

매개변수 형상모델링을 이용한 OpenFOAM 해석사례

김민우, 한국항공우주연구원



Contents

1

매개변수 형상모델링

2

OpenFOAM 해석자동화

배경) 매개변수 형상모델링

- 필요성
 - 개념설계 단계의 다양한 요구도를 검증하기 위한 다수의 형상 생성 필요
 - 그러나 형상생성작업(= CAD) 을 위한 시간은 부족함
- 실제 형상 변경 사례
 - Airfoil 형상 변경
 - 주익의 Span/Taper/Sweep/Incidence Angle 변경
 - 무게중심 조정을 위한 모터 위치 재배치
 - 프로펠러 지름 변경, 위치
 - 정적마진 확보를 위한 꼬리날개 위치/면적 변경
 - 내부 탑재물 배치 조정에 따른 Fuselage IML/OML 변경
- 형상변경에 소요되는 시간 절감 필요

Rhino 기반 매개변수 형상모델링

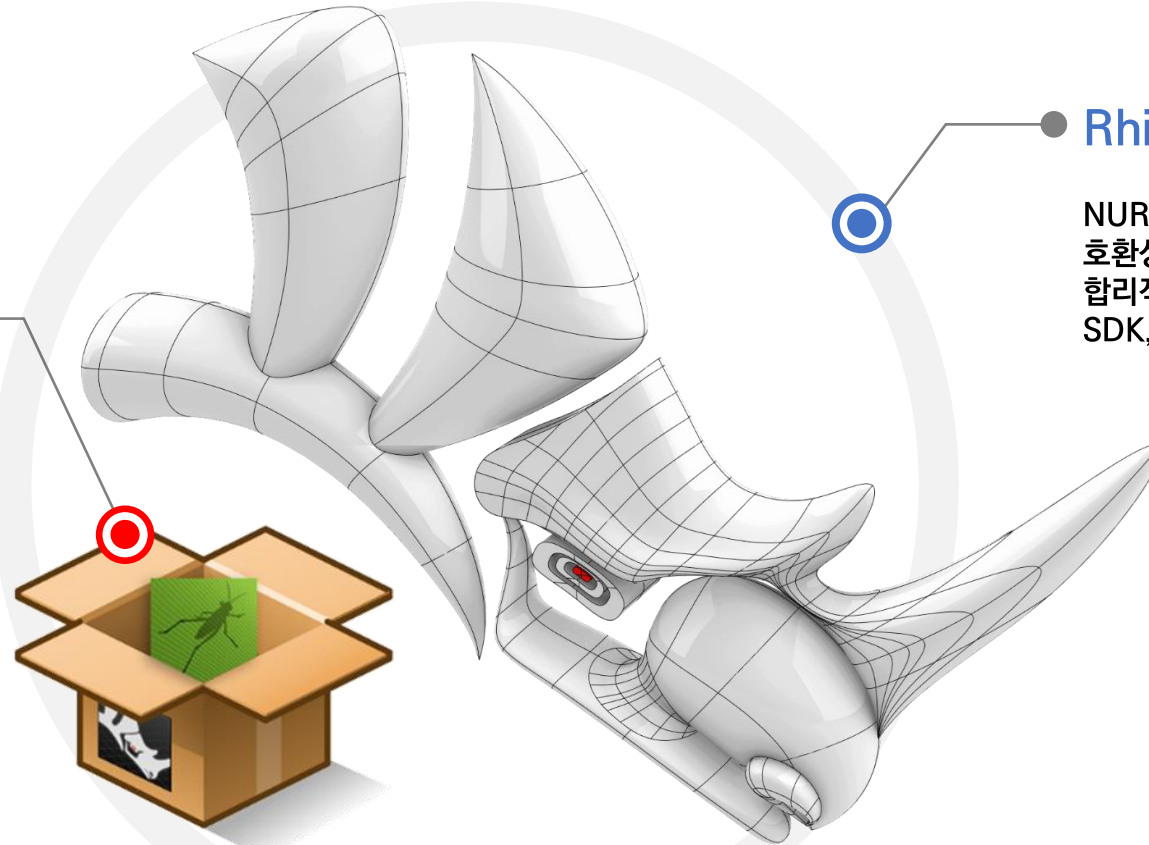
- 개발기간 : 2020.06~2021.02
- 개발 Platform :

Grasshopper

그래픽 인터페이스의 알고리즘 편집기
라이노에 Parameter 모델링기능 통합
프로그래밍 지식 없이 사용가능
복잡한 구속조건 3D모델링 구현

Rhinoceros

NURBS 기반 서피스, 솔리드 모델링
호환성 : STEP, IGES, STL 파일 등
합리적 가격
SDK, Script 지원 자동화 및 툴 개발 가능



- 개발기관 : **+plastic**
digital designers group

매개변수 형상모델링 목표

- NURBS 기반 육공분리합체 무인이동체 모델링
 - 동체, 날개, 랜딩기어, 배송 모듈 외형 모델
 - 동체 내부 FCC, Mission Computer, 항법장비, 배터리, 동력원 배치 Layout
 - 임무장비(배송모듈) 탑재 전/후 형상 모델링 및 간섭확인
 - LRU 배치 및 기체설계결과 기반 무게중심(CG) 계산
 - 외부에서 가져온 동력원, 모터, ESC, 프로펠러 등의 구성품 형상파일 자동 반영
- 매개변수 자동화처리
 - 매개변수간 Input/Output 스크립트 구성
 - 대상 : 동체/날개/로터 및 Pusher/랜딩기어/ LRU 위치
- 대상 :
 - Tandem Wing, Lift + Cruise Type



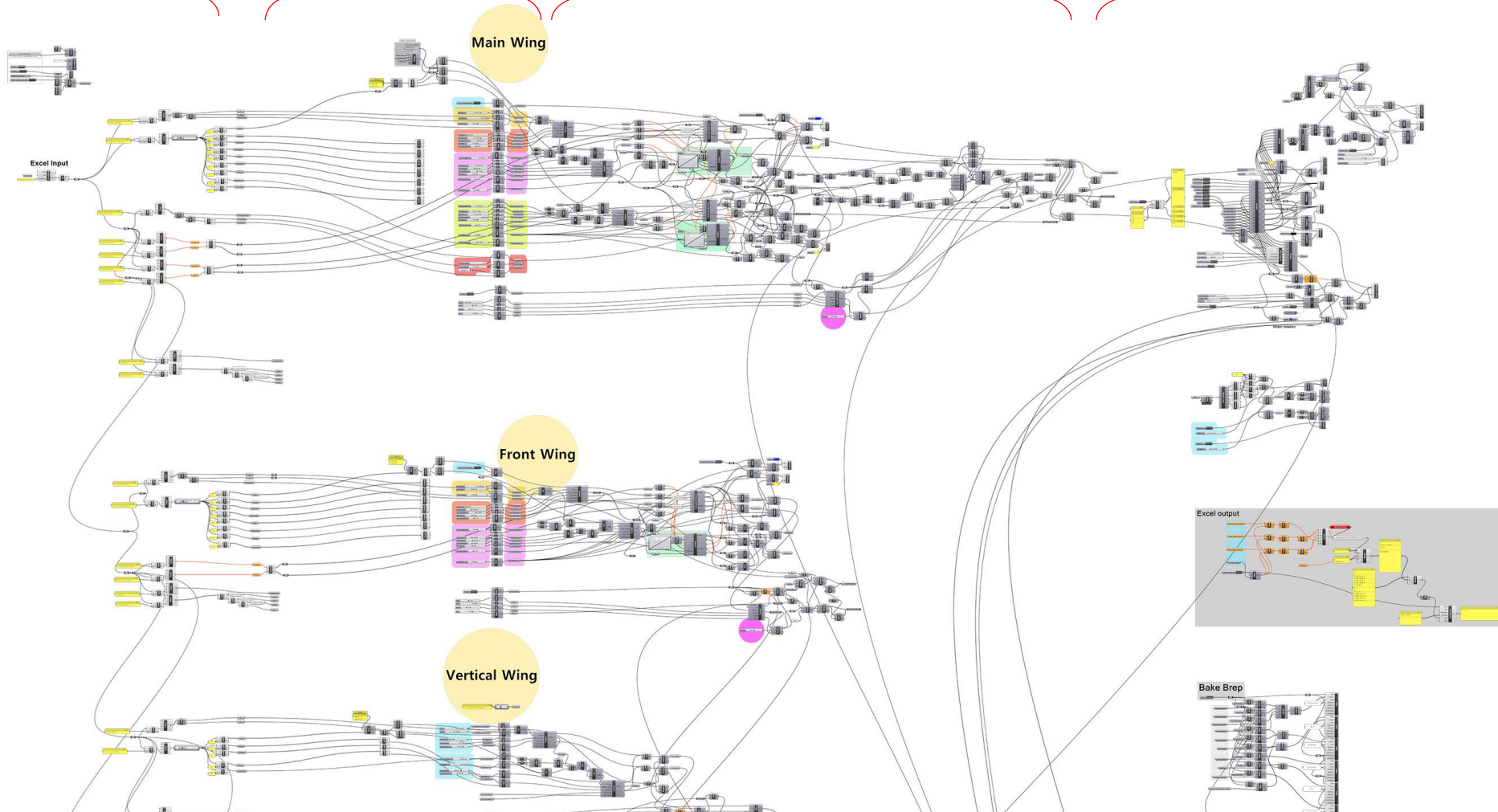
그래스호퍼 모델링 Overview

엑셀 입력단

GH내부입력

모듈별 모델링

디스플레이/ 개체 생성 처리



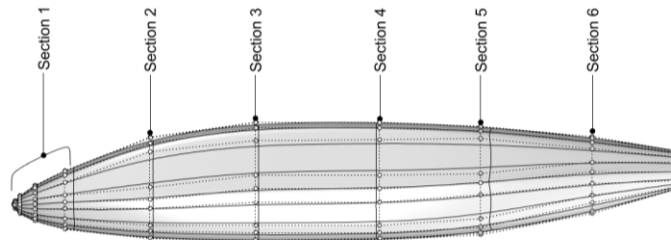
매개변수 형상모델링 입력부분

- 날개 변수
 - Chord, Thickness, Twist Angle,
 - Span Length, Sweepback, Dihedral
 - 전방 날개 트윙트 형상 적용 시 Twist Angle표 사용
 - 조종면
 - 시작, 끝 위치 및 갭(Clearance)
- 수직 날개
 - 날개형상 및 크기
- 동체
 - 동체 단면 형상 Weight 조절
 - 동체 표면 3차원 좌표 추출여부

Front Wing Parameters									
Position	X	776	Z	50	Attack Angle	0			
	Airfoil	Chord	Thickness	Twist angle	Span	Sweepback	Dihedral	Buffer zone	Twist ref. chord
Root	sd7032	348	79	0	0				25%
Tip	sd7032	256	35	0	1120	0.35	0		25%
FWD Control Surface								Use	TRUE
Span Start(%)		Span End(%)		Width(%)		Clearance			
30		80		23		1.5			

Vertical Stabilizer(Center Upper) Parameters						Use	TRUE	Vertical Control Surface		Use	TRUE
Position	X	2400	Z	196				Span Start(%)	Span End(%)	Width(%)	Clearance
	Airfoil	Chord	Thickness	Span	Sweepback			20	80	20	1.5
Root	NACA0012	590	74	0							
Top	NACA0012	304	38	573	35						

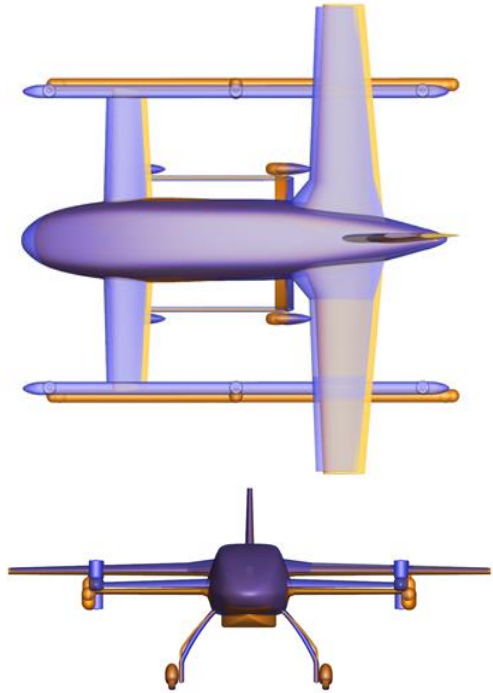
Vertical Stabilizer(Side Lower) Parameters						Use	TRUE	Vertical Control Surface		Use	TRUE
Position	X	2400						Span Start(%)	Span End(%)	Width(%)	Clearance
	Airfoil	Chord	Thickness	Span	Sweepback			20	80	20	1.5
Root	NACA0012	300	40	0							
Top	NACA0012	156	22	350	20						



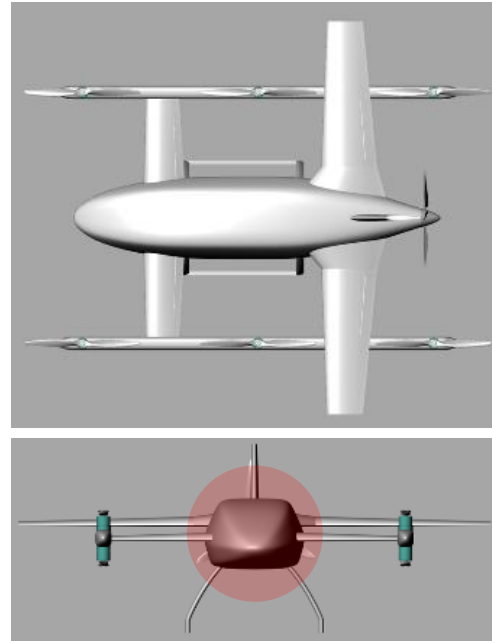
Fuselage shape control					
	Scale X	Scale Y	Scale Z	Move X	Move Z
Section 1	1	1	1	0	0
Section 2		1	1	0	0
Section 3		1	1	0	0
Section 4		1	1	0	0
Section 5		1	1	0	0
Section 6		1	1	0	0
Section 7		1.1	1.1	0	0

매개변수 형상모델링 이용한 설계적용

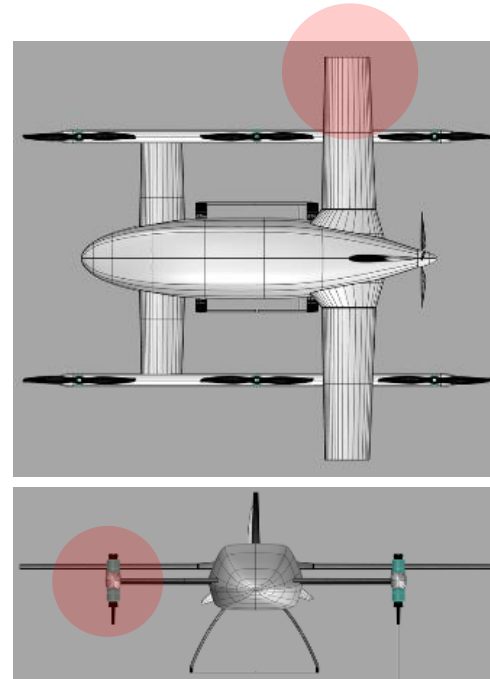
- 형상 변경예시



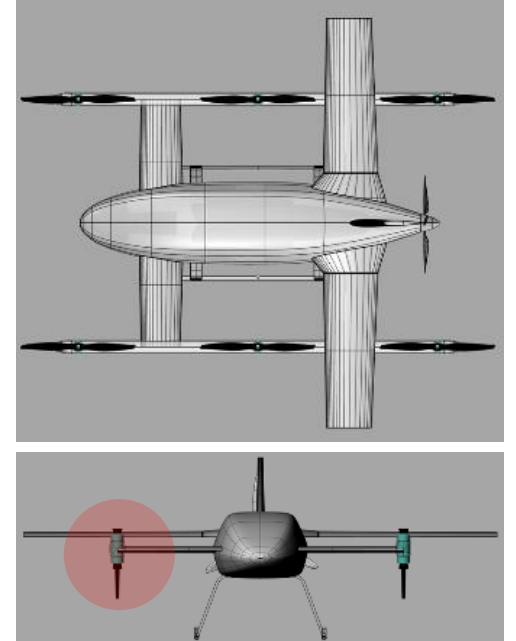
C003/C004



C005



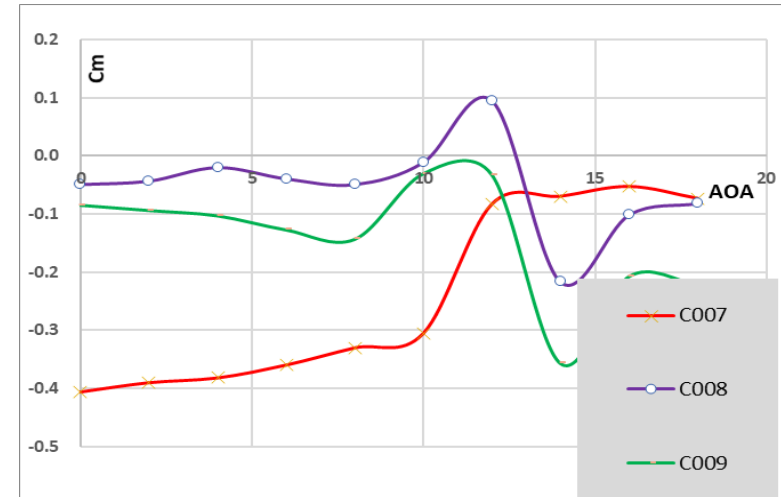
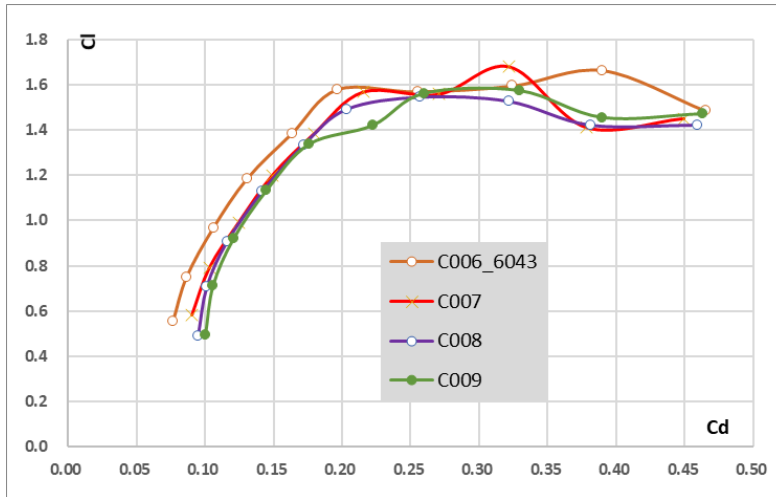
C006



C007

형상에 대한 공력해석

- 도출된 형상에 대해서 공력해석을 수행사례
 - CFD (Ansys FLUENT) analysis by expert in KARI !



- ??????

공력(유동)해석 접근성 확보 필요

- 형상생성 및 변경 관련 접근성은 확보된 것으로 생각됨
- 다만, 공력(유동) 해석 접근성을 추가적으로 확보
 - 격자생성의 어려움 해소
 - 해석조건 (경계조건, 초기조건, 난류모델링 등등) 등의 구성 간소화
 - 후처리 절차 소요시간 절감
- 매개변수 형상모델링 + OpenFOAM 연계
 - 기간 : 2021.07 ~ 2022.10
 - openFOAM v6, snappyHexMesh, BARAM
- 목표
 - 상세설계단계의 정밀한 해석결과 도출은 아님
 - 공력해석 분야의 전문가가 아닌 사람이 공력해석 DB 구축, 매개변수에 따른 경향성 파악을 수행

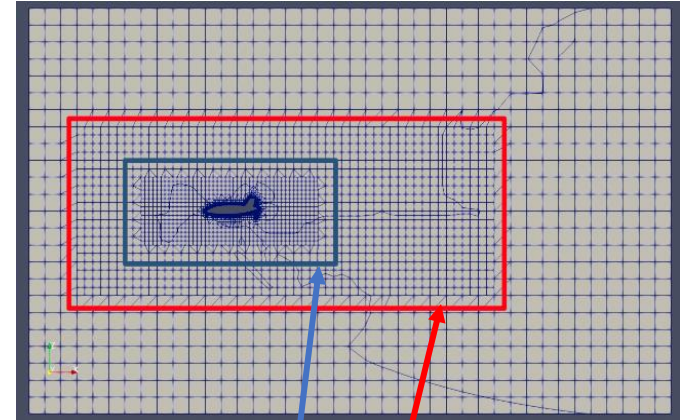
OpenFOAM 유동해석 자동화 절차



- 해석자동화, 후처리를 위한 python script 작성
- snappyHexMesh 설정
 - 입력 CAD File : STL File
 - 매개변수 형상모델링을 통해 각 부분(날개, 동체, Boom 등)의 개별 STL File 이 생성되도록 기능 추가
 - 해당 CAD 파일들을 OpenFOAM 입력파일로 적용
- OpenFOAM 설정
 - 엑셀 입력 변수 : (해석조건 : 밀도, 유속, 받음각), 설정값 (저장주기, CPU Core 등)
 - Propeller 는 Actuator Disk 설정
 - 그외의 설정값은 기본 해석을 통해 적절한 값으로 선정

해석자동화 진행절차

- STL File (ASCII) 들을 호출하고 개별 geometry 로 지정
- OpenFOAM 설정파일 생성
 - BARAM “createCase” 명령어로 case 생성
 - /system/ 이하 *Dict (blockMeshDict, snappyHexMeshDict …) 생성
 - blockMeshDict : 전체 비행체 BoundingBox 생성
 - snappyHexMeshDict 의 geometry 별 세부설정지정
 - 기본적인 크기가 다른 motor / Wing 등의 격자생성 옵션 설정



```
### create snappyHexMeshDict
snappyHexMeshDict = caseDir + '/system/snappyHexMeshDict'
write_header(snappyHexMeshDict, 'class', 'dictionary', 'object', 'snappyHexMeshDict')
f = open(snappyHexMeshDict, 'a')
f.write('castellatedMesh true;\n'
       'snap true;\n'
       'addLayers true;\n')
f.write('geometry\n'
       '{\n'
       '\n')
for i in stl_files:
    f.write('    ' + i + '\n'
           '{\n'
           '        type triSurfaceMesh;\n'
           '        name ' + i.split('.')[0] + ';\n'
           '    }\n')
f.write('    fine\n'
       '{\n'
       '        type searchableBox;\n'
       '        min (' + str(int(box[0] - L)) + ' ' + str(int(box[1] - B)) + ' ' + str(int(box[2] - D)) + ');\n'
       '        max (' + str(int(box[3] + L)) + ' ' + str(int(box[4] + B)) + ' ' + str(int(box[5] + D)) + ');\n'
       '    }\n')
f.write('    medium\n'
       '{\n'
       '        type searchableBox;\n'
       '        min (' + str(int(box[0] - 2*L)) + ' ' + str(int(box[1] - 3*B)) + ' ' + str(int(box[2] - 3*D)) + ');\n'
       '        max (' + str(int(box[3] + 4*L)) + ' ' + str(int(box[4] + 3*B)) + ' ' + str(int(box[5] + 3*D)) + ');\n'
       '    }\n')
f.write('}\n')
```

해석자동화 진행절차

- OpenFOAM 해석 설정
 - 엑셀 입력변수 활용
 - Density, kinematic viscosity, flow speed, AoA, AoS 등
 - “foamDictionary”, “createSettings” 명령어 script set 생성

```
os.system('foamDictionary -entry ' + zone[x] + ' -set "{}" ' + caseDir + '/system/fvOptions')
os.system('foamDictionary -entry ' + zone[x] + '.type -set actuationDiskSource ' + caseDir + '/system/fvOptions')
os.system('foamDictionary -entry ' + zone[x] + '.fields -set "(U)" ' + caseDir + '/system/fvOptions')
os.system('foamDictionary -entry ' + zone[x] + '.selectionMode -set cellZone ' + caseDir + '/system/fvOptions')
os.system('foamDictionary -entry ' + zone[x] + '.cellZone -set ' + zone[x] + ' ' + caseDir + '/system/fvOptions')
os.system('foamDictionary -entry ' + zone[x] + '.Cp -set ' + str(Cp[x]) + ' ' + caseDir + '/system/fvOptions')
os.system('foamDictionary -entry ' + zone[x] + '.Ct -set ' + str(Ct[x]) + ' ' + caseDir + '/system/fvOptions')
tmp_box = bounding_box(caseDir + '/constant/triSurface/' + tmp_file)
UPx = tmp_box[0]*3.0/4000.0 + tmp_box[3]/4000.0
UPy = tmp_box[1]*3.0/4000.0 + tmp_box[4]/4000.0
UPz = tmp_box[2]*3.0/4000.0 + tmp_box[5]/4000.0

magDiskDir = (float(diskDir[x][0])**2 + float(diskDir[x][1])**2 + float(diskDir[x][2])**2)**(0.5)
UPx = UPx - float(diskDir[x][0])/magDiskDir*0.01
UPy = UPy - float(diskDir[x][1])/magDiskDir*0.01
UPz = UPz - float(diskDir[x][2])/magDiskDir*0.01
upstreamPoint = '(' + str(UPx) + ' ' + str(UPy) + ' ' + str(UPz) + ')'
dirVector = '(' + str(-float(diskDir[x][0])/magDiskDir) + ' ' + str(-float(diskDir[x][1])/magDiskDir) + ' ' + str(-float(diskDir[x][2])/magDiskDir) + ')'
os.system('foamDictionary -entry ' + zone[x] + '.diskDir -set "' + dirVector + '" ' + caseDir + '/system/fvOptions')
os.system('foamDictionary -entry ' + zone[x] + '.diskArea -set ' + str(diskArea[x]) + ' ' + caseDir + '/system/fvOptions')
os.system('foamDictionary -entry ' + zone[x] + '.upstreamPoint -set "' + upstreamPoint + '" ' + caseDir + '/system/fvOptions')

flowVel = 'uniform (' + str(Ux) + ' ' + str(Uy) + ' ' + str(Uz) + ')'
os.system('foamDictionary -entry region0.freestream -set "{}" ' + caseDir + '/system/settings/boundaryConditions')
os.system('foamDictionary -entry region0.freestream.type -set "velocityInlet" ' + caseDir + '/system/settings/boundaryConditions')
os.system('foamDictionary -entry region0.freestream.U -set "' + flowVel + '" ' + caseDir + '/system/settings/boundaryConditions')
os.system('foamDictionary -entry region0.freestream.turbIntensity -set "uniform 0.01" ' + caseDir + '/system/settings/boundaryConditions')
os.system('foamDictionary -entry region0.freestream.viscosityRatio -set "uniform 10" ' + caseDir + '/system/settings/boundaryConditions')
os.system('foamDictionary -entry region0.flow.U -set "' + flowVel + '" ' + caseDir + '/system/settings/initialConditions')
os.system('sed -i "s/ABLWall;/adiabaticWall;/g" ' + caseDir + '/system/settings/boundaryConditions')
os.system('foamDictionary -entry convergenceCriteria.pressure -set "' + converge + '" ' + caseDir + '/system/settings/numericConditions')
os.system('foamDictionary -entry convergenceCriteria.momentum -set "' + converge + '" ' + caseDir + '/system/settings/numericConditions')
os.system('foamDictionary -entry convergenceCriteria.turbulence -set "' + converge + '" ' + caseDir + '/system/settings/numericConditions')
os.system('foamDictionary -entry improveStability -set "true" ' + caseDir + '/system/settings/numericConditions')
os.system('setConditions -case ' + caseDir + ' > ' + caseDir + '/Log.setConditions 2>&1')
```

해석자동화 진행절차

- OpenFOAM 후처리 자동화
 - 해석 조건(AoA, AoS, 유동속도) 에 대한 별도의 caseDir 별로 해석결과 도출
 - '/postprocessing/' 이하 파일 Parsing

```
for x in betalList:
    CmList = []
    CdList = []
    ClList = []
    AOA = []
    for y in alphaList:
        AOA.append(float(y))

        dataDir = head+'_AoA_'+y+'_AoS_'+x+'/data-Umag_'+Umag+'_AoA_'+y+'_AoS_'+x

        with open(dataDir+'forceCoeffs/0/forceCoeffs.dat', 'r') as f:
            tmp = f.readlines()[1:]
            CmList.append(tmp.split()[1])
            CdList.append(tmp.split()[2])
            ClList.append(tmp.split()[3])

        partList = []
        for i in os.listdir(dataDir):
            if os.path.isdir(os.path.join(dataDir, i)):
                if i == 'forceCoeffs':
                    continue
                else:
                    partList.append(i)

        excelFile = head + '_AoA_' + y + '_AoS_' + x + '/Forces.xlsx'
        wb = xl.Workbook()
        ws = wb.active

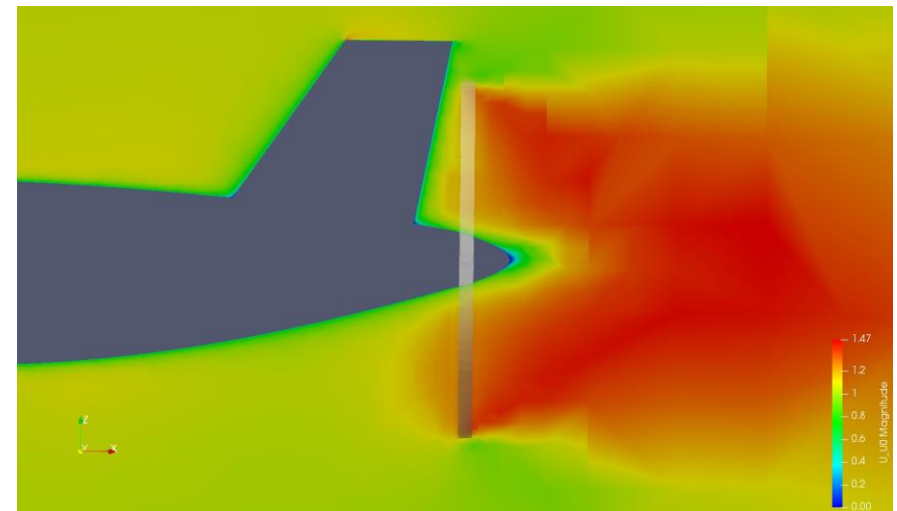
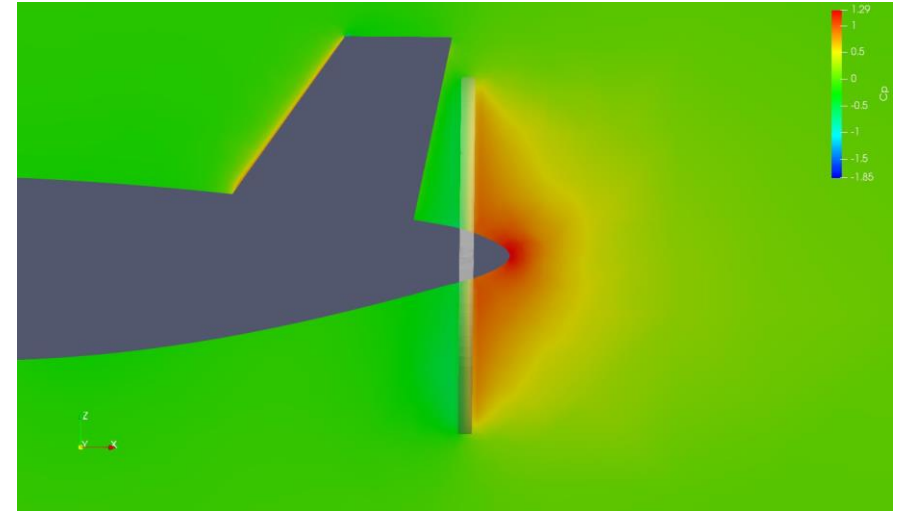
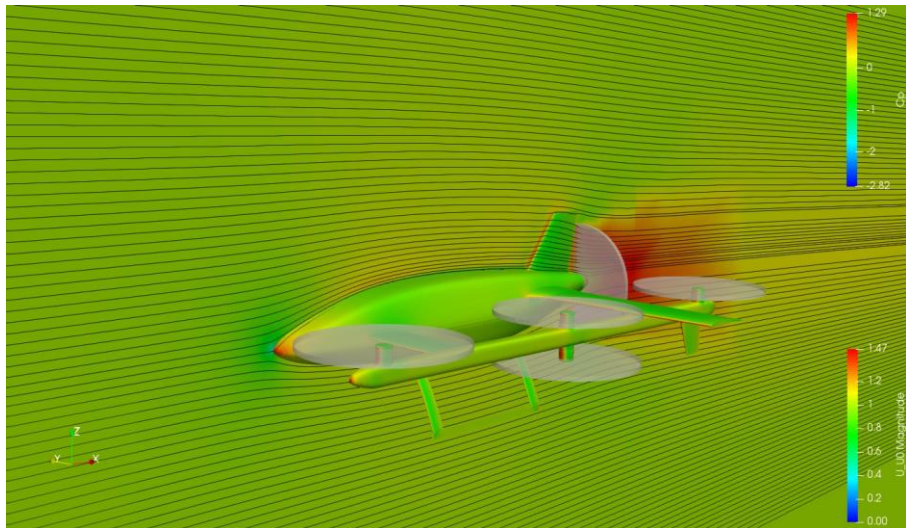
        ws['B1'] = 'CFX [N]'
        ws['C1'] = 'CFY [N]'
        ws['D1'] = 'CFZ [N]'
        ws['E1'] = 'CMX [N-m]'
        ws['F1'] = 'CMY [N-m]'
        ws['G1'] = 'CMZ [N-m]'

        ind = 1
        for i in partList:
            ind = ind + 1
            ws['A'+str(ind)] = i
            with open(dataDir+'/' + i + '/0/forces.dat', 'r') as f:
                tmp1 = f.readlines()[1:]
                ws['B'+str(ind)] = float(tmp1.replace('(', '').replace(')', '').split()[1])+float(tmp1.replace('(', '').replace(')', '').split()[4])
                ws['C'+str(ind)] = float(tmp1.replace('(', '').replace(')', '').split()[2])+float(tmp1.replace('(', '').replace(')', '').split()[5])
                ws['D'+str(ind)] = float(tmp1.replace('(', '').replace(')', '').split()[3])+float(tmp1.replace('(', '').replace(')', '').split()[6])
                ws['E'+str(ind)] = float(tmp1.replace('(', '').replace(')', '').split()[10])+float(tmp1.replace('(', '').replace(')', '').split()[13])
                ws['F'+str(ind)] = float(tmp1.replace('(', '').replace(')', '').split()[11])+float(tmp1.replace('(', '').replace(')', '').split()[14])
                ws['G'+str(ind)] = float(tmp1.replace('(', '').replace(')', '').split()[12])+float(tmp1.replace('(', '').replace(')', '').split()[15])
```

- 해석결과 별로 엑셀 파일로 저장
- 해석변수 별 Graph (Lift/Drag/PM)

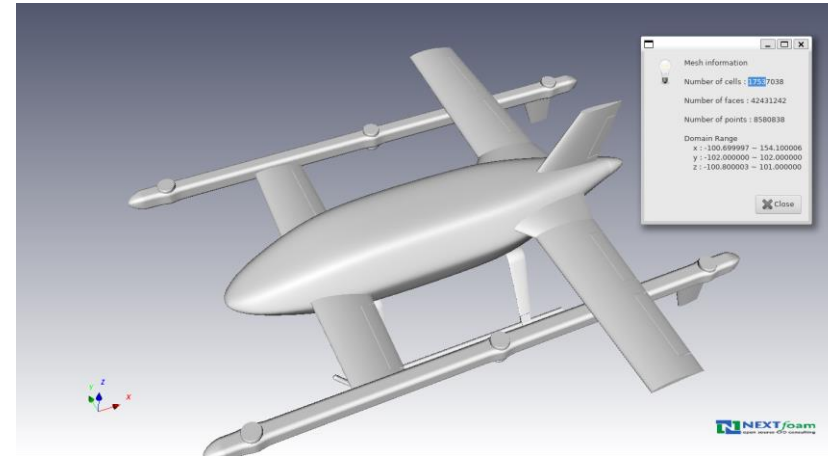
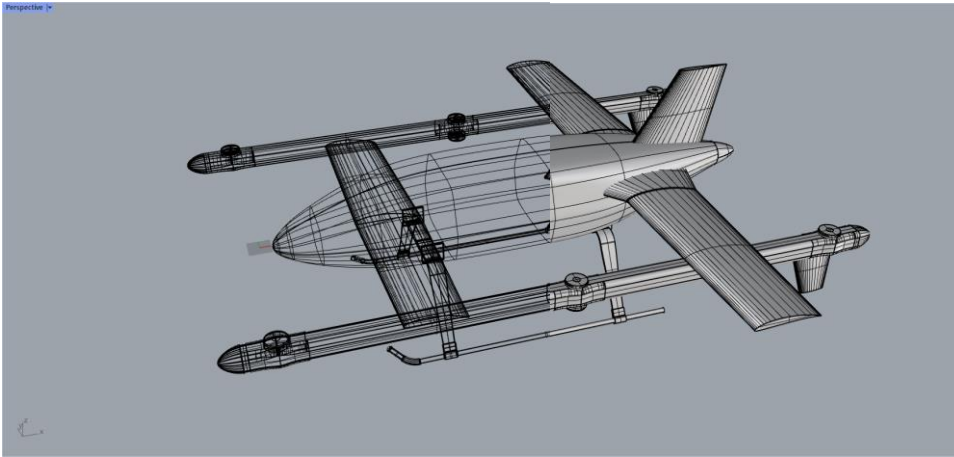
해석사례 (1)

- 기본 매개변수 형상모델링 해석 예시
 - Freestream (5 0 0)
 - 좌 : 동체 표면 압력/속도 Contour
 - 우 : Pusher(Actuator Disk) 압력과 속도



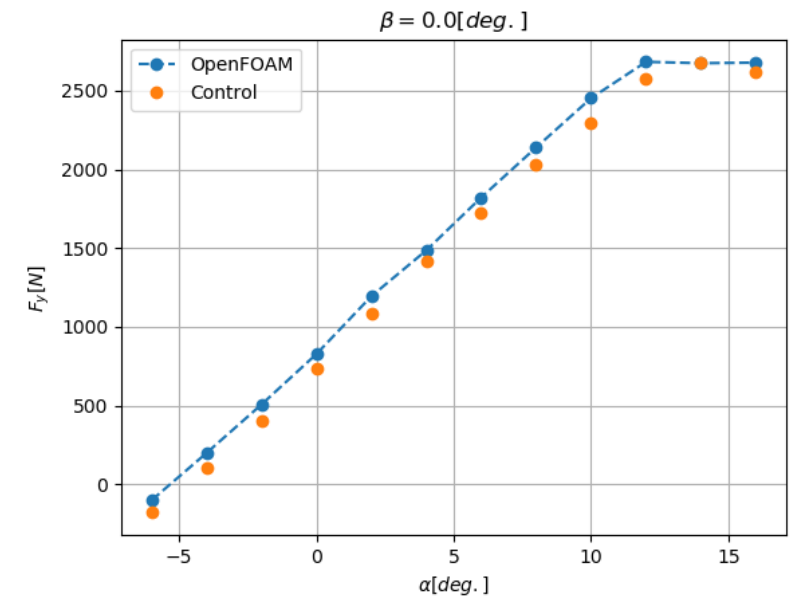
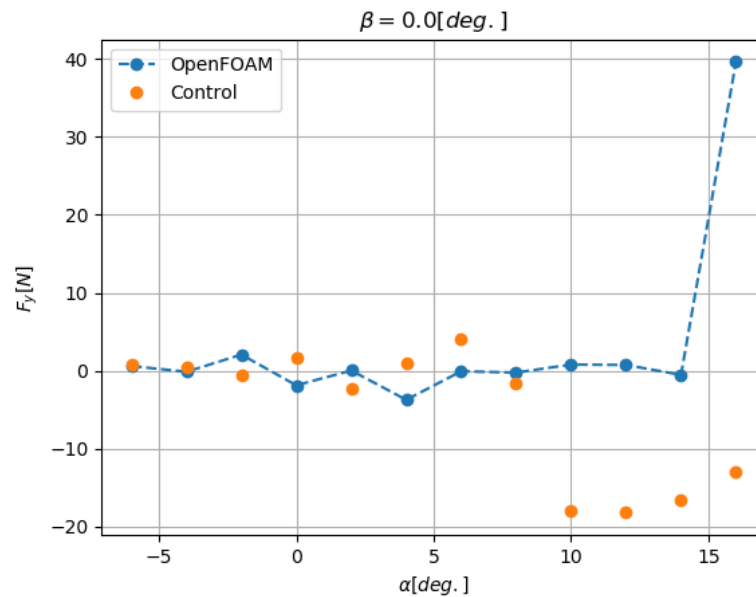
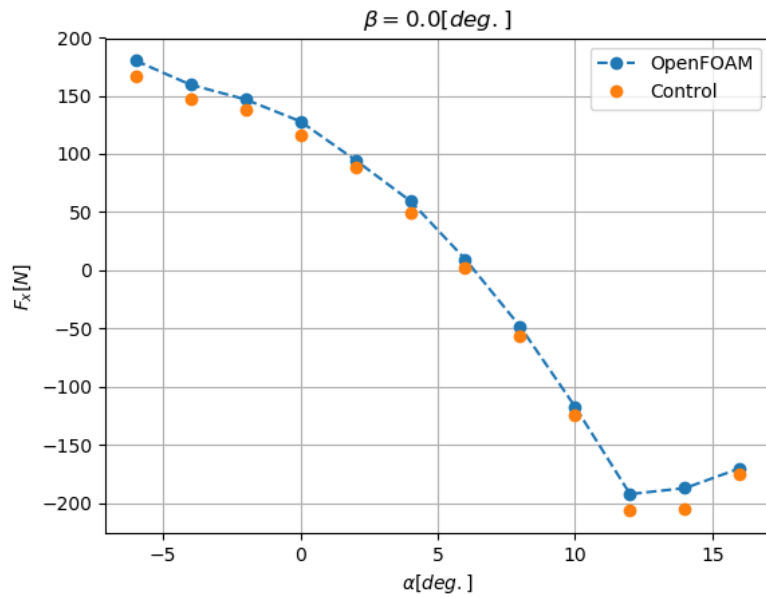
해석사례 (2)

- 유동해석 자동화 결과와 Fluent 해석결과 비교
 - CFD 해석과 비교를 위해 별도의 CAD File SET 제공
- 좌 : STL File , 우 : Fluent Mesh (17M)



해석사례 (2)

- 해석결과 비교
 - AoS 0
 - Lift/Drag (F_x , F_z) 기존 해석결과와 비슷한 경향성을 확인



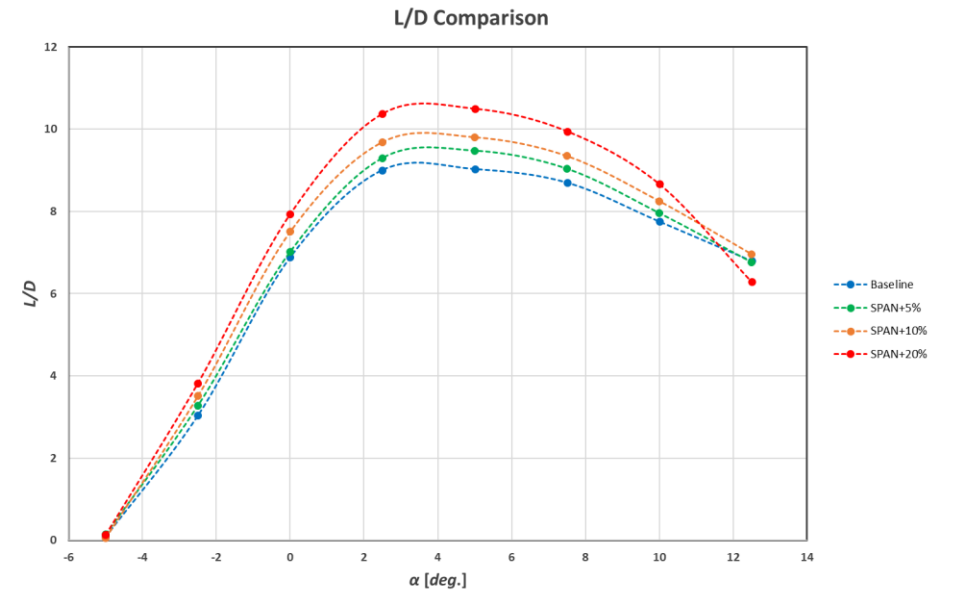
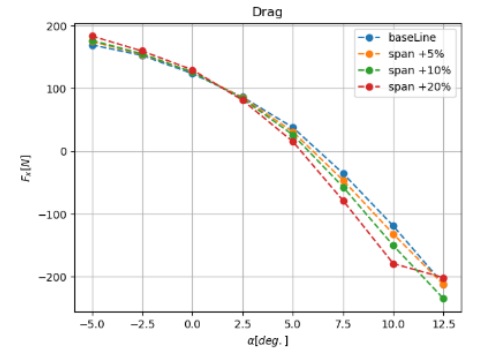
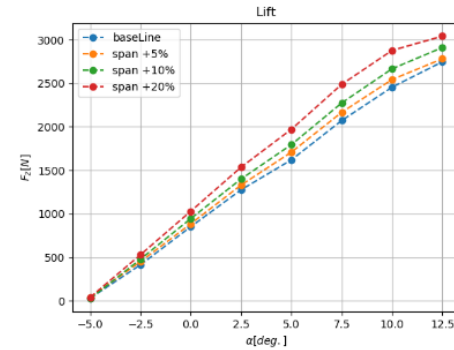
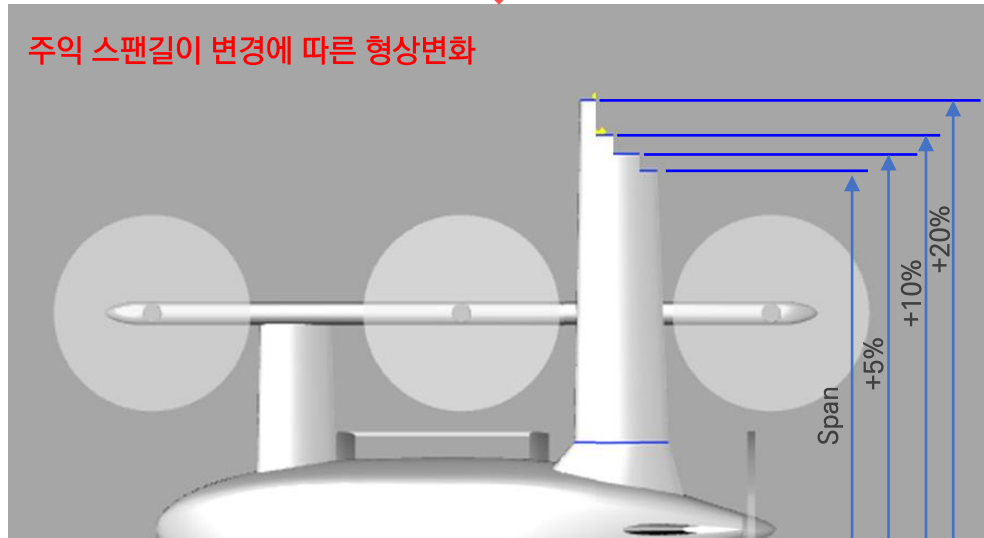
해석사례 (3)

- 매개변수 모델링 + 해석자동화 연계
 - 간단한 Example : 주익길이 변경에 따른 해석자동화 적용

AFT Wing Parameters										
Position	X	2109	Z	164	Attack Angle	0				
	Airfoil	Chord	Thickness	Twist angle	Span	Sweepback	Dihedral	Buffer zone	Twist ref. chord	
Root	sg6043	991	99	0	0				25%	
Mid	sg6043	486	58	0	453	29	0	5	25%	
Tip	sg6043	399	48	0	1400	0.3	0		25%	
Buffer Zone Shape		Chord_210811			AFT Control Surface		Use	FALSE		
	W2-3	991	149		Span Start(%)	Span End(%)	Width(%)	Clearance		
Num Sections	10	486	75		10	90	30	1.5		
Blend Bulge	0.5	311	30							
Continuity	Tangent									



주익 스패ん길이 변경에 따른 형상변화



결언

- 사용자의 개입(intervention) 을 최소화할 수 있는 자동화 절차 개발
 - 매개변수를 활용하여 형상을 모델링하고 해당 형상을 OpenFOAM 해석에 적용
 - Excel 등의 사용자 친화적 Input/Output 인터페이스 활용
 - 자동화를 위한 해석설정에 따라 해석 정밀도의 수준이 제한적임
- 비전공자 혹은 비전문가도 활용할 수 있는 설계프로세스의 모듈로 적용
 - 개념설계 혹은 초기설계 단계에서의 공력 DB 구축 간소화
 - 도출된 결과는 다른 해석모듈의 입력 데이터로 적용
- 기타 (개선목표)
 - Python script 내 하드코딩 부분 refactoring 진행중
 - 다른 해석모듈과의 연계를 위한 인터페이스 보완

감사합니다