Test Class

```
// parasite Drag

LSAerodynamicsManager.CalcCdvsAlpha theCDWingArrayCalculator = theLSAnalysis
.new CalcCdvsAlpha();

LSAerodynamicsManager.CalcCDAtAlpha theCDWingCalculator = theLSAnalysis
.new CalcCDAtAlpha();

parasiteCDWingCleanArray = new double [nValueAlpha];

Double [] cDWingTemp =  theCDWingArrayCalculator.calculateCDParasiteFromAirfoil(
                        alphaMinWing, alphaMaxWing, nValueAlpha);
for (int i=0; i< parasiteCDWingCleanArray.length; i++){
parasiteCDWingCleanArray[i] =(double)cDWingTemp[i];
}

// Induced Drag

inducedCDWingArray = theCDWingArrayCalculator.calculateCDInduced(
                     alphaMinWing, alphaMaxWing, nValueAlpha);

// Total drag

cDWingCleanArray = new double [nValueAlpha];
for (int i = 0; i<nValueAlpha; i++){
cDWingCleanArray[i] = parasiteCDWingCleanArray[i] +
                      inducedCDWingArray[i];
}
```

LSAerodynamicManager

CalcCDvsAlpha

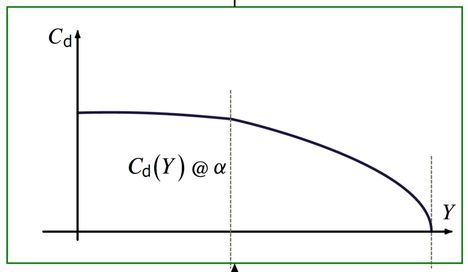
This class calculates the drag coefficient for an array of alpha values.

calculateCDParasiteFromAirfoil

This method calculates the distribution of lift coefficient using Nasa Blackwell method and, starting from local, calculates the local cd calling a method from an airfoil related class.

calculateCDInduced

This method calculates the induced drag coefficient using the output of other method.



CalcCLvsAlpha

CL

CalcCDAtAlpha

This class calculates the drag coefficient of a winh having angle of attack as input.
The method of this class calls an object of CalcCDDistribution in order to evaluate the distribution of drag coefficient along the semspan.

AlphaEffective

calculateAlphaEffective

get AlphaInduced

α_i

NasaBlackwell

MyAerodynamics

calculateCdatClLinear

calculateCdatAlpha

$$CD = CD_{min} + (CL - CL_{CD_{min}})^2 + k$$