Appendix C

Tcl Commands for the upU example

C.1 The model Command

The model command defines the ModelBuilder object to be constructed.

model BasicBuilder -ndm ndm? <-ndf ndf?>

The string -ndm followed by a integer defines the dimension of the problem, i.e. 1D, 2D or 3D problem. The string -ndf followed by a integer defines the number of degree-of-freedom at a node. The angle bracket around the -ndf ndf? means these arguments are optional. By default, the number of degree-of-freedom at a node depends on the dimension of the problem. For ndm=1, ndf=1; for ndm=2, ndf=3; for ndm=3, ndf=6.
For upU element, each node has 3 solid displacements, one pore pressure, and 3 fluid displacements, so the degree-of-freedom of each node is 7. The ModelBuilder command for upU element is:

model BasicBuilder -ndm 3 -ndf 7

C.2 The node Command

The node command is used to construct the Node object.

node nodeTag? (ndm coordinates?) <-mass (ndf values?)>

The nodeTag defines the integer tag that uniquely identifies the node object among all other nodes in the model. The number of ndm coordinates are used to define the spacial location of the node. An optional string -mass followed by the ndf values allows the analyst to define the nodal mass in different degree-of-freedom with the node.
The node command for upU element is:

node nodeTag x_coordinate y_coordinate z_coordinate

C.3 The fix Command

The fix command is used to construct the single-point boundary conditions.

fix nodeTag? (ndf values?)

where the nodeTag is the node need to be constrained. The ndf values can be 1 or 0. If the value of ith ndf is specified as 1, then the degree-of-freedom at that direction is constrained. Otherwise 0 means the it is left free.
The fix command for upU element is:

fix nodeTag (7 values (1 or 0))

for example
fix 1 1 1 1 1 1 1 1
fix all the degree-of-freedom of node no. 1.

fix 5 1 1 0 0 1 1 0
fix the solid and fluid displacement in x and y directions, while the z direction of solid and fluid and pore pressure are left free.

C.4 The nDMaterial Command

The nDMaterial command is used to construct a NDMaterial object.

nDMaterial materialType <specific material args>

NDMaterial objects represent stress-strain relationships at the integration points of continuum and force-deformation elements.
The materialType specifies the material type. Current the following type are available: ElasticIsotropic3D, J2Plastcity, Template3Dep and Bidirectional. The specific material args specifies the material property. For simplified, elastic isotropic material is used in upU examples.
The Elastic Isotropic Material command:

nDMaterial ElasticIsotropic matTag? E? ν? ρ

constructs an ElasticIsotropic material object with elastic modulus E, Poisson ratio ν and mass density ρ. The argument matTag is used to uniquely identify this object from others in the BasicBuilder object.
The following command:

nDMaterial ElasticIsotropic3D 1 17400 0.35 2.0

set the Young's modulus to 17400, Poisson ration to 0.35 and mass density to 2.0.

C.5 Element Command

The element command is used to construct an Element object.

element eleType <specific element type args>

The second argument is the element type. Currently there are following element types in OpenSees: truss, elasticBeamColumn, nonlinearBeamcolunm, beamWithHinges, zeroLength, zeroLengthSection, zeroLengthND, quad, and brick element. The element name for the upU is given as Brick8N_u_p_U (for 8 nodes brick upU element) and Brick20N_u_p_U (for 20 nodes brick upU element). The arguments in the angle bracket specify the properties of each type of element.
For upU element, the element command is:(for example 20 nodes upU)

element Brick20N_u_p_U eleTag? 1st_node? 2nd_node?...20th_node? materialID? x_body_force? y_body_force? z_body_force? porosity? alpha? solid_density? fluid_density? x_permeability? y_permeability? z_permeability? solid_bulk_modulus? fluid_bulk_modulus? pressure?

One upU example the element command looks like:
element Brick20N_u_p_U 1 5 6 7 8 1 2 3 4 13 14 15 16 9 10 11 12 17 18 19 20 1 0.0 0.0 -9.81 0.8 1.0 2.0 1.0 1.0e-5 1.0e-5 1.0e-5 1.0e3 1.0e8 0
The element is 20 node upU element Brick20N_u_p_U. The element tag is 1. The nodes number of the element is 5 6 7 8 1 2 3 4 13 14 15 16 9 10 11 12 17 18 19 20. The material tag is 1. The body force in x and y direction are zeros, while in z direction is -9.81. The porosity of the element is 0.8, and alpha (in eqn(**??**) is 1.0. The solid density is 1.8, and fluid density is 1.0. The permeability in x, y and z directions are all 1.0e-5. The bulk modulus of solid is 1.0e3 and for fluid is 1.0e8, and the pressure is zero.

C.6 The pattern Command

The pattern command is used to construct the LoadPattern object. There are Load and Constraint objects for the pattern.

pattern patternType patternTag? <arguments for pattern type>

Currently, there are the valid types of pattern are: Plain, UniformExcitation, and MultipleSupport. The Plain pattern has the form:

pattern Plain patternTag? {TimeSeriesType and Args}{
load ...
sp...
}

The argument Plain is used to construct an ordinary LoadPattern object with a unique patternTag. The fourth argument is a list to construct the TimeSeries object associated with the LoadPattern object. The arguments load and sp are used to create a nodal load and single-point constraint.
In upU example, the Plain pattern of Linear type:

pattern Plain 2 Linear {
load 5 0 0 $p 0 0 0 0
load 6 0 0 $p 0 0 0 0
load 7 0 0 $p 0 0 0 0
load 8 0 0 $p 0 0 0 0
load 13 0 0 $np 0 0 0 0
load 14 0 0 $np 0 0 0 0
load 15 0 0 $np 0 0 0 0
load 16 0 0 $np 0 0 0 0
}

create a load pattern with vertical load $p(depends on the value p, if use command set p 5, then $p=5) acting on nodes 5 6 7 and 8, and $np acting on nodes 13 14 15 and 16.

C.7 The system Command

The system command is used to construct the LinearSOE and a LinearSolver objects to store and solve the system of equations. The valid types of systemType are: BandGeneral, BandSPD, ProfileSPD, SparseGeneral, UmfPack, and SparseSPD.
In upU example, UmfPack system command is used.

system UmfPack

to construct a general sparse system of equations which will be factored and solved using UMFPACK solver.

C.8 The constraints Command

The constraints command is used to construct the ConstraintHandler object. The ConstrainHandler object determines how the constraint equations are enforced in the analysis. The valid

constraintType are: Plain, Penalty, Lagrange, and Transformation.
In upU example the Penalty constraints command is used.

The Penalty constraints command

constraints Penalty alphaSP? alphaMP?

is used to construct a PenaltyConstraintHandler which uses the penalty method to constraint. alphaSP and alphaMP are factors that are used when single-point or multi-point constraints are added into the system equations.
In upU example both alphaSP and alphaMP are 1e12.

C.9 The test Command

The test command is used to construct ConvergenceTest object, to determine if convergence has been achieved. The valid convergenceTestType are: NormUnbalance, NormDispIncr and EnergyIncr.
The Norm Displacement Increment Test command is used in upU example.
The Norm Displacement Increment Test command

test NormDispIncr tol? maxNumIter? <printFlag?>

is used to construct a CTestNormDispIncr object which tests if the second-norm of the displacement vector in the LinearSOE object is less than tolerance tol. The maxNumIter is the maximum number of iterations that will performed. The printFlag is used to indicate how to print out the convergence massage. 1 will print the message on each step, and 2 will print when convergence is achieved.

C.10 The integrator Command

The integrator command is used to construct the Integrator object.

integrator integratortType <args for integrator type>

The valid integrator Types for static analysis are: LoadControl, DisplacementControl, MinUnbalDispNorm, Arclength, and arclength1. The valid integratorTypes for dynamic analysis are: Newmark, Newmark1, HHT, and HHT1.
For static analysis in upU example, integrator of LoadControl is used.

integrator LoadControl dlambda1? Jd? minLambda? maxLambda?

This command construct a object of type LoadControl. The argument dlambda1 is the first

load increment in the next invocation of the analysis command. Jd is the argument which related the load increment at ith iteration to the load increment at (i-1)th iteration. Arguments minLambda and maxLambda are used to specified the minimum and maximum load increment. For dynamic analysis in upU example, integrator of Newmark is used.

integrator Newmark gamma? beta?

The argument gamma and beta are two numbers defining the two Newmark parameters γ and β.

C.11 The analysis Command

The analysis command is used to construct the Analysis Object. The valid analysisTypes are: Static and Transient.
The Static analysis command

analysis Static

is used to construct a staticanalysis object to perform a static analysis.
The Transient analysis command

analysis Transient

is used to construct a DirectIntegrationAnalysis object to perform a dynamic analysis.

C.12 The recorder Command

The recorder command is used to construct a Recorder object to record the information of interest. The MaxNodeDisp, Element and Node can be recorded.
The command

recorder Node fileName responseType <-time> -node node1? ... -dof dof1?...

is used to construct a recorder of NodeRecorder to record the node informations at node1 ... The results are saved in the file fileName. The responseType can be disp(displacement), vel(velocity), accel(acceleration), incrDisp(incremental displacement), incrVel(incremental velocity) and incrAccel(incremental acceleration). The dof can be any degree-of-freedom of the nodes which are recorded.

C.13 The wipe Command

The wipe command

wipe

is used to destroy all the exiting objects, and start over a new analysis.