

LUSAS 기술자료

LPI를 사용한 EXCEL 활용

(결과 정렬)

2005. 6.

에프이에이코리아(주)

1. 개요

USAS는 LPI(LUSAS Programmable Interface)를 제공하고 있는데, 이것은 LUSAS Modeler에서 이루어지는 모델링 단계에서부터 결과출력까지의 전 과정을 Visual Basic Script를 사용하여 자동화할 수 있는 기능이다. 이와 관련된 내용은 LUSAS Modeler Manual이나, 홈페이지(<http://www.lusas.co.kr>)에 게시된 온라인강좌 등을 참조하면 된다.

본 기술노트에서는 Visual Basic Script를 사용하여 LUSAS 해석결과 중에서 필요한 값을 추출하여 EXCEL의 지정한 위치에 옮기도록 하는 작업을 다루어보고자 한다.

이를 통하여 다양한 하중케이스에 대한 결과값을 일괄정렬하거나, 여러 그룹들에서 발생하는 최대값들을 사용자의 작업 없이 한번에 EXCEL에 일목요연하게 정리되도록 하는 등으로 응용할 수 있게 되어 해석 후 Post-processing에 필요한 시간을 혁신적으로 단축시킬 수 있게 될 것이다.

1.1. 사용할 모델

3경간연속교를 빔 요소로 구성하고, 이동하중을 재하시킨 모델을 사용한다.

1.2. 작성할 VBS가 수행할 작업 목표

각 경간 중앙 절점에서의 연직 처짐과 흐모멘트의 변화를 각 하중 이동 단계별로 엑셀 파일에 자동 출력한다.

2. 모델의 준비

본 기술노트에서 사용한 모델은 아래와 같이 3경간 교량을 보요소로 이상화한 것으로 하며 기술노트와 함께 제공되는 모델 파일을 사용하기로 한다.



그림 1 기본모델 형상

2.1. 요소의 종류 및 개수

2차원 보요소(BEAM)를 사용하였으며, 경간장 30m/40m/30m을 각각 10개의 요소로 분할하였다.

2.2. 제원 (기하/재료)

콘크리트로 가정하고, 모든 단면은 균일한 것으로 단순화하였다.

2.3. 하중조건

DB24하중을 10단계에 걸쳐서 이동재하 하였다.

2.4. 구속조건

1경간 시점부를 고정단으로 하고, 나머지는 X방향 가동단으로 하였다.

3. 해석의 수행

해석을 수행하고 결과파일을 모델파일 위에 불러들이면 아래와 같이 처짐도 등을 확인할 수 있다.

아래 작업창에 표시된 숫자는 각 경간 중앙의 절점번호이다.

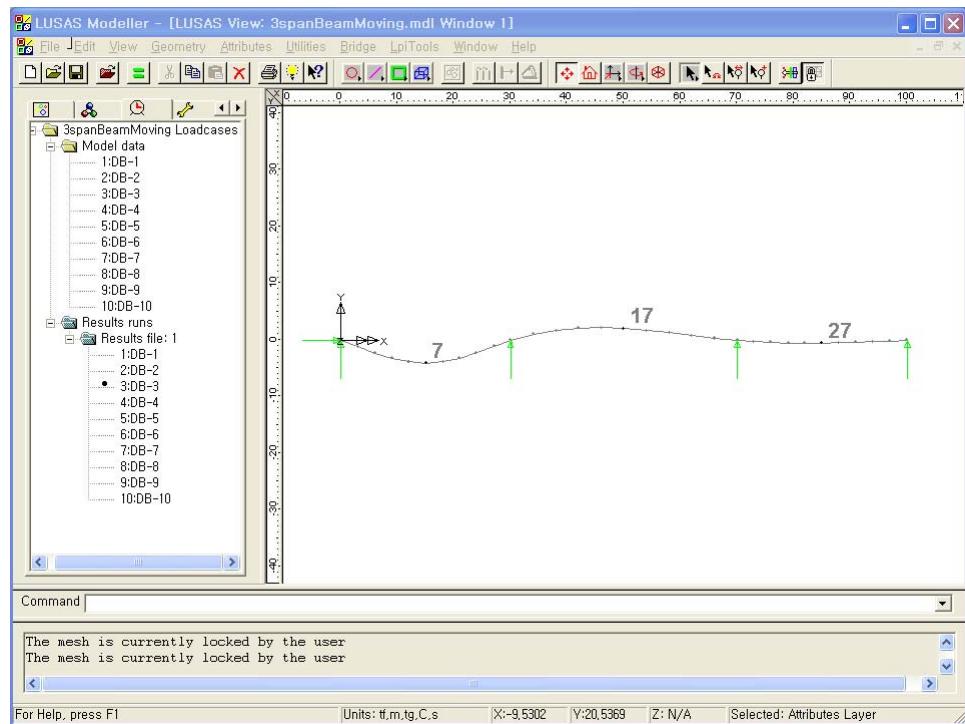


그림 2 해석 수행 후 결과 파일을 불러온 상태

4. VBScript 작성

이제 각 경간 중앙 절점의 처짐과 힘모멘트값의 변화를 EXCEL 파일에 기록하도록 하는 비쥬얼베이직 스크립트 (VBS, Visual Basic Script)를 작성하도록 한다.

4.1. 엔진선언 및 주요변수 선언

4.1.1. 엔진선언 및 엑셀Object의 정의

LUSAS LPI에서 사용할 언어가 VBS라는 것을 선언하기 위해서 [\$ENGINE=VBSCRIPT] 문장을 사용한다.

주의할 것은 이 문장은 반드시 대문자를 사용하여야 하며, 이 문장 앞에는 다른 어떤 문자도 있어서는 안된다.

엑셀을 프로그램 내에서 호출하기 위해서는 엑셀 Object를 정의하여야 하며, 아래와 같이 지정한다.
(외따옴표는 프로그램의 설명을 추가할 때 사용하는 것으로, 프로그램 실행시에는 무시된다.)

```
$ENGINE=VBSCRIPT  
  
'엑셀 Object 정의  
Set ExcelApp = CreateAutomationObject("Excel.Application")  
ExcelApp.DefaultFilePath = Path
```

표 1 엔진선언 및 EXCEL Object 정의

4.1.2. 사용할 EXCEL 파일명과 시트명 지정

기존에 만들어져 있는 EXCEL 파일을 호출하여 사용하고자 하는 경우는 아래와 같이 작성한다.

엑셀파일을 작업을 위해서 연다고 표현을 하였지만, 수행과정에서 화면상에 나타나지는 않고 작업만 수행된다.

```
'파일명 지정 (현재 작업디렉토리에 할 경우는 경로를 기록하지 않고 파일명만 입력하면 된다.)  
ExcelFileName = "c:LUSAS135\PROJECTS\RESULT.xls"  
  
'기존 엑셀파일 열기  
Set WorkBook = ExcelApp.WorkBooks.open(ExcelFileName)  
  
'열려진 엑셀파일의 첫 번째 시트를 WorkSheet1이라는 이름의 작업대상(Object)로 지정  
Set WorkSheet1= WorkBook.WorkSheets(1)  
  
'열려진 엑셀파일의 두번째 시트를 WorkSheet2이라는 이름의 작업대상(Object)로 지정  
Set WorkSheet2 WorkBook.WorkSheets(2)  
  
'....이와 같이 여러개의 시트를 대상으로 작업할 수 있도록 사전에 Object를 선언할 수 있다.
```

표 2 기존 EXCEL 파일에 기록하려고 할 경우 엑셀관련 Object 선언 방법

새로운 파일을 만들어서 결과를 기록하고 저장하고자 한다면 아래와 같이 기록한다.

```
'파일명 지정 (현재 작업디렉토리에 할 경우는 경로를 기록하지 않고 파일명만 입력하면 된다.)  
ExcelFileName = "c:LUSAS135\PROJECTS\RESULT.xls"  
'엑셀파일 생성시키기  
Set WB = ExcelApp.WorkBooks.Add  
'생성된 새 엑셀파일의 첫 번째 시트를 WorkSheet1 이라는 이름의 작업대상(Object)로 지정  
Set WorkSheet1= WorkBook.WorkSheets(1)  
'생성된 새 엑셀파일의 두번째 시트를 WorkSheet2이라는 이름의 작업대상(Object)로 지정  
Set WorkSheet2 WorkBook.WorkSheets(2)  
'....이와 같이 여러개의 시트를 대상으로 작업할 수 있도록 사전에 Object를 선언할 수 있다.
```

표 3 새로운 EXCEL 파일을 생성시켜 기록하려고 할 경우 엑셀관련 Object의 선언 방법

이 때 새로이 생성된 EXCEL 파일에는 3개의 시트까지 사용할 수 있게 된다.

3개 이상의 시트를 대상으로 작업을 수행해야 할 경우에는 [엑셀파일Object].Sheets.Add 명령을 사용하여 추가하면 되는데, 아래와 같이 사용하면 된다.

```
ExcelFileName = "c:LUSAS135\PROJECTS\RESULT.xls"  
Set WB = ExcelApp.WorkBooks.Add  
Set WorkSheet1= WorkBook.WorkSheets(1)  
Set WorkSheet2 WorkBook.WorkSheets(2)  
Set WorkSheet3 WorkBook.WorkSheets(3)  
WorkBook.Sheets.Add  
Set WorkSheet4 WorkBook.WorkSheets(4)  
WorkBook.Sheets.Add  
Set WorkSheet5 WorkBook.WorkSheets(5)  
....
```

표 4 EXCEL 파일 내 작업시트의 추가방법

LPI를 사용한 EXCEL 활용 (결과정렬)

작업시트의 이름을 지정하려고 하면, [엑셀시트Object].Name=[시트명] 과 같이 사용하면 되는데, 본 예제에서 첫 번째 시트에는 각 절점의 처짐값을, 두 번째 시트에는 각 절점의 흐름모멘트값을 기록할 것이며, 이름을 각각 [처짐]과 [흐모멘트]로 명명하기로 하면 아래와 같다.

```
ExcelFileName = "c:LUSAS135\PROJECTS\RESULT.xls"  
Set WB = ExcelApp.WorkBooks.Add  
Set WorkSheet1= WorkBook.WorkSheets(1)  
Set WorkSheet2 WorkBook.WorkSheets(2)  
WorkSheet1.Name = "처짐"  
WorkSheet2.Name = "흐모멘트"
```

표 5 EXCEL 시트의 정의 및 이름 변경

4.2. LUSAS 결과파일 지정 및 데이터베이스 Object 정의

해석결과파일을 지정하는 방법은 아래와 같다.

현재 모델러에 모델파일과 결과파일이 불려져 있는 상태에서 VBS를 실행시켜 결과처리작업을 하려고 하는 것이라면 이 단계는 필요가 없을 것이다. 그러나, 여러 개의 파일을 순차적으로 불러들여서 작업을 할 필요가 있는 경우 등에는 요긴하게 사용될 수 있을 것이다.

```
'결과파일명 지정
ResultFileName = "c:LUSAS135\PROJECTS\3spanBeamMoving.mys"
'모델러에 불려진 모델링 파일의 내용을 의미하는 Object의 정의
Set db=getDatabase()
'결과파일 모델러에 불러오기
db.openResults ResultFileName
```

표 6 해석결과파일 불러오기

모델에 있는 그룹들을 결과출력에도 활용을 하고자 하면 모델과 결과를 같이 불러와야 할 것이다. 이럴 경우에는 먼저 모델파일을 불러들인 다음, 결과파일을 그 위에 추가적으로 열 수 있도록 아래와 같이 지정한다.

```
'모델파일명 지정
ModelFileName = "c:LUSAS135\PROJECTS\3spanBeamMoving.m01"
'모델파일 불러오기,false는 읽기전용이 아니라는 의미
opendatabase ModelFileName, false
'모델러에 불려진 모델링 파일의 내용을 의미하는 Object의 정의
Set db=getDatabase()

'결과파일명 지정
ResultFileName = "c:LUSAS135\PROJECTS\3spanBeamMoving.mys"
'열려져 있는 모델파일 위에 결과파일을 추가적으로 불러오기
db.openResults ResultFileName
```

표 7 모델 위에 해석결과파일을 추가로 불러오기

이상과 같이 함으로써 모델러에는 결과파일 혹은 모델파일과 결과파일이 열리게 되며, 이 모델링/결과와 관련된 모든 데이터베이스를 활용하기 위해서 getDatabase() 라는 LUSAS가 제공하는 Method를 사용하여 db 라는 이름의 데이터베이스 Object가 정의되었다.

4.3. 결과출력대상 절점번호 지정

결과출력을 할 대상 절점은 7, 17, 27번이므로 아래와 같이 배열변수를 지정하여 저장하기로 한다.

```
'배열변수 선언  
Redim targetnode(2)  
'배열변수에 결과를 얻을 절점번호를 저장  
targetnode(0) = 7  
targetnode(0) = 17  
targetnode(0) = 27
```

표 8 결과출력 대상 절점번호 지정

4.4. 하중케이스별 결과값을 엑셀에 기록

4.4.1. 결과출력 대상 하중케이스의 순차적 변경

이제 하중케이스를 바꾸어 가며 작업을 할 수 있도록 하는 for-next 루프 문장을 구성해 보면 아래와 같다.

```
For i=1 to 10 step 1  
    processcommand("SET LOADCASE LCID=&i&" IEIGEN=-1 IHARMS=-1 IRES=0")  
    '  
    '결과얻어내고 엑셀에 기록하게 하는 내용 추가  
    '  
Next
```

표 9 하중케이스를 변경시켜가며 작업을 하는 것

4.4.2. 하중케이스별 처짐/모멘트 기록

지정된 하중케이스 내에서 LUSAS가 제공하는 Method를 사용하여 지정된 절점의 처짐과 모멘트값을 얻어내는 문장은 아래와 같다. (자세한 내용은 LUSAS Modeler User Manual 8장, Programmable Interface를 참조)

```
'절점번호 n를 가진 절점Object 선언
Set node = db.getNodeByNumber(n)
'n번 절점의 Y방향의 처짐값 DY
DY = node.getResults("DISPLACEMENT", "DY")
'n번 절점의 휨모멘트값 Mz
Mz = node.getResults("STRESS", "Mz")
```

표 10 지정한 절점번호에서의 결과값을 변수로 저장

얻어진 결과를 엑셀파일의 특정 시트에 기록하는 방법은 4.1.2에서 정의한 엑셀시트Object에 행 번호(rownum)와 열 번호(colnum)를 부여하고 기록될 값을 지정하는 것으로 아래와 같다.

```
[처짐] 시트에 앞으로 표시될 값의 제목을 먼저 기록하도록 한다.
WorkSheet1.Cells(1,1).Value = "하중케이스번호"
WorkSheet1.Cells(1,2).Value = "7번절점의 처짐"
WorkSheet1.Cells(1,3).Value = "17번절점의 처짐"
WorkSheet1.Cells(1,4).Value = "27번절점의 처짐"

[휨모멘트] 시트에 앞으로 표시될 값의 제목을 먼저 기록하도록 한다.
WorkSheet2.Cells(1,1).Value = "하중케이스번호"
WorkSheet2.Cells(1,2).Value = "7번절점의 모멘트"
WorkSheet2.Cells(1,3).Value = "17번절점의 모멘트"
WorkSheet2.Cells(1,4).Value = "27번절점의 모멘트"
```

표 11 엑셀시트에 값을 기록하는 예

4.4.3. 하중케이스별 처짐/모멘트 기록

4.4.1~4.4.2까지의 내용을 종합하여 엑셀에 각 위치에 결과를 출력하게 하는 스크립트를 완성하면 아래와 같다.

'[처짐] 시트에 앞으로 표시될 값의 제목을 먼저 기록하도록 한다.

```
WorkSheet1.Cells(1,1).Value = "하중케이스번호"  
WorkSheet1.Cells(1,2).Value = "7번절점의 처짐"  
WorkSheet1.Cells(1,3).Value = "17번절점의 처짐"  
WorkSheet1.Cells(1,4).Value = "27번절점의 처짐"
```

'[휨모멘트] 시트에 앞으로 표시될 값의 제목을 먼저 기록하도록 한다.

```
WorkSheet2.Cells(1,1).Value = "하중케이스번호"  
WorkSheet2.Cells(1,2).Value = "7번절점의 모멘트"  
WorkSheet2.Cells(1,3).Value = "17번절점의 모멘트"  
WorkSheet2.Cells(1,4).Value = "27번절점의 모멘트"
```

For i=1 to 10 step 1

```
'하중케이스 재설정(하중 케이스 번호는 i라는 변수로 표현)  
processcommand("SET LOADCASE LCID=&i& IEIGEN=-1 IHARMS=-1 IRES=1")
```

'[처짐]시트와 [휨모멘트]시트에 하중케이스 번호기록

```
WorkSheet1.Cells(i+1,1).Value = i  
WorkSheet2.Cells(i+1,1).Value = i
```

'targetnode(0),targetnode(1),targetnode(3)으로 정의된 각 경간 중앙절점 번호에 대해 결과 출력

For j=0 to 2 step 1

```
Set node = db.getNodeByNumber(targetnode(j))  
DY = node.getResults("DISPLACEMENT", "DY")  
Mz = node.getResults("STRESS", "Mz")  
WorkSheet1.Cells(i+1,j+2).Value = DY  
WorkSheet2.Cells(i+1,j+2).Value = Mz
```

Next

Next

표 12 하중케이스별로 처짐과 모멘트를 엑셀시트에 기록

4.5. EXCEL 저장

앞 단계에서 각 하중케이스에 대하여 3개의 절점에서의 처짐과 모멘트를 각각 [처짐]시트와 [휨모멘트]시트에 기록을 하였다. 아래과 같이 스크립트를 추가하면, 앞서 지정한 엑셀파일 이름으로 저장된다.

```
'엑셀 기록 내용 저장하기  
WorkBook.SaveAs(ExcelFileName)  
ExcelApp.Quit
```

표 13 하중케이스별로 처짐과 모멘트를 엑셀시트에 기록

4.6. 주요변수 초기화

이상으로 필요한 작업은 모두 완료되었지만, 처음에 부여하였던 주요 변수들을 초기화시켜주지 않으면 메모리에 상주하여 다음 작업시에나 정상적인 엑셀작업에 있어 불필요한 오류메시지를 유발할 수 있으므로 아래와 같이 변수나 오브젝트들을 초기화 시킴으로써 작업이 완료된다.

```
'주요변수 초기화  
Set db= Nothing  
Set node = Nothing  
Set ExcelApp = Nothing
```

표 14 주요변수를 초기화하는 과정

4.7. 작성된 VB스크립트 저장

이상으로 스크립트의 작성이 마무리 되었다.

작성된 내용은 ResultToExcel.vbs 라는 파일로 저장한다. 확장자명은 vbs 로 한다.

5. VBScript 실행

작성한 스크립트를 실행시키는 방법은 아래 그림과 같이 하여 파일을 선택하면 된다.

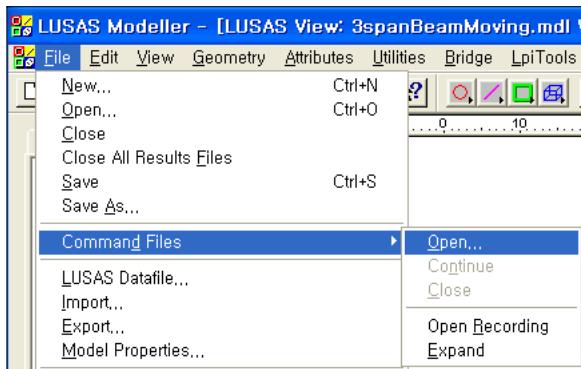


그림 2 스크립트실행 툴바

그림 1 스크립트 실행을 위한 메뉴

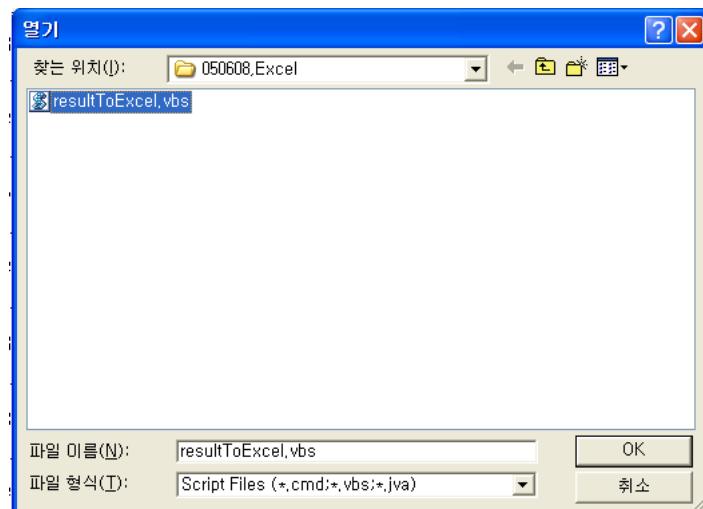


그림 3 실행할 스크립트 파일 선택

6. 처리결과 확인

엑셀파일을 저장하도록 지정한 디렉토리에서 result.xls파일을 확인하면 아래와 같이 결과가 정렬되어 있는 것을 확인할 수 있다.

	A	B	C	D	E	F
1	하중케이스번호	7번절점의 처짐	17번절점의 처짐	27번절점의 처짐		
2	1	-6.21E-06	2.4E-06	-5.39998E-07		
3	2	-0.032218265	0.012901567	-0.002902848		
4	3	-0.045962154	0.023695464	-0.00533147		
5	4	0.00082204	-0.003749521	0.0008612		
6	5	0.02484506	-0.065402325	0.016111015		
7	6	0.019555569	-0.076146258	0.025497269		
8	7	0.003971577	-0.017365461	0.009955007		
9	8	-0.004870301	0.021645819	-0.036962177		
10	9	-0.003957296	0.017588012	-0.042812735		
11	10	-2.15999E-06	9.59998E-06	-2.484E-05		
12						
13						
14						
15						
16						
17						
18						

그림 4 하중케이스별 처짐 저장상태

	A	B	C	D	E	F
1	하중케이스번호	7번절점의 모멘트	17번절점의 모멘트	27번절점의 모멘트		
2	1	-0.009200006	0.001999997	-0.000799997		
3	2	-54.63217895	10.7513059	-4.300515147		
4	3	-96.61530537	19.74621991	-7.898474717		
5	4	1.135408961	-3.384739047	1.275851992		
6	5	36.80749682	-75.17633314	23.86817003		
7	6	28.9712135	-99.34705444	37.77373206		
8	7	5.883818249	-15.42463052	14.72755123		
9	8	-7.215260929	18.03818257	-68.7665565		
10	9	-5.862660981	14.65667703	-79.40466199		
11	10	-0.00319999	0.007999987	-0.036800023		
12						
13						
14						
15						
16						
17						
18						

그림 5 하중케이스별 모멘트 저장상태