Parte II

ANEXOS

ANEXOS

A. Código de OpenFOAM

a. blockMeshDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

//******declaracion Grading*********

#include        "include/GradingConditions"
cornerCalculated_Grading #calc "1.000000000001/$BL_Corner_Grading";

//******declaracion variables geometricas principales*********

#include        "include/GeometricConditions"

Pi 3.141592654;
DgToRad #calc "$Pi/180";

//******declaración de distancias dependientes******************
B #calc "$lon-$R-$C*cos($alpha*$DgToRad)";
Fd #calc "10*$L";
Bd #calc "20*$L";
Ud #calc "8*$L";

//COTAS HORIZONTALES**************************
x13 #calc "$CapL*(1-sin($alpha*$DgToRad))";
Ax #calc "$Fd-$CapL";
Bx #calc "$Fd+$R-($R+$CapL)*0.707107";
Cx #calc "$Fd+$R-$R*0.707107";
Dx #calc "$Fd+$R";
Ex #calc "$Dx+$B";
EEx #calc "$Ex+$x13";
Fx #calc "$Fd+$lon";
Gx #calc "$Fx+$CapL";
Hx #calc "$Fx+$Bd";


//COTAS VERTICALES*****************************
y20 #calc "$CapL*(1-sin($alpha*$DgToRad))";
Ay #calc "$Dd+$R-($R+$CapL)*0.707107";
By #calc "$Dd+$R";
Cy #calc "$Dd+$R+$D";
Dy #calc "$Cy+($R+$CapL)*0.707107";
Fy #calc "$Dd-$CapL";
Gy #calc "$Dd+$L";
Hy #calc "$Cy+$R*0.707107";
Iy #calc "$Gy-$C*tan($alpha*$DgToRad)";
IIy #calc "$Iy+$y20";
Jy #calc "$By-$R*0.707107";
```

```
Ky #calc "$Iy+$CapL";
Ly #calc "$Dd+$L+$CapL";
Ey #calc "$Gy+$A";
My #calc "$Gy+$Ud";

convertToMeters 0.001;

vertices
(       //0 1 2 3 4 5 6 7
        ($Ax $Cy $nzp)
        ($Fd $Cy $nzp)
        ($Cx $Hy $nzp)
        ($Bx $Dy $nzp)
        ($Ax $Cy $zp)
        ($Fd $Cy $zp)
        ($Cx $Hy $zp)
        ($Bx $Dy $zp)
        //8 9 10 11
        ($Dx $Ly $nzp)
        ($Dx $Gy $nzp)
        ($Dx $Ly $zp)
        ($Dx $Gy $zp)
        //11 12 13 14
        ($Ex $Gy $nzp)
        ($Ex $Ly $nzp)
        ($Ex $Gy $zp)
        ($Ex $Ly $zp)
        //16 17 18 19
        ($Fx $Iy $nzp)
        ($Fx $Ky $nzp)
        ($Fx $Iy $zp)
        ($Fx $Ky $zp)
        //20 21 22 23
        ($Gx $IIy $nzp)
        ($Gx $Ky $nzp)
        ($Gx $IIy $zp)
        ($Gx $Ky $zp)
        //24 25 26 27
        ($Hx $IIy $nzp)
        ($Hx $Ky $nzp)
        ($Hx $IIy $zp)
        ($Hx $Ky $zp)

//*************28-39 plano nzp
        (0 $Cy $nzp)
        (0 $Dy $nzp)
        (0 $Ey $nzp)
        ($Bx $Ey $nzp)
        ($Dx $Ey $nzp)
        ($EEx $Ey $nzp)
        ($Fx $Ey $nzp)
        ($Fx $Ly $nzp)//35
        ($Gx $Ly $nzp)
        ($Gx $Ey $nzp)
        ($Hx $Ly $nzp)
        ($Hx $Ey $nzp)
//*********** 40-51 plano zp
        (0 $Cy $zp)
        (0 $Dy $zp)
        (0 $Ey $zp)
        ($Bx $Ey $zp)
        ($Dx $Ey $zp)
        ($EEx $Ey $zp)
        ($Fx $Ey $zp)
        ($Fx $Ly $zp)
        ($Gx $Ly $zp)
        ($Gx $Ey $zp)
        ($Hx $Ly $zp)
        ($Hx $Ey $zp)

//***Puntos PARTE DE INFERIOR MALLA*****************************
```

```
        //************Plano nzp
        (0 $By $nzp)//52
        ($Ax $By $nzp)
        ($Fd $By $nzp)
        ($Dx $Dd $nzp)
        ($Ex $Dd $nzp)
        ($Fx $Dd $nzp)//57
        ($Gx $Dd $nzp)
        ($Hx $Dd $nzp)

        (0 $Ay $nzp)//60
        ($Bx $Ay $nzp)
        ($Dx $Fy $nzp)
        ($Ex $Fy $nzp)
        ($Fx $Fy $nzp)//64
        ($Gx $Fy $nzp)
        ($Hx $Fy $nzp)

        (0 0 $nzp)//67
        ($Bx 0 $nzp)
        ($Dx 0 $nzp)
        ($Ex 0 $nzp)
        ($Fx 0 $nzp)//71
        ($Gx 0 $nzp)
        ($Hx 0 $nzp)

        //*****************Plano zp

        (0 $By $zp)//74
        ($Ax $By $zp)
        ($Fd $By $zp)
        ($Dx $Dd $zp)
        ($Ex $Dd $zp)
        ($Fx $Dd $zp)
        ($Gx $Dd $zp)
        ($Hx $Dd $zp)

        (0 $Ay $zp)//82
        ($Bx $Ay $zp)
        ($Dx $Fy $zp)
        ($Ex $Fy $zp)
        ($Fx $Fy $zp)
        ($Gx $Fy $zp)
        ($Hx $Fy $zp)

        (0 0 $zp)//89
        ($Bx 0 $zp)
        ($Dx 0 $zp)
        ($Ex 0 $zp)
        ($Fx 0 $zp)//93
        ($Gx 0 $zp)
        ($Hx 0 $zp)


        ($Cx $Jy $nzp)
        ($Cx $Jy $zp)

//***modificaciones************************************************
        //puntos de celdas superiores************
        //************plano nzp
        (0 $My $nzp)//98
        ($Bx $My $nzp)
        ($Ex $My $nzp)
        ($Gx $My $nzp)
        ($Hx $My $nzp)

        (0 $My $zp)
        ($Bx $My $zp)
        ($Ex $My $zp)
        ($Gx $My $zp)
        ($Hx $My $zp)
```

```
        //****************
        ($Dx $My $nzp)//108
        ($Dx $My $zp)
//****************
        ($Fx $My $nzp)//110
        ($Fx $My $zp)
);


blocks
(
//*******PARTE SUPERIOR************************************************
//*****capa limite superior del Ahmed y cola
        hex (0 1 2 3 4 5 6 7) ($BL_CellsY $BL_CornerCells 1) simpleGrading ($BL_Grading
1 1) //0
        hex (3 2 9 8 7 6 11 10) ($BL_CellsY $BL_CornerCells 1) simpleGrading ($BL_Grading
$cornerCalculated_Grading 1) //1
        hex (8 9 12 13 10 11 14 15) ($BL_CellsY $BL_UpperCells 1) simpleGrading
($BL_Grading 1 1) //2
        hex (13 12 16 17 15 14 18 19) ($BL_CellsY $BL_UpWakeCells 1) simpleGrading
($BL_Grading 1 1) //3

//*****bloques desde el techo a la capa límite
        hex (28 0 3 29 40 4 7 41) ($frontCellsX $BL_CornerCells 1) simpleGrading
($frontGradingX 1 1) //6
        hex (17 16 24 25 19 18 26 27) ($BL_CellsY $wakeCellsX 1) simpleGrading
($BL_Grading $wakeGradingX 1) //15

//****bloques frontal y dorsal del centro del Ahmed******************
        hex (52 53 0 28 74 75 4 40) ($frontCellsX $BL_FrontCells 1) simpleGrading
($frontGradingX 1 1) //16
        hex (53 54 1 0 75 76 5 4) ($BL_CellsY $BL_FrontCells 1) simpleGrading ($BL_Grading
1 1) //17
        //hex (57 65 20 16 79 87 22 18) ($BL_CellsY $BL_LowWakeCells 1) simpleGrading
(#calc "1/($BL_Grading)" 1 1) //18
        hex (57 59 24 16 79 81 26 18) ($wakeCellsX $BL_LowWakeCells 1) simpleGrading
($wakeGradingX 1 1) //19

//***********PARTE INFERIOR*******************************************
        hex (60 61 53 52 82 83 75 74) ($frontCellsX $BL_CornerCells 1) simpleGrading
($frontGradingX 1 1) //20
        hex (61 96 54 53 83 97 76 75) ($BL_CellsY $BL_CornerCells 1) simpleGrading
($BL_Grading 1 1) //21
        hex (62 55 96 61 84 77 97 83) ($BL_CellsY $BL_CornerCells 1) simpleGrading
($BL_Grading $BL_Corner_Grading 1) //22
        hex (64 57 55 62 86 79 77 84) ($BL_CellsY $BL_LowerCells 1) simpleGrading
($BL_Grading 1 1) //24
        hex (67 68 61 60 89 90 83 82) ($frontCellsX $floorCellsY 1) simpleGrading
($frontGradingX $floorGradingY 1) //27
        hex (68 69 62 61 90 91 84 83) ($BL_CornerCells $floorCellsY 1) simpleGrading
($cornerCalculated_Grading $floorGradingY 1) //28
        hex (69 71 64 62 91 93 86 84) ($BL_LowerCells $floorCellsY 1) simpleGrading (1
$floorGradingY 1) //30
        hex (71 73 66 64 93 95 88 86) ($wakeCellsX $floorCellsY 1) simpleGrading
($wakeGradingX $floorGradingY 1) //32

//***********CELDAS SUPERIORES MODIFICACION***************************
        hex (29 3 99 98 41 7 104 103) ($frontCellsX $upperCellsY 1) simpleGrading
($frontGradingX $upperGradingY 1) //33
        hex (8 13 100 108 10 15 105 109) ($BL_UpperCells $upperCellsY 1) simpleGrading
(1 $upperGradingY 1) //34

        hex (17 25 102 110 19 27 107 111) ($wakeCellsX $upperCellsY 1) simpleGrading
($wakeGradingX $upperGradingY 1) //36
        hex (3 8 108 99 7 10 109 104) ($BL_CornerCells $upperCellsY 1) simpleGrading
($cornerCalculated_Grading $upperGradingY 1) //37


        //hex (17 16 20 21 19 18 22 23) (20 20 1) simpleGrading (1 1 1) //4
        //hex (21 20 24 25 23 22 26 27) (20 30 1) simpleGrading (1 1 1) //5
        //hex (29 3 31 30 41 7 43 42) (30 10 1) simpleGrading (1 1 1) //7
```

```
        //hex (31 3 8 32 43 7 10 44) (10 20 1) simpleGrading (1 1 1) //8
        //hex (32 8 13 33 44 10 15 45) (10 20 1) simpleGrading (1 1 1) //9
        //hex (33 13 20 37 45 15 22 49) (10 20 1) simpleGrading (1 1 1) //10
        //hex (13 17 35 13 15 19 47 15) (20 20 1) simpleGrading (1 1 1) //11
        //hex (35 17 21 36 47 19 23 48) (20 20 1) simpleGrading (1 1 1) //12
        //hex (36 21 25 38 48 23 27 50) (20 30 1) simpleGrading (1 1 1) //13
        //hex (34 35 36 37 46 47 48 49) (10 20 1) simpleGrading (1 1 1) //14
        //hex (63 56 55 62 85 78 77 84) (20 60 1) simpleGrading (1 1 1) //23
        //hex (65 58 57 64 87 80 79 86) (20 20 1) simpleGrading (1 1 1) //25
        hex (64 66 59 57 86 88 81 79) ($wakeCellsX $BL_CellsY 1) simpleGrading
($wakeGradingX $BL_Grading 1) //26
        //hex (69 70 63 62 91 92 85 84) (60 80 1) simpleGrading (1 1 1) //29
        //hex (71 72 65 64 93 94 87 86) (20 80 1) simpleGrading (1 1 1) //31
        hex (13 17 110 100 15 19 111 105) ($BL_UpWakeCells $upperCellsY 1) simpleGrading
(1 $upperGradingY 1) //35
);

edges
(
//**********Arcos Superiores del bloque 0 y 1
   arc 1 2 (#calc "$Fd+$R-$R*cos(20.0*$DgToRad)" #calc "$Cy+$R*sin(20.0*$DgToRad)" $nzp)
        arc 5 6 (#calc "$Fd+$R-$R*cos(20.0*$DgToRad)" #calc "$Cy+$R*sin(20.0*$DgToRad)"
$zp)
        arc     0     3     (#calc     "$Fd+$R-($R+$CapL)*cos(20.0*$DgToRad)"     #calc
"$Cy+($R+$CapL)*sin(20.0*$DgToRad)" $nzp)
        arc     4     7     (#calc     "$Fd+$R-($R+$CapL)*cos(20.0*$DgToRad)"     #calc
"$Cy+($R+$CapL)*sin(20.0*$DgToRad)" $zp)

        arc 2 9 (#calc "$Fd+$R-$R*sin(20.0*$DgToRad)" #calc "$Dd+$D+$R+$R*0.939693" $nzp)
        arc       6       11       (#calc       "$Fd+$R-$R*sin(20.0*$DgToRad)"       #calc
"$Dd+$D+$R+$R*cos(20.0*$DgToRad)" $zp)
        arc       3       8       (#calc       "$Fd+$R-($CapL+$R)*0.342020"       #calc
"$Dd+$D+$R+($CapL+$R)*cos(20.0*$DgToRad)" $nzp)
        arc       7       10       (#calc       "$Fd+$R-($CapL+$R)*0.342020"       #calc
"$Dd+$D+$R+($CapL+$R)*cos(20.0*$DgToRad)" $zp)

//**********Arcos Superiores del bloque 21 y 22

        arc    54    96    (#calc    "$Fd+$R-$R*cos(20.0*$DgToRad)"    #calc    "$Dd+$R-
$R*sin(20.0*$DgToRad)" $nzp)
        arc    76    97    (#calc    "$Fd+$R-$R*cos(20.0*$DgToRad)"    #calc    "$Dd+$R-
$R*sin(20.0*$DgToRad)" $zp)
        arc   53   61   (#calc   "$Fd+$R-($R+$CapL)*cos(20.0*$DgToRad)"   #calc   "$Dd+$R-
($R+$CapL)*sin(20.0*$DgToRad)" $nzp)
        arc   75   83   (#calc   "$Fd+$R-($R+$CapL)*cos(20.0*$DgToRad)"   #calc   "$Dd+$R-
($R+$CapL)*sin(20.0*$DgToRad)" $zp)


        arc    55    96    (#calc    "$Fd+$R-$R*sin(20.0*$DgToRad)"    #calc    "$Dd+$R-
$R*cos(20.0*$DgToRad)" $nzp)
        arc    77    97    (#calc    "$Fd+$R-$R*sin(20.0*$DgToRad)"    #calc    "$Dd+$R-
$R*cos(20.0*$DgToRad)" $zp)
        arc    62    61    (#calc    "$Fd+$R-($CapL+$R)*0.342020"    #calc    "$Dd+$R-
($CapL+$R)*cos(20.0*$DgToRad)" $nzp)
        arc    84    83    (#calc    "$Fd+$R-($CapL+$R)*0.342020"    #calc    "$Dd+$R-
($CapL+$R)*cos(20.0*$DgToRad)" $zp)

);

boundary
(
    Ahmed
    {
        type wall;
        faces
        (
            (1 2 6 5)
            (2 9 11 6)
            (9 12 14 11)
            (12 16 18 14)
            (16 57 79 18)
```

```
        (57 55 77 79)
        (55 96 97 77)
        (96 54 76 97)
        (54 1 5 76)
      );
}

inlet
{   type patch;
    faces
    (   (98 29 41 103)
        (29 28 40 41)
        (28 52 74 40)
        (52 60 82 74)
        (60 67 89 82)   );
}

outlet
{
    type patch;
    faces
    (
    (25 102 107 27)//36
    (24 25 27 26)//15
    (59 24 26 81)
    (73 66 88 95)
            (66 59 81 88)
    );
}

upperWall
{   type patch;
    faces
    (
        (99 98 103 104)
        (108 99 104 109)
        (100 108 109 105)
        (102 110 111 107)
            (100 105 111 110)
    );
}

lowerWall
{   type wall;
    faces
    (
                    (67 68 90 89)
                    (68 69 91 90)
                    (69 71 93 91)
                    (71 73 95 93)
    );
}

frontAndBack
{   type empty;
    faces
    (
        (98 99 3 29) (41 7 104 103)
                    (29 3 0 28) (40 4 7 41)
                    (28 0 53 52) (74 75 4 40)
                    (52 53 61 60) (82 83 75 74)
                    (60 61 68 67) (89 90 83 82)

                    (99 108 8 3) (7 10 109 104)
                    (3 8 9 2) (6 11 10 7)
                    (0 3 2 1) (5 6 7 4)
                    (0 1 54 53) (75 76 5 4)
                    (53 54 96 61) (83 97 76 75)
                    (61 96 55 62) (84 77 97 83)
                    (61 62 69 68) (90 91 84 83)
```

```
                        (108 100 13 8) (10 15 105 109)//34
                        (8 13 12 9) (11 14 15 10)
                        (55 57 64 62) (84 86 79 77)//24
                        (62 64 71 69) (91 93 86 84)

                        (110 102 25 17) (27 107 111 19)//36
                        (12 13 17 16) (18 19 15 14)//3
                        (17 25 24 16) (18 26 27 19)//15
                        //(16 20 65 57) (79 87 22 18)//18
                        (16 24 59 57) (81 26 18 79)//19
                        (64 66 73 71) (93 95 88 86)//32

                        (57 59 66 64) (86 88 81 79)//26
                        (100 110 17 13) (19 111 105 15)//35
        );
    }
);

mergePatchPairs ( );
// ************************************************************************* //
```

b. GeometricConditions

El parámetro *alpha* es el encargado de modificar el ángulo de inclinación.

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
/* Cambiar Geometría
        -Ahmed body (lon,L,D,R,C,alpha)
        -grosor de capa límite (CapL)
        -ángulo del chaflan (alpha)
        -profundidad de celda (zp y nzp)
        -Distancia ente el suelo y el Ahmed (Dd)  */
lon 1044;       //longitud
L 288;          //altura (distancia característica)
D 88;           //altura morro plano
R 100;          //radio de curvatura
C 222;          //longitud del chaflan
alpha 35.0;     //ángulo de caida

CapL 25;//grosor de capa límite
Dd 50;          //Distancia ente el suelo y el Ahmed

nzp -50;//grosor negativo de celda en el plano Z
zp 50;          //grosor positivo de celda en el plano Z

A 100; //no afecta
#inputMode           merge
// ************************************************************************* //
```

c. GradingConditions de la malla fina

```
/*--------------------------------*- C++ -*----------------------------------
*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------
*/
//***Cambiar numero de Celdas de cada Cara y Grading

//Version 8****************************************[125.300 cells]
/**/

//Boundary-Layer <cells> <grading>
BL_CellsY 20; BL_Grading 0.2; //normal a la capa límite
BL_CornerCells 20; BL_Corner_Grading 1; //celdas en las esquinas
BL_UpperCells 150;    //celdas perpendiculares a la BL del techo del ahmed
BL_LowerCells250;     //celdas perpendiculares a la BL de la base del ahmed
BL_FrontCells 20;     //celdas verticales en el morro plano del ahmed
BL_LowWakeCells 90;   //celdas perpendiculares a la BL en la cola (plano
vertical)
BL_UpWakeCells 70;    //celdas perpendiculares a la BL en la cola (plano
inclinado)

//distances Ahmed-boundaries <cells> <grading>
frontCellsX 1; frontGradingX 1;     //celdas horizontales desde el muro
frontal hasta el ahmed
wakeCellsX 1; wakeGradingX 1;       //celdas horizontales del ahmed hasta la
salida
upperCellsY 1; upperGradingY 1;     //celdas verticales entre el ahmed y el
muro superior
floorCellsY 1; floorGradingY 1;     //celdas verticales entre el ahmed y el suelo


#inputMode        merge

// ************************************************************************* //
```

d. GradingConditions de la malla gruesa

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
//***Cambiar numero de Celdas de cada Cara y Grading

//5****************************************[27.250 cells]
/**/
        BL_CellsY 5; BL_Grading 1; //normal a la capa límite

        BL_CornerCells 20;  BL_Corner_Grading 1;//esquinas capa límite
        BL_UpperCells 100;      //
        BL_LowerCells 200;

        BL_FrontCells 20;
        BL_LowWakeCells 30;
        BL_UpWakeCells 30;

        frontCellsX 50; frontGradingX 0.03; //
        wakeCellsX 150; wakeGradingX 20;
        upperCellsY 40; upperGradingY 50;
        floorCellsY 5; floorGradingY 1;

#inputMode           merge
// ************************************************************************* //
```

e. FunctionObjects en ControlDict

Se ha implementado una modificación del código para poder extraer las fuerzas de sustentación y resistencia aerodinámicas, así como sus coeficientes.

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  3.0.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
#include "include/initialConditions"

application     pimpleFoam;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime         6;
```

```
deltaT          1.5e-05;

writeControl    timeStep;

writeInterval   1500;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

functions
(
forceCoeffs1
    {
        type        forceCoeffs;                                    //   Report coefficient
of forces
        functionObjectLibs ( "libforces.so" );
                outputControl timeStep;
        outputInterval 100;                                         //  iteration  at  which
output will be reported
        log          yes;                                                  //true; //yes
        patches ( Ahmed );                                          //   All   patches   to
compute forces on
        pName       p;
        UName       U;
    // Optional writing of force volume fields
                writeFields         yes;
    // Indicates incompressible
                rhoName rhoInf;
        // Redundant for incompressible
        rhoInf      $rhoFlow;           // is the free stream density.
        liftDir       ( 0. 1 0 );               // lift direction
        dragDir    ( 1 0. 0 );          // drag direction
        // Axle midpoint on ground
        CofR        (3.402  0.194 0);   // center of rotation
        pitchAxis  ( 0 0 1 );  // pitch axis
        magUInf     $flowUinf;   // inflow velocity
        // Wheelbase length
         lRef        1.044;        // Length
    //  Estimated Frontal Area
                Aref        0.0288;
    }

forces
{
        type forces;
        functionObjectLibs ( "libforces.so" );
        outputControl timeStep;
        outputInterval 100;             // iteration at which output will be reported
        // Optional writing of force volume fields
        writeFields         yes;

                patches ( Ahmed );              // All patches to compute forces on
        pName       p;
        UName       U;
        log         true;
                rhoName rhoInf;                         // Indicates incompressible
        rhoInf      $rhoFlow;           // is the free stream density.// Redundant for
incompressible
        CofR        ( 3.402  0.194 0 );   // center of rotation
```

```
}
);
// ************************************************************************* //
```

f. Script para monitorizar residuales de Ux y Uy

```
#!/bin/bash

gnuplot -persist > /dev/null 2>&1 << EOF
set logscale y
set title "Residuals"
set ylabel 'Residuals'
set xlabel 'Iteration'
plot "< cat log2_10cores | grep Uy | cut -d' ' -f9 | tr -d ','" title 'Uy' with lines ,
"< cat log2_10cores | grep Ux | cut -d' ' -f9 | tr -d ','" title 'Ux' with lines

EOF
```

g. Script Allresults

Su función es la de generar un sumario de datos específicos tras la simulación así como graficar y guardar imágenes de los residuales, el yPlus, los coeficientes y las fuerzas.

```
#!/bin/sh
cd ${0%/*} || exit 1    # Run from this directory

# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions
. /opt/openfoam30/etc/bashrc
#------------------------------------------------------------
log=log
coeffs=0/forceCoeffs.dat

reconstructPar -newTimes
yPlus -newTimes
foamLog $log

#------------------------------------------------------------
echo "--------------------------"

#simTime VS computationTime
tail -40 log | grep -w 'Time' | echo "1st log Time:" $(cut -d' ' -f3) && tail -30 log |
grep -w 'ClockTime' | echo "1st log Computation time:" $(cut -d' ' -f8)

echo "--------------------------"
#simTime VS computationTime
tail -40 $log | grep -w 'Time' | echo "Total sim Time:" $(cut -d' ' -f3) && tail -30 $log
| grep -w 'ClockTime' | echo "Computation time:" $(cut -d' ' -f8)

echo "--------------------------"
#deltaT
cat system/controlDict | grep 'deltaT' | cut -d';' -f1

#sacar la media de la segunda columna ($2)
echo "CoMax medio:" $(awk '{ total += $2 } END { print total/NR }' logs/CourantMax_0)

echo "--------------------------"
#sacar columna de yPlus medio al final
echo "last yPlus avg:" $(patchAverage -latestTime yPlus Ahmed | grep 'Average' | cut -d'
' -f12)

#sacar columna de yPlus medio en cada tiempo
echo "yPlus mean:" $(patchAverage yPlus Ahmed | grep 'Average' | cut -d' ' -f12 | awk '{
total += $1 } END { print total/NR }')
echo "--------------------------"
```

```
tail -309 postProcessing/forceCoeffs1/$coeffs > postProcessing/forceCoeffs1/lastCoeffs

#CDmean
echo "CD mean:" $(awk '{ total += $3 } END { print total/NR }'
postProcessing/forceCoeffs1/lastCoeffs)

#CLmean
echo "CL mean:" $(awk '{ total += $4 } END { print total/NR }'
postProcessing/forceCoeffs1/lastCoeffs)


echo "--------------------------"
./plot_forces.py

tail -300 forces.txt > lastForces
echo "Drag force:" $(awk '{ total += $3 } END { print total/NR }' lastForces)

echo "Lift force:" $(awk '{ total += $2 } END { print total/NR }' lastForces)

patchAverage yPlus Ahmed | grep -w 'Time' | cut -d' ' -f3 |tail -n +2 >
postProcessing/times && patchAverage yPlus Ahmed | grep -w 'Average' | cut -d' ' -f12 >
postProcessing/yPlusValues && paste postProcessing/times postProcessing/yPlusValues >
postProcessing/yPlusVStime && rm postProcessing/times postProcessing/yPlusValues &&
gnuplot plotYplus

gnuplot Residuals.gnu
gnuplot plotCD


# ------------------------------------------------------------------ end-of-file
```

B. Mallas empleadas

a. Malla fina para flujo turbulento (v.3)

Número total de celdas: 125.300

```
//****************************************[125300 cells]
/**/
       BL_CellsY 20; BL_Grading 0.2; //normal a la capa límite

       BL_CornerCells 20;  BL_Corner_Grading 1; //esquinas redondeadas
capa límite
       BL_UpperCells 150;    //techo del ahmed capa límite
       BL_LowerCells 250;    //suelo del ahmed capa límite

       BL_FrontCells 20;     //morro plano capa limite
//divisiones perendiculares a la BL en la cola
       BL_LowWakeCells 90;
       BL_UpWakeCells 70;

       frontCellsX 100; frontGradingX 0.03; //
       wakeCellsX 350; wakeGradingX 60;
       upperCellsY 80; upperGradingY 50;
       floorCellsY 10; floorGradingY 1;
```

b. Malla fina para flujo Laminar (v.3)

Número total de celdas: 27.500

- GradingConditions:

```
//****************************************[27.250 cells]
/*
       BL_CellsY 5; BL_Grading 1; //normal a la capa límite

       BL_CornerCells 20;  BL_Corner_Grading 1;   //esquinas      capa
límite
       BL_UpperCells 100;    //
       BL_LowerCells 200;

       BL_FrontCells 20;
       BL_LowWakeCells 30;
       BL_UpWakeCells 30;

       frontCellsX 50; frontGradingX 0.03; //
       wakeCellsX 150; wakeGradingX 20;
       upperCellsY 40; upperGradingY 50;
       floorCellsY 5; floorGradingY 1;
*/
```