



Developing confidence in critical state soil mechanics

7. Visual Basic for Applications (VBA)

Dr. Dawn Shuttle, PEng

Mike Jefferies, PEng

January, 2015



Soil mechanics has to wait 30 mins...

- Need to know some VBA techniques
- Packaged for you in “open code”
- Quickly overview VBA and then you jump in...



Bad practice – prevalent in geotech !

- Programming equations in a worksheet is like writing computer code from the 1960's... what does a variable named “C3” stand for ?
- Modern computer languages evolved to use “plain english” variables for both readability and maintainability... “C3” => “eta_max”
- Modern languages also allow comments adjacent to variable names for enhanced readability.... ‘eta_max *as per eqn 6.73*’
- Microsoft has provided a series of cheap, easy to use languages
 - ‘Quick Basic” QB45 (*ye olde DOS days*)
 - Visual Basic VB3 (*Windows 3.11*)
 - Now bundle an ‘integrated development environment’ with Office: VBA



Accessing VBA

“Alt” + “F11”

The screenshot shows the Microsoft Excel interface with the 'Tools' menu open. The 'Macro' submenu is expanded, showing options: 'Macros...', 'Record New Macro...', 'Visual Basic Editor', 'Add-Ins...', 'Customize Keyboard...', and 'Solver...'. The 'Visual Basic Editor' option is highlighted in blue. The background spreadsheet contains the following data:

NorSand material properties...			
CSL parameters			
Γ	=	0.817	---
λ_{10}	=	0.05	
λ	=	0.021739	base e
Plasticity			
Mtc	=	1.3	
N	=	0	(range 0 -
H	=	50	(typ 100 -
χ	=	3	usually tai
Elasticity			
Ir	=	150	---
(G...)	=	50	MPa
v	=	0.2	
κ	=	0.005435	
Initial soil state...			
ψ	=	0.01	
p	=	500	kPa
k0	=	1	
(e0...)	=	0.51627	
(sig1...)	=	500	
Loading Path Slope...			
slope	=	3.00E+30	3 for CID
			1E30 for constant p'

The graphs show: 1) deviator stress (q) vs axial strain (e1: %) with 'CamClay' and 'closed form' curves; 2) deviator stress (q) vs mean effective stress (p); 3) volumetric strain (εv: %) vs axial strain (e1: %).

“Enable Macros”



Code window

```
Module1 (Code)
CamClay_CIU

' set plastic shear strain increment
depG = 0.05 / (NumTheoPts - 1)

' and now for the euler integration
For j = 2 To NumTheoPts
    ' compute normalized stresses
    Kmax = sigM * (1 + e) / kappa
    ' elastic modulus varies with loadstep
    Dp = (Mtc - eta)
    ' plastic strain increments for the load step
    depV = Dp * depG
    H = (1 + e) / (lambda - kappa)
    ' apply the hardening rule
    dPcPc = H * depV
    Pc = Pc * (1 + dPcPc)
    sigMdot = -Kmax * depV
    ' undrained conditons
    sigM = sigM + sigMdot
    eta = Mtc * (1 + Log(Pc / sigM))
    ' YS equation applied to hardened YS
    sigQ = eta * sigM
    ' new deviator stress

    ' update strains
    dep1 = depG ' undrained has depV=0
    ep1 = ep1 + dep1
    epV = 0

    ' and put the results in output array...
    ' (the 100 factor is conversion of strain to %)
    NorResults(j, eps1) = 100 * ep1
    NorResults(j, epsV) = 100 * epV
    NorResults(j, p) = sigM
    NorResults(j, q) = sigQ
Next j
End Sub
```



Modelling interface

NorSand_txL_r18_nerlerk270.xls

Search in Sheet

Home Layout Tables Charts Formulas Data Review

V34

Soil properties....

CSL parameters

- $\Gamma = 0.855$ ---
- $\lambda = 0.021$ on base e

Plasticity

- $M_{ic} = 1.27$
- $N = 0.35$ (typ 0.2 - 0.3)
- $\chi_{ic} = 4.3$ often taken as 4
- $H_0 = 100 > H_c = 92$
- $H_v = 500$
- ($H = H_0 - H_v \cdot \psi...$) 70 (typ 60 - 400)

Elasticity

- $G_{max} @ p_0 = 60$ MPa
- $G_{exp} = 0.5$ elastic exponent
- $\nu = 0.2$
- ($\nu_r...$) 120 ---
- $\kappa = 0.01116$ ---

Initial soil state...

- $\psi_0 = 0.06$
- $\Rightarrow e_0 = 0.785$
- $p_0 = 500$ kPa
- $K_0 = 1$ ---
- ($sig1...$) 500
- OCR ("R") = 1.02 ---

Update Model

Strain Mode

Engineering NOTE: most labs report engineering (= "small") strain (change choice inside VBA)

Choose Test to Plot

Choice is: 7

Plot Data

DRAINED TXL....

To run this spreadsheet edit the independent (blue) material properties and then select the "Update Model" button
 Values shown in brackets after the blue model parameters are calculated equivalents that may be of interest

GNU License NOTES Data Summary Params and Plots Undrained plots TxL SimResults CIU_G101 CIU_G103 CI

Normal View Ready Sum=0



Structure of Programs

- On button click...
 - Get values from input worksheet
 - Clear output worksheet
 - Do all the clever things all in VBA (subroutines and functions)
 - Put results in single master array
 - Image that array back to excel output worksheet
 - Excel graphs plotted from output worksheet (update automatically)



Initial window in VBA

```
(General)
Option Explicit
Option Base 1
Const eps1 = 1
Const epsV = 2
Const p = 3
Const q = 4
Const sigma1 = 5
Const sigma2 = 6
Const state = 7
Const NumTheoPts = 700      ' number of points for the theory
Const T_30deg = 0.5236      ' T_30deg as radians
Dim NorResults(700, 7)

Sub ApplyConsistency(M, psi, p, dPiOverPi, ds2plas, nu, eta, sig1, sig2, sig3)
'-----
' this sub gets to the new stress state by applying the consistency condition
' for plane strain with dsig3=0 (corresponds to Cornforth's experiments)
'
' jan/00  mgj
'-----
Dim a, b, c, x As Double      ' local variables defined in derivation
Dim Dp, q, dsig1, dsig2 As Double
Dim partialdqs1, partialdqs2 As Double

' first we calculate the coefficients
q = eta * p
partialdqs1 = (2 * sig1 - sig2 - sig3) / (2 * q)
partialdqs2 = (2 * sig2 - sig3 - sig1) / (2 * q)

a = partialdqs1 + nu * partialdqs2

b = -ds2plas / p

Dp = M - eta
c = (1 + nu) * Dp / 3

(Declarations)
(Declarations)
ApplyConsistency
ApplySSuConsistency
ArrayFill
ArrayFill_PS
asin
CamClay_CID
CamClay_CIU
epsilonG
epsilonQ
GetStrainRateRatios
Lode
```




Functions vs Subroutines

- Functions return a single value based on their input values
 - Function M_i (M_c , χ , l_{ode} , ψ)
 - Function $e_crit(\text{sigM}, \dots)$
- Subroutines do not return anything... Results held in variables
 - Used for most of the critical state modelling



My bad programming practice...

- ComSci disciples like “encapsulation”
 - Everything used in a function/subroutine is passed to in in the “call”
 - Very “safe” programming practice
 - But, leads to very long winded code to read
 - **StrainRates** (ep1, ep2,ep3, sig1, sig2, sig3, NorProp(), Dp, z2, z3)
- Global variable allow easy, english-like code
 - **GetPlasticStrainRates**
 - Inputs, properties, outputs must be held “globally”
 - Accessible to any routine
 - But, can be modified by any routine (accidentally...)



Load up NorSand VBA

- In the “Session 8” folder
- NorSandM_txl_FRS.xls
- Open and “Enable Macros”