Scilab Textbook Companion for Principles Of Geotechnical Engineering by B. M. Das[1]

Created by
Madhusudhan Reddy
C++
Civil Engineering
IIT BOMBAY
College Teacher
D. B. Phatak
Cross-Checked by
Chaitanya

May 24, 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: Principles Of Geotechnical Engineering

Author: B. M. Das

Publisher: Cenage & learning, New Delhi

Edition: 7

Year: 2013

ISBN: 978-1133108665

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

Chapter 2

weight volume relationships

Scilab code Exa 2.2 solved

```
clc
//initialisation of variables
//from graph
d= 0.15 //mm
w= 0.17 //mm
a= 0.27 //mm
//calculations
C= a/d
c= w^2/(a*d)
//results
printf ('uniformity coefficient = % f ',C)
printf ('coefficient of gradation = % 2f ',c)
```

Chapter 3

weight volume relationships

Scilab code Exa 3.2 solved

```
clc
//initialisation of variables
V= 1.2 //m^3
M= 2350 //Kg
w= 0.086
G= 2.71
W= 1000 //kg/m^3
//calculations
R= M/V
D= M/((1+w)*V)
e= (G*W/D)-1
n= e/(e+1)
S= (w*G/e)*100
v= (M-(M/(1+w)))/W
//results
printf ('moist density = % f kg/m^3 ',R)
printf ('dry density = % f kg/m^3 ',D)
printf ('void ratio = % 3f ',e)
printf ('porosity = % 3f ',n)
printf ('Degree of saturation = % 3f ',S)
printf ('volume of water in soil sample = % 3f m^3 '
```

```
,v)
```

Scilab code Exa 3.3 solved

```
clc
//initialisation of variables
n= 0.4
G= 2.68
w= 0.12
R= 1000 //kg/m^3
V= 10 //m^3
//calculations
d= G*R*(1-n)*(1+w)
s= ((1-n)*G+n)*R
M= s-d
m= M*V
//results
printf ('mass of water to be added for full
    saturation = %f kg ',m)
```

Scilab code Exa 3.4 solved

```
clc
//initialisation of variables
d= 16.19 //kN/m^3
w= 0.23
W= 9.81 //kN/m^3
//calculations
R= d*(1+w)
G= d/(W-d*w)
e= w*G
//results
printf ('satuarated unit weight = %2f kN/m^3 ',R)
```

```
printf ('specific gravity = % 2f ',G)
printf ('void ratio = % 2f ',e)
```

Scilab code Exa 3.5 solved

```
clc
//initialisation of variables
G= 2.68
w= 0.12
d= 1794.4 //kg/m^3
W= 1000 //kg/m^3
emax= 0.75
emin= 0.4
//calculation
e= (G*W*(1+w)/d)-1
D= ((emax-e)/(emax-emin))*100
//results
printf ('relative density of compaction in
    percentage = % f ',D)
```

Chapter 4

plasticity and structure of soil

Scilab code Exa 4.1 solved

```
clc
//initialisation of variables
V1= 24.6 //cm^3
V2= 15.9 //cm^3
M1= 44 //g
M2= 30.1 //g
W= 1 //g/cm^3
//calculations
SL= (((M1-M2)/M2)*100)-(((V1-V2)/M2)*W*100)
//results
printf ('shrinkage limit of the soil = % 2f ',SL)
```

Chapter 6

soil compaction

Scilab code Exa 6.2 solved

```
clc
//initialisation of variables
G= 2.6
LL= 20
P= 20
//calclations
R= (4804574*G-195.55*(LL)^2+156971*(P)^0.5-9527830)
   ^0.5
n= (1.195e-4)*((LL)^2)-1.964*G-(6.617e-5)*(P)+7.651
w= %e^n
//results
printf ('maximum dry density = % f kg/m^3 ',R)
printf ('optimum moisture content = % 2f ',w)
```

Scilab code Exa 6.3 solved

```
clc
//initialisation of variables
```

```
do= 1570 //kg/m^3
mo= 0.545 //kg
M1= 7.59 //kg
M2= 4.78 //kg
M3= 3.007 //kg
w= 0.102 //
dmax= 19 //KN/m^3
//calculations
Ms= M1-M2
Mc= Ms-mo
Vh= Mc/do
Dc= M3/Vh
Du= Dc*9.81/1000
f= Du/(1+w)
Rc= f*100/dmax
//results
printf ('dry unit weight of compaction in the field
    = % 2f kN/m^3 ',f)
printf ('relative compaction in the field = % f ',Rc
    )
```

Scilab code Exa 6.4 solved

```
clc
//initialisation of variables
D1= 0.36 //mm
D2= 0.52 //mm
D5= 1.42 //mm
//calculations
Sn= 1.7*(sqrt((3/(D5)^2)+(1/(D2)^2)+(1/(D1)^2)))
//results
printf ('sustainabilty number = % f ',Sn)
```

Chapter 7

permiability

Scilab code Exa 7.1 solved

```
clc
//initialisation of variables
L= 30 //cm
A= 177 //cm^2
h= 50 //cm
Q= 350 //cm^3
t= 300 //sec
//claculations
k=Q*L/(A*h*t)
//results
printf ('hydraulic conductivity = %3f cm/sec ',k)
```

Scilab code Exa 7.2 solved

```
clc
//initialisation of variables
L= 203 //mm
A= 10.3 //cm^2
```

```
a= 0.39 //cm^2
h0= 508 //mm
h180= 305 //mm
t= 180 //sec
//calculations
k= 2.303*a*L*log10(h0/h180)/(A*t)
//results
printf ('hydraulic conductivity of sand = % 2f in/
   sec ',k)
```

Scilab code Exa 7.3 solved

```
clc
//initialisation of varilables
k= 3e-7 //cm/sec
n= 0.0911e-4 //g*sec/cm^2
dw= 1 //g/cc
//calculations
K= k*n/dw
//results
printf ('absolute premeability = % 4f cm^2 ',K)
```

Scilab code Exa 7.4 solved

```
clc
//initialisation of variables
k= 5.3e-5 //m/sec
H= 3 //m
a= 0.139 //radians
//calculations
A= H*cos(a)
i= sin(a)
q= k*i*A*3600
```

```
//results
printf ('rate of seepage = % 4f m^3/hr/m ',q)
```

Scilab code Exa 7.5 solved

```
clc
//initialisation of variables
L= 50 //m
k= 0.08e-2//m/sec
h= 4 //m
H1= 3 //m
H= 8 //m
a= 0.139 //radians
//calculations
i= h*cos(a)/L
A= H1*cos(a)
q= k*i*A
//results
printf ('flow rate = % 2f m^3/sec/m ',q)
```

Scilab code Exa 7.6 solved

```
clc
//initialisation of variables
k1= 0.02 //cm/sec
e1= 0.5
e2= 0.65
//calculations
k2= k1*(e2^3/(1+e2))/(e1^3/(1+e1))
//results
printf ('hydraulic conductivity at void ratio of
   0.65 = % 2f cm/sec ',k2)
```

Scilab code Exa 7.8 solved

```
clc
//initialisation of variables
e= 0.6
D10= 0.09 //mm
//calculations
k= 2.4622*(D10^2*(e^3/(1+e)))^0.7825
//results
printf ('hydraulic conductivity = % 4f cm/sec ',k)
```

Scilab code Exa 7.9 solved

```
clc
//initialisation of variables
e= 0.6
D10= 0.09 //mm
D60= 0.16 //mm
//calculations
Cu=D60/D10
k= 35*(e^3/(1+e))*(Cu^0.6)*(D10^2.32)
//results
printf ('hydraulic conductivity = % 3f cm/sec ',k)
```

Scilab code Exa 7.10 solved

```
clc
//initialisation of variables
k1= 0.302e-7 //cm/sec
```

```
k2= 0.12e-7 //cm/sec
e1= 1.1
e2= 0.9
e= 0.75
//calcualtions
n= (log10((k1/k2)*((1+e1)/(1+e2))))/log10(e1/e2)
C= k1/(e1^n/(1+e1))
k= C*(e^n/(1+e))
//results
printf ('hydraulic conductivity = %e cm/sec ',k')
```

Scilab code Exa 7.11 solved

```
clc
//initialisation of variables
H1= 2 //m
H2= 3 //m
H3= 4 //m
k1= 1e-4 //cm/sec
k2= 3.2e-2 //cm/sec
k3= 4.1e-5 //cm/sec
//calculations
H= H1+H2+H3
Kh= (1/H)*((k1*H1)+(k2*H2)+(k3*H3))
Kv= H/((H1/k1)+(H2/k2)+(H3/k3))
P= Kh/Kv
//results
printf ('ration of equivalent hydraulic conductivity
    = % 2f ',P)
```

Scilab code Exa 7.12 solved

```
clc
```

```
//initialisation of variables
H= 450 //mm
h= 150 //mm
k1= 1e-2 //cm/sec
k2= 3e-3 //cm/sec
k3= 4.9e-4 //cm/sec
h1= 300 //mm
//calculations
Kv= H/(h*(1/k1+1/k2+1/k3))
i= h1/H
q= Kv*i*100*3600
//results
printf ('rate of water supply = %2f cm/hr ',q)
```

Chapter 8

seepage

Scilab code Exa 8.1 solved

```
clc
//initialisation of variables
H1= 12 //in
H2= 20 //in
z= 8 //in
h1= 24 //in
h= 20 //in
k1= 0.026 //in/sec
D= 3 //in
//calculations
k2= H2*k1/((z/(1-h/h1))-H1)
i= h1/(H1+H2)
A= %pi/4*D^2
keq= (H1+H2)/((H1/k1)+(H2/k2))
q= keq*A*i*3600
//results
printf ('rate of water flow = % 2f in^3/hr ',q)
```

Scilab code Exa 8.2 solved

```
clc
//initialisation of variables
Nd= 6
H1= 5.6 //m
H2= 2.2 //m
k= 5e-5 //cm/sec
dL= 4.1 //m
//calculations
H= (H1-H2)/Nd
h1= 5.61-H
h2= 5.61-5*H
q= 2.38*(H1-H2)*k/Nd
i= H/dL
//results
printf ('at point a,water will rise to height of = %
    3f m ',h1)
printf ('at point b,water will rise to height of = %
    3f m ',h2)
printf ('total rate of seepage per unit lenghth = %e
    m^3/sec/m ',q)
printf ('average hydraulic gradient at c = % 3f ',i)
```

Scilab code Exa 8.3 solved

```
clc
//initialisation of variables
k1= 5.67 //ft/day
k2= 11.34 //ft/day
//from graph
Nd= 8
Nf= 2.5
H= 20
//calculations
q= sqrt(k1*k2)*H*Nf/Nd
//results
```

```
printf ('average rate of flow = % 2f ft^3/day/ft ',q
   )
```

Scilab code Exa 8.4 solved

```
clc
//initialisation of variables
B= 6 //m
L= 120 //m
s= 3 //m
T= 6 //m
x= 2.4 //m
H= 5 //m
k= 0.008 //cm/sec
//calculations
b=B/2
a1= b/T
a2= s/T
a3= x/b
Q= 0.378*k*H*L*36*24
//results
printf ('seepage under the dam = % 2f m^3/day ',Q)
```

Scilab code Exa 8.5 solved

```
clc
//initialisation of variables
b= %pi/4 //degrees
a= %pi/6.//degrees
B= 10 //ft
H= 20 //ft
h= 25 //ft
k= 2e-4 //ft/min
```

```
//calculations
r= H/tan(b)
d= 0.3*r+(h-H)/tan(b)+B+h/tan(a)
L= d/cos(a)-sqrt((d/cos(a))^2-(H/sin(a))^2)
q= k*L*tan(a)*sin(a)*24*60
//results
printf ('seepage rate = % 4f ft^3/day/ft ',q)
```

Chapter 9

in situ stresses

Scilab code Exa 9.1 solved

```
clc
//initialisation of variables
Ds= 16.5 //kN/m^3
S= 19.25 //kN/m^3
g= 9.8 //kN/m^3
h1= 6 //m
h2= 13 //m
//at point A
Sa= 0
Ua= 0
Sa1= 0
//at point B
Sb= h1*Ds
Ub= 0
Sb1= Sb-Ub
//at point C
Sc= h1*Ds+h2*S
Uc= h2*g
Sc1= Sc-Uc
//results
printf ('total pressure at C= %2f kN/m^3 ',Sc)
```

```
printf ('pore water pressure at C = % 2f kN/m^3 ',Uc
   )
printf ('effective stress at point C= % 2f kN/m^3 ',
   Sc1)
```

Scilab code Exa 9.2 solved

```
clc
//initialisation of variables
h= 20 //ft
g= 120 //kg/ft^3
h1= 12 //ft
w= 62.4 //kg/ft^3
//calculations
H= h-(h1*w/g)
//results
printf ('maximu depth that can be made in clay = % 2
   f ft ',H)
```

Scilab code Exa 9.3 solved

```
clc
//initialisation of variables
G= 2.68
e= 0.52
g= 9.81 //kN/m^3
h1= 0.7 //m
h2= 1 //m
h3= 1.5 //m
h4= 2 //m
//calculations
//for soil A
sa= (G+e)*g/(1+e)
```

```
//point a
Sa= h1*g+h2*sa
u= (h2+h1+h3/2)*g
Es= Sa-u
//point b
sb= h1*g+h4*sa
ub= (h4+h1+h3)*g
Eb= sb-ub
i= h3/2
s= i*g
//results
printf ('effective stress at point a= %2f kN/m^2 ',Es)
printf ('effective stress at point b= %2f kN/m^2 ',Eb)
printf ('seepage force per unit voume = %2f kN/m^3 ',s)
```

Scilab code Exa 9.4 solved

```
clc
//initialisation of variables
C0= 0.357
H1= 30.5 //ft
H2= 5 //ft
w= 62.4 // lb/ft^3
D= 20
g= 112 // lb/ft^3
//calculations
G= g-w
FS= D*G/(C0*w*(H1-H2))
//results
printf ('safety factor = %2f ',FS)
```

Chapter 10

stress in a soil mass

Scilab code Exa 10.1 solved

```
clc
//initialisation of variables
sx= 2000 //lb/ft^3
sy= 2500 //lb/ft^3
T= 800 //lb/ft^3
t= 0.348//radians
//calculations
s1= (sx+sy)/2+sqrt(((sy-sx)/2)^2+T^2)
s2= (sx+sy)/2-sqrt(((sy-sx)/2)^2+T^2)
sn= (sx+sy)/2+(sy-sx)*cos(2*t)/2-T*sin(2*t)
Tn= (sy-sx)*sin(2*t)/2+T*cos(2*t)
//results
printf ('principle stress s1 = % 2f lb/ft^3 ',s1)
printf ('principle stress s2 = % 2f lb/ft^3 ',s2)
printf ('normal stress = % 2f lb/ft^3 ',sn)
printf ('shear stress = % 2f lb/ft^3 ',Tn)
```

Scilab code Exa 10.3 solved

```
clc
//initialisation of variables
x= 3 //m
y= 4 //m
P= 5 //kN
z= 2 //m
//calculations
r= sqrt(x^2+y^2)
k= r/z
I= 3/(2*%pi*((r/z)^2+1)^2.5)
s= P*I/z^2
//results
printf ('verticle stress increase at 2m = % 4f kN/m
   ^3 ',s)
```

Chapter 11

Compressibility of Soil

Scilab code Exa 11.1 solved

```
clc
Tz=150
b=1
l=2
z=5*b
Es= (10000*2 + 8000*1 +12000*2)/5
a=4
H=z
m=l/b
n=2*H/b
F1=0.641 //from tables 11.1 and 11.2
F2=0.031
u=0.3
Is= F1 + ((2-u)/(1-u))*F2
If=0.71 //from  table 11.3
Sef= Tz *a*b/l *(1-u^2)*Is*If/Es
Ser=0.93*Sef
printf('The elastic settlement at the centre of
    foundation = %f m',Ser)
```

Scilab code Exa 11.2 solved

```
clc
// one value of e is done
Gs=2.75
A=30.68
Ms=128
p=1
Hs=Ms/(A*Gs*p)
H=2.540
Hv=H-Hs
e=Hv/Hs
printf('the value of e for give values = %f',e)
```

Scilab code Exa 11.3 solved

```
clc
e11=0.9
e21=0.8
T2=4
T1=2
Cc= (e11-e21)/log10(T2/T1) // from loading branch
e1=0.67
e2=0.655
Cs=(e1-e2)/log10(T2/T1)
k=Cs/Cc
T3=12
e3=e11-Cc*log10(T3/T1)
printf('Compression index Cc= %f\n',Cc)
printf(' Cs/Cc = %f\n',k)
printf(' e3 = %f',e3)
```

Scilab code Exa 11.4 solved

```
clc
Gd=14
Gss=18
Gsc=19
Gw=9.81
To= 2*Gd+4*(Gss-Gw)+2*(Gsc-Gw)
LL=40
Cc=0.009*(LL-10)
H=4
T=100
e=0.8
Sc= Cc*H*log10((To+T)/To)/(1+e)
printf('a)Primary Consolidation Sc = %f m\n',Sc)

Tc=190
Cs=Cc/6
Sc= Cs*H*log10((To+T)/To)/(1+e)
printf(' b)Primary Consolidation Sc = %f m\n',Sc)

Tc=170
Sc= Cc*H*log10((To+T)/Tc)/(1+e)+ Cs*H*log10(Tc/To)
    /(1+e)
printf(' c)Primary Consolidation Sc = %f m\n',Sc)
```

Scilab code Exa 11.5 solved

```
clc
Gs=18
Gw=9.81
H=10
```

```
eo=1.1
To=5*(Gs-Gw)
T1=48
T=To+T1
e1=1.045 // void ratio corresponding to T
e=eo-e1
Sc=H*e/(1+eo)
printf('The settlement in the field Sc = %f m',Sc)
```

Scilab code Exa 11.6 solved

```
clc
T=8.5
eo=0.8
Cc=0.28
To=2650
T1=970
C1=0.02
t2=5
t1=1.5
H=8.5*12
epr=Cc*log10((To+T1)/To)
ep=eo-epr
C2=C1/(1+ep)
Sc=epr*H/(1+eo)
Ss=C2*H*log10(t2/t1)
TS=Sc+Ss
printf('Total consolidation settlement of the clay =
    %f in',TS)
```

Scilab code Exa 11.7 solved

```
clc
```

```
//T50 = Cvtlab /H^2 lab = Cvtfield?H^2 fiels
tl=140
Hf=3
Hd=0.025/2
tf=tl*Hf^2/Hd^2
k=tf/(3600*24)
printf('t field = %f days',k)
```

Scilab code Exa 11.8 solved

```
clc
//Tv is directly proportional to U^2
t1=93.333
U2=30
U1=50
t2=t1*U2^2/U1^2
printf('t2 = %f days',t2)
```

Scilab code Exa 11.9 solved

```
clc
t90=75*24*60*60 // time in sec
T90=0.848
Hd=1.5*100 //in cm
Cv=T90*Hd^2/t90
printf('Cv = %f cm^2/sec',Cv)
```

Scilab code Exa 11.10 solved

```
clc
```

```
To=3000 // lb/ft^2
eo=1.1
e1=0.9
e=eo-e1
ea=(eo+e1)/2
T1=3000 // lb/ft^2
T=1 // in
t = 2 // min
m=(e/T1)/(1+ea)
U=50
Tv=0.197
Gw=62.4 //lb/ft^3
Cv=Tv*(T/(2*12)^2)/t
k=Cv*m*Gw *10^7
printf('a)k = %f x10^-7 ft/min\n',k)


U=60
Tv=0.286
H=6
t60=Tv*H^2/(Cv*60*24)
printf(' b)t60 =%f days',t60)
```

Scilab code Exa 11.11 solved

```
clc
t50=19
Hd=2.24/2
Cv=0.197*Hd^2/t50
printf('Cv = %f cm^2/min',Cv)
```

Scilab code Exa 11.12 solved

```
clc
LL=40
Cc=0.009*(LL-10)
H=10*12
eo=1.0
Gss=120
Gsc=110
Gd=100
To=10*Gd +10*(Gss-62.4)+10*(Gsc-62.4)/2

Tt=0.408
Tm=0.232
Tb=0.019
Tav= (Tt+4*Tm+Tb)/6
Sc=Cc*H*log10((To+Tav*1000)/To)/(1+eo)
printf('Sc = %f in',Sc)
```

Scilab code Exa 11.13 solved

```
clc
H = 6
Cc = 0.28
eo = 0.9
Cv = 0.36
To=210
Tp=115
Sc= Cc*H*log10((To+Tp)/To)/(1+eo)
t2=9
Hd=3
Tv=Cv*t2/Hd^2
U=0.67
Tf=0.677*Tp
printf('Tf = %f kN/m^2',Tf)
```

Chapter 12

Shear Strength of Soil

Scilab code Exa 12.1 solved

```
clc
D=50 // in mm
A= %pi/4 *(D/1000)^2
// solving for test 1
N=150
Sp=157.5
Sr=44.2
Tf=Sp/A
Tr=Sr/A
// from graph
k=tand(27)
k1=tand(14.6)

printf('Peak strength Tf = 40+ t*%f\n',k )
printf(' Residual strength Tr = t*%f',k1)
```

Scilab code Exa 12.2 solved

```
clc
T3=16 // lb/in^2
Tf=25 // lb/in^2
T1=T3+Tf
a= asind((T1-T3)/(T1+T3)) // Mohr's circle
printf('a)Angle of friction ,a = %f\n',a)
b= 45+ a/2
printf(' b)Angle b that the failure plane makes with the major principal plane = %f',b)
```

Scilab code Exa 12.3 solved

```
clc
T1=41
T3=16
a=58
T=(T1+T3)/2 + (T1-T3)*cosd(2*a)/2
tf=(T1-T3)*sind(2*a)/2
printf('a)the normal stress T = %f lb/in^2',T)
printf(' and the shear stress tf = %f lb/in^2',tf)
```

Scilab code Exa 12.4 solved

```
clc
//For normally consolidated clay, c' = 0.
a=30
T3=10
T1=T3*(tand(45+a/2))^2
Tf=T1-T3
printf('The deviator stress at failure = %f lb/in^2',Tf)
```

Scilab code Exa 12.5 solved

```
clc
T13=70
T1f=130
T11=T13+T1f

T23=160
T2f=223.5
T21=T23+T2f

a= 2*(atand(((T11-T21)/(T13-T23))^0.5)-45)
c= (T11-T13*(tand(45+a/2))^2)/(2*tand(45+a/2))
printf('the shear strength parameter c = %f kN/m^2',
    c)
```

Scilab code Exa 12.6 solved

```
clc
T3=12
Tf=9.1
T1=T3+Tf
u=6.8
a= asind((T1-T3)/(T1+T3))

a1= asind((T1-T3)/(T1+T3-2*u))

printf('a)Consolidated-undrained angle of shearing
    resistance = %f degrees\n',a)
printf(' b)Drained friction angle = %f degrees',a1)
```

Scilab code Exa 12.7 solved

```
clc
T3=12
a=27.8
T1=T3*(tand(45+a/2))^2
Tf=T1-T3
printf('the deviator stress at failure = %f lb/in^2'
    ,Tf)
```

Scilab code Exa 12.8 solved

```
clc
PI=28
OCR=3.2
To=160
Kn=0.11+0.0037*PI
Ko=OCR^0.8 * Kn
Cu=Ko*To
printf('the average undrained shear strength of the
    clay = %f kN/m^2',Cu)
```

Chapter 13

Lateral Earth Pressure

Scilab code Exa 13.1 solved

```
clc
OCR=2
a=30
Ko=(1-sind(a))*(OCR)^sind(a)
//at z=0
To1=0
Th1=0
u1=0
//at z=10
To2=10*100
Th2=Ko*To2
u2=0
//at z=15
To3= 10*100+5*(122.4-62.4)
Th3=Ko*To3
u3=5*62.4
//Lateral force Po =Area 1 + Area 2+ Area3+ Area 4
Po =(1/2)*10*707+5*707+(1/2)*5*212.1+(1/2)*5*312
z=((3535)*(5+10/3)+3535*(5/2)+530.3*(5/3)+780*(5/3))
   /Po
printf('z = %f ft',z)
```

Scilab code Exa 13.2 solved

```
clc
//c=0
a=36
G=16
Ka=(1-sind(a))/(1+sind(a))
//at z=0 Tp=0
z=6
To=G*z
Ta=Ka*To
Pa=z*Ta/2

printf('a)Rankine active force per unit length of
    the wall = %f kN/m',Pa)
printf(' and the location of the resultant is z = 2m
    \n')


p=36
G=16
Kp=(1+sind(a))/(1-sind(a))
//at z=0 Tp=0
z=6
To=G*z
Tp=Kp*To
Pp=z*Tp/2

printf(' b)Rankine passive force per unit length of
    the wall = %f kN/m',Pp)
printf(' and the location of the resultant is z = 2m
    ')
```

Scilab code Exa 13.3 solved

```
clc
H=12
a=20
b=20
G=115
c=30
Oa= asind(sind(a)/sind(c))-a+2*b
Ka= (cosd(a-b)*sqrt(1+(sind(c))^2-2*sind(c)*cosd(Oa)
    ))/((cosd(b))^2*(cosd(a)+sqrt((sind(c))^2-(sind(a
    ))^2)))
Pa=G*H^2*Ka/2
B= atand((sind(c)*sind(Oa))/(1-(sind(c)*cosd(Oa))))
printf('The active force Pa per unit length of the
    wall = %f lb/ft\n',Pa)
printf( ' The resultant will act a distance of 12/3
    = 4 ft above the bottom of the wall with B = %f
    degree',B)
```

Scilab code Exa 13.4 solved

```
clc
a=30
Ka1=(1-sind(a))/(1+sind(a))
a=35
Ka2=(1-sind(a))/(1+sind(a))
//at z=0 so T0=0
//atz=3
To=3*16
Ta1=Ka1*To
Ta2=Ka2*To
```

```
// At z=6
To=3*16+3*(18-9.81)
Ta2=Ka2*To

Pa =(1/2)*3*16+3*13.0+ (1/2)*3*36.1
z= (24 *(3+3/3)+39.0*(3/2)+54.15*(3/3))/Pa
printf('The force per unit length of the wall = %f kN/m\n',Pa)
printf (' The location of the resultant = %f m ',z)
```

Scilab code Exa 13.5 solved

```
clc
Ka= (tand(1))^2
G=16.5
cu=10
H=6
//at z=0
z=0
Ta=G*z-2*cu
//zt z=6
z=6
Ta=G*z-2*cu

zo=2*cu/G
// Before the tensile crack occurs
Pa= G*H^2/2 - 2*cu*H
printf('Pa before the tensile crack occurs = %f kN/m\n',Pa)
//After the tensile crack occurs
Pa=(H-zo)*Ta/2
printf(' Pa after the tensile crack occurs = %f kN/m ',Pa)
```

Scilab code Exa 13.6 solved

```
clc
H=15
a=10
G=118
b=20
C=250
Zo=2*C*sqrt((1+sind(b))/(1-sind(b)))/G
//at z=0 Ta=0
//at z=15
z=15
K=0.3
Ta=G*z*K*cosd(a)
Pa=(H -Zo)*Ta/2
printf('The Rankine active force Pa on the retaining
    wall after the tensile crack occurs = %f lb/ft',
   Pa)
```

Scilab code Exa 13.7 solved

```
clc
c=30
b=15
a=10
Ka=0.3872 // from table 13.8
H=4
G=15
Pa=G*H^2*Ka/2
printf('The active force per unit length Pa = %f kN/
   m\n',Pa)
```

```
printf(' The resultant will act at a vertical distance equal to H/3 = 4/3 = 1.33 m above \n the bottom of the wall and will be inclined at an angle of 15to the back face of the wall.')
```

Scilab code Exa 13.9 solved

```
clc
kh=0.2
kv=0
H=4
a=0
b=0
c=15
d=30
G=15.5
B= atand(kh/(1-kv))
b1=b+B
a1=a+B
Ka=0.452
Pa=G*H^2*Ka/2
Pae=Pa*(1-kv)*((cosd(b1))^2/((cosd(b))^2*(cosd(B))^2))
Ka=0.3014
Pa=G*H^2*Ka/2
P1=Pae-Pa
z= ((Pa*H/3)+P1*0.6*H)/Pae
printf('Pae = %f kN/m\n',Pae)
printf(' Z = %f m',z)
```

Scilab code Exa 13.10 solved

```
clc
```

```
H=28
C=210
b=10
G=118
c=20
kh=0.1
Ka=tand(45-c/2)
zo=2*C/(G*(Ka))
n=zo/(H-zo)
Nac=1.60
Nav=0.375
L=1.17
Pae= G*(H-zo)^2*(L*Nav)-C*(H-zo)*Nac
printf('The magnitude of the active force , Pae = %f
    lb/ft',Pae)
```

Chapter 14

Lateral Earth Pressure Curved Failure Surface

Scilab code Exa 14.1 solved

```
clc
G=15.7
a=0
b=15
c=30
H=3
Kp=4.977 // from table 13.9
Pp=Kp*G*H^2/2
printf('a)the passive force = %f kN/m\n',Pp)
// for part b
Kp=4.53
Pp=Kp*G*H^2/2
printf(' b)the passive force = %f kN/m\n',Pp)
// for part c
Kp=4.13
Pp=Kp*G*H^2/2
printf(' c)the passive force = %f kN/m\n',Pp)
//for part d
Kp=4.56
```

```
Pp=Kp*G*H^2/2
printf(' d)the passive force = %f kN/m\n',Pp)
```

Scilab code Exa 14.2 solved

```
clc
G=16
H=7
c=30
Ta=0.65*G*H*(tand(45-c/2))^2
A=Ta*3*3/4
B1=Ta*3-54.61
C=Ta*4*4/4
B2=Ta*4-97.08
s=2
As=A*s
Bs=(B1+B2)*s
Cs=C*s
printf( 'The strut loads at level A = %f kN\n',As)
printf( ' The strut loads at level B = %f kN\n',Bs)
printf( ' The strut loads at level C = %f kN',Cs)
```

Chapter 15

Slope Stability

Scilab code Exa 15.1 solved

```
clc
Gs=17.8
Gw=9.81
C=10
c=20
b=15
H=6
G=Gs-Gw
Fs= C/(Gs*H*cosd(b)*cosd(b)*tand(b))+G*tand(c)/(Gs*
   tand(b))
printf('a)The factor of safety = %f \n',Fs)
Fs=2
H=2.247/(Fs-0.61)
printf(' b)H= %f m',H)
```

Scilab code Exa 15.2 solved

```
clc
```

```
G=105
c=15
C=600
b=45
Fs=3
Cd=C/Fs
c1= atand(tand(c)/Fs)

H= 4*Cd*(sind(b)*cosd(c1)/(1-cosd(b-c1)))/G
printf('The depth of the cut slope = %f ft',H)
```

Scilab code Exa 15.3 solved

```
clc
Cu=40
G=17.5
b=60
a=35
c=72.5
m=0.195
Hc=Cu/(G*m)
r=Hc/(2*sind(a)*sind(c/2))
BC=Hc*((1/tand(a))-(1/tand(b)))
printf('a)The maximum depth Hc = %f m\n',Hc)
printf(' b)The radius , r = %f m\n',r)
printf(' c)The distance BC.= %f m',BC)
```

Scilab code Exa 15.4 solved

```
clc
Gs=17.29
d=9.15
d1=6.1
```

```
D=d/d1
a=40
m=0.175
b=40
H=6.1
Cu=H*Gs*m
printf('a)The undrained cohesion of the clay Cu = %f
    kN/m^2\n',Cu)
printf(' b)The nature of the critical circle is
    midpointcircle\n')
d=1.5
b=40
n=0.9
D1=n*H
printf(' c)Distance = %f m',D1)
```

Scilab code Exa 15.5 solved

```
clc
Fs=1
b=56
Kh=0.25
M=3.66
Cu=500
G=100
Hc=Cu*M/G
printf('a)The maximum depth = %f ft\n',Hc)
Fs=2
H=Cu*M/(G*Fs)
printf(' b)H= %f ft',H)
```

Scilab code Exa 15.6 solved

```
clc
b=45
c=20
C=24
G=18.9
m=0.06
Hc=C/(G*m)
Cd=G*Hc*m
Fc=C/Cd
printf('a)Critical height of slope = %f \n',Hc)
```

Scilab code Exa 15.7 solved

```
clc
FSs=1
c=20
G=18.9
C=24
Hcr=C/(G*tand(c)*0.17)
printf('a)Critical height Hc = %f m\n',Hcr)
H=10
k=C/(G*H*tand(c))
Fs=4*tand(c)
printf(' b)Fs = %f',Fs)
```

Scilab code Exa 15.8 solved

```
clc
W=22.4
C=20
a=70
s=sind(a)
c=cosd(a)
```

```
l=2.924
Wn=W*s
Wn1=W*c
//doing this to all values
F1=30.501
F2=776.75
F3=1638
Fs=(F1*C+F3*tand(C))/F2
printf('Fs = %f',Fs)
```

Scilab code Exa 15.9 solved

```
clc
C=20
G=18.5
r=0.25
H=21.62
C=25
b= atand(0.5)
//from table 15.3
m=1.624
n=1.338
Fs=m-n*r
printf(' The value of Fs for D= 1 is %f',Fs)
```

Scilab code Exa 15.11 solved

```
clc
C=20
G=18.5
H=21.62
c=25
r=0.25
```

```
Fs=3.1*tand(c)
printf('Fs = %f',Fs)
```

Chapter 16

Soil Bearing Capacity for Shallow Foundations

Scilab code Exa 16.1 solved

```
clc
c=20
// from table 16.1
Nc=17.69
Nq=7.44
Ng=3.64

Df=3
G=110
q=G*Df

C=200
B=4

Qu= C*Nc+q*Nq+G*B*Ng/2

Fs=3
Qall=Qu/Fs
printf('Qa = %f lb/ft^2',Qall)
```

Scilab code Exa 16.2 solved

```
clc
G=18.15
qa=30000*9.81/1000

Nc=57.75
Nq=41.44
Ng=45.41
C=0
q=G*1
B=1
(1.3*C*Nc+q*Nq+0.4*G*B*Ng)*B^2/3 == qa
B= sqrt(294.3/(250.7+109.9))
printf(' B = %f m',B)
```

Scilab code Exa 16.3 solved

```
clc
B=1.2
L=1.2
c=32
C=0
Df=1
G=16
Nq=23.18
Ng=22.02
Nc=1
Lqs=1+0.1*B*(tand(45+c/2))^2/L
Lgs=Lqs
Lqd=1+0.1*Df*tand(45+c/2)/B
```

```
Lgd=Lqd
Lcs=1
Lcd=1
Gs=19.5
q=0.5*G+0.5*(Gs-9.81)
Qu= C*Lcs*Lcd*Nc+q*Lqs*Lqd*Nq+(Gs-9.81)*Lgs*Lgd*B*Ng
   /2
Qa=Qu/3
Q=Qa*B^2
printf('the gross load = %f kN',Q)
```

Scilab code Exa 16.4 solved

```
clc
e=0.1
B=1
X=B-2*e
Y=1.5
B1=0.8
L1=1.5
c=30
Df=1
Nq=18.4
Ng=15.668
q=1*18
G=18
Lqs=1+e*(B1/L1)*(tand(45+c/2))^2
Lgs=Lqs
Lqd=1+e*(Df/B1)*tand(45+c/2)
Lgd=Lqd
qu=q*Lqs*Lqd*Nq+Lgs*Lgd*G*B1*Ng/2
Qu=qu*B1*L1
printf('The magnitude of the gross ultimate load =
   %f kN',Qu)
```

Scilab code Exa 16.5 solved

```
clc
B=1.5
Df=0.75
e=0.1*B
G=17.5
c=30
C=0
q=G*Df
Nq=18.4
Ng=15.668
Lqd=1+0.1*(Df/B)*tand(45+c/2)
Lgd=Lqd
Quc=q*Nq*Lqd+Lgd*B*Ng/2
k=0.8
a=1.754
Qua=Quc*(1-a*(e/B)^k)
printf('The gross ultimate load per unit length = %f
    kN',Qua)
```

Scilab code Exa 16.6 solved

```
clc
Qup=280
Bp=0.7 // in m
Bf=1.5
Quf=Qup*Bf/Bp
printf('The ultimate bearing capacity = %f kN/m^2',
   Quf)
```

Scilab code Exa 16.7 solved

```
clc
a=2500
//doing for the first values only
Bf=4
Bp=0.305
q=a/Bf^2
Sep=4
Sef=Sep*(2*Bf/(Bf+Bp))^2
printf('Sef = %f mm',Sef)
```

Chapter 18

Subsoil Exploration

Scilab code Exa 18.1 solved

```
clc
//solving for z=5 only
To=0.275
Cn=To^(-0.5)
N60=8
N160=Cn*N60
printf('(N1)60 = %f',N160)
```

Scilab code Exa 18.2 solved

```
clc
z=5
To=0.275
Cn=2/(1+To)
N60=8
N160=Cn*N60
printf('(N1)60 = %f',N160)
```

Scilab code Exa 18.3 solved

```
clc
pa=1 // 14.7 lb/in^2 = 1ton/ft^2
To=0.275 // ton/ ft^2
N60=8
c= atand((N60/(12.2+20.3*(To/pa)))^0.34)
printf('The average soil friction angle = %f',c)
```