

Isight Tips : 使用 script 客製化 Abaqus component

簡介

在 Isight 中，利用 Abaqus component 可以達到 Isight 與 Abaqus solvers 及 Abaqus/CAE 資料交換的目標。Abaqus component 藉由掃描 (*.cae) 檔，可得到模型中輸入的參數。其中支援的參數非常多元，如圖表 1 列。

Category	Parameters
Part features	Numerical features, such as diameter and extrusion depth
Geometry features	Sketcher dimensions Extrusion depth Extrusion direction
Material properties	Material table data
Section properties	Numerical values, such as shell thickness
Interaction properties	Friction coefficient
Loads	Magnitudes Concentrated force components Body force or gravity components Connector force components Connector moment components
Boundary conditions	
Beam profiles	

圖表 1、Abaqus component 支援之參數(Isight 2017)

本文將介紹如何使用 script 客製化 Abaqus component，支援更多元的參數形式。

註：本公司關於 Python script 部分不提供技術服務，需要進一步支援可用專案方式執行之。

Isight runtime 執行時，Abaqus component 會自動產生“execInp.py”的 Python script 檔案於 Isight 執行目錄，script 會做下列的工作：

1. 開啟指定的 Abaqus/CAE 模型檔的複本。
2. 更改由 Isight 中設定好的 Abaqus input 參數。
3. 如果有變動到幾何，則重新產生幾何。
4. 如果需要重新 mesh，則重新 mesh 有線元素模型。
5. 重新產生 Abaqus/CAE 內定義所有 job 的 input 檔案。
6. 儲存 Abaqus/CAE 模型檔。

● 針對 CAE files 之客製化

Script 中預留了兩個擴充模組檔，可供客製化使用。透過編輯此二檔案，可以匯入原 Abaqus component 尚未支援的參數。

1. userscript_cae_pre.py：此模組是完成上述“execInp.py”步驟 1~2。
2. userscript_cae_post.py：此模組是完成上述“execInp.py”步驟 4~5。

execInp.py 中匯入 userscript_cae_pre.py 的程式碼如下：

```
if os.path.exists('userscript_cae_pre.py'):
    import userscript_cae_pre
    userscript_cae_pre.runUserScript(mdb, values)
```

匯入 userscript_cae_post.py 的程式碼如下：

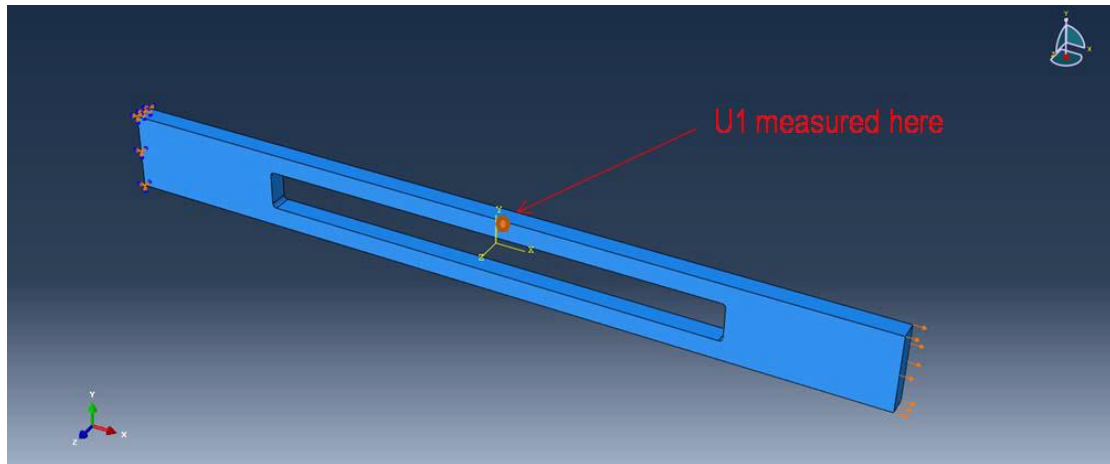
```
if os.path.exists('userscript_cae_post.py'):
    import userscript_cae_post
    userscript_cae_post.runUserScript(mdb, values)
```

其中 mdb 是模型的資料庫，values 是 input 的陣列。

● 針對 odb files 之客製化

可以修改 userscript_odb.py 來修改輸出的變數。

範例

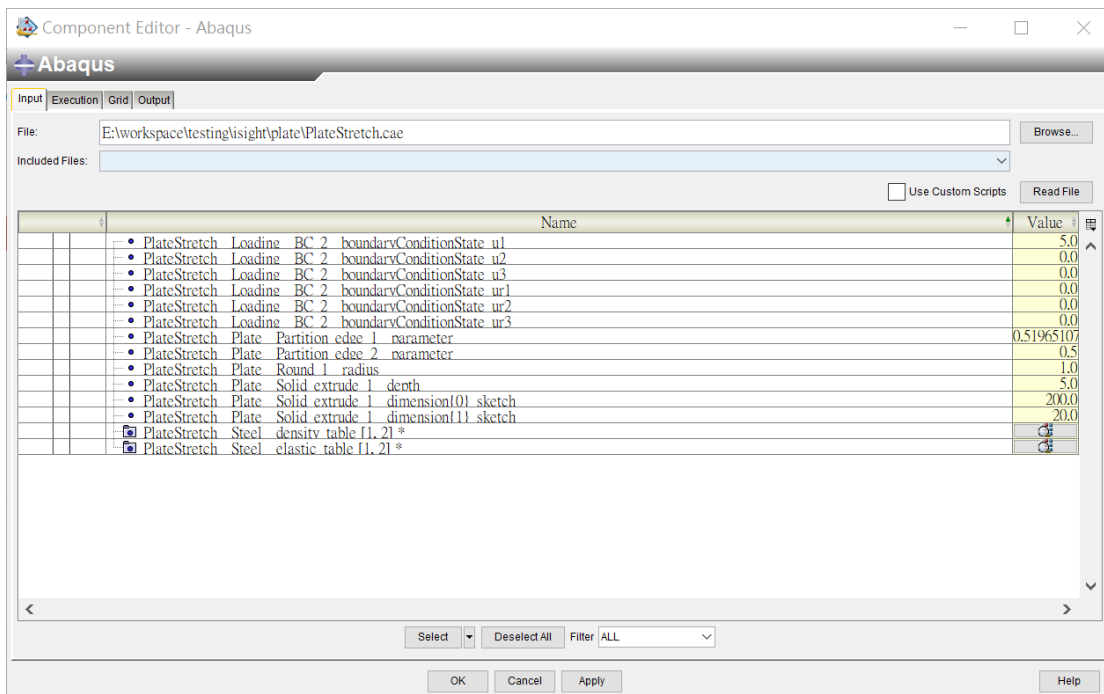


圖表 2、平板拉伸 CAE 模型檔案

以一平板拉伸之模型為例，分別對 CAE 檔以及 ODB 檔修改參數：

1. 藉由改變 MeshSize 做敏感度分析，看對中心點的位移結果會產生多大的影響。
2. 將 output U1 值單位由 mm 轉換成 inches。

若直接使用 Isight 之 Abaqus component 掃描此.cae 檔案，會發現 Mesh Size 尚不能被主動添加成一個變數，如圖表 3。



圖表 3、Isight Abaqus component 中 input 介面

MeshSize

首先需要在 .CAE 之同一資料夾內建立 userscript_cae_pre.py，並且寫入刪除、重新 mesh 之 script 如下。

```
# userscript_cae_pre.py

from abaqus import *

def runUserScript(mdb, values):

    p = mdb.models['PlateStretch'].parts['Plate']

    p.deleteMesh() #刪除 mesh

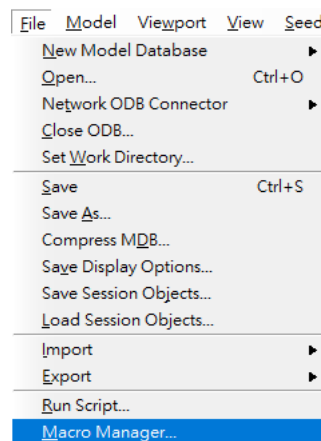
    p.seedPart(size=float(values['MeshSize']), deviationFactor=0.1, minSizeFactor=0.1) #變數名稱為 MeshSize

    p.generateMesh() #重新 mesh

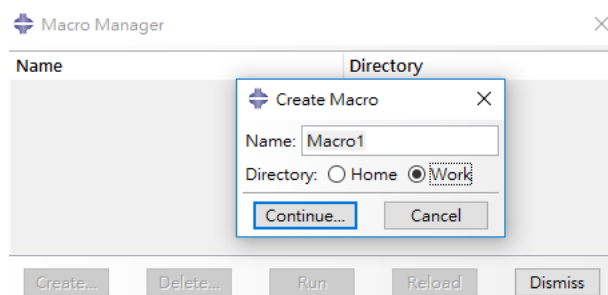
    p.regenerate() #重新產生 part

    mdb.models['PlateStretch'].rootAssembly.regenerate() #重新產生 assembly
```

若不知道相關 script 代碼，可由 Abaqus/CAE 之 File>Macro Manager 中錄製相關步驟並修改，如圖表 4。新增一 Macro 於家目錄或是工作目錄之下，並重新操作更改 MeshSize、重新 mesh 之功能，如圖表 5。



圖表 4、CAE 中開啟 Macro Manager 之方法



圖表 5、新錄製 Macro 於工作目錄之下

開啟 abaqusMacros.py，找到錄製之 macro，刪除 import 資料、並將多餘的視角程式碼刪除，將 mesh size 的地方改為變數 float(values['變數名稱'])，並複製至 userscript_odb.py 的"def

runUserScript(mdb, values):”行之下，即完成編輯 userscript_cae_pre.py。

```
# abaqusMacros.py
# -*- coding: mbcs -*-
# Do not delete the following import lines

from abaqus import *
from abaqusConstants import *
import __main__

def Macro1():

    import section

    import regionToolset

    import displayGroupMdbToolset as dgm

    import part

    import material

    import assembly

    import step

    import interaction

    import load

    import mesh

    import optimization

    import job

    import sketch

    import visualization

    import xyPlot

    import displayGroupOdbToolset as dgo

    import connectorBehavior

    p = mdb.models['PlateStretch'].parts['Plate']

    session.viewports['Viewport: 1'].setValues(displayedObject=p)

    session.viewports['Viewport: 1'].partDisplay.setValues(mesh=ON)

    session.viewports['Viewport: 1'].partDisplay.meshOptions.setValues(meshTechnique=ON)

    session.viewports['Viewport: 1'].partDisplay.geometryOptions.setValues(referenceRepresentation=OFF)

    p = mdb.models['PlateStretch'].parts['Plate']

    p.deleteMesh()

    p = mdb.models['PlateStretch'].parts['Plate']

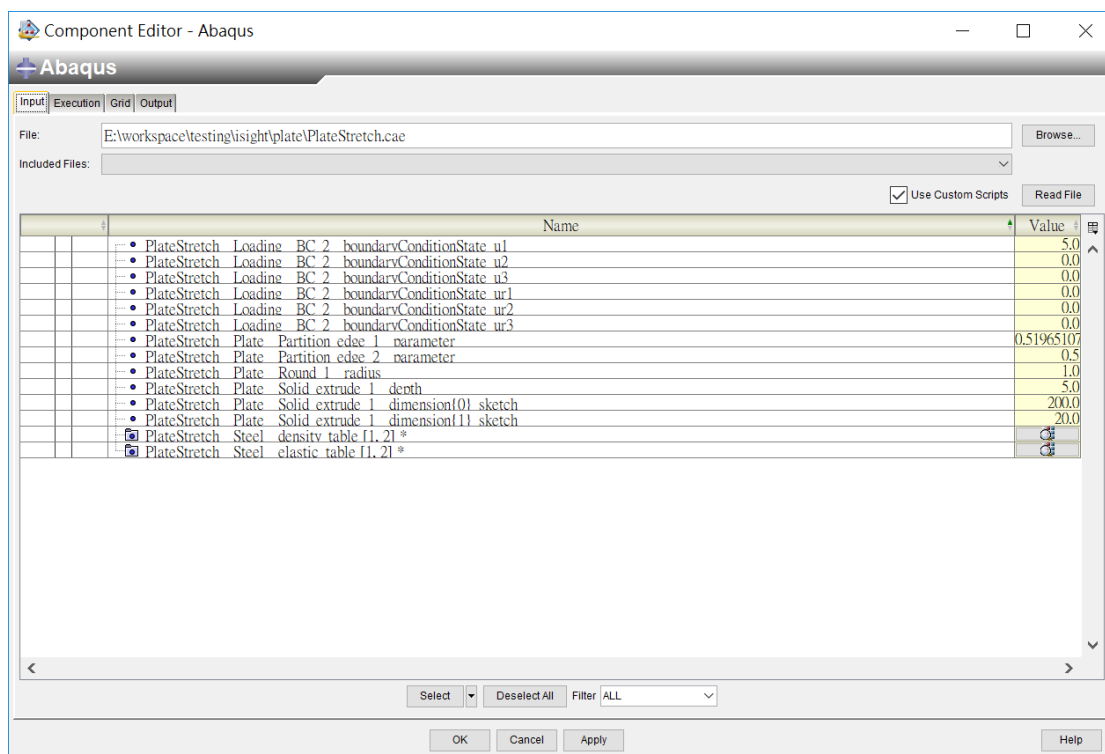
    p.seedPart(size=2.0, deviationFactor=0.1, minSizeFactor=0.1)

    p = mdb.models['PlateStretch'].parts['Plate']

    p.generateMesh()
```

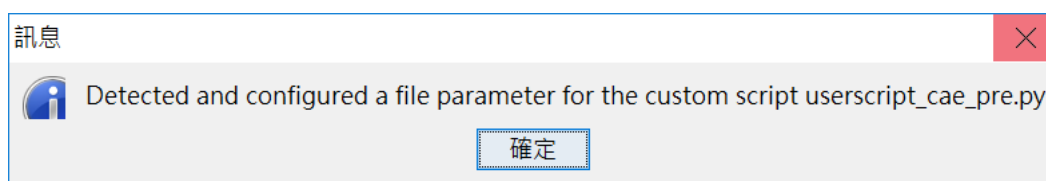
士盟科技 CAE 團隊

回到 Isight 的 Abaqus component，勾選 Use Custom Scripts，重新 Read File，如圖表 6。



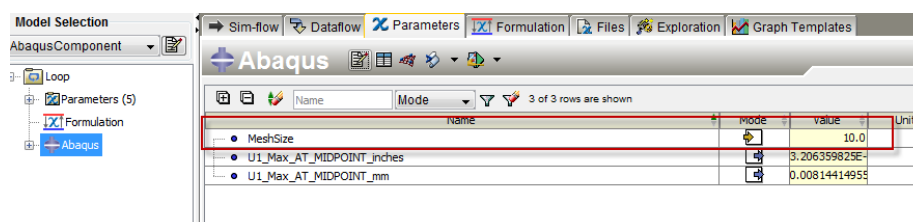
圖表 6、Isight Abaqus component 中 input 介面

若讀取成功則會出現以下訊息，如圖表 7：



圖表 7、讀取 userscript_cae_pre.py 成功訊息

手動添加 MeshSize 參數於 Parameters 中，如圖表 8。



圖表 8、手動添加參數。

更改後處理

userscript_odb.py 創造兩個客製化的 U1 參數，單位各為 mm 以為 inches。Isight 以“user_params.txt”辨認客製化的輸出參數。首先需要在 .ODB 之同一資料夾內建立

userscript_odb.py

```
# userscript_odb.py

import odbAccess

from odbAccess import *

import __main__

import operator

if __name__ == '__main__':

    odbName = "PlateStretch.odb" # 欲修正的.odb 之檔案名稱

    myOdb = openOdb(odbName,readOnly=True) # 以唯讀方式開啟 odb 檔案

    U1_Measure = myOdb.rootAssembly.instances['PLATE-1'].nodeSets['U1_MEASURE'] # 存取模型中的集合 U1_MEASURE

    U1_Max_AT_MIDPOINT_mm = myOdb.steps['Loading'].frames[-1].fieldOutputs['U'].getScalarField(componentLabel = 'U1').getSubset(region =
U1_Measure).values[0].data # 由 odb 檔讀取最後一個 frame 集合 U1_MEASURE 之 U1 資料

    U1_Max_AT_MIDPOINT_inches = 0.0393701 * U1_Max_AT_MIDPOINT_mm # 將存取出來的資料作單位轉換

    paramsFile=open('user_params.txt','w') # 打開 Isight 儲存客製化使用者變數的檔案

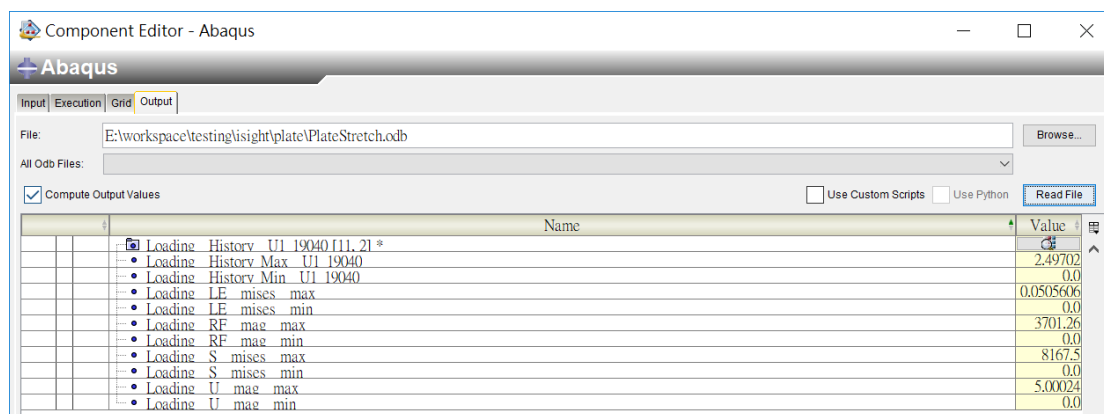
    paramsFile.write("U1_Max_AT_MIDPOINT_mm"+"\\t"+ str(U1_Max_AT_MIDPOINT_mm)+"\\n") # 寫入 mm 單位之輸出值

    paramsFile.write("U1_Max_AT_MIDPOINT_inches"+"\\t"+ str(U1_Max_AT_MIDPOINT_inches)+"\\n") # 寫入 inches 單位之輸出值

    paramsFile.close() # 關閉 user_params.txt

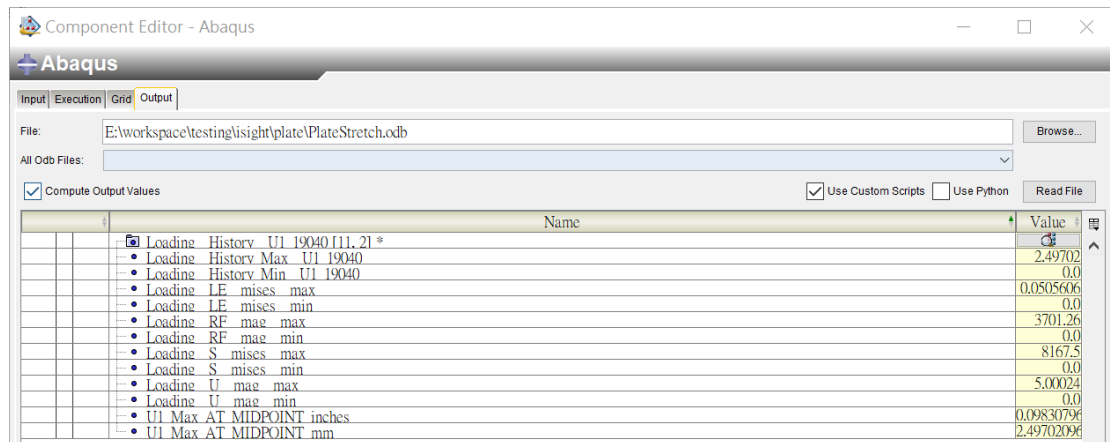
    myOdb.close() # 關閉 odb 檔案
```

使用 userscript_odb.py 前，Abaqus Component 並不會讀到客製化參數，如圖表 9。



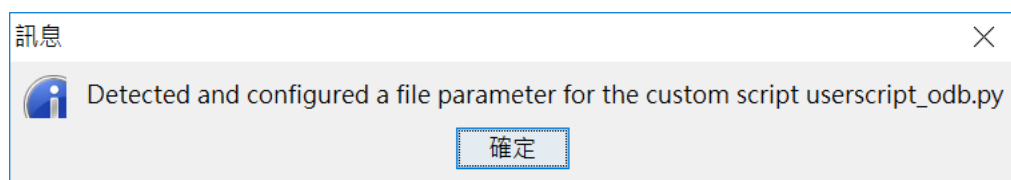
圖表 9、Isight Abaqus component 中 output 介面

回到 Isight 的 Abaqus component，勾選 Use Custom Scripts，重新 Read File，會新增兩個參數，如圖表 10。



圖表 10、Isight Abaqus component 中 output 介面

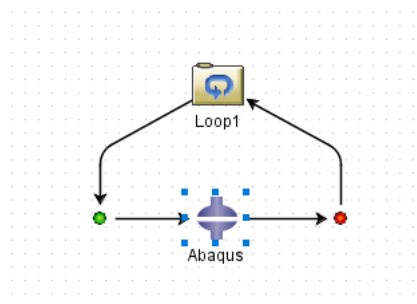
若讀取成功則會出現以下訊息：



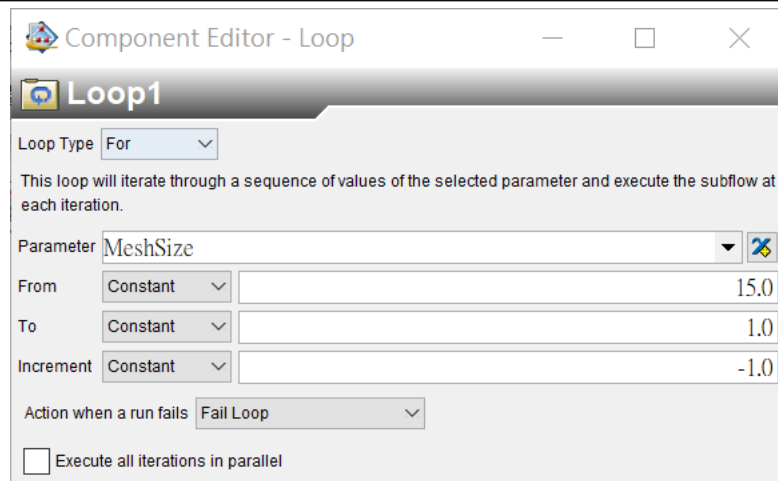
圖表 11、讀取 userscript_odb.py 成功訊息

Isight scenario

設定完 Abaqus Component 之後，設定 MeshSize 由 15 遞減至 1 的迴圈，如圖表 12、圖表 13。



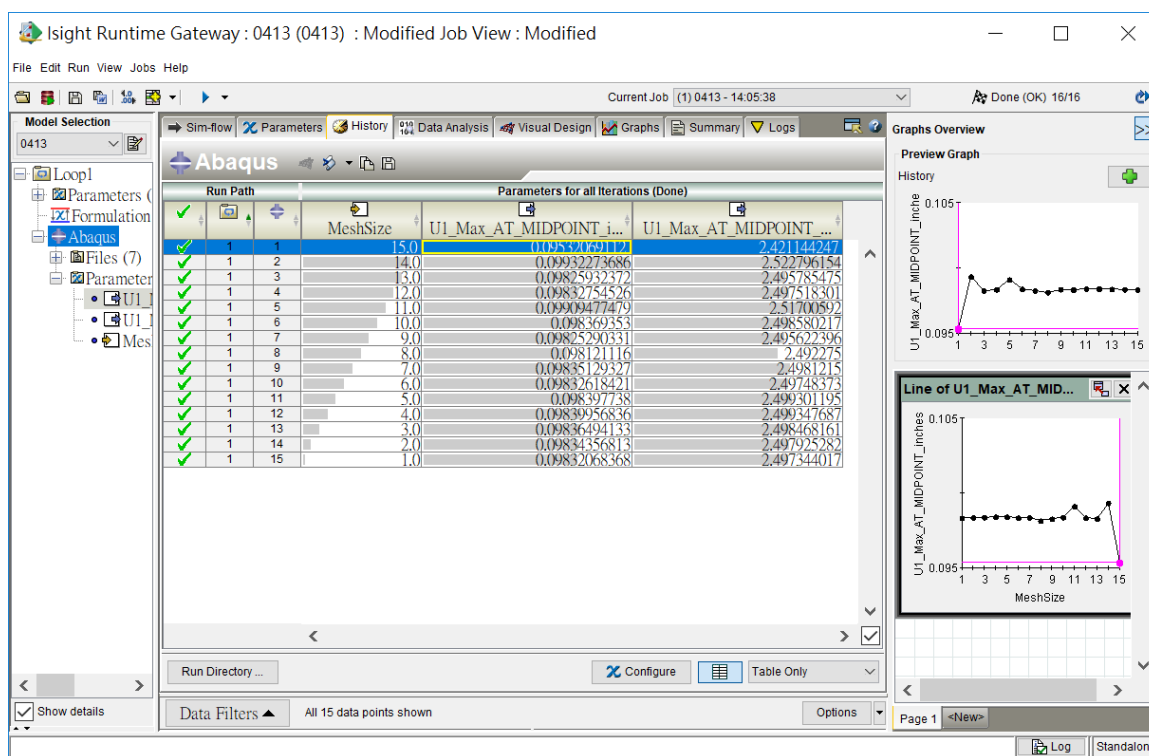
圖表 12、Isight design gateway 之介面



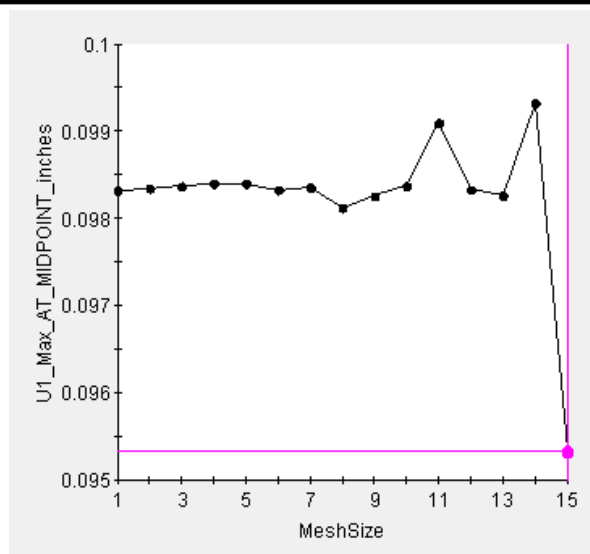
圖表 13、Loop 中之設定

分析結果

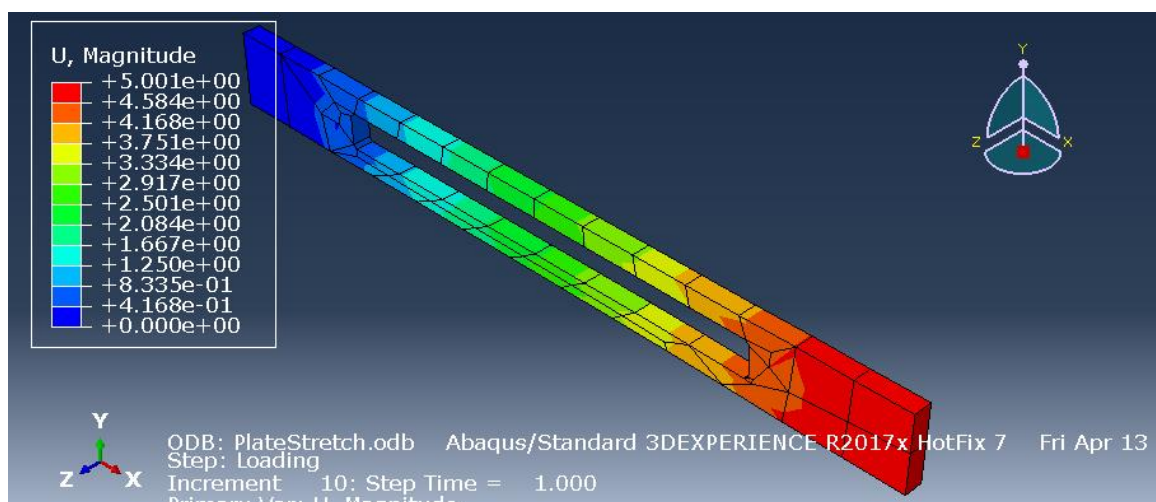
由 Isight 介面中可直接得知 MeshSize 對兩個客製化參數 U1 值的影響，除了數值之外也能以二維圖顯示之，請見圖表 14、圖表 15。也可以針對特定結果開啟 Abaqus .odb 檔接續後處理之設定，如圖表 16、圖表 17、圖表 18、圖表 19。



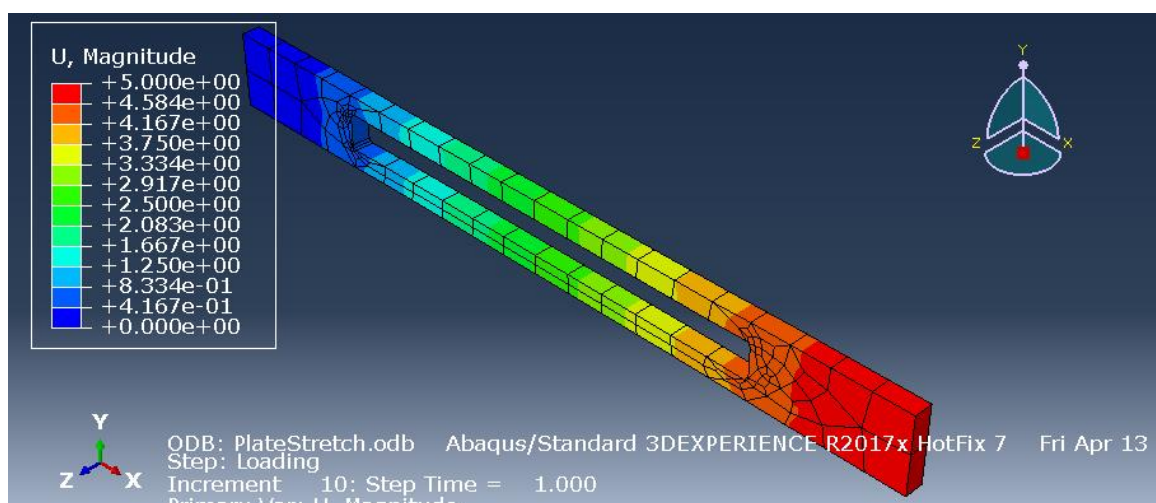
圖表 14、Isight Runtime Gateway 觀看結果表格及圖表。



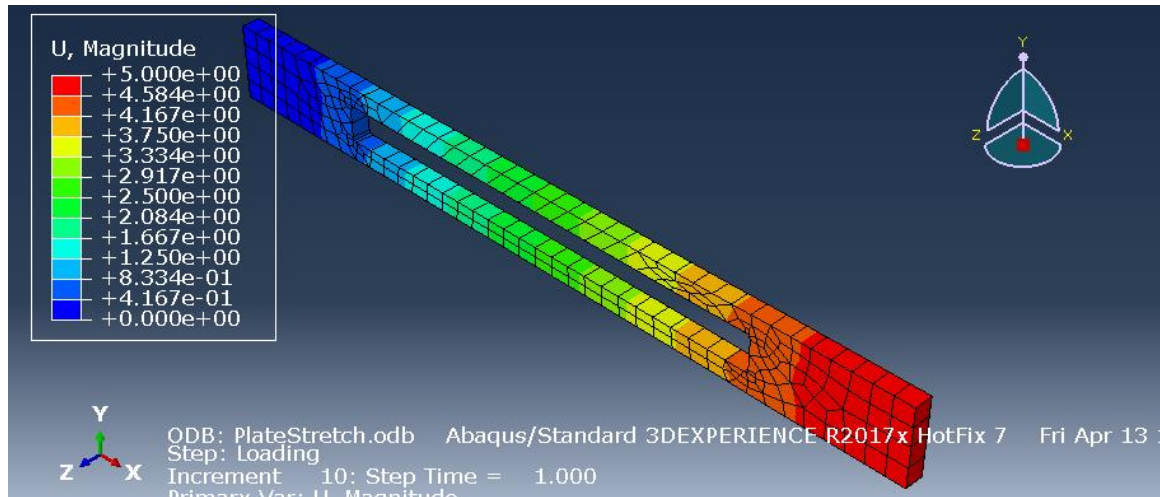
圖表 15、Isight 輸出之 U1-inches/MeshSize 圖



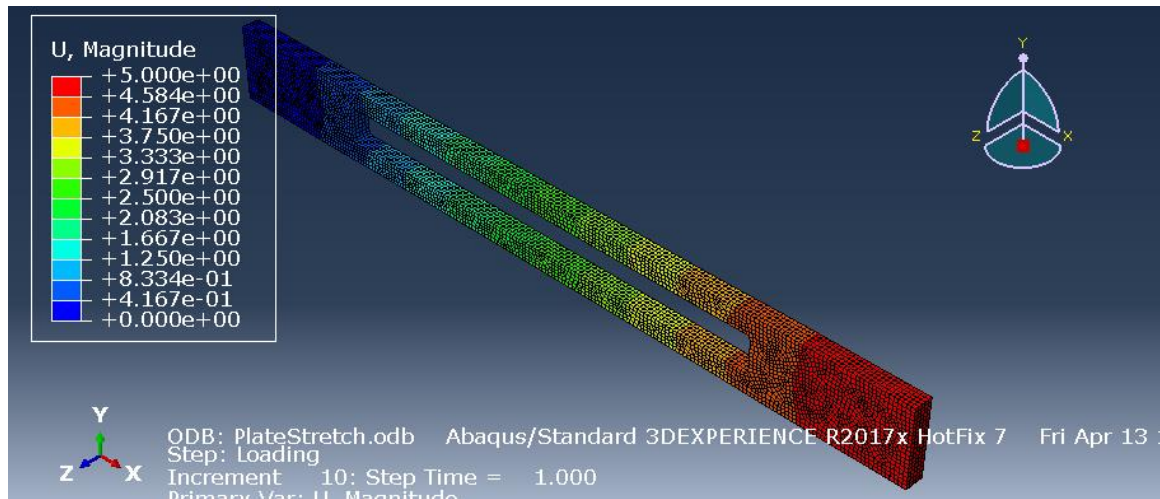
圖表 16、MeshSize=15 之位移量 Contour



圖表 17、MeshSize=10 之位移量 Contour



圖表 18、MeshSize=5 之位移量 Contour



圖表 19、MeshSize=5 之位移量 Contour

文、應用工程師 Freda