**ICMAC TexGen Workshop 12th July 2018**

* General introduction
  + Overview slide - 2
* Open TexGen
  + Windows – make sure have Controls, Outliner and Log windows open
  + Controls – most commonly used are Textiles and Modeller
  + Outliner – shows yarns and nodes
  + Log window
    - Demonstrate Python console
    - Python output – TexGen is written in C++ but functions are wrapped to give access via Python scripting. GUI calls via Python & output recorded here
    - TexGen output – shows messages output by TexGen – useful to see if any error messages
* Most common starting point – 2D Wizard with default settings
  + Yarn spacing is distance between centrelines of yarns
  + The yarn height will be the fabric thickness / 2
    - Show plain weave without refine
    - Weave pattern dialog
    - Create weave
    - Defaults to elliptical cross-sections
  + Python output shows commands which have been executed
  + Show outliner
  + Render x-ray, nodes, paths
  + Render interference
  + Right button – zoom
  + Centre button – pan
* Slides showing how yarns are built up
  + TexGen class hierarchy – 3/4
  + Modelling theory – 5-13
* Show same plain weave with refine
  + Gap size if want space between yarn surfaces
  + Cross-sections have been changed
  + Show changed to interp between nodes
  + Show use of options in modeller window to assign sections etc
* Slides showing CTextileWeave class
  + Classes inherited from CTextile – 14
  + CTextileWeave2D - 15
* 2D wizard 4x4 textile
  + Show selecting bars – ctrl-click to select multiple
  + Change 2nd & 4th yarns to 0.9 width, 0.08 height
* Save textile
  + File->SaveTexGenFile
  + Edit file – standard XML file
  + Can edit file
  + Show weave data part of file
* Manual creation of yarns
  + *Record script?*
  + Create empty textile
  + Create default yarn
  + Select & duplicate yarn – use button or ctrl-d
  + Drag to y = 5
  + Select yarn 0 from outliner & duplicate
  + Select node 1 using outliner
  + Type x = 0, y = 10
  + Select yarn 2 using outliner & duplicate
  + Drag to x = 5
  + Select node 1 in each yarn & insert node
  + Select nodes 0 & 2 of yarns 0 & 3 and node 1 of yarns 1 & 2
  + Type z = 2
  + Select all yarns
  + Assign section
  + Change to lenticular, width = 4, height = 2
  + Create box domain max x & y = 10, min z = -1, max z = 3
  + Trim to domain
  + Select all yarns
  + Assign repeats x = 10 & y = 10
  + Show render interference depth – look at node 1
  + Select yarn 1
  + Assign section – change to interpolate between nodes
  + Select yarn 1 – change distortion ( to 0.1 or 0.2?)
  + Show interference depth
  + Select hybrid section – divide into 2 horizontally
  + Leave top section
  + Lower section – height = 1.8
  + Show interference depth
* Python editing of textile
  + Textile = GetTextile(‘name’)
  + yarn = textile.GetYarn(1)
  + yarn.SetResolution(20) (or numSlaveNodes, numSectionPoints)
  + newNode = CNode( XYZ(5, 5, 2.4)
  + yarn.ReplaceNode(1, newNode )
  + Scripting guide <https://github.com/louisepb/TexGenScriptingGuide>
* Export options
  + *Yarn properties*
* 3D textiles
  + *Slides showing CTextileWeave3D*
  + *Diagram showing grid structure*
  + *All use same base class – can’t create using this in GUI*
  + 3D weave wizard
  + Start with orthogonal
  + 4 wefts, 4 layers
  + Weft spacing needs to be sufficient to accommodate height of binder
  + Explain warp/binder ratio
  + *Find yarn properties etc to demonstrate refine*
  + Demonstrate layer to layer
    - Increased number of layers governs max number of binder layers
    - Warp layers linked to number of weft layers