AceGen - Getting started - table of contents

- double-click the cells on the right to “Open/Close Group” and see the content of each step and a link to the notebook
- use search function to e.g. find steps working with “tensor” or using the “deformation gradient”

In[ ]:=
```
SetDirectory[NotebookDirectory[]];
ShowStepLinkAndContent[] := (
   (* Get the name of the Step folder from the previous section cell *)
   path = StringJoin[NotebookRead[PreviousCell[]][[1, 1]]];
   path = StringReplace[path, "\"" → ""];
   (* Search for notebook file in path,
   but exclude variants that start with "_" *)
   path_file = First[Complement[FileNames["*.nb", path], FileNames["_*.nb", path]]];
   Print[Hyperlink[path_file]];
   Print[
    First[NotebookImport[path_file, "Abstract" → "Text"], "no content available"]]
  );
```

"Step001_Basics"

In[ ]:=
```
ShowStepLinkAndContent[]
```

Step001_Basics /AceGen -InCalcOut .nb

content :
- basic usage of AceGen
- compute scalar output y based on input x and derivative dy/dx
- scalars
- input/output
- SMSInitialize
- SMSModule
- SMSReal
- SMSD
- SMSExport
- SMSPrintMessage
- SMSWrite

"Step002_SetDelayed"

In[ ]:= **ShowStepLinkAndContent []**

Step002_SetDelayed /AceGen-SetDelayed .nb

content :
- use of Mathematica 's SetDelayed ":=" operator to create function-like construct
- organise equations
- accessibility of variables inside SetDelayed
- use of modulePureLocal
- scalars
- SMSInitialize , SMSModule , SMSReal , SMSPrintMessage , SMSWrite

"Step020_SMSIf_Conditions"

In[ ]:= **ShowStepLinkAndContent []**

Step020_SMSIf _Conditions /AceGen-SMSIf .nb

content :
- basic usage of SMSIf for conditions
- compute scalar output y based on input x and derivative dy/dx
- scalars
- input/output
- SMSIf , SMSElse , SMSEndIf
- SMSInitialize , SMSModule , SMSReal , SMSD , SMSExport , SMSPrintMessage , SMSWrite

"Step101_MaterialModel-linearElastic"

In[ ]:= **ShowStepLinkAndContent []**

Step101_MaterialModel -linearElastic /AceGen-LinearElasticity .nb

content :
- linear elastic material model
- input is deformation gradient as 3x3 matrix
- output is stress as 6x1 vector and
  stress-strain tangent as fourth order tensor (3x3x3x3)
- tensor/vector
- input/output
- SMSFreeze
- Symmetric
- SMSInitialize , SMSModule , SMSReal , SMSD , SMSExport , SMSPrintMessage , SMSWrite

"Step112_MaterialModel-FiniteElastoPlasticity-spectralDecomposition"

In[ ]:= **ShowStepLinkAndContent []**

Step112_MaterialModel -FiniteElastoPlasticity -spectralDecomposition /AceGen-FiniteElastoPlasticity -spectralDecomposition .nb

content :

- material model
- multiplicative plasticity
- spectral decomposition , use of eigenvalues /eigenvectors
- SMSCall
- SMSInitialize , SMSModule , SMSReal , SMSD , SMSExport , SMSPrintMessage , SMSWrite
- AceGen building time : ~20 s (Mode : Debug), ~45s (Mode : Optimal)

"Step115_MaterialModel-FiniteElastoPlasticity-exponentialMap-kinematicHardening"

In[ ]:= **ShowStepLinkAndContent []**

Step115_MaterialModel -FiniteElastoPlasticity -exponentialMap -kinematicHardening /Pmulti_expM_HisokinAF .nb

content :

- material model
- multiplicative plasticity
- exponential map
- kinematic hardening
- modular concept
- SMSInitialize , SMSModule , SMSD , SMSDo , SMSCall
- AceGen building time : ~100 s (Mode : Debug), ~300 s (Mode : Optimal)

"Step215_ElementFormulation3D_Q1X_callMatMod"

In[ ]:= **ShowStepLinkAndContent []**

Step215_ElementFormulation3D_Q1X_callMatMod/Q1X_callMatMod.nb

content:
- element formulation
- geometrically nonlinear
- linear ansatz function
- 3D
- full or reduced integration (no stabilisation)
- optionally F-bar approach to avoid volumetric locking
- call to external material model, such as "Pmulti_eig" from Step112_MaterialModel -FiniteElastoPlasticity -spectralDecomposition
- SMSIf, SMSCall, SMSD, SMSFreeze
- AceGen building time: ~10 s (Mode: Debug), ~20 s (Mode: Optimal)