P=1.5;                                //pressure head at d/s
    end of floor
t=P/(rho-1);
t=round(t*10)/10;
mprintf("\n\nfloor thickness at downstream side of
   sloping glacis=%f m.",t);
D=rb-sh_up;                         //depth of u/s scour
   hole above bed level
a=1.5*D;
a=round(a*10)/10;
mprintf("\n\nminimum length of upstream launching
   apron=%f m.",a);
mprintf("\nprovide 1.5 m thick apron for length of 5
```

```
    m.");
D=249.6-241.5;
a=1.5*D;
mprintf("\n\nminimum length of downstream launching
   apron=%f m.",a);
mprintf("\nprovide 1.5 m thick apron for length of
   12 m.");
```

Scilab code Exa 12.5 EX12 5

```


//example 12.5
//calculate uplift pressure at the junction of inner
    faces of pile with weir floor using Khosla
   theory
clc;funcprot(0);
//given
b=16;        //total length of floor
d=5;         //depth of downstream pile
D=4;         //depth of upstream pile
H=2.5;       //head created by weir

//pressure at E
alpha=b/d;
lambda=(1+(1+alpha^2)^0.5)/2;
fie=acos((lambda-2)/lambda)/%pi;
C=19*(D/b)^0.5*((d+D)/b);
fie=fie*100-C;
P=H*fie/100;
P=round(P*1000)/1000;
mprintf("Pressure at E=%f m.",P);

//pressure at C1
alpha=b/D;
```

```
lambda=(1+(1+alpha^2)^0.5)/2;
fie=acos((lambda-2)/lambda)/%pi;
fic=1-fie;              //by principle reversibility
    of flow
C=19*(d/b)^0.5*((d+D)/b);
fic=fic*100+C;
P=fic*H/100;
P=round(P*1000)/1000;
mprintf("\n Pressure at C=%f m.",P);
```

Scilab code Exa 12.6 EX12 6

```


//example 12.6
//calculate floor thickness at mid length and at
    junction with u/s and d/s cut-off walls
clc;funcprot(0);
//given
b=13;           //length of floor
d=2;            //depth of downstream wall
D=1.5;         //depth of upstream cut-off
rho=2.24;       //relative density
H=1.5;

//at junction of d/s cut-off with floor
alpha=b/d;
lambda=(1+(1+alpha^2)^0.5)/2;
fie=acos((lambda-2)/lambda)/%pi;
C=19*(D/b)^0.5*((d+D)/b);
fie=fie*100-C;
P=H*fie/100;
t=P/(rho-1);
t=round(t*10)/10;
mprintf("floor thickness at junction of d/s cut-off
```

```
with floor=%f m.",t);

//at junction of u/s cut-off with floor
alpha=b/D;
lambda1=(1+(1+alpha^2)^0.5)/2;
fie=acos((lambda1-2)/lambda1)/%pi;
fic=1-fie;                //by principle reversibility of flow
C=19*(D/b)^0.5*((d+D)/b);
fiec=fic*100+C;
P=fiec*H/100;
t=0.3;                 //this the uplift will be counter balanced by downward weigth of impounded water
mprintf("\nfloor thickness at junction of u/s cut-off with floor=%f m.",t);

//at mid-length
P=(1.08+0.489)/2;               //assuming linear variation
t=P/(rho-1);
t=round(t*100)/100;
mprintf("\nfloor thickness at mid-length=%f m.",t);

//exit gradient
G=H/(d*%pi*(lambda)^0.5);
G=round(G*1000)/1000;
//since G<0.18
mprintf("\n G=%f. <0.18./nfloor is safe against failure by piping.",G);
```

Scilab code Exa 12.7 EX12 7

```


```

```
//example12.7
//calculate heigth of weir to be built
clc;funcprot(0);
//given
B=30;                //stream width
D=3;                 //stream depth
V=1.25;              //mean velocity
Cd=0.95;             //discharge coefficient
Q=B*D*V;
C=2*Cd*(2*9.81)^0.5/3;
x=4-(Q/(C*B))^(2/3);
x=round(x*1000)/1000;
mprintf("heigth of weir to be built=%f m.",x);
```

Scilab code Exa 12.8 EX12 8

```


//example 12.8
//calculate uplift pressure at two cut-off
clc;funcprot(0);
//given
b=50;           //length of floor
d=8;            //depth of downstream pile
D=8;            //depth of upstream pile
H=5;            //effective head
tu=1;            //floor thickness at upstream
td=2;            //floor thickness at downstream

//downstream cut-off
alpha=b/d;
lambda=(1+(1+alpha^2)^0.5)/2;
fie=acos((lambda-2)/lambda)/%pi;
fid=acos((lambda-1)/lambda)/%pi;
Ct=(fie-fid)*td/d;
```

```
C=19*(D/b)^0.5*((d+D)/b);
fie=fie*100-C-Ct*100;
P=H*fie/100;
P=round(P*100)/100;
mprintf("Pressure at downstream cut-off=%f m.",P);

//upstream cut-off
fie=acos((lambda-2)/lambda)/%pi;
fid=acos((lambda-1)/lambda)/%pi;
fic1=1-fie;
fid1=1-fid;
Ct=(fic1-fid1)*td/d;
C=-19*(D/b)^0.5*((d+D)/b);
fic1=fic1*100-C-Ct*100;
P=H*fic1/100;
P=round(P*100)/100;
mprintf("\nPressure at upstream cut-off=%f m.",P);
G=H/(d*%pi*(lambda)^0.5);
mprintf("\nExit Gradient=%f.",G);
```

Scilab code Exa 12.9 EX12 9

```


//example 12.9
//calculate depth of downstream cut-off
clc;funcprot(0);
//given
Q=1000;         //discharge of river
L=256;          //crest length of diversion
f=1.1;          //silt factor
seg=1/6;        //safe exit gradient
hfl=103;        //high flood level
cf=100;         //reduced level of downstream
   concrete floor
```

```
H=2.4;          //maximum static head of weir
b=40;           //length of concrete floor

q=Q/L;
R=1.35*(q^2/f)^(1/3);
rld=hfl-1.5*R;
d=cf-rld;
d=round(d*100)/100;
mprintf("depth of downstream cut-off=%f m.",d);

alpha=b/d;
lambda=(1+(1+alpha^2)^0.5)/2;
G=H/(d*%pi*(lambda)^0.5);
//since G<seg
mprintf("\n G=%f. <1/6./nfloor is safe against
    failure by piping.",G);
```

Scilab code Exa 12.10 EX12 10

```


//example 12.10
//calculate critical exit gradient and factor of
    safety of system
clc;funcprot(0);
//given
b=60;        //length of floor
H=6;         //static head of weir
d=6;         //downstream depth of pile
n=0.3;       //porousity of soil particles
G=2.7;       //relative density of soil particles

alpha=b/d;
lambda=(1+(1+alpha^2)^0.5)/2;
Ge=H/(d*%pi*(lambda)^0.5);
```

```
e=n/(1-n);
chg=(G-1)/(1+e);
f=chg/Ge;
f=round(f*100)/100;
mprintf("critical exit gradient=%f.\nfactor of
   safety of system=%f.",chg,f);
```

Scilab code Exa 12.11 EX12 11

```


//example 12.11
//design a vertical drop weir on Bligh's theory
//test floor by Khosla's theory
clc;funcprot(0);
//given
Q=2800;                  //maximum flood discharge
hfl=285;                 //H.F.L before
   construction
hw=278;                  //minimum water level
fsl=284;                 //F.S.L of canal
c=12;                    //coefficient of creep
flux=1;                  //allowable afflux
Ge=1/6;                  //permissible exit
   gradient
rho=2.24;                //specific gravity of
   concrete

//Hydraulic calculation
L=4.75*Q^0.5;
q=Q/L;
q=round(q*10)/10;
mprintf("Hydraulic calculation:");
mprintf("\ndischarge per unit width of river=%f
   cumecs.",q);
```

```
f=1;
R=1.35*(q^2/f)^(1/3);
R=round(R*100)/100;
mprintf("\nregime scour depth=%f m.",R);
V=q/R;                          //regime velocity
vh=V^2/(2*9.81);               //velocity head
l_down=hfl+vh;
l_up=l_down+flux;
hfl_up=l_up-vh;
hfl_down=hfl-0.5;
hfl_down=round(hfl_down*100)/100;
mprintf("\nactual d/s H.F.L allowing 0.5 m for retrogation=%f m.",hfl_down);
K=(q/1.7)^(2/3);
cl=l_up-K;                      //crest level
cl=round(cl*100)/100;
mprintf("\ncrest level=%f m.",cl);
pl=fsl+0.5;                     //pond level
s=hfl_down-cl;                       //heigth of shutter
mprintf("\nheigth of shutter=%f m.",s);
rl_up_pile=hfl_up-1.5*R;       //R.L of bottom u/s pile
d_up_cut=hw-276;                 //depth of upstream cut-off
mprintf("\ndepth of upstream cut-off=%f m.",d_up_cut);
mprintf("\n provide concrete cut off 2 m depth.");
rl_bot_ds=hfl_down-2*R;
Hs=hfl_down-hw;                  //seepage head
Hc=cl-hw;                        //heigth of crest
mprintf("\nR.L of gates crest=%f m.",Hs);
mprintf("\nHeigth of crest=%f m.",Hc);

//design of weir wall
d=hfl_up-cl;
a=d/(rho)^0.5;
a=3*d/(2*rho);                  //from sliding consideration
a=s+1;                          //from practical
```

```
consieration
a=a+1;
mprintf("\n\ndesign of weir wall:")
mprintf("\nprovide top width of %i m.",a);
Mo=9.81*Hs^3/6;                    //overtirning moment
//equating the moment of resistance to overturning moment and putting the values we get
y=poly([-1.084,0.020,0.039],'x','c');
b=roots(y);
//we get b= - 5.5347261 and 5.0219056
//taking
b=5;
//when weir is submerged
C=0.58;
d=(q^2/((2*C/3)^2*2*9.81))^(1/3);
Mo=9.81*d*Hc^2/2;
//from equation of moment of resistence we get
y=poly([-77.55,3,1],'x','c');
b=roots(y);
//we get b= - 10.433085 and 7.4330846
//taking
b=8;
mprintf("\nbottom width=%i m.",b);

//design of impervious and pervious aprons
C=12;
L=C*Hs;
mprintf("\n\ndesign of impervious and pervious aprons:");
mprintf("\ntotal creep length=%i m.",L);
l1=2.21*C*(Hs/13)^0.5;
l1_=l1+1;
mprintf("\nlength of downstream impervious apron=%i m.",l1_);
d1=hw-276;
d2=hw-271;
l2=L-l1-(b+2*d1+2*d2);
mprintf("\nlength of upstream impervious apron=%i m.
```

```
",l2);
l3=18*C*(Hs*q/975)^0.5;
mprintf("\ntotal length of d/s apron=%i m.",l3);            //calculation is wrong in book
l=l3-l1;
le=l/2;
le=round(le*100)/100;
mprintf('\nprovide filter of length %f m. and launching apron of length %f m.',le,le);
t=d2*10^0.5/le;
mprintf("\nthickness of launching apron in horizontal position=%f m.",t);
mprintf("\nprovide launching apron of thickness 1.5 m.");
T=2*d1;
V=d1*10^0.5;
ta=V/T;
ta=round(ta*10)/10;
mprintf("\nthickness of apron in horizontal position=%f m.",ta);
Hr=Hs-Hs*(4+33+8)/L;
t=4*Hr/(3*(rho-1));
t=round(t*10)/10;
mprintf('\nprovide thickness of %f m from d/s of weir wall to point 6 m from it.',t);
Hr=Hs-Hs*(4+33+8+6)/L;
t=4*Hr/(3*(rho-1));
t=round(t*10)/10;
mprintf("\nprovide thickness of %f m from 6 m to 12 m from d/s end of weir wall.",t);
Hr=Hs-Hs*(4+33+8+12)/L;
t=4*Hr/(3*(rho-1));
t=round(t*10)/10;
mprintf("\nprovide thickness of %f m for rest of length of weir floor.",t);

//check by khosla's theory
b=33+8+19;            //total horizontal length of
```

```
   impervious floor
d=7;                    //depth of downstream pile
alpha=b/d;
n=0.14;                  //n=1/%pi*(lambda)^0.5;
Ge=Hs*n/d;
mprintf("\n\ncheck by Khosla theory:");
mprintf("\nexit gradient=%f. < 1/6\n hence safe",Ge);
alpha_=d/b;
fic1=0.83;fid1=0.88;
corec_c1=(fid1-fic1)*100/2;
bdash=b;
d=2;D=7;
C1=19*(D/bdash)^0.5*(d+D)/b;
fic1=fic1*100+corec_c1+C1;
Pc=Hs*fic1/100;                         //pressure head at C
alpha_=d/b;
fie2=0.31;fid2=0.21;
corec_e1=(fie2-fid2)*1.7*100/7;
bdash=b;
d=7;D=2;
C1=19*(D/bdash)^0.5*(d+D)/b;
fie2=fie2*100-corec_e1-C1;              //in book 3.53 value is wrong
Pe=Hs*fie2/100;                          //pressue head at E
//assuming linear variation of pressure for intermediate points
Pa=Pc-(Pc-Pe)*(33+8)/b;
t=Pa/1.24;
Pa=round(Pa*100)/100;
t=round(t*100)/100;
mprintf("\npressure at d/s of weir wall=%f m.",Pa);
mprintf("\nthickness at d/s of weir wall=%f m. < thickness by Bligh theory;\nhence safe.",t);
Pb=Pc-(Pc-Pe)*(33+8+6)/b;
t=Pb/1.24;
```

```
Pa=round(Pa*100)/100;
t=round(t*100)/100;
mprintf("\npressure at 6 m from d/s of weir wall=%f
   m.",Pb);
mprintf("\nthickness at 6m from d/s of weir wall=%f
   m. < thickness by Bligh theory;\nhence safe.",t);
Pc=Pc-(Pc-Pe)*(33+8+12)/b;
t=Pc/1.24;
Pa=round(Pa*100)/100;
t=round(t*100)/100;
mprintf("\npressure at 12 m from d/s of weir wall=%f
    m.",Pc);
mprintf("\nthickness at 12m from d/s of weir wall=%f
    m. > thickness by Bligh theory;\nhence unsafe.",
   t);
mprintf("\nhence increase th ethickness to 1.9 m for
    a length of 7 m of impervious floor.");
```

Scilab code Exa 12.12 EX12 12

```


//example 12.12
//calculate
//number of gates required for the barrage
//head regulator if each gate has 10 m clear span(
   neglect end contractions and approach velocity)
//length and R.L of basin floor if silting basin is
   provided downstream of barrage
clc;funcprot(0);
//given
Lmax=212;            //maximum reservior level
Lp=211;              //pond level
hfl=210;             //downstream high flood level in
    the river
```

```
Qmax=3500;            //maximum design flood discharge
Lcrest=207;           //crest level of the barrage
Lcrest_r=208;         //crest level of head regulator
Cd=2.1;               //coefficient of discharge for barrage
Cd_r=1.5;             //coefficient of discharge for head regulator
rbl=205;              //river bed level
Q=500;                //design discharge of main canal

//design of water way for barrage during flood
H=Lmax-Lcrest;
L=Qmax/(Cd*H^1.5);
//which gives L=149.07.
//provide 15 bays of 10m clear span
mprintf("nunmber of gates for the barrage=15.");

//design of waterway for canal head regulator
H=Lp-Lcrest_r;
L1=Q/(Cd_r*H^1.5);
//which gives L=64.2
//hence provide 7 bays of 10 m each
mprintf("\n\nnunmber of gates for the head regulator=7.");

//design of stilling basin
Hl=Lmax-hfl;
q=Qmax/L;
yc=(q^2/9.81)^(1/3);
Z=Hl/yc;
//since Z<1
Y=1+0.93556*Z^0.368;
y2=Y*yc;
Lc=5*y2;
Lc=round(Lc*10)/10;
mprintf("\n\nLength of cistern=%f m.",Lc);
Ef2=yc*(Y+1/(2*Y^2));
j=hfl-Ef2;
```

```
j=round(j*10)/10;
mprintf("\nR.L of cistern=%f m.",j);
```

Chapter 14

IRRIGATION CHANNEL 1 SILT THEORIES

Scilab code Exa 14.1 EX14 1

```


//example 14.1
//design irrigation channel on Kennedy's theory
clc;funcprot(0);
//given
Q=45;         //discharge
N=0.0225;     //rogosity coefficient
m=1.05;       //critical velocity ratio
S=1/5000;     //bed slope

D=2;  //assume
Vo=0.55*m*D^0.64;
A=Q/Vo;
//for a trapezoidal section
B=(A-0.5*D^2)/2;
P=B+D*5^0.5;
R=A/P;
C=(23+1/N+0.00155/S)*(R*S)^0.5/(1+(23+0.00155/S)*N/R
```

```
   ^0.5);
V=C*(R*S)^0.5;
//Vo<V

//assume D=2.2
D=2.2;
Vo=0.55*m*D^0.64;
A=Q/Vo;
B=(A-0.5*D^2)/D;
P=B+D*5^0.5;
R=A/P;
C=(23+1/N+0.00155/S)*(R*S)^0.5/(1+(23+0.00155/S)*N/R
   ^0.5);
V=C*(R*S)^0.5;

//ratio of V and Vo is almost equal to 1
B=round(B*10)/10;
 mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

Scilab code Exa 14.2 EX14 2

```


//example 14.2
//design an irrigation canal for given data
clc;funcprot(0);
//given
Q=14;          //discharge
m=1;           //critical velocity ratio
r=5.7;         //B/D

D=(Q/(0.55*6.2))^(1/2.64);
B=D*r;
R=(B*D+D^2/2)/(B+D*5^0.5);
```

```
Vo=0.55*m*D^0.64;
//applying kutters formula; V=C(RS)^0.5
//where C=(23+1/N+0.00155/S)*(R*S)
    ^0.5/(1+(23+0.00155/S)*N/R^0.5);
//assuming S^0.5=y
y=poly([-1.98D-5,1.55D-3,-0.954,67.5],'x','c');
roots(y);
//taking real values of y
S=0.0139906^2;
B=round(B*100)/100;
D=round(D*100)/100;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 14.3 EX14 3

```


//example 14.3
//design a channel on Kennedy's theory
clc;funcprot(0);
//given
Q=45;                //discharge
m=1.05;              //critical velocity ratio
N=0.025;             //rugosity coefficient
S=1/5000;            //bed slope

l=S*Q^0.02/(N^2*m^2.02);
//from fig.14.3 we get r=10
//solving the equation by trial and error method we
    get
r=9.7;
D=(1.818*Q/(m*(r+0.5)))^(1/2.64);
B=r*D;
```

```
V=Q/(D^2*(r+0.5));
Vo=0.55*D^0.64*m;
B=round(B);
D=round(D*100)/100;
V=round(V*1000)/1000;
Vo=round(Vo*1000)/1000;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nVelocity through the channel section=%f m
   /s.",V);
mprintf("\nVo=%f m/s.\nHence Safe",Vo);
```

Scilab code Exa 14.4 EX14 4

```


//example 14.4
//design channel using method of curve fitting based
    onKennedy's theory
clc;funcprot(0);
//given
Q=45;           //discharge
N=0.0225;       //rugosity coefficient
m=1.05;         //critical velocity ratio
S=1/5000;       //Bed slope

r=(1.607*S^1.63*Q^0.033/(N^3.26*m^3.293)-0.258)
   ^(-0.915);
D=(1.818*Q/(m*(r+0.5)))^(1/2.64);
B=r*D;
B=round(B);
D=round(D*100)/100;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

Scilab code Exa 14.5 EX14 5

```
//example 14.5
//design channel using curve of CWPC for B/D ratio
clc;funcprot(0);
//given
Q=45;          //discharge
N=0.0225;      //rugosity coefficient
m=1.05;        //critical velocity ratio

r=(15+6.44*Q)^0.382;
S=(N^2/1.338*Q^0.02)*(0.258+(15+6.44*Q)^(-0.417))
    ^0.6135;
D=(1.818*Q/(m*(r+0.5)))^(1/2.64);
B=r*D;
B=round(B);
D=round(D*100)/100;
mprintf("Bed slope=%f.",S);
mprintf("\nWidth of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

Scilab code Exa 14.6 EX14 6

```


//example 14.6
//design the channel section using the following
    data and calculate logitudnal section
clc;funcprot(0);
//given
Q=30;      //discharge
```

```
f=1;        //silt factor
s=1/2;      //side slope

V=(Q*f/140)^(1/6);
A=Q/V;
P=4.75*Q^0.5;
D=(P-(P^2-6.944*A)^0.5)/3.472;
B=P-2.236*D;

R=5*V^2/(2*f);
S=f^(5/3)/(3340*Q^(1/6));
B=round(B*100)/100;
D=round(D*100)/100;
mprintf("Bed slope=%f.",S);
mprintf("\nWidth of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

Scilab code Exa 14.7 EX14 7

```


//example 14.7
//design a channel in alluvial soil using tractive
   force approach
clc;funcprot(0);
//given
Q=45;           //discharge
S=1/4800;       //bed slope
N=0.0225;       //rogosity coefficient
sigma=0.0035;   //permissible tractive stress
s=1/2;          //side slope
gamma_w=9.81;    //unit weigth of water

R=sigma/(gamma_w*S);
V=R^(2/3)*S^0.5/N;
```

```
A=Q/V;
P=A/R;
y=poly([-49,28.61,-1.736],'x','c');
D=roots(y);
//we get D=14.539034 and 1.9413812
//taking D=1.9413812
D=1.9413812;
B=28.61-2.23*D;
B=round(B*100)/100;
D=round(D*100)/100;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

Scilab code Exa 14.8 EX14 8

```


//example 14.8
//designa channel section by Kennedy theory
clc;funcprot(0);
//given
Q=28;           //discharge
m=1;            //critical velocity ratio
r=7.6;          //B/D

D=(Q/4.46)^(1/2.64);
B=r*D;
R=0.823*D;
V=0.55*(D)^0.64;

//applying kutters formula; V=C(RS)^0.5
//where C=(23+1/N+0.00155/S)*(R*S)
    ^0.5/(1+(23+0.00155/S)*N/R^0.5);
//we get equation in S
//assuming S^0.5=y
```

```
y=poly([-1.42D-5,1.55D-3,-0.885,67.4],'x','c');
roots(y);
//taking real values of y
S=0.0126305^2;
B=round(B*10)/10;
D=round(D*100)/100;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 14.9 EX14 9

```


//example 14.9
//design the channel section and calculate discharge
clc;funcprot(0);
//given
r=5.7;          //B/D
S=1/5000;       //bed slope
N=0.0225;       //rogosity coefficient
m=1;            //critical velocity ratio(assumed)

//applying kutters formula; V=C(RS)^0.5
//where C=(23+1/N+0.00155/S)*(R*S)
    ^0.5/(1+(23+0.00155/S)*N/R^0.5);
//we get equation in d as
//38.88*D^0.64-66.5*D^0.5+30.37*D^0.14=0
//solving it by trial and error method
//we get D=1.7 m.
D=1.7;
B=r*D;
V=0.55*m*(D)^0.64;
A=B*D+D^2/2;
Q=A*V;
```

```
Q=round(Q*100)/100;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\n Discharge=%f cumecs.",Q);
```

Scilab code Exa 14.10 EX14 10

```


//example 14.10
//design irrigation channel  according to Laecy silt
    theory
clc;funcprot(0);
//given
Q=15;                    //discharge
f=1;                     //laecy silt factor
s=1/2;                   //channel side slope

V=(Q*f^2/140);
A=Q/V;
R=5*V^2/(2*f);
//using the value of A in equations we get,
//equation in D as
y=poly([-21.765,18.336,-1.73],'x','c');
D=roots(y);
//we get D=9.2368003 and 1.3620436.
//taking
D=1.3620436;
B=18.336-D*2.23;
P=4.75*Q^0.5;
S=1/(3340*Q^(1/6));
B=round(B*10)/10;
D=round(D*100)/100;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

```
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 14.11 EX14 11

```


//example 14.11
//find channel section and discharge
clc;funcprot(0);
//given
S=1/5000;               //bed slope
s=1/2;                  //side slope
f=0.9;                    //laecy silt factor

Q=(f^(5/3)/(3340*S))^6;
R=f^3/(4980*S)^2;
P=4.75*Q^0.5;
A=P*R;
//using the value of A and P in equations we get,
//equation in D as
y=poly([-6.961,9.41,-1.73],'x','c');
D=roots(y);
//we get D=4.5561754 and 0.8831309.
//taking
D=0.8831309;
B=9.41-D*2.23;
B=round(B*100)*100;
D=round(D*100)/100;
Q=round(Q*1000)/1000;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\n Discharge=%f cumecs.",Q);
```

Scilab code Exa 14.12 EX14 12

```


//example 14.12
//calculate quantity of bed load moved by the Meyer−Peter equation
clc;funcprot(0);
//given
gamma_w=9.81;          //unit weigth of water
D=3;                   //depth of channel
d=0.3;                 //grain size
k=1.5;                 //size of roughness of channel bed
S=1/4400;              //bed slope
G=2.65;               //specific gravity
tau_b=gamma_w*D*S;
N1=d^(1/6)/24;
N=k^(1/6)/24;
gamma_s=gamma_w*G;
tau_c=0.047*(gamma_s-gamma_w)*d/1000;
r=(N1/N)^1.5;
q=47450*(tau_b*r-tau_c)^1.5;
q=round(q*100)/100;
mprintf("quantity of bed load moved=%f kN/m/hr.",q);
```

Scilab code Exa 14.13 EX14 13

```


//example 14.13
//calculate bed load transported by channel by einstein equation
clc;funcprot(0);
//given
```

```
gamma_w=9.81;            //unit weigth of water
D=3;                     //depth of channel
d=0.3;                   //grain size
k=1.5;                   //size of roughness of channel
    bed
S=1/4400;                //bed slope
G=2.65;                 //specific gravity

N1=d^(1/6)/24;
N=k^(1/6)/24;
r=(N1/N)^1.5;
R1=3*r;
si=(G-1)*d/(1000*R1*S);
//hence we get
fi=7;
q=3600*fi*G*gamma_w*(G-1)^0.5*(gamma_w)^0.5*(d/1000)
   ^1.5;
q=round(q*10)/10;
mprintf("quantity of bed load moved=%f kN/m/hr.",q);
```

Scilab code Exa 14.14 EX14 14

```


//example 14.14
//calculate concentration of suspended load
clc;funcprot(0);
//given
gamma_w=9.81;            //unit weigth of water
D=3;                     //depth of channel
d=0.3;                   //grain size
k=1.5;                   //size of roughness of channel
    bed
S=1/4400;                //bed slope
G=2.65;                 //specific gravity
```

```
V=0.03;                     //fall velocity
c_=400;                      //concentration at 0.3 m above bed
a=0.3;
y=1;
k_=0.4;                    //van karman's constant

N1=d^(1/6)/24;
N=k^(1/6)/24;
r=(N1/N)^1.5;
R1=3*r;
V_=(gamma_w*R1*S)^0.5;
c=c_*((a/y)*(D-y)/(D-a))^(V/(V_*k_));
c=round(c*10)/10;
mprintf("concentration of suspended load=%f ppm.",c);
```

Scilab code Exa 14.15 EX14 15

```


//example 14.15
//design an irrigation channel by Meyer peter equation
clc;funcprot(0);
//given
Q=45;                    //discharge
c=55;                    //bed load concentraion
d=0.3;                   //average grain diameter
gamma_w=9.81;            //unit weigth of water
G=2.67;
f=0.964;

c=c*Q*gamma_w*3600/1000000;
P=4.75*Q^0.5;
```

```
//taking channel width as B=28 m(slightly less than
    P)
B=28;
qs=c/B;
//assuming effective grain diameter k=0.4 mm
ks=0.4D-3;
N1=ks^(1/6)/24;
sf=1.76*d^0.5;
N=0.0225*sf^0.25;
r=N1/N;
tau_c=0.047*gamma_w*(G-1)*d/1000;
tau_b=r^1.5*((qs/47450)^(2/3)+tau_c);
//from Manning's formula we get on simplification
R=(0.000992*1000/0.525)^(3/7);
S=0.525/(1000*R);
//solving equation of R for trapezoidal section of
    side slope 1/2 we get
y=poly([-36.792,25.06,0.5],'x','c');
D=roots(y);
//we get D= -51.547499  and 1.4274989
//taking
D=1.4274989;
D=round(D*100)/100;
mprintf("Width of channel section=%i m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 14.16 EX14 16

```


//example 14.16
//design an irrigation channel by Einstein equation
clc;funcprot(0);
//given
```

```
Q=45;                   //discharge
c=55;                   //bed load concentraion
d=0.3;                  //average grain diameter
gamma_w=9.81;           //unit weigth of water
G=2.67;                 //specific gravity of soil
f=0.964;                //silt factor

//taking channel width as B=28 m(slightly less than P)
B=28;
qs=c/B;

fi=(qs/(gamma_w*G))*(1/(G-1))^0.5*(1000000000/(gamma_w*d^3))^0.5;
//from fig. 14.6 we get value of sci
//using the sci equation and Manning formula and on simplifications we get
R=(2.4296)^(3/7);
S=0.4083/(1000*1.463);
//solving equation of R for trapezoidal section of side slope 1/2 we get
y=poly([-40.96,24.73,0.5],'x','c');
D=roots(y);
//we get D= -51.064253  and 1.6042534
//taking
D=1.6042534;
D=round(D*10)/10;
mprintf("Width of channel section=%i m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 14.17 EX14 17

```
//example 14.17
//design a channel for non−alluvial deposites
clc;funcprot(0);
//given
Q=45;                    //discharge
S=1/4000;                //bed slope
v=0.9;                   //permissible velocity
N=0.025;                 //rogosity coefficient

A=Q/v;
R=(v*N/S^0.5)^1.5;
P=A/R;
//let us provide a trapezoidal section
//from equation of Area and Perimeter of trapezoid
y=poly([50,-29.45,1.828],'x','c');
D=roots(y);
//from which we get D=14.181815 and 1.9286881
//taking
D=1.9286881;
B=P-2*1.41*D;
D=round(D*100)/100;
mprintf("Width of channel section=%i m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

Scilab code Exa 14.18 EX14 18

```


//example14.18
//design non−allvial channel using Bazin's formula
clc;funcprot(0);
//given
Q=15;                    //discharge
V=0.75;                  //mean velocity
s=1;                     //side slope
```

```
K=1.3;                 //bazin's coefficient
//width is five times its depth

A=Q/V;
D=(A/6)^0.5;
B=5*D;
P=B+2*D*1.41;
R=A/P;
C=87/(1+K/(R)^0.5);
S=(V/C)^2/R;
B=round(B*10)/10;
D=round(D*100)/100;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 14.19 EX14 19

```



//example14.19
//determine dimension of channel using chezy's
   equation
//calculate the value of manning n
clc;funcprot(0);
//given
Q=21.5;             //discharge
S=1/2500;           //slope of bottom
C=70;
r=1/1.73;
//taking R=0.5*D
//and keeping it in Q=V*A;where V=C(RS)^0.5 and A=D
   ^2(2*(4/3)^0.5-1/3^0.5);
D=(21.5/1.7146)^(1/2.5);
```

```
B=2*D*((4/3)^0.5-(1/3)^0.5);
B=round(B*100)/100;
D=round(D*100)/100;
mprintf("side slope=%f.",r);
mprintf("\nWidth of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);

R=0.5*D;
V=C*(R*S)^0.5;
n=R^(2/3)*S^0.5/V;
n=round(n*1000)/1000;
mprintf("\n\nvalue of manning n=%f.",n);
```

Scilab code Exa 14.20 EX14 20

```


//example14.20
//design a regime channel
clc;funcprot(0);
//given
Q=100;            //discharge
f=1.1;            //silt factor
s=1/2;            //side slope

V=(Q*f^2/140)^(1/6);
A=Q/V;
P=4.75*Q^0.5;
D=(P-(P^2-6.944*A)^0.5)/3.472;
B=P-2.236*D;
R=5*V^2/(2*f);
S=f^(5/3)/(3340*Q^(1/6));
B=round(B*10)/10;
D=round(D*100)/100;
mprintf("Width of channel section=%f m.",B);
```

```
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 14.21 EX14 21

```


//example 14.21
//design a channel using Laecy theory
clc;funcprot(0);
//given
Q=40;                    //discharge
s=1;                     //side slope
md=0.8;                  //average size of base material

f=1.76*(md)^0.5;
V=(Q*f^2/140)^(1/6);
A=Q/V;
P=4.75*Q^0.5;
//from equations of Area and perimeter of trapezoidal section;we get
y=poly([42.41,-30.04,1.828],'x','c');
D=roots(y);
//we get D=14.873416 and 1.5598447
//taking
D=1.5598447;
B=A/D-D;
R=5*V^2/(2*f);
S=f^(5/3)/(3340*Q^(1/6));
B=round(B*100)/100;
D=round(D*100)/100;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 14.22 EX14 22

```
//example14.22
//calculate bed width and floor depth
clc;funcprot(0);
//given
Q=30;                  //discharge
V=1;                   //velocity of flow

A=Q/V;
//perimeter of section=30/D-D/2
//taking its derivative w.r.t to D
D=1/((1.914/30)^0.5);
//from equation of area
B=30/D-D/2;
B=round(B*10)/10;
D=round(D*100)/100;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

Scilab code Exa 14.23 EX14 23

```
//example 14.23
//determine whether flow is critical or sub-critical
clc;funcprot(0);
//given
Q=17;               //discharge
B=6;                //base of channel
s=1/2;              //side slope
```

```
D=1.5;              //depth of channel

A=D*((B+B/s)/2);
V=Q/A;
P=B+2*((D/s)^2+D^2)^0.5;
R=A/P;
F=V/(9.81*R)^0.5;        //froud number
F=round(F*100)/100;
//since F<1;
mprintf("Froud number=%f.\nF<1.\nFlow is sub-critical",F);
```

Scilab code Exa 14.24 EX14 24

```


//example14.24
//calculate normal depth and average shear stress at channel bed
clc;funcprot(0);
//given
B=3.5;              //bottom width of channel
n=0.016;            //manning n
S=2.6/10000;        //bed slope
Q=8;                //discharge
lfs=1;              //left side slope
rhs=1.5;            //rigth side slope
gamma_w=9.81;       //unit weigth of water

//using the equation of area and perimeter of trapezoidal section;Manning's formula and V=Q/A we get D as
//Manning formula: V=R^(2/3)*S^0.5/n
//(D*(3.5+1.25*D))^2.5=78.281+71.951*D
//solving it by trial and error method;we get
```

```
D=1.5;
R=(D*(3.5+1.25*D))/(3.5+3.217*D);
tau=gamma_w*R*S*1000;
tau=round(tau*100)/100;
mprintf("Depth of section=%f m.",D);
mprintf("\nAverage shear stress at channel bed=%f N/
   square-mm.",tau);
```

Scilab code Exa 14.25 EX14 25

```


//example 14.25
//calculate bed load transported by the channel in
   tonnes per day
clc;funcprot(0);
//given
S=1/5000;           //bed slope
B=40;               //width of channel
D=2.6;              //depth of channel
d=0.38;             //mean diameter of bed material
n=0.021;            //Manning n
D65=0.64D-3;        //bed material size(m)
w=1000;             //density of water
//B/D as large tau_c=0.075*d;
tau_c=0.075*d;
tau_b=w*D*S;
N1=(D65)^(1/6)/24;
r=N1/n;
qs=4700*24*(tau_b*r^1.5-tau_c)^1.5/1000;
qs40=qs*40;
mprintf("bed load transported by the channel =%i t/m
   /day.",qs40);
```

Scilab code Exa 14.26 EX14 26

```
//example 14.26
//calculate bed width of channel;also check depth
    using Kutter equation
clc;funcprot(0);
//given
Q=5;                  //discharge
S=0.2/1000;           //bed slope
m=0.8;                //critical velocity ratio
s=1/2;                //side slope of chanel
C=30;
//assuming
D=1;
Vo=0.55*m*D^0.64;
A=Q/Vo;
B=A/D-(s*D);
P=B+2.43*D;
R=A/P;
V=C*(R*S)^0.5;
//Vo>V
//hence take second trial
D=0.8;       //assume
Vo=0.55*m*D^0.64;
A=Q/Vo;
B=A/D-(s*D);
P=B+2.43*D;
R=A/P;
V=C*(R*S)^0.5;
//again Vo>V
//hence we take third trial
D=0.7;
```

```
Vo=0.55*m*D^0.64;
A=Q/Vo;
B=A/D+(s*D);
P=B+2.43*D;
R=A/P;
V=C*(R*S)^0.5;
B=round(B*100)/100;
//Vo is almost equal to V;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

Scilab code Exa 14.27 EX14 27

```


//example 14.27
//design irrigation channel by Kennedy method
clc;funcprot(0);
//given
Q=50;            //discharge
r=2.5;           //B/D ratio
m=1.1;           //critical velocity ratio
N=0.025;         //rogosity coefficient
s=0.5;           //side slope of channel

//using the equation of Vo and Q=A*V;we get
D=(Q/1.815)^(1/2.64);
B=r*D;
R=(B*D+0.5*D^2)/(B+2.236*D);
Vo=0.55*m*D^0.64;

//applying kutters formula; V=C(RS)^0.5
//where C=(23+1/N+0.00155/S)*(R*S)
    ^0.5/(1+(23+0.00155/S)*N/R^0.5);
//assuming S^0.5=y
```

```
y=poly([-3.737D-7,2.46D-5,-0.0199,1],'x','c');
roots(y);
//taking real values of y
S=0.0196171 ^2;
B=round(B*100)/100;
D=round(D*100)/100;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 14.28 EX14 28

```


//example 14.28
//design a regime channel using Laecy's theory
clc;funcprot(0);
//given
Q=35;              //discharge
f=0.9;             //silt factor
s=1/2;             //side slope

V=(Q*f/140)^(1/6);
A=Q/V;
P=4.75*Q^0.5;
D=(P-(P^2-6.944*A)^0.5)/3.472;
B=P-2.236*D;

R=5*V^2/(2*f);
S=f^(5/3)/(3340*Q^(1/6));
D=round(D*100)/100;
mprintf("Bed slope=%f.",S);
mprintf("\nWidth of channel section=%i m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

Scilab code Exa 14.29 EX14 29

```


//example 14.29
//design an irrigation canal for given data
clc;funcprot(0);
//given
Q=15;          //discharge
m=1;           //critical velocity ratio
r=5.7;         //B/D

D=(Q/(0.55*6.2))^(1/2.64);
B=D*r;
R=(B*D+D^2/2)/(B+D*5^0.5);
Vo=0.55*m*D^0.64;
//applying kutters formula; V=C(RS)^0.5
//where C=(23+1/N+0.00155/S)*(R*S)
    ^0.5/(1+(23+0.00155/S)*N/R^0.5);
//assuming S^0.5=y
y=poly([-2D-5,1.55D-3,-0.968,67.5],'x','c');
roots(y);
//taking real values of y
S=0.0141937^2;
B=round(B*100)/100;
D=round(D*100)/100;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 14.30 EX14 30

```
//example 14.30
//Design a section of unlined canal in a loomy soil
clc;funcprot(0);
//given
Q=50;              //discharge
V=1;               //permissible velocity
s=2;               //side slope
r=6;               //B/D ratio
N=0.0225;          //rogosity coefficient

A=Q/V;
D=(A/(r+2))^0.5;
B=r*D;
P=B+2*(5*D^2)^0.5;
R=A/P;
S=(V*N/R^(2/3))^2;
mprintf("Width of channel section=%i m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 14.31 EX14 31

```
//example 14.31
//calculate concentration of suspended load at depth
clc;funcprot(0);
//given
gamma_w=9.81;          //unit weigth of water
D=5;                   //depth of channel
d=0.3;                 //grain size
k=1.5;                 //size of roughness of channel
    bed
```

```
S=1/4000;                   //bed slope
G=2.65;                    //specific gravity
V=0.02;                    //fall velocity
c_=1000;                     //concentration at 0.3 m above bed
a=0.3;
y=2.5;
k_=0.4;                   //van karman's constant

R=5;                      //R=D for wide channel
V_=(gamma_w*R*S)^0.5;
c=c_*((a/y)*(D-y)/(D-a))^(V/(V_*k_));
mprintf("concentration of suspended load=%i ppm.",c);
```

Scilab code Exa 14.32 EX14 32

```


//example 14.32
//calculate dimension of channel if it is design on the basis of Laecy theory and Kennedy's theory
clc;funcprot(0);
//given
Q=40;          //discharge
f=1;           //silt factor

//Laecey's theory
V=(Q*f/140)^(1/6);
A=Q/V;
P=4.75*Q^0.5;
D=(P-(P^2-6.944*A)^0.5)/3.472;
B=P-2.236*D;

R=5*V^2/(2*f);
```

```
S=f^(5/3)/(3340*Q^(1/6));
B=round(B);
D=round(D*100)/100;
mprintf("\n\nBy Laecey theory:");
mprintf("\nBed slope=%f.",S);
mprintf("\nWidth of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);

//Kennedy's theory
r=B/D;
m=1;          //critical velocity ratio
N=0.0225;    //rogosity coefficient
//using equation of area of trapezoidal section;Vo
    =0.55mD^0.64 and Q=A*Vo

D=(Q/8.058)^(1/2.64);
B=r*D;
B=round(B);
D=round(D*100)/100;
mprintf("\n\nBy Kennedy theory:");
mprintf("\nWidth of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

Scilab code Exa 14.33 EX14 33

```


//example 14.33
//design Laecey regime channel
clc;funcprot(0);
//given
A=100000;              //culturable area(hectare)
IR=0.4;                //intensity of irrigation in
    kharif season
IK=0.3;                //intensity of irrigation in
```

```
    rabi season
OR=1800;             //outlet discharge factor in
    kharif season
OK=800;              //outlet discharge factor in
    kharif season
l=0.1;              //conveyance loss
md=0.328;            //average diameter of material

AR=A*IR;            //area under rabi
AK=A*IK;            //area under kharif
Qr=AR/OR;
Qk=AK/OK;
Q=1.1*Qk;
f=1.76*(md)^0.5;
V=(Q*f^2/144)^(1/6);
A=Q/V;
P=4.75*(Q)^0.5;
D=(P-(P^2-6.944*A)^0.5)/3.472;
B=P-2.236*D;
S=f^(5/3)/(3340*Q^(1/6));
B=round(B*10)/10;
D=round(D*100)/100;
mprintf("\nBed slope=%f.",S);
mprintf("\nWidth of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
```

Scilab code Exa 14.34 EX14 34

```


//example 14.34
//calculate concentration at point 10 cm above the
    bed
clc;funcprot(0);
//given
```

```
D=2.8;          //depth of flow
c_=700;         //concentration at 30 cm below water
   surface
y=0.1;
a=D-0.3;
e=0.4;          //exponent in rouse equation;

c=c_*(a*(D-y)/(y*(D-a)))^e;
mprintf("concentration at point 10 cm above the bed=
   %i ppm.",c);
```

Scilab code Exa 14.35 EX14 35

```


//example 14.35
//design the distributory using Laecey theory
clc;funcprot(0);
//given
f=0.85;             //silt factor
AR=3600;            //area for rabi
AK=1400;            //area for kharif
delta_r=0.135;      //kor depth for rabi
delta_k=0.19;       //kor depth for kharif
tr=4;               //kor period for rabi
tk=2.5;             //kor period for kharif
Du_r=8.64*tr*7/delta_r;      //duty for rabi
Du_k=8.64*tk*7/delta_k;     //duty for kharif
q_r=AR/Du_r;                //discharge for rabi
q_k=AK/Du_k;                //discharge for kharif
Q=q_r;                     //since q_r>q_k
V=(Q*f^2/144)^(1/6);
A=Q/V;
P=4.75*(Q)^0.5;
D=(P-(P^2-6.944*A)^0.5)/3.472;
```

```
S=f^(5/3)/(3340*Q^(1/6));
P=round(P*100)/100;
D=round(D*100)/100;
mprintf("\nBed slope=%f.",S);
mprintf("\nPerimeter of channel section=%f m.",P);
mprintf("\nDepth of channel section=%f m.",D);
```

Chapter 15

IRRIGATION CHANNEL 2 DESIGN PROCEDURE

Scilab code Exa 15.1 EX15 1

```


//example 15.1
//design a channel by Kennedy theory using Garret's
   diagram
clc;funcprot(0);
//given
Q=7;                    //full supply discharge
N=0.0225;               //rogosity coefficient
S=1/4444;               //bed slope
m=1;                    //critical velocity ratio
s=1/2;                  //side slope

//Values of B and D are obtained by Garret's diagram
    fig. 15.3(b) and tabulated as below
B=[6 7 6.75];               //width of bed from Garret
   diagram
D=[1.5 1.35 1.38];          //depth of bed from Garret
   diagram
```

```
Vo=[0.72 0.673 0.685];    //from Garret diagram

mprintf("Bed width          Depth            Ratio of V/Vo:               Remarks");
for i=1:3
    A(i)=B(i)*D(i)+D(i)^2/2;        //Area
    V(i)=Q/A(i);                    //Velocity
    r(i)=V(i)/Vo(i);                        //ratio V/Vo
    r(i)=round(r(i)*1000)/1000;
    if i==1 then
    s='small';
else
    if (i==2) then
    s='more';

else
    s='satisfactory';
end
end
    mprintf("\n%f           %f           %f                        %s",B(i),D(i),r(i),s);
end
mprintf("\nHence, B=%f m; D=%f m.",B(3),D(3));
```

Scilab code Exa 15.2 EX15 2

```


//example 15.2
//design an irrigation channel in alluvial soil by Laecy's theory
clc;funcprot(0);
//given
Q=15;                 //Full supply discharge
```

```
f=1;                    //silt factor
s=1/2;                  //side slope of channel

//from Laecey regime channel (Fig.15.4(b)) B and D
   is obtained as;
B=15.1;
D=1.38;
//also from Fig.15.5 we get slope as
S=0.19/1000;
mprintf("Width of channel section=%f m.",B);
mprintf("\nDepth of channel section=%f m.",D);
mprintf("\nBed slope=%f.",S);
```

Scilab code Exa 15.3 EX15 3

```


//example 15.3
//design and prepare the longitudnal section;
   schedule of area statistics and channel dimension
    of irrigation channel
clc;funcprot(0);
//given
dl=157.7;                    //datum level
fsl=157;                    //full supply level of parent
    channel
bl=156;                     //bed level of parent channel
kor_r=4;                    //kor period of rabi
kor_k=2.5;                  //kor period of kharif
kord_r=13.4;                //kor depth of rabi
kord_k=19;                  //kor depth of kharif
s=0.5;                      //side slope
m=1;                        //critical velocity ratio
N=0.0225;                   //Kutter n
qo_r=8.64*7*kor_r*100/kord_r;  //outlet discharge
```

```
for rabi(calculation is wrong in book)
qo_k=8.64*7*kor_k*100/kord_k;  //outlet discharge for kharif(calculation is wrong in  book)
ca=16000;                  //culturable commanded area
Ir=0.3;                   //intensity of irrigation in rabi
Ik=0.125;                 //intensity of irrigation in rabi
Ar=Ir*ca;                 //area under rabi
Ak=ca*Ik;                 //area under kharif
q_r=Ar/qo_r;
q_k=Ak/qo_k;
q_r=round(q_r*100)/100;
q_k=round(q_k*100)/100;
mprintf("discharge neede for rabi crop=%f cumecs.",q_r);
mprintf("\ndischarge neede for kharif crop=%f cumecs.",q_k);
mprintf("\noutlet discharge factor adopted=%i hectares per cumecs.",qo_r);
//at km 5
ca=8000;            //culturable area
Ar=Ir*ca;                 //area under rabi
q_r=Ar/qo_r;
l=0.5                  //total loss after 5 km
q=q_r+l;                //total discharge
dq=1.1*q;               //desigm discharge
S=1/4000;              //slope
B=[5.5 4.9 4.55];    //Bed width
D=[0.73 0.79 0.84]; //water depth
Vo=[0.448 0.472 0.488];  //critical velocity
mprintf("\n\nBed width      water depth       area       velocity       critical velocity     C.V.R");
for i=1:3
    A(i)=B(i)*D(i)+D(i)^2/2;
    V(i)=dq/A(i);
    m(i)=V(i)/Vo(i);
```

```
    A(i)=round(A(i)*100)/100;
    V(i)=round(V(i)*1000)/1000;
    m(i)=round(m(i)*100)/100;
    mprintf("\n%f        %f        %f        %f          %f
                  %f",B(i),D(i),A(i),V(i),Vo(i),m(i))
        ;
end
B=4.55;D=0.84;
mprintf("\nhence take B=%f .; D=%f m.",B,D);
//at km 4
q=round(q*100)/100;
mprintf("\ndischarge at 5 km=%f cumecs.",q);
ca=10000;            //culturable area
Ar=Ir*ca;                //area under rabi
q_r=Ar/qo_r;
l=0.5                //total loss below 5 km
P=B+D*5^0.5;         //wetted perimeter
l1=P*1000*2/1000000;   //loss between 5 km and 4km
l2=l1+l;
q=q_r+l2;
dq=1.1*q;
q=round(q*1000)/1000;
mprintf("\ndischarge at 4 km =%f cumecs",q);
mprintf("\nother discharge are calculated and are
   tabulated as:");
x=[0:1:5];
A1=[4800 4200 3600 3300 3000 2400];
A2=[2000 1750 1500 1375 1250 1000];
S=[22.5 22.5 22.5 24 24 25];
B=[5.5 5.2 4.85 4.7 4.55 4.55];
D=[1.04 1.007 0.975 0.945 0.915 0.840];
dq=[3.56 3.17 2.8 2.6 2.4 2.02];
V=[0.570 0.555 0.538 0.530 0.521 0.484];
m=[1.015 1 1 1 1 0.992];
mprintf("\n\nBelow km      area to irrigate rabi
         area to irrigate kharif      bed slope
          bed width           water depth
   design discharge       velocity         C.V.R");
```

```
for i=1:6
    mprintf("\n%i                    %i
                                   %i
        %f               %f                 %f                 %f
                   %f            %f",x(i),A1(i),A2(i),S(i)
        ,B(i),D(i),dq(i),V(i),m(i));
end
```

Scilab code Exa 15.4 EX15 4

```


//example 15.4
//calculate the economical depth of cutting for
   cross section of channel
clc;funcprot(0);
//given
B=5;                    //bed width
t=2;                    //top width of banks
h=2.92;                 //heigth of banks from bed
n=1.5;

//sectional area of digging=sectional area of two
   banks
//By+zy^2=2(h-y)+2n(h-y)^2
//substituting the values and on simplificatio we
   get
s=poly([18.59,-13.26,1],'x','c');
y=roots(s);
//from this we get y=11.666556 and 1.5934436.
//taking
y=1.5934436;
y=round(y*10)/10;
mprintf("economical depth of cutting=%f m.",y);
```

Chapter 16

WATERLOGGING AND CANAL LINING

Scilab code Exa 16.1 EX16 1

```
//example 16.1
//design a trapezoidal concrete lined channel
clc;funcprot(0);
//given
Q=100;                  //discharge
S=25/100000;              //bed slope
N=0.016;                //rogsity coefficient
s=1.5;                  //side slope
V=1.5;                  //limiting velocity

//using manning's equation  V=(R^2/3*S^1/2)/N;
R=(V*N/(S^0.5))^(1.5);       //hydraulic mean depth

//for s=1.5;
theta=acot(1.5);
A=Q/V;
P=A/R;
```

```
//using equation of area and perimeter of trapezium
//perimeter of trapezium=b+2d(theta+cot(theta));
//area of trapezium=bd+d^2(theta+cot(theta));
//we get
y=poly([31.9,-17.1,1],'x','c');
d=roots(y);
//we get D=14.968917 and 2.1310826.
//taking
d=2.1310826;
b=P-4.18*d;
b=round(b*100)/100;
d=round(d*100)/100;
mprintf("required bed width=%f m.",b);
mprintf("\nrequired bed depth=%f m",d);
```

Scilab code Exa 16.2 EX16 2

```


//example 16.2
//design a trapezoidal concrete lined channel
clc;funcprot(0);
//given
Q=100;                    //discharge
S=25/100000;                //bed slope
N=0.016;                  //rogsity coefficient
s=1.5;                    //side slope
r=8;                      //b/d ratio

//using manning equation V=(R^2/3*S^1/2)/N;
//Perimeter=A/R
//V=Q/A and on simplification we get
d=((101/10.09)*(12.18/10.09)^(2/3))^(3/8);
b=r*d;
b=round(b);
```

```
d=round(d*100)/100;
mprintf("required bed width=%f m.",b);
mprintf("\nrequired bed depth=%f m",d);
```

Scilab code Exa 16.3 EX16 3

```


//example 16.3
//design a concrete lined channel
clc;funcprot(0);
//given
Q=45;                //discharge
S=1/10000;           //bed slope
s=5/4;               //side slope
N=0.018;             //rogosity coefficient(manning N)

//channel is assumed to be of triangular section
theta=acot(s);
//using manning equation V=(R^2/3*S^1/2)/N;
//V=Q/A;
//perimeter of trapezium=b+2d(theta+cot(theta));
//area of trapezium=bd+d^2(theta+cot(theta));
//we get
d=(Q*2.86/1.925)^(3/8);
d=round(d*100)/100;
mprintf("\nrequired depth of triangular channel=%f m
   ",d);
```

Scilab code Exa 16.4 EX16 4

```


```

```
//example 16.4
//design a concrete lined channel of trapezoidal
   section
clc;funcprot(0);
//given
Q=250;                //discharge
S=1/6000;             //bed slope
s=1.5;                //side slope
d=3;                  //limiting depth
N=0.015;              //rogosity coefficient

//using Perimeter=A/R;
//perimeter of trapezium=b+2d(theta+cot(theta));
//area of trapezium=bd+d^2(theta+cot(theta));
//Q=A*V; and on simplification
//we get
//(3b+18.81)^5/3/(b+12.54)^2/3=290.47;
//solving it by trial and error method we get
b=44.6;
mprintf("required bed width=%f m.",b);
mprintf("\nrequired bed depth=%i m",d);
```

Scilab code Exa 16.5 EX16 5

```


//example 16.5
//calculate spacing of drains
clc;funcprot(0);
//given
H=10;                //depth of impervious stratum from
    top soil
D=1.8;              //position of drain below top soil
   surface
Hw=1.5;             //depth of highest point of water
```

```
k=1D-4;          //permeability constant

//since water has to be removed in 24 hours
b=H-Hw;
a=H-D;
L=(4*k*(b^2-a^2)*100*24*3600/0.8)^0.5;
mprintf("drains should be spaced at %i m c/c.",L);
```

Scilab code Exa 16.6 EX16 6

```


//example 16.6
//calculate permeability coefficient
clc;funcprot(0);
//given
L=30;           //spacing between drans
Q=4D-6;         //discharge
a=8;
b=8.3;

k=1000000*Q*L/(4*(b^2-a^2));
k=round(k*100)/100;
mprintf("permeability coefficient=%fD-6 m/sec.",k);
```

Scilab code Exa 16.7 EX16 7

```


//example 16.7
//calculate annual average rainfall
clc;funcprot(0);
//given
```

```
L=50;           //spacing between drains
k=1D-5;         //permeability coefficient
a=10;
b=10.3;

Q=4*k*(b^2-a^2)/L;
Pav=Q*24*3600*100*100/L;
mprintf("annual average rainfall=%i cm",Pav);
```

Scilab code Exa 16.8 EX16 8

```


//example16.8
//calculate ratio of discharge at A and B;ratio of
   average rainfall at A and B
clc;funcprot(0);
//given
r1=2;          //ka/kb
r2=1/1.5;      //La/Lb
r3=5/6;        //(b^2-a^2)a/((b^2-a^2)b)

Rq=r1*r3/r2;
Rp=Rq/r2;
mprintf("ratio of discharge at A and B=%f.",Rq);
mprintf("\nratio of average rainfall at A and B=%f."
   ,Rp);
```

Scilab code Exa 16.9 EX16 9

```


//example 16.9
```

```
//decide whether it is economically feasible to provide canal lining
clc;funcprot(0);
//given
li=2.5;                    //seepage loss for lined channel
p1=25;                     //wetted perimeter for lined channel
t=12;                      //thickness of concrete lining
lf=0.02;                   //seepage loss for unlined channel
p2=20;                     //wetted perimeter for unlined  channel

//assume 1 km length of canal
//annual benifit

//(1).seepage
A1=p1*1000;                  //area of wetted perimeter
li=li*p1/1000;               //seepage loss
A2=p2*1000;                 //area of wetted perimmeter for unlined channel
lf=p2*lf/1000;             //seepage loss for unlined channel
s=li-lf;                    //saving in water loss
a1=s*p1*100000;             //annual revenue saved

//(2)maintainence
a2=0.4*25000;              //saving in maintainance cost
ts=a1+a2;                  //total annual benifit

//annual cost
A1=p2*1000;                //area of lining for unlinrd canal
C=100*A1;                  //cost of lining
//interest rate is 6%
i=0.06;
N=50;
```

```
a=(C*i*(i+1)^N)/((1+i)^N-1);  //annual cost of
    lining or capital recovery factor
bcr=ts/a;                     //benifit cost ratio
bcr=round(bcr*1000)/1000;
mprintf("\nBenifit cost ratio=%f.",bcr);
//as bcr>1
mprintf(" ;Since it is more than 1.\nHence, it is
    economically justifiable.");
```

Scilab code Exa 16.10 EX16 10

```


//example 16.10
//calculate the required depth of water to be
    applied
clc;funcprot(0);
//given
Ecd=20;             //electrical conductivity of
    drainage water
Eci=1.5;           //m mho/cm
Dc=55.5;           //consumptive use

Lr=Eci/Ecd;
D=Dc/(1-Lr);
mprintf("required depth of water to be applied=%i mm
    .",D);
```

Scilab code Exa 16.11 EX16 11

```


//example 16.11
```

```
//calculate the required depth of water to be applied
clc;funcprot(0);
//given
Eci=1.4;          // m mho/cm
Ece=11;           //saturated extract of soil
Dc=85;            //consumptive use requirement of crop

//let us assume Ecd=2Ece
Lr=Eci/(2*Ece);
Di=Dc/(1-Lr);
Di=round(Di*10)/10;
mprintf("required depth of water to be applied=%f mm.",Di);
```

Scilab code Exa 16.12 EX16 12

```


//example 16.12
//calculate average boundary shear stress;
//percentage of earth work is saved in lined section
clc;funcprot(0);
//given
s=1.5;          //side slope
Q=15;           //discharge
S=1/4000;       //bed slope
Nl=0.014;        //manning n for lined channel
Nu=0.028;        //manning n for ulined channel
fb=0.75;           //free board

//considering the perimeter of trapezoidal section
//taking minimum perimeter for given area
//i.e dP/dD=0
```

```
//we get
//A=2.1D^2; R=D/2; and P=4.2D

//for linrd channel
//Q=AR^(2/3)*S^0.5
//substituting above values we get
D=(10.0396)^(3/8);
B=0.6*D;
R=D/2;
tau=9.81*R*S*1000;
tau=round(tau*1000)/1000;
mprintf("for lined canal:");
mprintf("\naverage boundary shear stress=%f N/square
    m.",tau);
Dc=D+fb;            //total depth of cutting
A1=(B+1.5*Dc)*Dc;

//for unlined channel
//Q=AR^(2/3)*S^0.5
//substituting above values we get
D=3.08;
B=0.6*D;
R=D/2;
tau=9.81*R*S*1000;
tau=round(tau*100)/100;
mprintf("\n\nfor unlined canal:");
mprintf("\naverage boundary shear stress=%f N/square
    m.",tau);
Dc=D+fb;            //total depth of cutting
A2=(B+1.5*Dc)*Dc;
per=(A2-A1)*100/A2;
per=round(per*100)/100;
mprintf("\n\npercent saving of earth=%f percent.",
   per);
```

Scilab code Exa 16.13 EX16 13

```


//example 16.13
//design a lined canal
clc;funcprot(0);
//given
Q=100;                    //discharge
S=1/2500;                 //bed slope
V=2;                      //maximum permissible velocity
n=0.013;                  //manning n
s=1.25;                   //side slope

A=Q/V;
//from manning formula  V=(R^2/3*S^1/2)/N;
R=(V*n/S^0.5)^1.5;
P=A/R;

//now using the equation of area and perimeter of
   trapezoid
//area=D(B+2.5D)
//perimeter=B+3.2D;
//we get
y=poly([50,-33.73,1.95],'x','c');
D=roots(y);
//we get D=15.660087 and 1.6373489
//taking
D=1.6373489;
B=P-3.2*D;
B=round(B*10)/10;
D=round(D*100)/100;
mprintf("required bed width=%f m.",B);
mprintf("\nrequired bed depth=%f m",D);
```

Scilab code Exa 16.14 EX16 14

```


//example 16.14
//calculate to what extent discharge can be
   increased without changing bed slope
clc;funcprot(0);
//given
B=5;           //bed width
D=2;           //bed depth
S=1/1600;      //bed slope
n=0.015;       //manning n

A=B+2*D;       //area of lining
//let B1 and D1 be new  width and depth of bed
//for getting maximum discharge we diffrentiate Q
   and equating it to zero
//Q=S^0.5*B1D1^5/3/n
//we get
D1=45/16;
B1=9-2*D1;
Q1=S^0.5*B1*D1^5/3/n;
D1=round(D1*10000)/10000;
mprintf("new width of bed=%f m.",B1);
mprintf("\nnew depth of bed=%f m.",D1);
mprintf("\n maximum discharge=%f cumec.",Q1);
R=D;
V=R^(2/3)*S^0.5/n;
F=V/(9.81*D)^0.5;         //froud number
R=D1;
V=R^(2/3)*S^0.5/n;
F=V/(9.81*D1)^0.5;         //froud number
mprintf("\nFroud number is less than 1 in both case
   .\nHence,flow doesnot change from sub-critical to
    super critical.");
```

Scilab code Exa 16.15 EX16 15

```


//example 16.15
//calculate maximum carrying capacity of canal
//area to be irrigated
clc;funcprot(0);
//given
B=5;            //bed width
D=2.5;          //bed depth
s=1.5;          //side slope
S=1/1000;       //bed slope
n=0.016;         //manning n
k=10;           //kor period
d=150;          //field irrigation requirement

theta=acot(s);
A=B*D+D^2*(theta+1/tan(theta));
P=B+2*D*(theta+1/tan(theta));
R=A/P;
Q=A*R^(2/3)*S^0.5/n;
V=Q*k*24*3600;    //volum of water supply by channel
A=V*10/(d*10000);
Q=round(Q*100)/100;
A=round(A)*100;
mprintf("maximum carrying capacity of canal=%f cumec
    .",Q);
mprintf("\nArea to be irrigated=%f hectares.",A);
```

Chapter 17

CANAL OUTLETS

Scilab code Exa 17.1 EX17 1

```


//example 17.1
//calculate discharge through the outlet
clc;funcprot(0);
//given
D=100.0;          //F.S.L of distributory
wc=99.90;       //F.S.L of water course
L=9;            //length of pipe
d=20;           //diameter of pipe
f=0.005;      //coefficient of friction
g=9.81;       //acceleration due to gravity

H=D-wc;         //working head
C=(d/((1.5*d/(400*f)+L)*f))^0.5/20;
A=%pi*d^2/(4*10000);
q=C*A*(2*g*H)^0.5;
q=round(q*10000)/10000;
mprintf("discharge through the outlet=%f cumec.",q);
```

Scilab code Exa 17.2 EX17 2

```
//example 17.2
//design a submerged pipe
clc;funcprot(0);
//given
q=0.04;          //discharge through outlet
D=100.0;          //F.S.L of distributing canal
wc=99.90;       //F.S.L of water course
dep=1.1;        //full supply depth distributing
   canal
C=0.7;          //average value of coefficient of
   discharge
g=9.81;       //acceleration due to gravity

H=D-wc;           //available head
A=q/(C*(2*g*H)^0.5);
d=(4*A/%pi)^0.5*100;
d=round(d*10)/10;
mprintf("diameter of pipe required=%f cm.",d);
mprintf("\nuse pipe of diameter 25 cm.");
```

Scilab code Exa 17.3 EX17 3

```
//example 17.3
//design submerged pipe
clc;funcprot(0);
//given
q=0.04;          //discharge through outlet
```

```
D=100.0;           //F.S.L of distributing canal
wc=99.90;        //F.S.L of water course
dep=1.1;         //full supply depth distributing
   canal
f=0.01;           //coefficient of friction
g=9.81;       //acceleration due to gravity
L=9;          //Length of pipe

H=D-wc;          //working head
//first trial
//taking d=22.8 cm
d=22.8;
C=(d/((1.5*d/(400*f)+L)*f))^0.5/20;
A=q/(C*(2*g*H)^0.5);
d=(4*A/%pi)^0.5*100;
//second trial
C=(d/((1.5*d/(400*f)+L)*f))^0.5/20;
A=q/(C*(2*g*H)^0.5);
d=(4*A/%pi)^0.5*100;
d=round(d*100)/100;
mprintf("diameter of pipe required=%f cm.",d);
mprintf("\nprovide diameter of pipe as 25 cm.");
```

Scilab code Exa 17.4 EX17 4

```


//example 17.4
//design an open flume outlet
clc;funcprot(0);
//given
Q=0.06;            //discharge
D=0.85;            //full supply depth
Hw=15;              //available working head
Bt=7;C=1.6;       //let us choose
```

```
H=(Q*100/(C*Bt))^(2/3);
mh=0.2*H;          //minimum modular head
mh=round(mh*1000)/1000;
mprintf("minimum modular head=%f m. < available
   working head.\nhemce,design is safe.",mh);
o=H/D;
o=round(o*1000)/1000;
mprintf("\nsetting of outlet=%f. <0.9.\nhence,outlet
    will work as hyper propotional outlet.",o);
```

Chapter 18

CANAL REGULATION WORKS

Scilab code Exa 18.1 EX18 1

```
//example 18.1
//design Sarda type fall
clc;funcprot(0);
//given
Q=40;                  //full supply discharge
sl_u=218.3;            //supply level at upstream
sl_d=216.8;            //supply level at downstream
D=1.8;                //suplly depth
L=26;                 //bed width
bl_u=216.5;           //bed level upstream
bl_d=215;             //bed level downstream
drop=1.5;

//from  the eqauation; Q=1.99LH^1.5*(H/B)^(1/6);
//B=0.55*(H+d)^0.5;
//H+d=drop+D;
//we get
```

```
H=(0.774)^0.6;
d=3.3-H;
Hc=D-H;
d=round(d*100)/100;
H=round(H*100)/100;
Hc=round(Hc*100)/100;
mprintf("H=%f m.\nd=%f m.",H,d);
mprintf("\ncrest height above bed=%f m.",Hc);

//adopt trapezoidal crest
B=1;                    //top width
mprintf("\n\nD/S batter=1:3; U/S batter=1:8.");
Va=Q/((27+D)*D);
vh=Va^2/(2*9.81);
tel_up=sl_u+vh;
crest=sl_u-H;
E=sl_u-crest;
mprintf("\nR.L of crest=%f m.",crest);
mprintf('\nE=%f m.',E);
//design of cistern
x=(E*drop)^(2/3)/4;                //depth of cistern
lc=5*(E*drop)^0.5;                 //length of cistern
cb=bl_d-x;
x=round(x*100)/100;
cb=round(cb*1000)/1000;
lc=round(lc*10)/10;
mprintf("\n\ndepth of cistern=%f m.",x);
mprintf("\nlength of cistern=%f m.",lc);
mprintf("\nR.L of bed of cistern=%f m.",cb);
mprintf("\nkeep cistern at R.L 214.69.");
//design of impervious floor
Hs=2.44;                //seepage head
c=8;                   //Bligh's coefficient
li=Hs*c;
d1=1;d2=1.6;
vl=2*(d1+d2);
lh=li-vl;
```

```
mprintf("\n\ndesign of impervious floor:");
mprintf("\nprovide upstream cut-off=%i m.;
   downstream cut-off=%f m.",d1,d2);
mprintf("\nlength of horizontal impervious floor=%f
   m.",lh);
mprintf("\nprovide 15 m length impervious floor.");
ld=2*(D+1.2)+drop;
mprintf("\nminimum length of impervious floor to the
    d/s of toe of crest wall=%f m.",ld);
mprintf("\nprovide ld=8 m.");
bl=15-8;
mprintf("\nthe balance of the length %i m is to be
   provided under and u/s of the crest.",bl);

tcl=15+2*(1+16);
mprintf("\n\nuplift pressure is counter balanced by
   weigth of water.\n hence provide thickness of 0.4
    m.");
rho=2.24;
static=2.44*(1-0.446)+x;
t=static/(rho-1);
t=round(t*100)/100;
mprintf("\nfor other points; thickness required =%f
   m.",t);
mprintf("\nprovide thickness of 1.40 m.");
mprintf("\nat downstream end of floor provide
   thickness of 0.6 m overlaid by 0.2 m brick
   pitching.");

n=d2/(Hs*5);                //n=1/%pi*(lambda)^0.5
//from khosla exit curve we get
alpha=10.5;
lambda=(1/(%pi*n))^2;
alpha=((2*lambda-1)^2-1)^0.5;
b=alpha*d2;
b=round(b*100)/100;
mprintf("\n\nchecking of floor thickness by khosla
   theory:");
```

```
mprintf("\nlength of floor provided=%f m. > length
    by Bligh theory.",b);
b=15;
d2=1.8;
alpha=b/d2;
n=0.145;
Ge=Hs*n/d2;
Ge=round(Ge*10)/10;
mprintf("\nexit gradient after increase in depth cut
    -off=%f. which is in permissible limit",Ge);
mprintf('\nprovide depth cut-off to 1.8 m.');
//calculation of pressure
mprintf("\n\ncalculation of pressure:");
mprintf("\nU/S cut-off:");
d1=1;
b=15;
alpha_=d1/b;
fic1=100-24;
fid1=100-17;
t=0.4;
fic1=fic1+(fid1-fic1)*t/d1;
mprintf("\ncorrected fic1=%f percent.",fic1);
mprintf("\nD/S cut-off wall:");
d2=1.8;
b=15;
alpha_=d1/b;
fie2=31;
fid2=21.5;
t=0.6;
fie2=fie2-(fie2-fid2)*t/1.8;
fie2=round(fie2*10)/10;
mprintf("\ncorrecte fie2=%f percent.",fie2);
//calculation of thickness
mprintf("\n\nprovide a minimum thickness of 0.4 m
    for u/s floor.");
pre=fie2+(fic1-fie2)*8/b;
static=pre*Hs/100+x;
t=static/(rho-1);
```

```
t=round(t*100)/100;
mprintf("\nthickness at d/s toe of crest=%f m.",t);
mprintf("\nprovide thickness of 1.4 m thick concrete
    overlaid by 0.2 m brick pitching.");
pre=fie2+(fic1-fie2)*5/b;
static=pre*Hs/100+x;
t=static/(rho-1);
t=round(t*100)/100;
mprintf("\nthickness at 3 m from d/s toe of crest=%f
    m.",t);
mprintf("\nprovide thickness of 1.2 m thick concrete
    overlaid by 0.2 m brick pitching.");
pre=fie2+(fic1-fie2)*2/b;
static=pre*Hs/100;                   //calculation is
   wrong in book
t=static/(rho-1);
t=round(t*100)/100;
mprintf("\nthickness at 6m from d/s toe of crest=%f
   m.",t);
mprintf("\nprovide thickness of 0.7 m thick concrete
    overlaid by 0.2 m brick pitching.");
//design of downstream wings
wing=6*(E*drop)^0.5;
hw=D+0.5;
mprintf("\n\nheigth of top of downstream wings above
    the bed=%f m.",hw);
projec=hw*3;
mprintf("\nlength of warped wing measured along
   centre line of canal=%f m.",projec);
//downstream pitching
l=9+2*1.5;
mprintf("\n\nlength of bed pitching=%f m.",l);
mprintf("\nlength of sloping pitching=7 m.\nlength
   of horizontal pitching=6 m.");
mprintf("\nprovide one toe wall of 1 m depth and 0.4
    m width.");
mprintf("\nside pitching is curtailed at 45 degree
   from the end of bed pitching in plan.\nsupprot
```

```
    the side pitching on toe wall 0.4 m thick and 1 m
     deep. ");
//energy dissipators
q=Q/L;
dc=(q^2/9.81)^(1/3);
mprintf("\n\nsize and position of friction blocks:")
    ;
L=2*dc;
w=dc;
h=dc;
di=1.5*dc;
L=round(L*10)/10;
w=round(w*10)/10;
h=round(h*10)/10;
di=round(di);
mprintf("\nlength of block=%f m.\nwidth of block=%f
    m.\nheight of block=%f m.\ndistance from toe of
    crest=%f m.",L,w,h,di);
mprintf("\nprovide two rows staggered ata distance
    of 1 m from toe of crest.");
mprintf("\nsize and position of cube blocks:");
L=D/10;
w=D/10;
h=w;
L=round(L*10)/10;
w=round(w*10)/10;
h=round(h*10)/10;
mprintf("\nlength of block=%f m.\nwidth of block=%f
    m.\nheight of block=%f m.",L,w,h);
mprintf("\nprovide two rows staggered at the end of
    impervious floor.");
//u/s approach
r=6*H;
mprintf("\n\nprovide wing wall segmental with 5 m
    radius subtending angle of 60 degree at the
    centre.");
```

Scilab code Exa 18.2 EX18 2

```
//example 18.2
//design an unflumed straight glacis non-meter fall
clc;funcprot(0);
//given
Q=40;                    //full supply discharge
sl_u=218.3;              //supply level at upstream
sl_d=216.8;              //supply level at downstream
D=1.8;                  //suplly depth
L=26;                   //bed width
bl_u=216.5;             //bed level upstream
bl_d=215;               //bed level downstream
drop=1.5;
Ge=1/6;                //permissible exit gradient

//design of crest
mprintf("design of crest:");
E=(Q/(1.84*L))^(2/3);
V=Q/((L+D)*D);
vh=V^2/(2*9.81);
tel_up=sl_u+vh;
cl=tel_up-E;
w=2*E/3;
w=round(w*10)/10;
mprintf("\nlength of crest=%f m.",L);
mprintf("\nwidth of crest=%f m.",w);
//design of cistern
q=Q/L;
H1=1.5;
//from blench curve
Ef2=1.44;
```

```
cistern=sl_d+0.03-1.25*Ef2;
mprintf("\n\nR.L of cistern=%f m. > d/s bed level.",
   cistern);
mprintf("\nkeep R.L of cistern at 214.5 m.");
l=6*Ef2;
mprintf("\nlength of cistern=%f m.",l);
mprintf("\nprovide cistern of 9 m length ");
d=bl_d-214.5;
mprintf("\ndepth of cistern=%f m.",d);

//design of impervious floor
d1=D/3;
mprintf("\n\ndesign of impervious floor:");
mprintf("\nprovide 0.4 m wide and 1 m deep curtain
   wall at u/s.");
d2=D/2;
mprintf("\nprovide 0.4 m wide and 1 m deep curtain
   wall at d/s.\nthe curtain wall will project the
   above the d/s bed by 0.18 m.");
Hs=cl-bl_d;
d2=1;
n=d2*Ge/Hs;          //n=1/(%pi*(lambda)^0.5)
//from khosla exit curves we get
alpha=40;
lambda=(1/(%pi*n))^2;
alpha=((2*lambda-1)^2-1)^0.5;
b=alpha*d2;
//since length is to excessive
d2=2;
n=d2*Ge/Hs;          //n=1/(%pi*(lambda)^0.5)
//from khosla exit curves we get
alpha=10;
lambda=(1/(%pi*n))^2;
alpha=((2*lambda-1)^2-1)^0.5;
b=alpha*d2+1;
mprintf("\ntotal length=%i m.\nlength of cistern=9 m
   .\nlength of d/s glacis=5.88 m.\nwidth of crest
   =0.6 m.\nlength of u/s glacis=0.47 m.\nbalance to
```

```
    be provided to u/s of the u/s glacis=4.05 m.",b)
   ;

//pressure calculations
mprintf("\n\npressure calculations:");
mprintf("\nupstream curtain wall:");
d1=1;b=20;
alpha_=d1/b;
t=0.3;
fic1=100-22;
fid1=100-15;
corec=(fid1-fic1)*t/d1
fic1=fic1+corec;
mprintf("\ncorrected fi_c1=%f percent.",fic1);
mprintf("\ndownstream curtain wall:");
d2=2;b=20;
alpha_=d2/b;
t=0.5;
fie=29;
fid=21;
corec=(fie-fid)*t/d2
fie=fie-corec;
mprintf("\ncorrected fi_e=%f percent.",fie);
mprintf("\ntoe of glacis:");
//assuming linear variation of pressure
p=fie+(80-fie)*9/20;
mprintf("\npressure at downstream of the glacis=%f
   percent.",p);

//floor thickness
rho=2.24;
mprintf("\n\nfloor thickness:\nprovide minimum
   thickness of 0.3 m at the u/s floor.");
static=p*2.44/100+(bl_d-214.5);
t=static/(rho-1);
t=round(t*100)/100;
mprintf("\nfloor thickness required at toe of glacis
   =%f m.\nprovide 1.5 m thick floor for length of 3
```

```
m.",t);
p=fie+(80-fie)*6/20;
static=p*2.44/100+(bl_d-214.5);
t=static/(rho-1);
t=round(t*100)/100;
mprintf("\nfloor thickness required at 3m from toe of glacis=%f m.\nprovide 1.3 m thick floor from 3 m to 6.5 m from toe of glacis.",t);
t=0.27*2.44/(rho-1);
t=round(t*100)/100;
mprintf("\nthickness of d/s end of cistern=%f m.\nprovide thickness of 0.6 m at d/s end of floor.",t);

//design of d/s protection
mprintf("\n\nno bed protection is needed as deflector wall is provided.");
sp=3*D;
mprintf("\nlength of side protection=%f m.\nprovide 5.5 m length of 20 cm thick brick pitching beyond impervious floor.\npitching will rest on toe wall 0.4 m wide and 0.9 m deep.\nprovide 0.4 m wide profile at the end of pitching",sp);
//design of u/s approach
mprintf("\n\nu/s wing wall is splayed at 45 degree from u/s end of impervious floor.\nextend 1 m into earthen banks from line of F.S.L.");
```

Scilab code Exa 18.3 EX18 3

```

//example 18.3
//design a cross -regulator and head regulatorfor a distributory channel
clc;funcprot(0);
```

```
//givrn
Q=100;              //discharge of parent channel
Qd=15;              //discharge ofdistributory
fsl_u=218.1;        //F.S.L of upstream parent channel
fsl_d=217.9;        //F.S.L of downstream of parent channel
bw_u=42;            //bed width of parent channel upstream
bw_d=38;            //bed width of parent channel downstream
hw=2.5;             //depth of water in parent channel
fsl_dis=217.1;      //F.S.L of distributory
hw_dis=1.5;         //depth of water in distributory
Ge=1/5;             //permissible exit gradient

//design of cross regulator
mprintf("DESIGN OF CROSS-REGULATOR::");
//design of crest and waterway
mprintf("\n\ndesign of crest and waterway:");
cl=fsl_u-hw;
h=fsl_u-fsl_d;
d=fsl_d-cl;
C1=0.557;C2=0.8;
L=Q/(2*C1*(2*9.81)^0.5*h^1.5/3+C2*d*(2*9.81*h)^0.5);
L=round(L*10)/10;
mprintf("\ncrest level=%f m.",cl);
mprintf("\nlength of crest=%f m.",L);
mprintf("\nprovide 4 bays of 7 m each with a clear water-way.");
tw=28+4.5;
mprintf("\nprovide 3 piers of 1.5 m width each.\ntotal width of cross regulator=%f m.",tw);
//design of d/s floor
L=28;
q=Q/L;
Hl=fsl_u-fsl_d;
Ef2=1.89;            //from blench curve
fl_d=fsl_d-Ef2;
```

```
mprintf("\n\ndesign of d/s floor:");
mprintf("\nd/s floor level=%f m.; which is higher
    than d/s bed level.\nadopt floor level =d/s bed
    level=215.40 m.",fl_d);
Ef1=Ef2+Hl;
//from specific energy curve
D1=0.7;D2=1.65;
cil=5*(D2-D1);      //cistern length
tl=2*16/3;
tl=round(tl*10)/10;
mprintf("\ncistern length =%f m.\nlength of d/s
    floor=%f m.",cil,tl);
//design of impervious floor
d1=hw/3+0.6;              //depth of u/s cut-off
w=0.5;                    //width of cut-off
d2=hw/2+0.6;              //deth of d/s cut-off
d2=2;                     //keep
Hs=fsl_u-(fsl_d-hw);      //maximum static head
n=Ge*d2/Hs;               //n=1/%pi*(lambda)^0.5;
//from exit gradient curves we get
alpha=8;n=0.148;
b=alpha*d2;
mprintf("\n\ndesign of impervious floor:");
mprintf("\ntotal length of impervious floor=%i m.;
    which is divided as-",b);
mprintf("\nd/s floor length=10.6 m.\nd/s glacis
    length with 2:1 slope=0.4 m.\nbalance to be
    provided upstream=5 m.");
d1=1.5;b=16;
alpha_=d1/b;
//hence
fic1=100-28;
fid1=100-19;
t=0.5;
fic1=fic1+(fid1-fic1)*t/d1;
mprintf("\n\npressure calculation:\nupstream cut-off
    :\npressure =%f percent.",fic1);
d2=2;b=16;
```

```
alpha_=d2/b;
//hence
t=0.6;
fie2=31;fid2=22;
fie2=fie2-(fie2-fid2)*t/d2;
mprintf("\ndownstream cut-off:\npressure=%f percent.
    ",fie2);
t=10.6;
p=fie2+(fic1-fie2)*t/b;
p=round(p*10)/10;
mprintf("\ntoe of glacis:\npressure=%f percent.",p);
mprintf("\n\nthickness of floor:\nminimu thickness
    for u/s floor=0.5 m.");
rho=2.24;
t=fie2*2.7/(100*(rho-1));
t=round(t*100)/100;
mprintf("\nthickness of floor near d/s cut-off=%f m
    .\nprovide 0.7 m thick floor for last 2.1 m
    length.",t);
t=1.6/(rho-1);
t=round(t*100)/100;
mprintf("\nthickness of floor at toe of glacis=%f m.
    ",t);
t=6.6;
p=fie2+(fic1-fie2)*t/b;
t=p*2.7/(100*(rho-1));
t=round(t*100)/100;
mprintf("\nthickness of floor at 4 m from toe of
    glais=%f m.\nprovide 1.1 m thick floor for next 2
     m length",t);
t=4.6;
p=fie2+(fic1-fie2)*t/b;
t=p*2.7/(100*(rho-1));
t=round(t*100)/100;
mprintf("\nthickness of floor at 6 m from toe of
    glais=%f m.\nprovide 0.9 m thick floor for next
    2.5 m length",t);

```

```
//design of u/s protection
d1=hw/3+0.6;
v=d1;
v=round(v*100)/100;
mprintf("\n\ndesign of u/s protection:\nvolume of block protection=%f cubic metre/metre.",v);
mprintf("\nkeep thickness of protection=1 m.\nprovide 0.8mx0.8mx0.6m thick concrete blocks over 0.4 m thick apron in length of 0.6 m.");
cu=2.25*d1;
cu=round(cu*100)/100;
mprintf("\ncubic content of launching apron=%f cubic metre/metre.\nprovide 1 m thick and 3.5 m long launching apron.",cu);
//design of d/s protection
d2=hw/2+0.6;
v=d2;
v=round(v*100)/100;
mprintf("\n\ndesign of d/s protection:\nvolume of inverted filter=%f cubic metre/metre.",v);
mprintf("\nkeep thickness of concrete block=0.6 m.\nprovide 2 rows of 0.8mx0.8mx0.6m thick concret blocks over 0.6 m graded filter for length of 1.6 m.");
cu=2.25*d2;
cu=round(cu*100)/100;
mprintf("\nlaunching apron volume=%f cubic metre/metre.\nprovide 1 m thick launching apron for length of 4.5 m.\nprovide a toe wall 0.4 m wide and 1.5 m deep between filter and launching apron.",cu);

//design of head regulator
mprintf("\n\n\nDESIGN OF DISTRIBUTORY HEAD REGULATOR::");
//design of crest and waterway
mprintf("\n\ndesign of crest and waterway:");
cl=fsl_u-hw+0.5;
```

```
h=fsl_u-fsl_dis;
d=fsl_dis-cl;
C1=0.557;C2=0.8;
L=Qd/(2*C1*(2*9.81)^0.5*h^1.5/3+C2*d*(2*9.81*h)^0.5);
L=round(L*100)/100;
mprintf("\ncrest level=%f m.",cl);
mprintf("\nlength of crest=%f m.",L);
mprintf("\nprovide 2 bays of 3.5 m each with a 1 m thick pier in between.");
tw=8;
mprintf("\ntotal width of cross regulator=%f m.",tw);
//design of d/s floor
L=7.5;
q=Q/L;
Hl=fsl_u-fsl_dis;
Ef2=1.58;          //from blench curve
fl_d=fsl_dis-Ef2;
mprintf("\n\ndesign of d/s floor:");
mprintf("\nd/s floor level=%f m.;\nkeepR.L of d/s floor=215.50 m.",fl_d);
Ef1=Ef2+Hl;
//from specific energy curve
D1=0.42;D2=2.55;
cil=5*(D2-D1);      //cistern length
tl=2*14/3;
mprintf("\ncistern length =%f m.",cil);

//design of impervious floor
d1=hw/3+0.6;            //depth of u/s cut-off
w=0.5;                  //width of cut-off
d2=hw_dis/2+0.6;            //deth of d/s cut-off
d2=2;                   //keep
Hs=fsl_u-215.5;       //maximum static head
n=Ge*d2/Hs;             //n=1/%pi*(lambda)^0.5;
//from exit gradient curves we get
alpha=7;n=0.154;
```

```
b=alpha*d2;
mprintf("\n\ndesign of impervious floor:");
mprintf("\ntotal length of impervious floor=%i m.;
   which is divided as-",b);
mprintf("\nlength below the toe of glacis=10.5 m\
   nlength of d/s glacis at 2:1 slope=1.2 m.\nwidth
   of crest=1 m.\nlength of u/s glacis at 1:1 slope
   =0.5 m.\nu/s floor:balnce=0.8 m.");
d1=1.5;b=16;
alpha_=d1/b;
//hence
fic1=100-28;
fid1=100-19;
t=0.5;
fic1=fic1+(fid1-fic1)*t/d1;
mprintf("\n\npressure calculation:\nupstream cut-off
   :\npressure =%f percent.",fic1);
d2=2;b=16;
alpha_=d2/b;
//hence
t=0.6;
fie2=31;fid2=22;
fie2=fie2-(fie2-fid2)*t/d2;
mprintf("\ndownstream cut-off:\npressure=%f percent.
   ",fie2);
t=10.6;
p=fie2+(fic1-fie2)*t/b;
p=round(p*100)/100;
mprintf("\ntoe of glacis:\npressure=%f percent.",p);
mprintf("\n\nthickness of floor:\nminimu thickness
   for u/s floor=0.5 m.");
rho=2.24;
t=p*2.6/(100*(rho-1));
t=round(t*100)/100;
mprintf("\nthickness under the crest=1 m.");
mprintf("\nthickness of floor at toe of glacis=%f m.
   ",t);
t=9.5;
```

```
p=fie2+(fic1-fie2)*t/b;
t=p*2.7/(100*(rho-1));
t=round(t*100)/100;
mprintf("\nthickness of floor at 2 m from toe of
    glais=%f m.\nprovide 1.1 m thick floor for next 4
     m length",t);
t=4.5;
p=fie2+(fic1-fie2)*t/b;
t=p*2.7/(100*(rho-1));
t=round(t*100)/100;
mprintf("\nthickness of floor at 6 m from toe of
    glais=%f m.\nprovide 0.9 m thick floor for next
    2.5 m length",t);
t=2;
p=fie2+(fic1-fie2)*t/b;
t=p*2.7/(100*(rho-1));
t=round(t*100)/100;
mprintf("\nthickness of floor at 8.5 m from toe of
    glais=%f m.\nprovide 0.7 m thick floor for next 2
     m length",t);

//design of upstream protection
d=hw/3+0.6;
d=round(d*10)/10;
mprintf("\n\ndesign of u/s protection:\nu/s scour
    depth=%f m.\nprovide same protection as in cross
    regulator",d);

//design of d/s protection
d2=hw_dis/2+0.6;
v=d2;
mprintf("\n\ndesign of d/s protection:\nvolume of
    inverted filter=%f cubic metre/metre.",v);
mprintf("\nkeep thickness of concrete block=0.5 m.\
    nprovide 2 rows of 0.8mx0.8mx0.5m thick concret
    blocks over 0.5 m thick graded filter.");
cu=2.25*d2;
mprintf("\nlaunching apron volume=%f cubic metre/
```

```
metre.\nprovide 1 m thick launching apron for
length of 3.5 m.\nprovide a masonary toe wall 0.4
 m wide and 1.2 m deep between filter and
launching apron.",cu);
```

Chapter 19

CROSS DRAINAGE WORKS

Scilab code Exa 19.1 EX19 1

```


//example 19.1
//design an expansion transition for canal by Mitra's method
clc;funcprot(0)
//given
Lf=16;                //length of flume
Bf=9;               //width of throat
Bo=15;              //width of canal

//width at any distance x from flumed section is given by
//Bx=Bo*Bf*Lf/(Lf*Bo-(Bo-Bf)x)
//on solving we get
//Bx=2160/(240-6x)

x=[2:2:16];         //distance
mprintf("width at any distance x from flumed section :");
for i=1:8
```

```
    Bx(i)=2160/(240-6*x(i));
    Bx(i)=round(Bx(i)*100)/100;
    mprintf('\n%f',Bx(i));
end
```

Scilab code Exa 19.2 EX19 2

```


//example 19.2
//design an expansion transition for canal by
   Chaturvedi's method
clc;funcprot(0);
//given;
Lf=16;                //length of flume
Bf=9;                 //width of throat
Bo=15;                //width of canal

x=[2:2:16];           //distance

//distance x is related as x=Lf*Bo^(2/3)(1-(Bf/Bx)
   ^1.5)/(Bo^1.5-Bf^1.5)
//on solving we get
//(Bf/Bx)^1.5=1-(x/29.893)   (relation is misprinted
    in book)
//let (Bf/Bx)^1.5=r

mprintf("width at any distance x from flumed section
   :");
for i=1:8
    r(i)=1-(x(i)/29.893);        //Bf/Bx^(1.5)
    R(i)=r(i)^(2/3);             //Bf/Bx
    Bx(i)=Bf/R(i);
    Bx(i)=round(Bx(i)*100)/100;
    mprintf("\n%f.",Bx(i));
```

```
end
```

Scilab code Exa 19.3 EX19 3

```
//example 19.3
//design a syphon aqueduct
clc;funcprot(0);
//given
Q=25;                //design discharge of canal
B=20;                //bed width of canal
D=1.5;               //depth of water in canal
bl=160;              //bed level of canal
hfq=400;            //high flood discharge of drainage
hfl=160.5;          //high flood level of drainage
bl_drain=158;       //bed level of drainage
gl=160;             //general ground level

//desing of drainage water-way
P=4.75*(hfq)^0.5;      //laecey P-Q formula
mprintf("design of drainage water-way:\nwetted
   perimeter of river=%i m.\nprovide 13 spans of 6 m
    each,separated by 12 piers each of 1.25 m thick.
   ",P);
t=78+15;
mprintf("\ntotal length of water-way=%i m.",t);
v=2;                  //velocity through syphon
hb=hfq/(78*v);
ac=hfq/(6*2.5*1.3);    //calculation is wrong in book
hb=round(hb*100)/100;
ac=round(ac*100)/100;
mprintf("\nheight of barrels=%f m.\nprovide
   rectangular barrels 6 m wide and 2.5 m high.\
   nactual velocity through barrels=%f m/sec.",hb,ac
```

```
);

//design of canal waterway
mprintf("\n\ndesign of canal waterway:\nType 3 aqueduct is adopted.");
l1=B-10;
l2=(20-10)*3/2;
mprintf("\nproviding a splay 2:1 in expansion,length of contraction transition=%i m.\nproviding a splay of 3:1 in expansion,length of expansion transition=%i m.",l1,l2);
mprintf('\nIn transition side slopes are warped from original slope of 1.5:1 to vertical.');

//design of levels of different sectionn
mprintf("\n\ndesign of levels of different sectionn:\nat section 4-4:");
A=(B+1.5*D);         //area
V=Q/A;               //velocity of flow
vh=V^2/(2*9.81);     //velocity head
ws=gl+D;             //R.L of water surface
tel=ws+vh;
tel=round(tel*1000)/1000;
mprintf("\nR.L of T.E.L=%f m.\n at section 3-3:",tel);
A=10*D;                        //area of trough
V=Q/A;                         //velocity
vh1=V^2/(2*9.81);              //velocity head
le=0.3*(vh1-vh);    //loss of head in expansion from section 3-3 to 4-4
tel=tel+le;
rlw=tel-vh1;
rlb=rlw-D;
tel=round(tel*1000)/1000;
rlb=round(rlb*1000)/1000;
mprintf("\nelevation of T.E.L=%f m.\nR.L of bed to maintain constant water depth=%f m.",tel,rlb);

```

```
//at section 2-2
R=A/P;
N=0.016;
S=V^2*N^2/R^(4/3);                  //from manning's formula
L=93;                               //length of trough
hl=L*S;                              //head loss
tel=tel+hl;
rlw=tel-vh1;
rlb=rlw-D;
tel=round(tel*1000)/1000;
rlb=round(rlb*1000)/1000;
mprintf("\nat section 2-2:\nR.L of T.E.L=%f m.\nR.L of bed to maintain constant water depth=%f m.",tel,rlb);

//at section 1-1
hl=0.2*(vh1-vh);    //loss of hed in contraction transition
tel=tel+hl;
rlw=tel-vh;
rlb=tel-D;
tel=round(tel*1000)/1000;
rlb=round(rlb*1000)/1000;
mprintf("\nat section 1-1:\nR.L of T.E.L=%f m.\nR.L of bed to maintain constant water depth=%f m.",tel,rlb);

//design of contraction transition
//it is designed on the basis of chaturvedi's formula
Bo=20;
Bf=10;
L=10;
//from chaturvedi formula we get relation between x and Bx as: x=15.45(1-(10/Bx)^1.5);
Bx=[10:1:20];
mprintf("\n\ndesign of contraction transition on the
```

```
    basis of chaturvedi formula:\nBx              x");
for i=1:11
    x(i)=15.45*(1-(10/Bx(i))^1.5);
    x(i)=round(x(i)*100)/100;
    mprintf("\n%i          %f",Bx(i),x(i));
end

//design of expansion transition on the basis of
    chaturvedi formula
L=15;
Bf=10;Bo=20;
//from chaturvedi formula we get relation between x
    and Bx as: x=23.15(1-(10/Bx)^1.5);
mprintf("\n\ndesign of expansion transition on the
    basis of chaturvedi formula:\nBx              x");
for i=1:11
    x(i)=23.15*(1-(10/Bx(i))^1.5);
    x(i)=round(x(i)*100)/100;
    mprintf("\n%i          %f",Bx(i),x(i));
end

//design of trough
mprintf("\n\ndesign of the trough:");
mprintf("\nflumed water way of canal=10 m.\ntrough
    carrying canal will divide into two compartments
    each 5 m wide an dseparated by 0.3 m thick
    partiions.\nheigth of trough will be = 2 m.\
    ntrough iss constructed using monolithic
    reinforced concrete.\nthe outer and inner walls
    ca be kept 0.4 m thick.\nthus,outer width of
    trough = 11.1 m.");

//head loss through syphon barrels
V=2.05;         //velocity through barrels
f1=0.505;       //coefficient of loss of head at
    entry
a=0.00316;b=0.030;
R=(6*2.5)/(2*(6+2.5));
```

```
f2=a(1+b/R);
L=11.1;           //length of barrel
h=(1+f1+f2*L/R)*V^2/(2*9.81);
hfl_up=hfl+h;
h=round(h*1000)/1000;
hfl_up=round(hfl_up*1000)/1000;
mprintf("\n\nhead loss through syphon barrels=%f m.\nupstream H.F.L=%f m.",h,hfl_up)

//uplift pressure on the roof
bt=gl-0.4;              //R.L of bottom of the trough
hl=0.505*V^2/(2*9.81);
u=hfl_up-hl-159.6;
up=u*9.81;
mprintf("\n\nuplift pressure on the roof=%f kN/square m.\ntrough slab is 0.4 m thick and exert a downward load of 9.42 kN.",up);
mprintf("\nth ebalance of the uplift pressure has to be resisted by bending action of trough slab.\nso,reinforcement has to be provided at the top of the slab.");

//uplift on the floor of the barrel and its design
//(a) static head
mprintf("\n\nuplift on the floor of the barrel and its design:\n(a) static head:");
bf=bt-2.5;            //R.L of barrel floor
t=0.8;              //tentative thickness of floor
bot=bf-t;
static=bl_drain-bot;
static=round(static*100)/100;
mprintf("\nstatic uplift on the floor=%f m.",static);

//(b) seepage head
L=10;             //length of u/s transition
bs=3;             //half the barrel span
df=11;            //end drainage floor
```

```
tcl=24;           //total creep length
tsh=161.5-bl_drain;  //total seepage head
rs=tsh*(1-13/tcl);    //residual seepage at B
tu=(static+rs)*9.81;
tu=round(tu*100)/100;
mprintf("\n(b) seepage head:\ntotal uplift=%f kN/
   square m.\nprovide thickness of floor 0.8 m",tu);
bending=tu-17.58;
bending=round(bending*100)/100;
mprintf("\nuplift to be resisted by bending action
   of floor=%f kN/square m.",bending);

//design of cut-off and protection works for
   drainage floor
mprintf("\n\ndesign of cut-off and protection works
   for drainage floor:");
Q=400;f=1;
R=0.47*(Q/f)^(1/3);
d_up=1.5*R;                        //depth of u/s cut-
   off
bot_up=hfl_up-d_up;               //R.L of bottom of u
   /s cut-off
d_down=1.5*R;                     //depth of d/s cut-
   off
bot_down=hfl-d_down;             //R.L of bottom of d/
   s cut-off
l_down=2.5*(bl_drain-bot_down);
l_down1=2*(bl_drain-bot_up);
bot_up=round(bot_up*100)/100;
bot_down=round(bot_down*100)/100;
l_down=round(l_down);
l_down1=round(l_down1);
mprintf("\nR.L of bottom of u/s cut-off=%f m.\nR.L
   of bottom of d/s cut-off=%f m.",bot_up,bot_down);
mprintf("\nlength of d/s protection consisting of 40
    cm brick pritching=%f m.\npitching is supported
   by toe wall 0.4 m wide and 1.5 m deep at its d/s
   end.\nlength of d/s protection consisting of 0.4
```

```
cm brick pritching=%f m.\npitching is supported
by toe wall 0.4 m wide and 1 m deep at its u/s
end.",l_down,l_down1);
```

Chapter 20

RIVER ENGINEERING

Scilab code Exa 20.1 EX20 1

```


//example 20.1
//design  a guide bank required for a bridge in a
   river
//calculate volume of stone required per m length of
    guide bank
clc;funcprot(0);
//given
Q=50000;          //discharge
f=1.1;            //silt factor
bl=130;           //bed level of river
hfl=140;          //high flood level
L=4.75*(Q)^0.5;
L=L+212;           //providing 20 percent more length
L_up=5*L/4;        //upstream length of guide bund
L_down=L/4;        //downstream length of guide bund
r_up=0.45*L;       //radius of upstream curved head
mprintf("upstream length of guide bund=%i m.",L_up);
mprintf("\ndownstream length of guide bund=%i m.",
   L_down);
```

```
mprintf("\nupstream radius of curved head=%i m.;it can be carved at 145 degrees.",r_up);
mprintf("\ndownstream radius of curved head=287m.;it can be carved at 60 degrees.");

fb=1.5;        //free board
ltop=fb+hfl;   //level of top of guide bund
mprintf("\n\nlevel of top of guide bund=%f m.",ltop);
mprintf("\nadopt top level=142 m.");
ltop=142;
Hr=ltop-bl;
mprintf("\nkeep top width=4 m. and side slope as 2:1.");
T=0.06*(Q)^(1/3);          //thickness of stone pitching
T=round(T*100)/100;
mprintf("\n\nThickness of stone pitching=%f m.",T);
R=0.47*(Q/f)^(1/3);        //depth of scour
Rmax=1.25*R;               //maximum scour
rl=hfl-Rmax;               //R.L at maximum anticipated cover
D=bl-rl;                   //depth of maximum scour
Lapron=1.5*D;
R=round(R*100)/100;
Lapron=round(Lapron*100)/100;
mprintf("\ndepth of scour=%f m.",R);
mprintf("\n\nfor straigtht reach of guide band:");
mprintf("\nlength of apron=%f m.",Lapron);
Rmax=1.5*R;
rl=hfl-Rmax;
D1=bl-rl;
Lapron=1.5*D1;
R=round(R*100)/100;
mprintf("\n\nfor curvilinear transition portion of guide band:");
mprintf("\nlength of apron=%f m.",Lapron);
T1=1.9*T;
```

```
T1=round(T1*10)/10;
mprintf("\nthickness of apron=%f m.",T1);
mprintf("\n\nvolume of stones:");
ss=5^0.5*(141-130)*T;
as=5^0.5*D*1.25*T;
ss=round(ss*100)/100;
as=round(as*100)/100;
mprintf("\nat shank:");
mprintf("\non slope=%f cubic metre/m.",ss);
mprintf("\non apron with a slope 2:1 =%f cubic metre/m.",as);

va=5^0.5*D1*1.25*T;
vs=ss;
vs=round(vs*100)/100;
va=round(va*100)/100;
mprintf("\nU/S andD/S curved portion:");
mprintf("\non slope=%f cubic metre/m.",vs);
mprintf("\non apron =%f cubic metre/m.",va);

ta=va/(1.5*D1);
ta=round(ta*10)/10;
mprintf("\n\nthickness of launching apron=%f m.",ta);
```

Chapter 21

WATER POWER ENGINEERING

Scilab code Exa 21.1 EX21 1

```


//example 21.1
//calculate
//total installed capacity
//load factor
//plant factor
//utilization factor
clc;funcprot(0);
//given
c=10000;                //capacity of each generator;
n=3;                //number of generator
l1=12000;             //initial load on plant
l2=26000;             //final load on plant

tc=n*c;
mprintf("Total installed capacity=%i kW.",tc);

avg=(l1+l2)/2;        //average load
```

```
pk=12;                    //peak load
lf=avg*100/pk;
lf=round(lf*10)/10;
mprintf("\nload factor=%f percent.",lf);

//take any time duration t hours
pf=avg*100/tc;
pf=round(pf*10)/10;
mprintf("\nplant factor=%f percent.",pf);

uf=pk*100/tc;
uf=round(uf*10)/10;
mprintf("\nutilization ratio=%f percent.",uf);
```

Scilab code Exa 21.2 EX21 2

```


//example 21.2
//calculate maximum generation capacity of generator
//pondage to be provided
clc;funcprot(0);
//given
Q=40;            // minimum flow in river
H=30;            //net head
lf=0.73;          //load factor
eita=0.6;        //plant efficiency

P=9.81*Q*H*eita;
pk=P/lf;
pk=round(pk*10)/10;
mprintf("maximum generation capacity of generator=%f
    kW.",pk);

pp=pk-P;              //power develop from pondage
```

```
Q=pp/(9.81*H*eita);
pr=Q*4*3600/10000;
pr=round(pr*10)/10;
mprintf("\nPondage required=%fD+4 cubic metre.",pr);
```

Scilab code Exa 21.3 EX21 3

```


//example 21.3
//calculate minimum discharge required in the stream
//maximum load factor
clc;funcprot(0);
//given
c=15000;            //installed capacity of plant
lf=0.3;             //load factor
eita=0.82;          //plant efficiency
H=25;               //working head

avg=c*lf;         //average power developed
Q=avg/(9.81*H*eita);
Q=round(Q*100)/100;
mprintf("minimum discharge required in the stream=%f
    cumecs.",Q);

Q=32;           //for second case
P=9.81*H*Q*eita;
lf=P*100/c;
lf=round(lf*10)/10;
mprintf("\nmaximum load factor=%f percent.",lf);
```