Università di Genova | DICCA DIPARTIMENTO DI INGEGNERIA CIVILE, CHIMICA E AMBIENTALE

18 OpenFOAM Workshop

PRELIMINARY STUDY FOR A COMPUTATIONAL FLUID DYNAMICS ANALYSIS ON THE INFLUENCE OF VARIABLE CANT ANGLE WINGLETS ON TOTAL DRAG REDUCTION

Giulia Innocenti[1], Joel Enrique Guerrero Rivas[2], Eric Segalerba[1]

(1) Dipartimento di Ingegneria Civile, Chimica e Ambientale, Università di Genova (Genova)

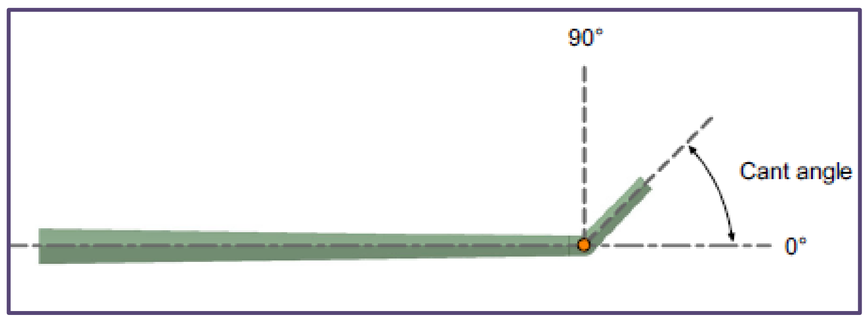
(2) Wolf Dynamics (Genova)

INTRODUCTION

The development of wingtip devices has been motivated by the need to reduce lift-induced drag in aircraft. Traditional winglets, fixed devices attached at the tips of the wings, have limitations in terms of lift-induced drag reduction and their inability to adapt to changing flight conditions. To address this issue, this study proposes the use of **variable cant angle** winglets that can be adjusted to **optimize drag reduction** under different flight conditions.

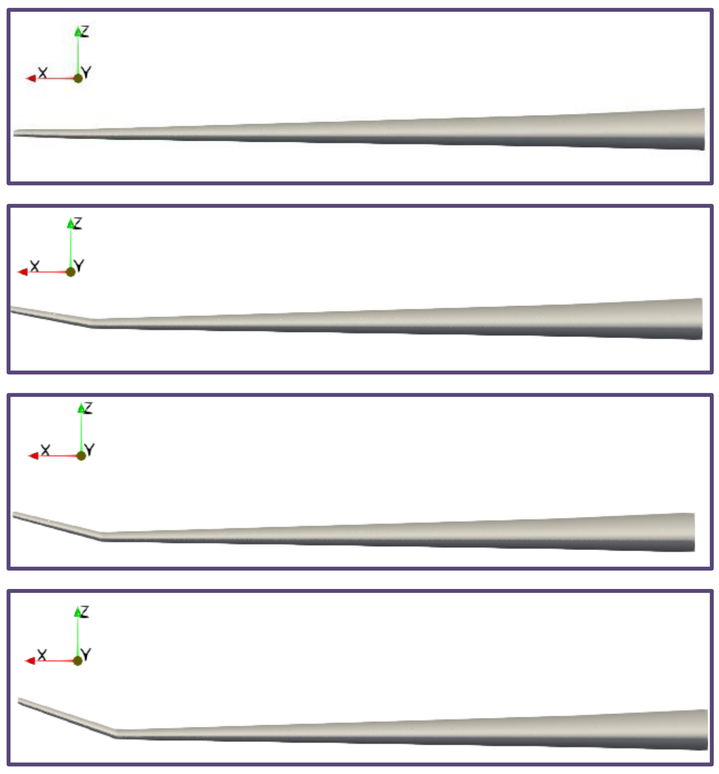
MORPHING APPROACH

The study employs computational fluid dynamics software, specifically OpenFOAM, to investigate the effects of varying the winglet cant angle modifying the wing geometry. An operator is used as a source to define the wing geometry. The solver is transient, allowing for the examination of **multiple winglet positions**.

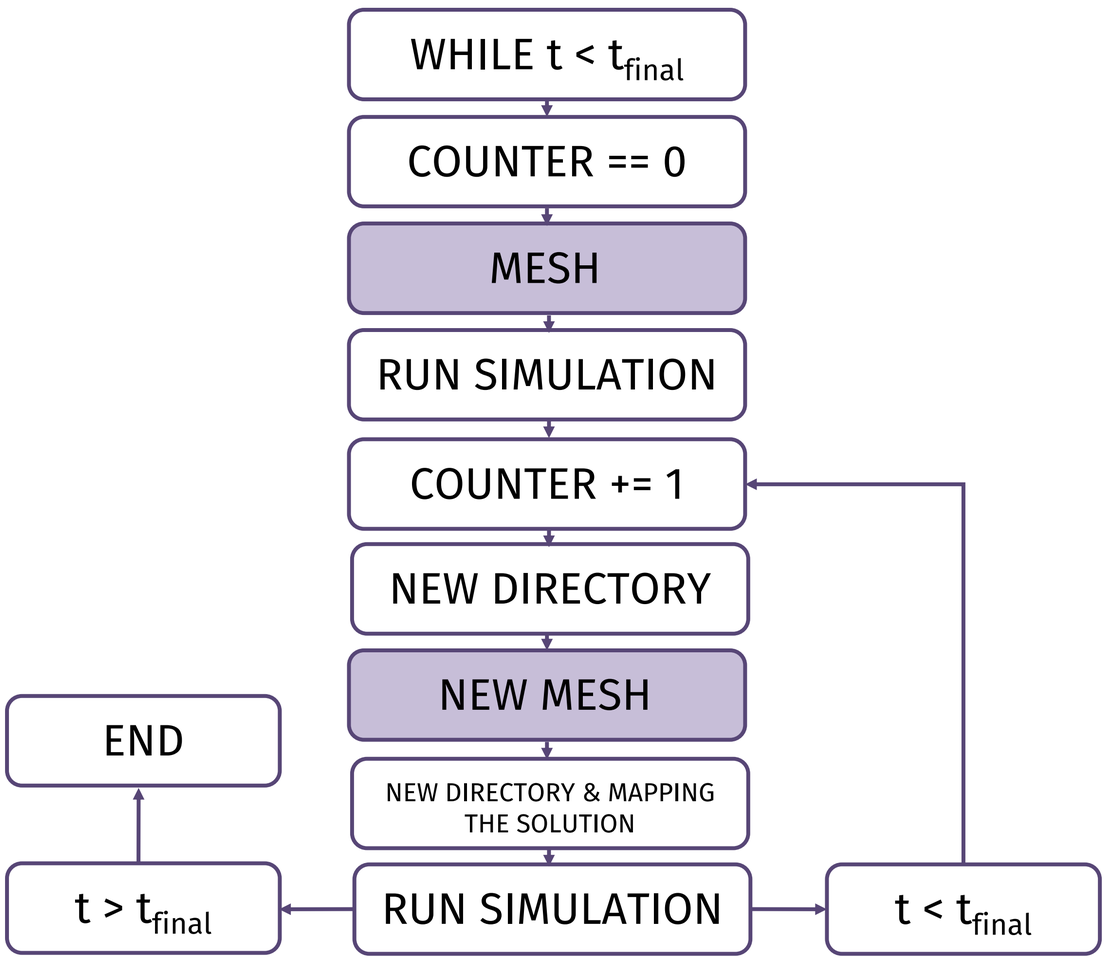


$$f_1(x,t) = \sin(\omega t) \cdot (x - x0)$$

$$f_2(x,t) = \sin(\omega t) \cdot \frac{(x - x0)}{1 + e^{-k(x-x0)}}$$



(RE)MESHING APPROACH




```
while t < tFinal:
    if counter == 0:
        print(f"----------> Remeshing iteration number: {counter}")
        #CREATE THE MESH - path = f"flappyBird_{counter}"
        Allmesh = f"./flappyBird_{counter}/Allmesh"
        os.system(Allmesh)
        os.system(f"sed -i 's/minX/left/'   {path}/constant/polyMesh/boundary")
        os.system(f"sed -i 's/maxX/right/'  {path}/constant/polyMesh/boundary")
        os.system(f"sed -i 's/minY/outlet/' {path}/constant/polyMesh/boundary")
        os.system(f"sed -i 's/maxY/inlet/'  {path}/constant/polyMesh/boundary")
        os.system(f"sed -i 's/minZ/bottom/' {path}/constant/polyMesh/boundary")
        os.system(f"sed -i 's/maxZ/top/'    {path}/constant/polyMesh/boundary")

        #RUN THE SIMULATION
        Allsolver = f"./{path}/Allsolver"
        os.system(Allsolver)

        #UPDATE THE COUNTER
        counter +=1
    else:
        print(f"------> Remeshing iteration number: {counter}")
        # FROM THE PREVIOUS DIRECTORY, COPY IN A NEW DIRECTORY -NEW NAME-
        newDirCmd = f"cp -r flappyBird_{counter-1} flappyBird_{counter}"
        os.system(newDirCmd)

        # EXTRACT THE GEOMETRY
        geoExtract = f"surfaceMeshTriangulate -case flappyBird_{counter} -patches '(wing)' wing_{counter}.stl > flappyBird_{counter}/log.surfaceMeshTriangulate"
        os.system(geoExtract)
        geoMove    = f"mv flappyBird_{counter}/wing_{counter}.stl flappyBird_{counter}/constant/triSurface/"
        os.system(geoMove)
        # CHANGE THE NAME OF snappyHexMeshDict AND surfaceFeaturesDict
        modSnappyHexMeshDict = f"sed -i 's/wing_{counter-1}/wing_{counter}/' flappyBird_{counter}/system/snappyHexMeshDict"
        modSurfaceDict       = f"sed -i 's/wing_{counter-1}/wing_{counter}/' flappyBird_{counter}/system/surfaceFeaturesDict"
        os.system(modSnappyHexMeshDict)
        os.system(modSurfaceDict)

        # CREATE A NEW MESH
        '''
        Allmesh = f"./flappyBird_{counter}/Allmesh"
        os.system(Allmesh)
        os.system(f"sed -i 's/minX/left/'   flappyBird_{counter}/constant/polyMesh/boundary")
        os.system(f"sed -i 's/maxX/right/'  flappyBird_{counter}/constant/polyMesh/boundary")
        os.system(f"sed -i 's/minY/outlet/' flappyBird_{counter}/constant/polyMesh/boundary")
        os.system(f"sed -i 's/maxY/inlet/'  flappyBird_{counter}/constant/polyMesh/boundary")
        os.system(f"sed -i 's/minZ/bottom/' flappyBird_{counter}/constant/polyMesh/boundary")
        os.system(f"sed -i 's/maxZ/top/'    flappyBird_{counter}/constant/polyMesh/boundary")
        '''

        # CREATE A NEW DIRECTORY SPECIFYING THE PREVIOUS MESH TIMESTEP
        os.system(f"foamCleanCase -case ./flappyBird_{counter}")
        prevTimeStep = f"foamListTimes -case ./flappyBird_{counter-1} | tail -n 1"
        newDirName   = subprocess.check_output(prevTimeStep, shell=True).decode('utf-8')
        newDirName.join(c for c in newDirName if c.isdigit() or '.')
        os.system(f"mkdir flappyBird_{counter}/{newDirName}")

        # MAPPING THE SOLUTION
        mapping = f"cd flappyBird_{counter} | mapFields ../flappyBird_{counter-1} -sourceTime {prevTimeStep} -mapMethod cellPointInterpolate -noFunctionObjects -consistent"
        os.system(mapping)

        # CHECK THE endTime FROM THE PREVIOUS controlDict
        oldEndTime = f"foamDictionary -entry endTime -value flappyBird_{counter-1}/system/controlDict"
        oldEndTime_v = subprocess.check_output(oldEndTime, shell=True)

        # MODIFY THE controlDict IN THE NEW DIRECTORY
        newEndTime_v = round(float(oldEndTime_v), 6) + interval
        newEndTime = f"foamDictionary -entry endTime -set {newEndTime_v} flappyBird_{counter}/system/controlDict"
        os.system(newEndTime)
        t = newEndTime_v

        # RUN THE SIMULATION
        Allsolver = f"./flappyBird_{counter}/Allsolver"
        os.system(Allsolver)

        # UPDATE THE COUNTER
        counter +=1
```

Università di Genova

Giulia Innocenti

giulia.innocenti@edu.unige.it

UniGe DICCA