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 !

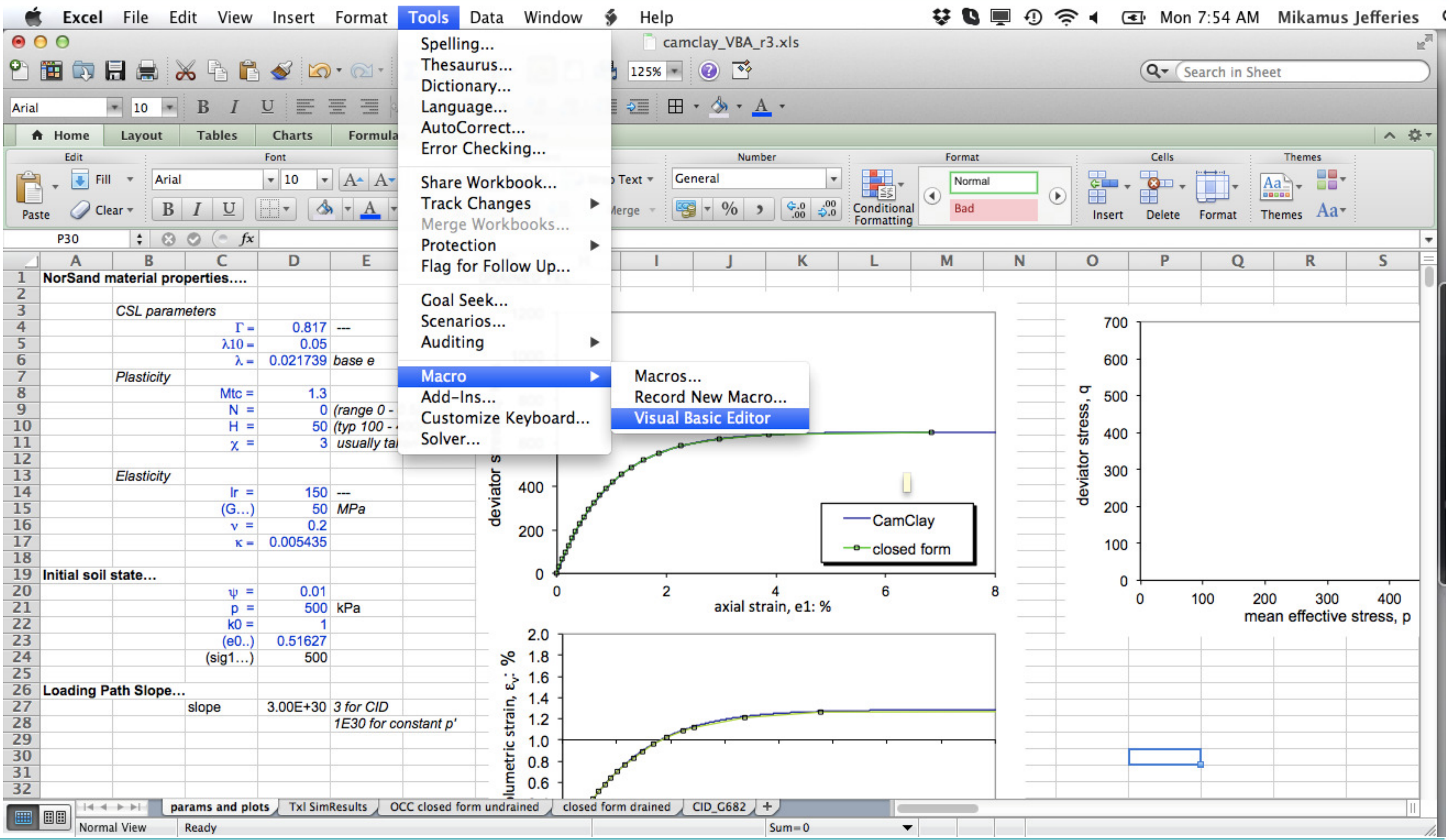
- Programming equations in a worksheet is like writing computer code from the 1960's… what does a variable named "C3" stand for ?

- 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"



“Enable Macros”

Code window

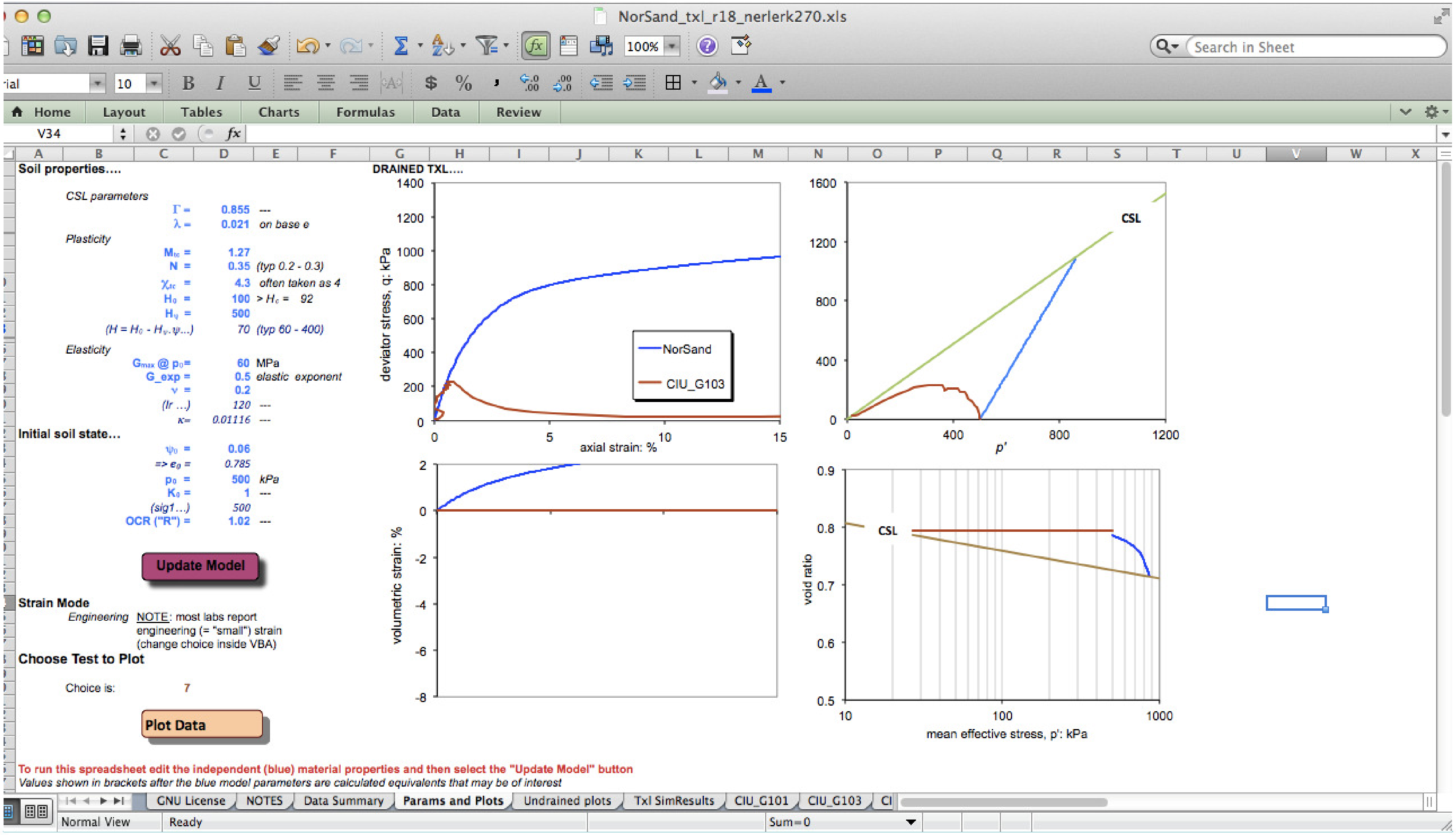
Module1 (Code)

(General) | 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
```

Ready | Sum=0

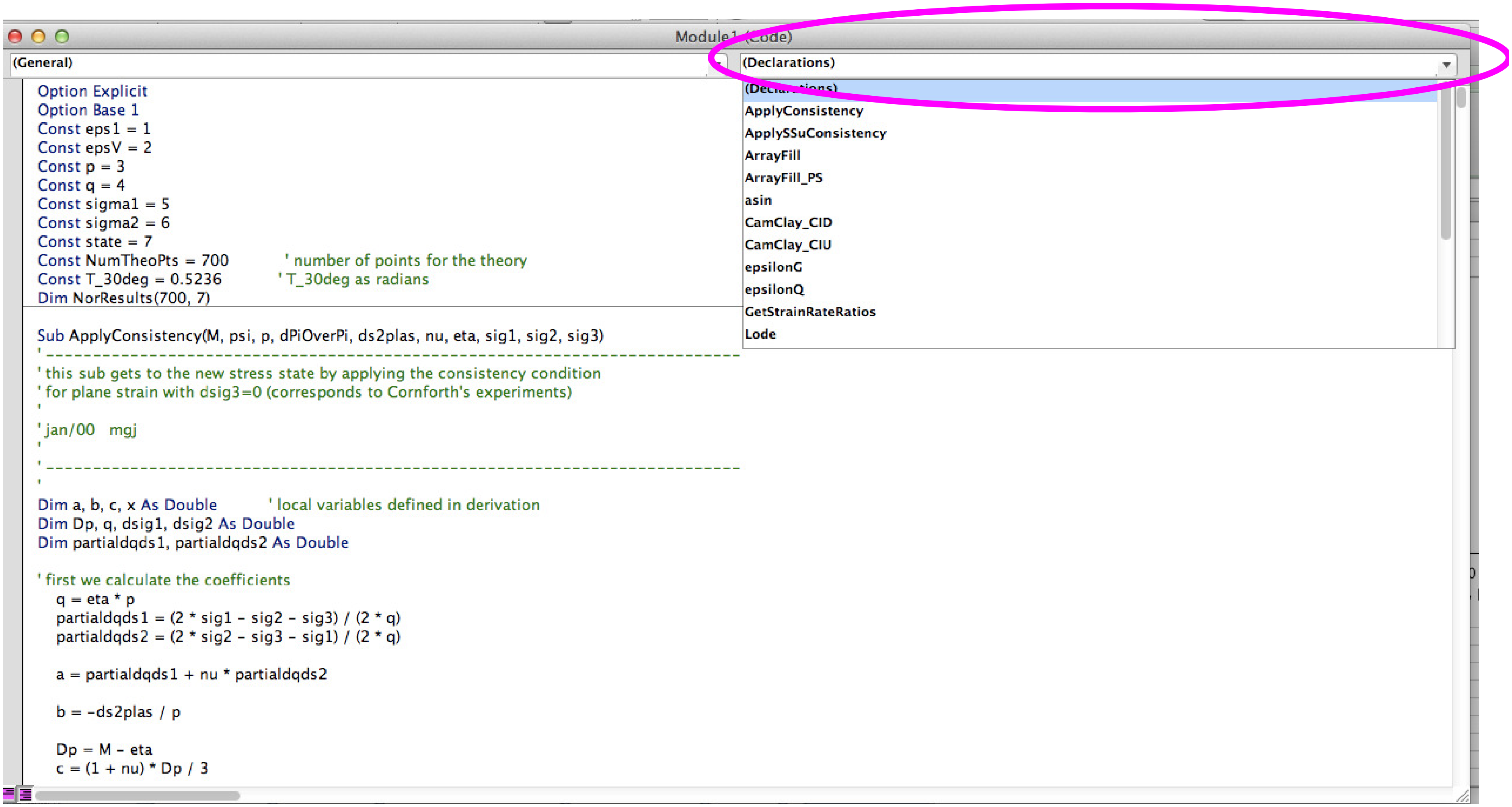
Modelling interface



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



Functions vs Subroutines

- Functions return a single value based on their input values
  - Function Mi (Mc, chi, lode, psi)
  - Function e_crit(sigM,….)

- 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’