Preliminary design of a high-altitude kite

De-rating the 2D kite section for a finite span

In the earlier papers in this series, I considered only the cross-section of a kite whose lifting surface is a loosely-held flexible membrane. The shape of the membrane is not determined by a rigid structure, but by the pattern of the airflow. The aerodynamics have been studied using a two-dimensional wind tunnel, and the resulting patterns of airflow have been assumed to obtain uniformly across the span of the kite.

Moving from a two-dimensional analysis to a three-dimensional analysis is going to force us to accept some simplifying assumptions. For the two-dimensional analysis, the kite section was modeled as a round one-inch diameter leading edge tube to which a 98-centimeter strip of ripstop nylon was attached. The trailing edge of the nylon sheet was held aft by "trailing edge strings" secured to the ends of straight carbon fiber ribs. The sole purpose of the ribs was to define the trailing edge of the section; they did not control the shape of the membrane. For the purposes of computational fluid dynamics, the membrane was divided into 500 discrete segments. The shape of the membrane was determined using a succession of iterations, each of which consisted of an OpenFoam run and a follow-up run of a shape-calculation program which calculates the shape the chain of 500 segments takes on under the influence of the forces calculated by OpenFoam.

I am not going to generalize the shape-calculation procedure from 2D to 3D in this paper. Instead, I am going to assume that the shape of the flexible membrane is the same across the span. One of the conclusions which can be drawn from the earlier papers is that the shape of the membrane stays relatively constant over a wide range of different flight conditions. The shape does change with wind conditions, of course, but the changes in shape are often in the order of fractions of millimeters.

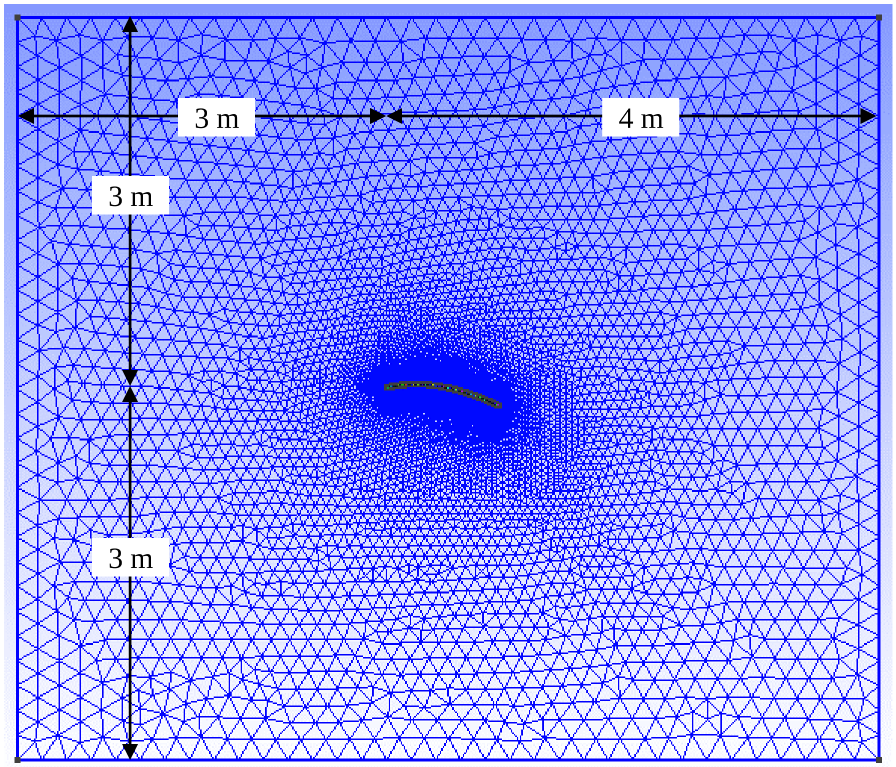
The second simplification I will make is to reduce the degree of discretization of the membrane. Instead of dividing the membrane into 500 segments (each with its own top and bottom surface), I will divide the membrane into about one-quarter of that number. Reducing the complexity of the cross-section is necessary to keep the size of the three-dimensional model within the bounds of one's computer.

The third simplification I will make is to remove the leading edge tube entirely. It is true that the wake of the leading edge tube has a significant impact on the airflow on the lower surface of the membrane. But the influence of the wake is still accounted for if we take the shape of the membrane as a given, as I propose to do. I made an attempt to model the leading edge tube with one-quarter of the number of points as was used in the 2D model, but the result was not satisfactory. A greater percentage of the points was needed. In the end, I dispensed with the leading edge tube entirely.

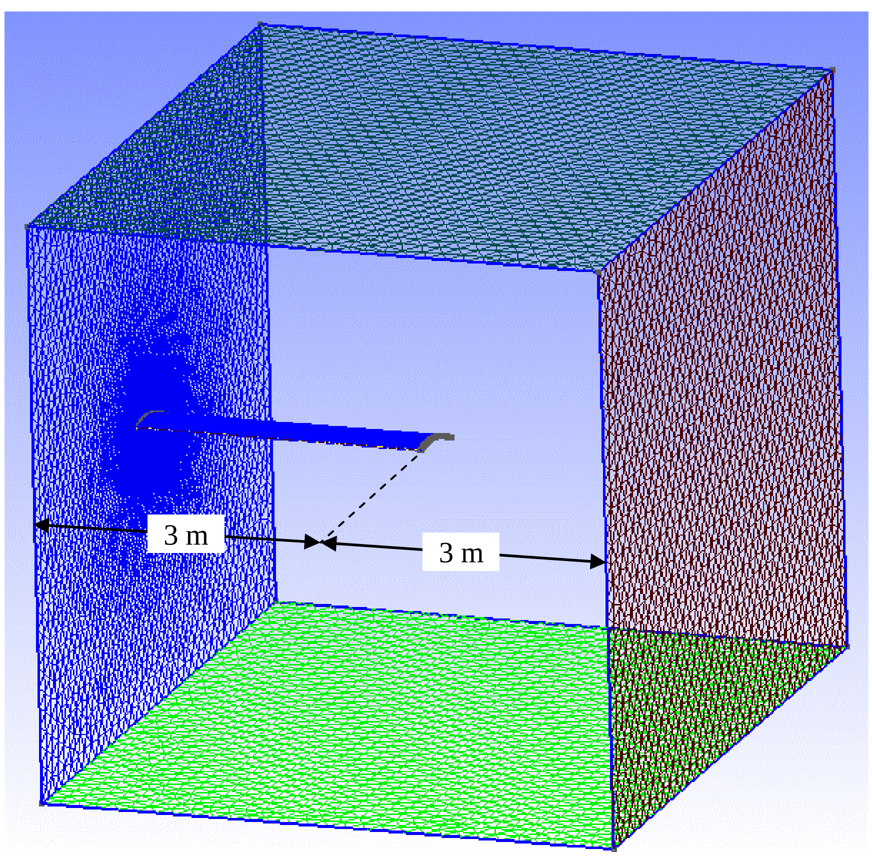
It is clear that the lift and drag forces calculated using the simplified model can not be used as estimates for the absolute lift and drag acting on the kite wing. But they can still be used to see how the lift and drag change as the span is changed. Because the lift (per meter of span) decreases as the span decreases, the fractions can be used to "de-rate" the lift and drag from their infinite-span ideal values.

All of the OpenFoam simulations done for this paper assume a 9° angle of attack and air conditions at 15,000 feet. I tested wing spans from one meter to ten meters. Since the chord length is approximately one meter, the corresponding aspect ratios vary from one to ten.

The following figure shows the virtual wind tunnel when it holds a wing with a six meter span. This is a view through the sides of the wind tunnel, looking straight down the wing, which is seen in cross-section. The origin of the $\hat{x}$-$\hat{y}$ plane of the wind tunnel is the leading edge of the free-flying segments of the membrane. The top and bottom of the wind tunnel are three meters above and below this origin.



The inlet of the wind tunnel is three meters upwind from the origin, and the outlet is four meters downwind. The following is a peek inside wind tunnel. The inlet and outlet are not shown, so one can see right through the wind tunnel. Only the left half of the wing is modeled. The right wall of the wind tunnel is the midplane of the wing, the vertical plane which slices the wing into two equal halves. The wind tunnel is six meters wide.



I refer to the mesh into which the right wall of the wind tunnel (which is the midplane of the wing) has been divided. The triangles in this mesh are much more regular than the randomly-oriented triangles which are normally used. The randomly-oriented triangles are the result of using a meshing process first described by Messr. Delaunay. The Delaunay algorithm is the default used by the meshing program GMesh. Unfortunately, it did not work on this physical model. It was able to mesh the plane surfaces, like the sides of the wind tunnel, but it could not figure out how to mesh the three-dimensional volume of air which is held in the interior of the wind tunnel. Perhaps GMesh was thinking hard, or perhaps it was just plain stumped, but I gave up waiting after a day (24 hours). As an alternative to Delaunay, I used the "Frontal" algorithm. It was able to mesh the interior, but still needed several hours to do so. In order to get the Frontal algorithm to run in three dimensions, it was necessary to use the Frontal algorithm for the two dimensional (that is, plane surface) meshing, too. The Frontal algorithm attacks a surface or volume with somewhat less imagination than the Delaunay algorithm, which leads to more regular triangulation, but was able to complete the task.

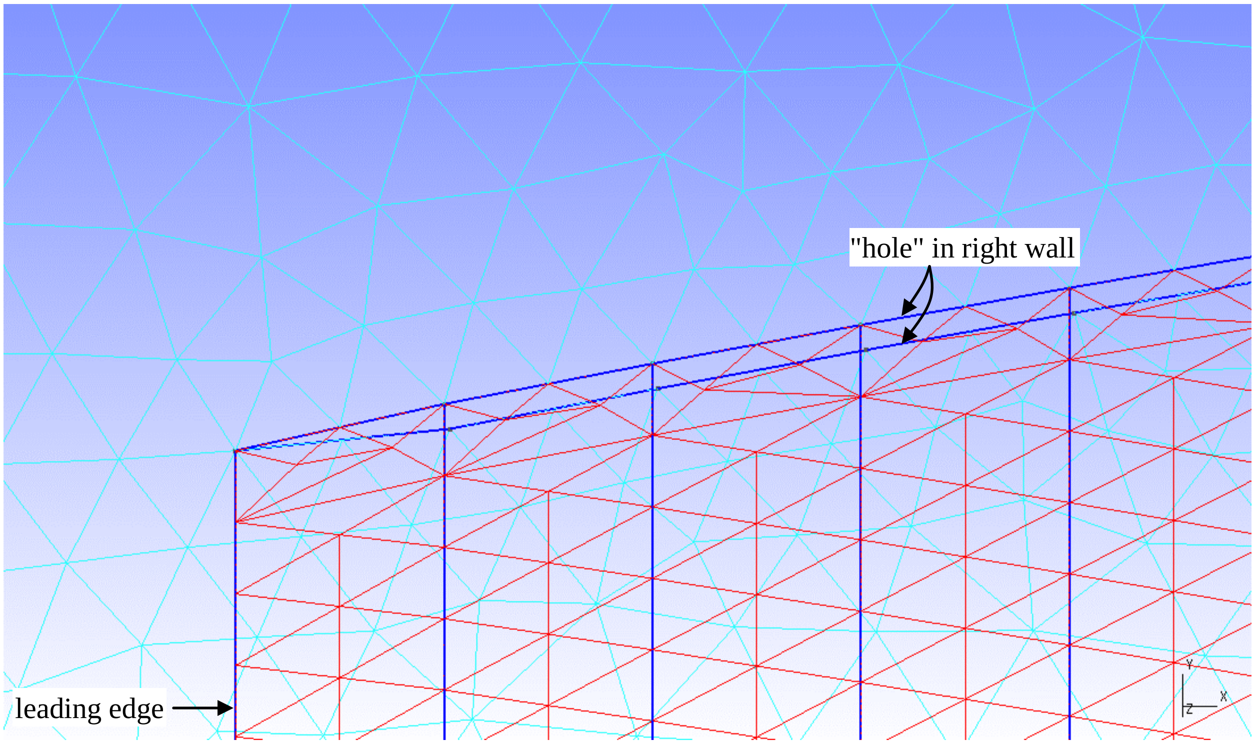
(Aside: This is the first time I have experienced a failure of the Delaunay algorithm,. I have used GMesh to produce meshes with up to 40,000,000 tetrahedra ~~without too much trouble~~ with no more than the usual amount of trouble and fiddling.)

It is possible that the trouble arises from the extreme difference between the mesh size needed on the wing tip and the mesh size used on the left wall of the wind tunnel, only three meters away. Let me explain why this might be so. To describe the shape of the membrane, I started with the co-ordinates of the surface determined in one of the earlier papers for the case of a 9° angle of attack, a 20 mph wind speed and the characteristics of air at 15,000 feet. The complete set of co-ordinates included: (i) top and bottom surfaces on a membrane one millimeter thick, (ii) the outside surface of the nylon wrapped about two-thirds of the way around the leading edge tube, and (iii) the uncovered part of the leading edge tube. To extract an outline from this previous set of co-ordinates, I chose as the first, or leading, point that point on the covered part of the leading edge tube which lay one segment's length upwind from the departure point (as defined in the earlier papers). With this as the leading point, I then selected every fourth hinge (a hinge is an inter-segment point of contact) along the top surface and every fourth hinge along the bottom surface. I then connected the two final points on the surface with a short vertical line, representing the trailing edge.

Since the membrane had a physical length of 98 centimeters, and was originally divided into 500 segments, each segment in the earlier papers was approximately two millimeters long. By now taking every fourth point, the segments were about eight millimeters long in the chordwise direction. This caused problems near the wing tip, since the membrane is only one millimeter thick. The length of segments in the horizontal direction, along the surface, was eight times greater than vertical thickness at the tip. Both GMesh and the `gmshToFoam` converter reported problems in the neighbourhood of the tip. I had to resort to the following device to permit the size of the mesh to vary more smoothly. I specified that the characteristic length to be used by GMesh for meshing at the wing tip was to be two millimeters.

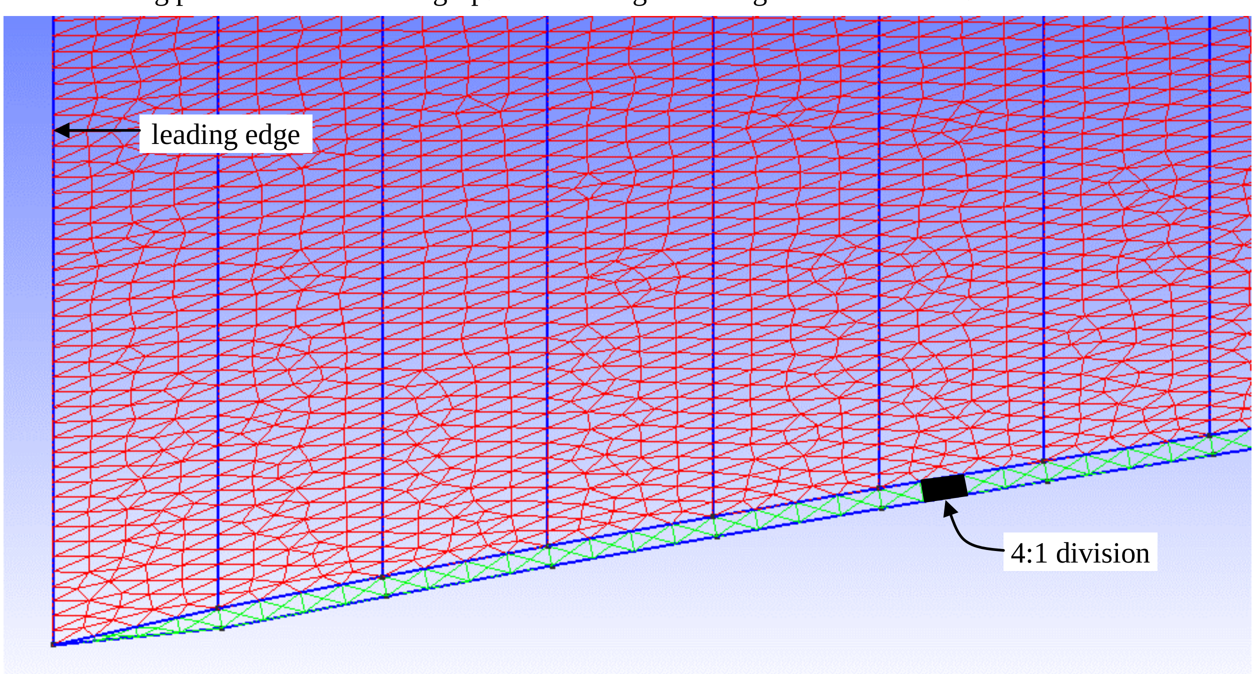
I encountered similar problems in the neighbourhood of the wing root. I specified the characteristic length to be used by GMesh for meshing at the vertices of the wind tunnel as 20 centimeters. Both GMesh and the `gmshToFoam` converter reported problems trying to marry the mesh on the right wall of the wind tunnel with the mesh on the surfaces of the membrane when the latter was meshed with triangles whose base was equal to the segment length. These problems ceased when I reduced the characteristic length on the surface of the membrane at the root by two. In other words, each segment on the siurface of the membrane at the wing root provided bases for two triangles.

The following two pictures show the mesh on the top surface of the membrane at the wing root and at the wing tip. The first picture shows the wing root.

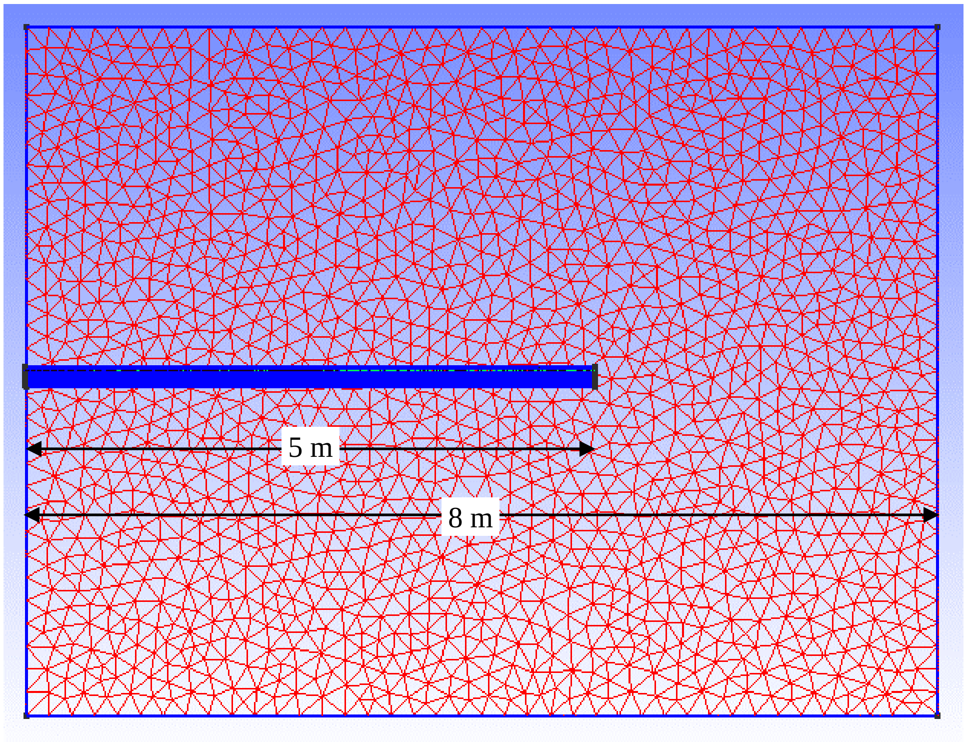


The picture shows only the top surface of the leading five segments. The red lines are the edges of the triangles which represent the top surface of the wing. The right wall of the wind tunnel, which is the midplane of the wing, is also shown, and its triangles are defined by light blue lines. The heavy blue lines are the lines of intersection between individual plane surfaces. Although the mesh on the bottom surface is not shown, the line of intersection between the bottom surface and the right wall is shown. The closed loop of lines around the wing root is defined as a "hole" in the right wall of the wind runnel. This hole extends all the way through the half-wing, but is closed at the other end by the surface which defines the wing tip. The volume which is filled with fluid (here, air) must be very carefully defined for OpenFoam. In any event, it can be seen in this picture that the mesh has been constructed so that each segment of the wing at the root is host to two triangles.

The following picture shows the wing tip. The leading seven segments are shown.



There is one last characteristic of the wind tunnel I must explain. In the picture above looking through the wind tunnel, from upwind to downwind, the six-meter wing span kite was shown. Only the left half of the wing, three meters long, was included. There was a three meter clear space between the wing tip and the left wall of the wind tunnel. In all of the three-dimensional simulations run for this paper, the clear space between the wing tip and the left wall was fixed at three meters. The following picture illustrates this. It shows the left half of the ten-meter wing span kite, seen from straight ahead. Including the three-meter clear space, the wind tunnel is eight meters wide.

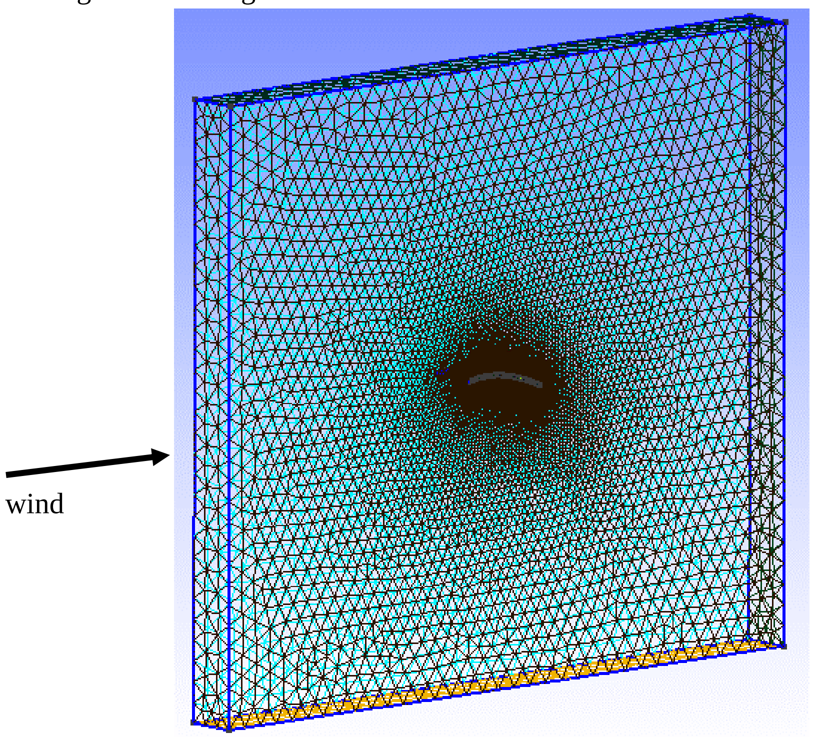


This picture shows one other thing as well. The red grid is the outlet of the wind tunnel (the inlet is not shown). The outlet was meshed using the Delaunay algorithm. The random orientation of the triangles is quite different from the regular array of triangles which the Frontal algorithm produces. This picture is for illustration only; as I said above, the Frontal algorithm was used for both 2D and 3D meshing.

The following table highlights some of the time and storage requirements for the 3D meshes used in this study. In all cases, the mesh produced by GMesh was optimized using four iterations of the Netgen optimization procedure. The sole fault detected by OpenFoam's `checkMesh` utility was the existence of a number of severely non-orthogonal faces. In no case did the number of severely-non-orthogonal faces prevent OpenFoam from declaring `Mesh OK`. All iterations of OpenFoam were carried out until all residuals were less than 10^{-5}.

Total wing span →	**2 meters**	**4 meters**	**6 meters**	**8 meters**	**10 meters**
Time required by 3D GMesh	332 sec	721 sec	1870 sec	2090 sec	2710 sec
Total badness after Netgen optimization	2,460,000	4,640,000	6,770,000	8,920,000	11,200,000
Severely non-orthogonal faces	145	255	406	537	555
Number of OpenFoam tetrahedra	1,960,000	3,690,000	5,380,000	7,090,000	8,930,000
Number of OpenFoam iterations	333	747	457	739	1,112
Time required to converge	4,380 sec	20,100 sec	19,200 sec	40,100 sec	84,800 sec

Because of the lack of a leading edge tube, use of a different turbulence model (to be described below) and other differences, the lift and drag figures which OpenFoam produces for this physical model will not be directly comparable to those described in the earlier papers. To provide a better basis for comparison, it would be useful to be able to model an infinite span wing using this physical model. It is impractical to continue to increase the span, testing 20 meters, 30 meters and so on. But, an infinite span can be and was modeled using the following virtual wind tunnel.



The tunnel is one meter thick. The airfoil pierces both side walls. The "hole" in the wind tunnel passes right through. In all other respects, including dimensions and the meshing algorithm, the wind tunnel is the same as used to model wings with finite span.

The apparent wind is perpendicular to the upwind face, or inlet, of the wind tunnel. For OpenFoam's purpose, the inlet was defined as a constant velocity face, across which the wind speed and direction are constant. The downwind face, or outlet, was defined as a constant pressure face, across which the pressure was set to a constant value. The top and bottom of the wind tunnel were defined as `symmetryPlanes`, across which all physical variables must remain unchanged. The left and right walls were also defined as `symmetryPlanes`.

Since the flow is low speed and incompressible, I again used OpenFoam's `simpleFoam` procedures. To model turbulence, I used OpenFoam's `kOmegaSST` procedures. I estimated initial parameters for time `0` in the OpenFoam simulations using the following independent guesses ...

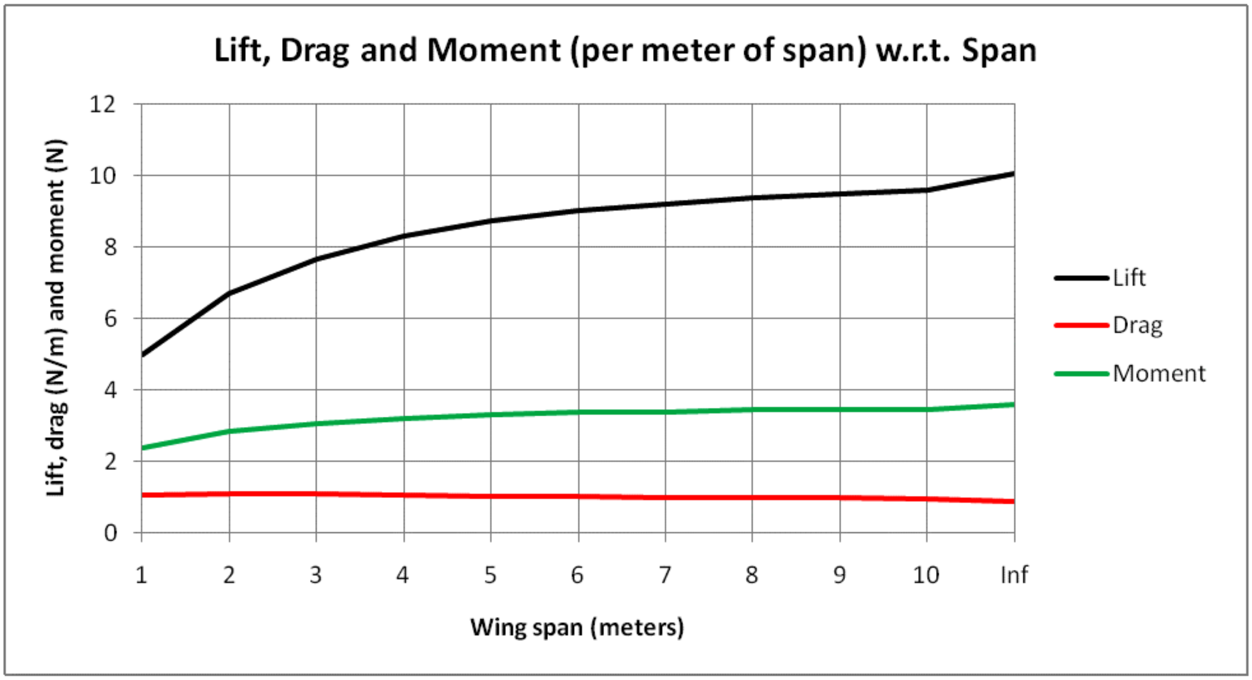
Quantity	**Formula**	**Value**
Fluid speed U	n/a	20 mph = 8.94 m/s
Turbulent intensity I	n/a	1% (relatively low)
Turbulence length scale l	n/a	0.5 m (half chord)

... and the following dependent guesses:

Quantity	Formula	Value
Turbulent viscosity $\tilde{\nu}$ (nut)	$\tilde{\nu} = \frac{3}{2}UIl$	0.0548
Turbulent energy k	$k = \frac{3}{2}(UI)^2$	0.0120
Turbulence length scale ω	$\omega = \frac{\sqrt{k}}{l}$	10.95

For the sake of completeness, I have attached as Appendix "A" a listing of the text file used by GMesh to produce the mesh for the base case. I have attached as Appendix "B" a listing of the 13 text files in the case directory for the base case which OpenFoam uses as inputs and for control.

The following graph shows the lift and drag forces, and the nose-down aerodynamic moment, acting on wings of various span. The forces are expressed in Newtons per meter of span and the moment is expressed in Newton-meters per meter of span.

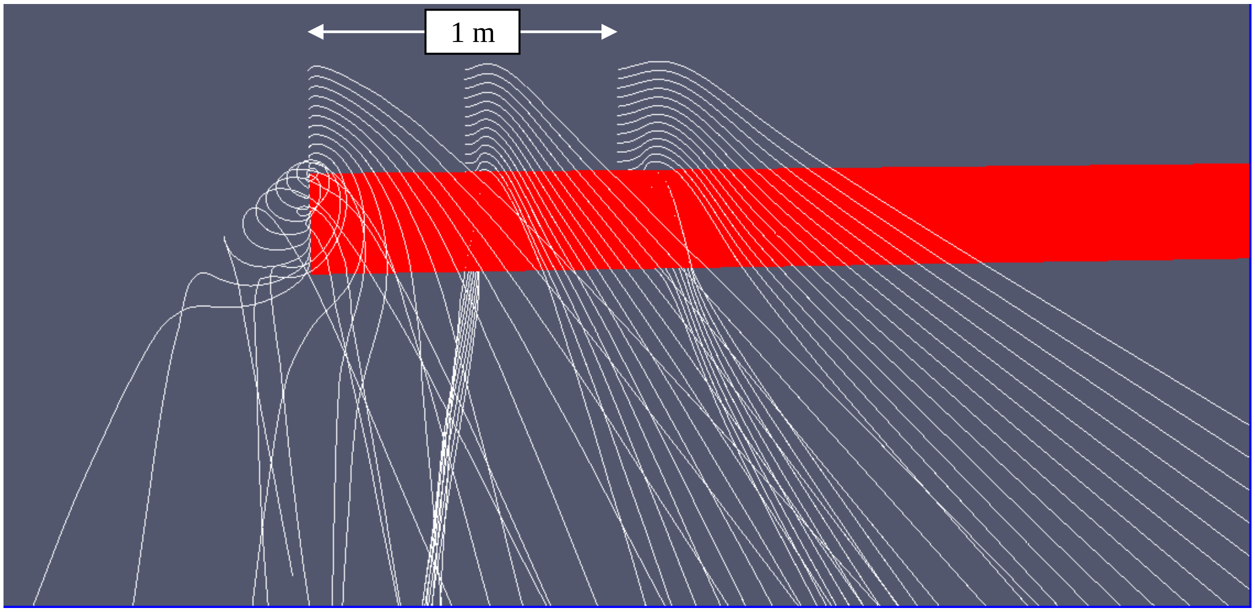


The horizontal axis states the full span of the wing, from one meter to ten meters. I have shown the result for an infinite span adjacent to that for the ten-meter span, but it should be understood that the infinite span represents the asymptote of the curve if it was extended indefinitely out towards the right.

I want to be clear about how the numbers should be interpreted. Consider the lift, which decreases as the span decreases. The lift decreases for two reasons: (i) with a shorter span, there is less lifting area (this effect is not shown in the graph), and (ii) the lift per meter of span also decreases as the span decreases (this is the effect shown). The drag (the red curve) exhibits a different tendency – the drag per meter of span increases slightly as the span decreases. But, that is not to say that the total drag on a shorter wing is greater than that on a longer wing.

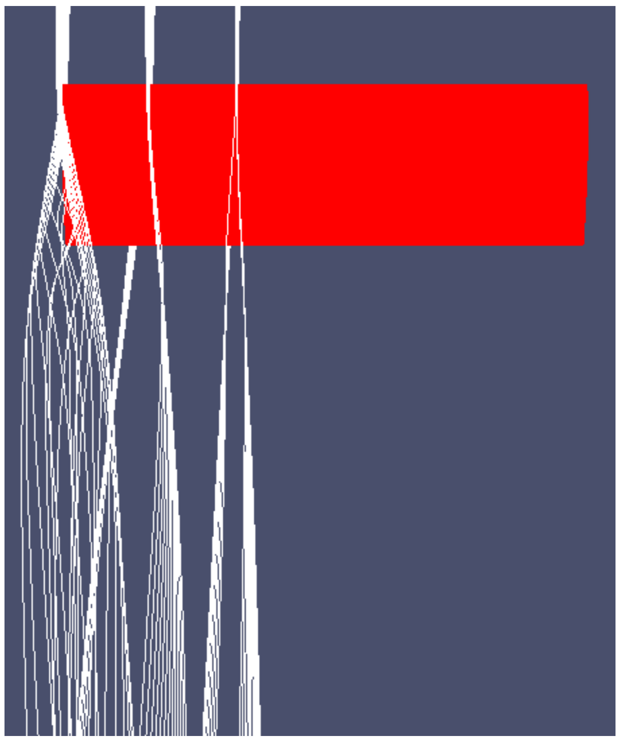
In so far as the lift is concerned, the reduction in "efficiency" is not too great until the span gets less than six meters. The lift per meter of span for a six-meter wing is about 90% of that for an infinite wing. As the wing span deceases below six meters, the efficiency gets progressively worse. A one-meter span wing has only about 50% of the efficiency of an infinite span wing.

The following picture shows certain streamtracers near the left wing tip of a six-meter wing.

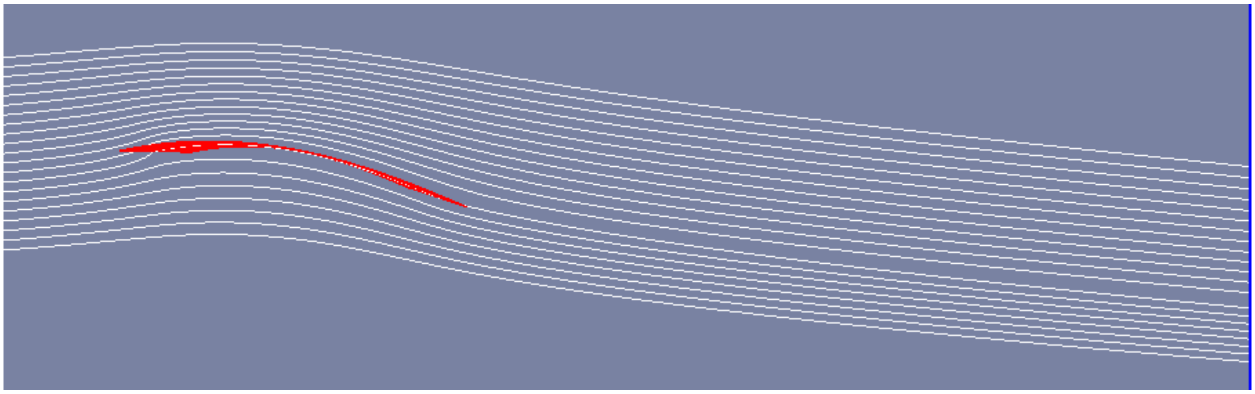


The "camera" with which this picture is taken is located five meters aft, and one-half meter above, the leading edge of the wing, in line with the wing tip. The three sets of streamtracers are generated from three vertical lines which are one-half meter upwind from the leading edge, and extend from 30 centimeters above the leading edge to 30 centimeters below. 20 individual streamtracers emanate from equally-spaced locations along each vertical line. The vertical lines are concentrated near the wing tip. The inboard vertical line (the right-most one in the picture) is located one meter in from the wing tip. The middle one is located one-half meter in from the wing tip and the third one is located right at the wing tip.

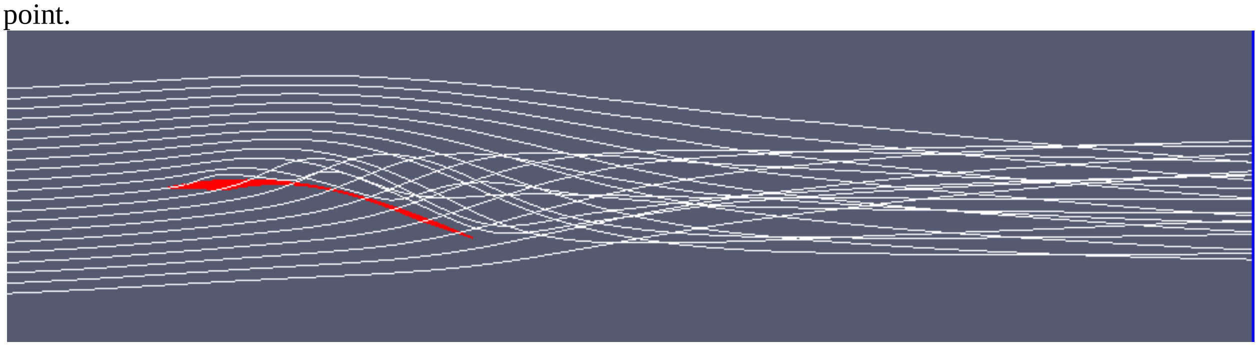
It seems that the effect of the wing tip is largely confined to the outboard meter or so of the wing. The picture here is a view of the left half of this wing from straight above. The inboard set of streamtracers is not disturbed very much from its vertical plane. The streamtracers which flow over the wing are deflected inboard; the streamtracers which flow under the wing are deflected outboard. That tendency is greater for the set of streamtracers which is one-half meter in from the wing tip, but the full vortex only arises with streamtracers at, or very near, the wing tip.



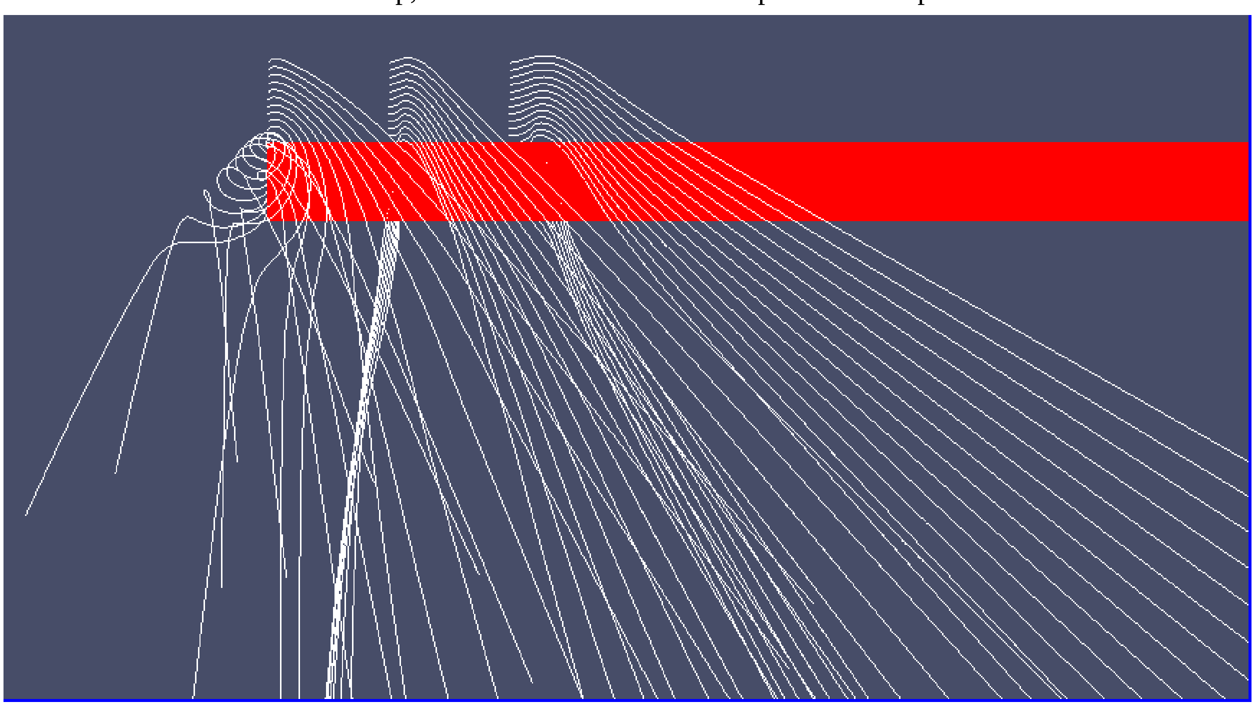
The following picture is a view down the wing from a vantage point to the left of the wing. Only the inboard set of streamtracers is shown. Seen in cross-section, the pattern of air flow depicted by this set of streamtracers seems close to what one would expect for a wing with infinite span. (The airfoil appears smeared out because paraFoam displays a perspective view rather than an orthogonal view.)



By way of contrast, the following picture shows the wing tip set of streamtracers from the same vantage point.



The following picture shows the airflow near the tip of a wing with a ten-meter span. The generating lines for these three streamtracers are exactly the same as those used for the six-meter wing above -- one-half meter upwind, extending 30 centimeters above and below the leading edge and located at spanwise locations one-meter in from the tip, one-half meter in from the tip and at the tip.



These streamtracers are extremely similar to the ones above for the six-meter span, adding weight to my assertion that the wing tip phenomenon is confined to the outer one meter of span.

Let me run through a couple a numerical examples, the results of which are summarized in the following table. In the earlier papers, we calculated that the lift, drag and nose-down moment for this kite section, at a 9° angle of attack in a 20 mph wind at 15,000 feet, were 10.6 N, 0.891 N and 3.80 N/m per meter of span, respectively, when modeled in two dimensions. The de-rating fractions for lift can be applied as follows:

Total wing span →	**4 meters**	**6 meters**	**8 meters**	**10 meters**
Unadjusted total lift (A)	4 × 10.6 N	6 × 10.6 N	8 × 10.6 N	10 × 10.6 N
De-rating fraction (B)	82.6%	89.7%	93.3%	95.3%
Adjusted total lift (A×B)	35.0 N	57.0 N	79.1 N	101.0 N

For drag, the de-rating fractions can be applied as follows:

Total wing span →	**4 meters**	**6 meters**	**8 meters**	**10 meters**
Unadjusted total drag (C)	4 × 0.891 N	6 × 0.891 N	8 × 0.891 N	10 × 8.891 N
De-rating fraction (D)	120.1%	115.4%	112.3%	110.0%
Adjusted total drag (C×D)	4.28 N	6.17 N	8.00 N	9.89 N

For the nose-down moment, the de-rating fractions can be applied as follows:

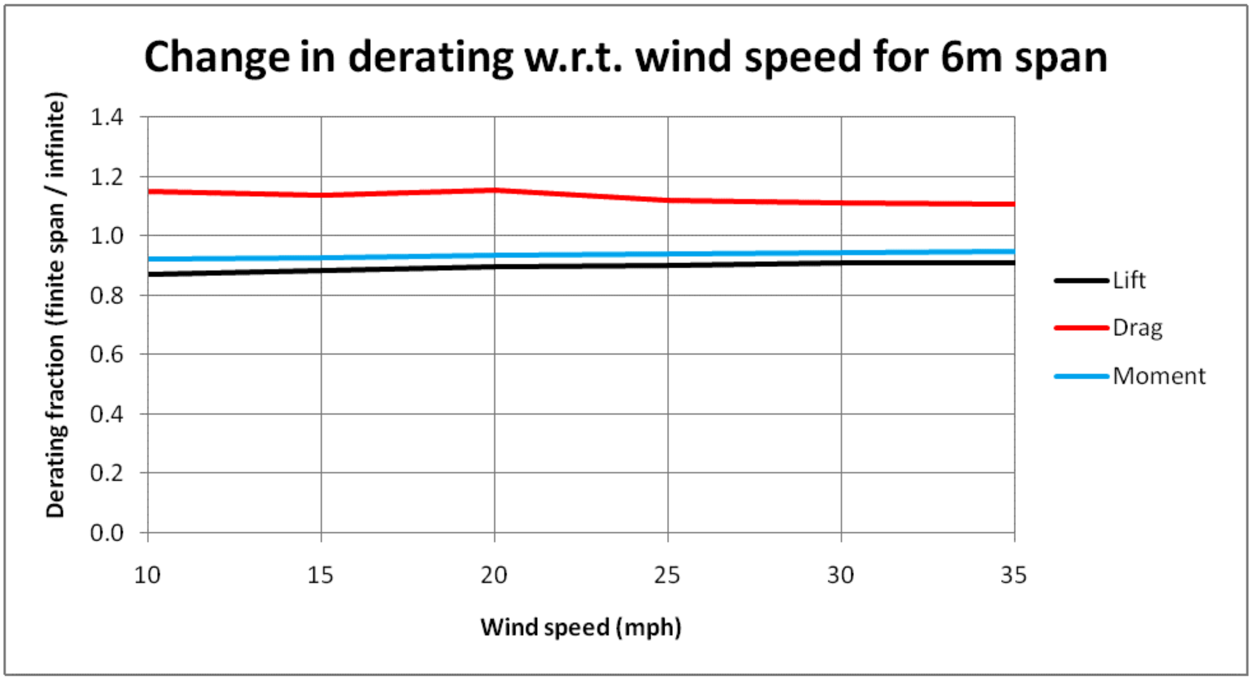
Total wing span →	**4 meters**	**6 meters**	**8 meters**	**10 meters**
Unadjusted total moment (E)	4 × 3.80 N	6 × 3.80 N	8 × 3.80 N	10 × 3.80 N
De-rating fraction (F)	89.3%	93.4%	95.5%	96.4%
Adjusted total moment (E×F)	13.6 Nm	21.3 Nm	29.0 Nm	101.0 Nm

The principal measure of efficiency of a wing is the lift-to-drag ratio. Since de-rating for finite span decreases the lift, but increases the drag, the lift-to-drag ratio is reduced "from both sides", from the two-dimensional ideal of 10.6 / 0.891 = 11.9, as follows:

Total wing span →	**4 meters**	**6 meters**	**8 meters**	**10 meters**
Adjusted total lift (A×B)	35.0 N	57.0 N	79.1 N	101.0 N
Adjusted total drag (C×D)	4.28 N	6.17 N	8.00 N	9.89 N
Adjusted L/D	8.18	9.24	9.89	10.2 N

Before moving on, I want to compare the results for infinite span obtained here with those obtained in the earlier papers. Using the `kOmegaSST` model in three dimensions, the lift and drag are 10.1 and 0.894 Newtons, respectively. Using the `Spalart-Allmaras` model in two dimensions (as in the earlier paprs), the lift and drag were 10.6 and 0.891 Newtons, respectively. These two completely different turbulence models give results which are comfortingly close.

I was also interested to examine how the de-rating fractions change with the wind speed. The runs shown in the following graph were all carried out with a six-meter span. The wind speed was varied over a range from 10 mph to 35 mph. In order to calculate the de-rating from infinite span to six meters, it was necessary to carry out two OpenFoam simulations for each wind speed, one using the infinite-span wind tunnel to calculate the base quantities and a second using the finite-span wind tunnel to calculate the finite-span results. The graph shows the de-rating fractions – the quotients obtined by dividing the finite-span result by the infinite-span result – for these wind speeds.



The de-rating fractions change only slightly with wind speed. To a very good approximation, the de-rating fractions at a 20 mph wind speed could be used for all practical wind speeds.

Before concluding, I want to mention the range of y^+ values observed. In the base case with the six meter wing, average y^+ values were near 4 on the upper surface near the leading edge, decreased to 0.25 at and around the trailing edge and increased to 1.5 on the lower surface near the leading edge. The average y^+ on the wing tip was 0.2.

Jim Hawley
September 2014

An e-mail setting out errors or omissions would be appreciated.

Appendix "A"

<u>Listing of "FiniteKiteSpan_6m.geo.txt" used in the base case</u>

```
// Membrane-surfaced kite with finite span
// Span (meters) = 6
Mesh.RandomFactor = 1e-8;
Geometry.AutoCoherence = 1;
Geometry.HighlightOrphans = 1;
Geometry.MatchGeomAndMesh = 1;
Geometry.Tolerance = 1e-10;
lcWingRoot = .00775;
lcWingTip = .002;
lcWT = .2;
HalfSpan = 3;
WTAhead = 3;
WTAstern = 4;
WTAbove = 3;
WTBelow = 3;
WTWidth = 6;
//
// Points around the airfoil on the wing root, clockwise
Point(1) = {.007603185, .0107885044, 0, lcWingRoot};
Point(2) = {.0153118257, .0126211632, 0, lcWingRoot};
Point(3) = {.0229994072, .0142043724, 0, lcWingRoot};
Point(4) = {.0306990611, .0157222297, 0, lcWingRoot};
Point(5) = {.0384095473, .01718112, 0, lcWingRoot};
Point(6) = {.0461298561, .0185849172, 0, lcWingRoot};
Point(7) = {.0538594913, .0199357669, 0, lcWingRoot};
Point(8) = {.0615981863, .021233961, 0, lcWingRoot};
Point(9) = {.0693458201, .0224787051, 0, lcWingRoot};
Point(10) = {.0771021557, .0236688415, 0, lcWingRoot};
Point(11) = {.0848669251, .024803342, 0, lcWingRoot};
Point(12) = {.0926397546, .02588155, 0, lcWingRoot};
Point(13) = {.1004202284, .026903179, 0, lcWingRoot};
Point(14) = {.1082079061, .0278682078, 0, lcWingRoot};
Point(15) = {.1160023786, .028776675, 0, lcWingRoot};
Point(16) = {.1238031778, .0296287514, 0, lcWingRoot};
Point(17) = {.1316098973, .0304245318, 0, lcWingRoot};
Point(18) = {.1394221276, .0311642491, 0, lcWingRoot};
Point(19) = {.1472394371, .031847827, 0, lcWingRoot};
Point(20) = {.1550613616, .0324756516, 0, lcWingRoot};
Point(21) = {.1628875651, .0330478958, 0, lcWingRoot};
Point(22) = {.1707176012, .0335645932, 0, lcWingRoot};
Point(23) = {.1785510977, .0340257737, 0, lcWingRoot};
Point(24) = {.1863876182, .0344317236, 0, lcWingRoot};
Point(25) = {.1942268078, .0347823894, 0, lcWingRoot};
Point(26) = {.2020682545, .0350779366, 0, lcWingRoot};
Point(27) = {.2099115172, .0353184339, 0, lcWingRoot};
Point(28) = {.217756277, .0355043505, 0, lcWingRoot};
Point(29) = {.2256021368, .035635526, 0, lcWingRoot};
Point(30) = {.2334487074, .0357120155, 0, lcWingRoot};
Point(31) = {.2412955629, .0357340585, 0, lcWingRoot};
Point(32) = {.2491424103, .0357019454, 0, lcWingRoot};
```

```
Point(33) = {.2569888189, .0356155058, 0, lcWingRoot};
Point(34) = {.2648344425, .0354749187, 0, lcWingRoot};
Point(35) = {.272678837, .0352805129, 0, lcWingRoot};
Point(36) = {.2805217341, .0350325871, 0, lcWingRoot};
Point(37) = {.2883627302, .0347311704, 0, lcWingRoot};
Point(38) = {.2962014753, .0343764898, 0, lcWingRoot};
Point(39) = {.30403761, .0339687441, 0, lcWingRoot};
Point(40) = {.3118708035, .0335082135, 0, lcWingRoot};
Point(41) = {.3197006823, .0329950307, 0, lcWingRoot};
Point(42) = {.3275269382, .0324295597, 0, lcWingRoot};
Point(43) = {.3353492382, .0318119316, 0, lcWingRoot};
Point(44) = {.3431672102, .0311423613, 0, lcWingRoot};
Point(45) = {.3509805555, .0304213095, 0, lcWingRoot};
Point(46) = {.358788988, .029649097, 0, lcWingRoot};
Point(47) = {.3665921818, .0288259313, 0, lcWingRoot};
Point(48) = {.3743898285, .0279522132, 0, lcWingRoot};
Point(49) = {.3821816386, .0270282936, 0, lcWingRoot};
Point(50) = {.3899673387, .0260545578, 0, lcWingRoot};
Point(51) = {.3977466608, .0250314212, 0, lcWingRoot};
Point(52) = {.4055193305, .0239591534, 0, lcWingRoot};
Point(53) = {.4132850704, .0228382036, 0, lcWingRoot};
Point(54) = {.4210436523, .0216689211, 0, lcWingRoot};
Point(55) = {.4287947896, .0204517184, 0, lcWingRoot};
Point(56) = {.4365382809, .0191870183, 0, lcWingRoot};
Point(57) = {.4442738875, .0178751933, 0, lcWingRoot};
Point(58) = {.452001371, .0165167843, 0, lcWingRoot};
Point(59) = {.459720574, .0151122739, 0, lcWingRoot};
Point(60) = {.4674312551, .0136620935, 0, lcWingRoot};
Point(61) = {.4751332518, .0121667949, 0, lcWingRoot};
Point(62) = {.4828264042, .0106268697, 0, lcWingRoot};
Point(63) = {.4905105258, .0090427256, 0, lcWingRoot};
Point(64) = {.4981854148, .0074148042, 0, lcWingRoot};
Point(65) = {.5058509699, .0057437852, 0, lcWingRoot};
Point(66) = {.5135070563, .0040302453, 0, lcWingRoot};
Point(67) = {.5211535623, .0022746569, 0, lcWingRoot};
Point(68) = {.5287903462, .0004774903, 0, lcWingRoot};
Point(69) = {.5364172833, -.0013607063, 0, lcWingRoot};
Point(70) = {.5440342939, -.0032393553, 0, lcWingRoot};
Point(71) = {.5516412693, -.00515791, 0, lcWingRoot};
Point(72) = {.5592381658, -.007115729, 0, lcWingRoot};
Point(73) = {.5668249072, -.0091122876, 0, lcWingRoot};
Point(74) = {.5744014293, -.011147056, 0, lcWingRoot};
Point(75) = {.5819676559, -.0132194398, 0, lcWingRoot};
Point(76) = {.5895236025, -.0153287899, 0, lcWingRoot};
Point(77) = {.5970692039, -.017474547, 0, lcWingRoot};
Point(78) = {.6046044589, -.0196561667, 0, lcWingRoot};
Point(79) = {.6121293287, -.0218730238, 0, lcWingRoot};
Point(80) = {.619643834, -.0241245048, 0, lcWingRoot};
Point(81) = {.6271480084, -.0264099405, 0, lcWingRoot};
Point(82) = {.6346418418, -.0287288136, 0, lcWingRoot};
Point(83) = {.6421253686, -.0310804244, 0, lcWingRoot};
Point(84) = {.6495986684, -.0334641374, 0, lcWingRoot};
Point(85) = {.6570617448, -.0358793701, 0, lcWingRoot};
Point(86) = {.6645146859, -.0383254728, 0, lcWingRoot};
```

```
Point(87) = {.6719575466, -.0408018205, 0, lcWingRoot};
Point(88) = {.6793904136, -.0433077817, 0, lcWingRoot};
Point(89) = {.6868133421, -.0458427793, 0, lcWingRoot};
Point(90) = {.6942264273, -.0484061367, 0, lcWingRoot};
Point(91) = {.7016298081, -.050997162, 0, lcWingRoot};
Point(92) = {.7090235598, -.0536153026, 0, lcWingRoot};
Point(93) = {.71640781, -.0562598948, 0, lcWingRoot};
Point(94) = {.7237826665, -.0589302795, 0, lcWingRoot};
Point(95) = {.7311483065, -.0616257679, 0, lcWingRoot};
Point(96) = {.7385048399, -.0643457909, 0, lcWingRoot};
Point(97) = {.7458523981, -.0670897132, 0, lcWingRoot};
Point(98) = {.7531911496, -.0698568517, 0, lcWingRoot};
Point(99) = {.7605212689, -.0726465196, 0, lcWingRoot};
Point(100) = {.7678429438, -.0754580557, 0, lcWingRoot};
Point(101) = {.7751563371, -.0782908153, 0, lcWingRoot};
Point(102) = {.7824616394, -.0811441158, 0, lcWingRoot};
Point(103) = {.789759068, -.0840172445, 0, lcWingRoot};
Point(104) = {.7970488365, -.0869095143, 0, lcWingRoot};
Point(105) = {.8043311511, -.0898202408, 0, lcWingRoot};
Point(106) = {.8116062409, -.0927487789, 0, lcWingRoot};
Point(107) = {.8188742848, -.0956944788, 0, lcWingRoot};
Point(108) = {.8261355551, -.0986565714, 0, lcWingRoot};
Point(109) = {.8333903201, -.1016342661, 0, lcWingRoot};
Point(110) = {.840638879, -.104626769, 0, lcWingRoot};
Point(111) = {.8478815203, -.1076332972, 0, lcWingRoot};
Point(112) = {.8551185101, -.110653124, 0, lcWingRoot};
Point(113) = {.8623501727, -.1136853722, 0, lcWingRoot};
Point(114) = {.8695768275, -.1167292296, 0, lcWingRoot};
Point(115) = {.8767988464, -.1197836849, 0, lcWingRoot};
Point(116) = {.8840166785, -.1228476564, 0, lcWingRoot};
Point(117) = {.8912307408, -.1259200813, 0, lcWingRoot};
Point(118) = {.8984415224, -.1289996928, 0, lcWingRoot};
Point(119) = {.9056496334, -.1320849352, 0, lcWingRoot};
Point(120) = {.9128558017, -.1351737867, 0, lcWingRoot};
Point(121) = {.912461712, -.1360928587, 0, lcWingRoot};
Point(122) = {.9052557938, -.1330041144, 0, lcWingRoot};
Point(123) = {.8980482683, -.1299191226, 0, lcWingRoot};
Point(124) = {.890838295, -.1268398564, 0, lcWingRoot};
Point(125) = {.8836252239, -.1237678538, 0, lcWingRoot};
Point(126) = {.8764085322, -.1207043666, 0, lcWingRoot};
Point(127) = {.8691877856, -.1176504496, 0, lcWingRoot};
Point(128) = {.8619625423, -.1146071871, 0, lcWingRoot};
Point(129) = {.8547324019, -.1115755775, 0, lcWingRoot};
Point(130) = {.8474970474, -.1085564334, 0, lcWingRoot};
Point(131) = {.8402561423, -.1055506265, 0, lcWingRoot};
Point(132) = {.8330094163, -.1025588807, 0, lcWingRoot};
Point(133) = {.8257565802, -.0995819784, 0, lcWingRoot};
Point(134) = {.8184973444, -.0966207164, 0, lcWingRoot};
Point(135) = {.8112314299, -.0936758801, 0, lcWingRoot};
Point(136) = {.8039585696, -.0907482403, 0, lcWingRoot};
Point(137) = {.7966785559, -.0878384343, 0, lcWingRoot};
Point(138) = {.7893911805, -.0849471148, 0, lcWingRoot};
Point(139) = {.7820962297, -.0820749626, 0, lcWingRoot};
Point(140) = {.7747934931, -.0792226652, 0, lcWingRoot};
```

```
Point(141) = {.7674827565, -.0763909357, 0, lcWingRoot};
Point(142) = {.7601638254, -.0735804543, 0, lcWingRoot};
Point(143) = {.7528365269, -.0707918612, 0, lcWingRoot};
Point(144) = {.745500685, -.0680258211, 0, lcWingRoot};
Point(145) = {.7381561233, -.0652830191, 0, lcWingRoot};
Point(146) = {.730802673, -.0625641375, 0, lcWingRoot};
Point(147) = {.723440192, -.0598698067, 0, lcWingRoot};
Point(148) = {.7160685676, -.0572005939, 0, lcWingRoot};
Point(149) = {.7086876472, -.0545571958, 0, lcWingRoot};
Point(150) = {.7012973021, -.0519402631, 0, lcWingRoot};
Point(151) = {.6938974068, -.0493504595, 0, lcWingRoot};
Point(152) = {.6864878825, -.0467883352, 0, lcWingRoot};
Point(153) = {.679068607, -.0442545871, 0, lcWingRoot};
Point(154) = {.6716394749, -.0417498872, 0, lcWingRoot};
Point(155) = {.6642004221, -.0392748084, 0, lcWingRoot};
Point(156) = {.6567513713, -.0368299848, 0, lcWingRoot};
Point(157) = {.649292258, -.0344160369, 0, lcWingRoot};
Point(158) = {.6418230142, -.03203362, 0, lcWingRoot};
Point(159) = {.6343436069, -.0296833061, 0, lcWingRoot};
Point(160) = {.6268539879, -.0273657397, 0, lcWingRoot};
Point(161) = {.6193541034, -.025081613, 0, lcWingRoot};
Point(162) = {.6118439615, -.022831442, 0, lcWingRoot};
Point(163) = {.6043235325, -.020615896, 0, lcWingRoot};
Point(164) = {.596792811, -.0184355917, 0, lcWingRoot};
Point(165) = {.5892518011, -.0162911433, 0, lcWingRoot};
Point(166) = {.581700533, -.0141831022, 0, lcWingRoot};
Point(167) = {.5741390451, -.0121120195, 0, lcWingRoot};
Point(168) = {.5665673551, -.010078552, 0, lcWingRoot};
Point(169) = {.5589855093, -.008083285, 0, lcWingRoot};
Point(170) = {.5513935768, -.0061267487, 0, lcWingRoot};
Point(171) = {.5437916361, -.0042094673, 0, lcWingRoot};
Point(172) = {.5361797441, -.0023320843, 0, lcWingRoot};
Point(173) = {.5285579922, -.0004951409, 0, lcWingRoot};
Point(174) = {.5209264711, .0013007834, 0, lcWingRoot};
Point(175) = {.5132852887, .0030551458, 0, lcWingRoot};
Point(176) = {.5056345782, .0047674785, 0, lcWingRoot};
Point(177) = {.4979744783, .0064373044, 0, lcWingRoot};
Point(178) = {.4903051142, .0080640499, 0, lcWingRoot};
Point(179) = {.4826265985, .0096470342, 0, lcWingRoot};
Point(180) = {.4749391041, .0111858225, 0, lcWingRoot};
Point(181) = {.4672428164, .0126800085, 0, lcWingRoot};
Point(182) = {.4595379106, .0141290984, 0, lcWingRoot};
Point(183) = {.4518245597, .0155325395, 0, lcWingRoot};
Point(184) = {.4441029681, .0168899083, 0, lcWingRoot};
Point(185) = {.4363733363, .0182007155, 0, lcWingRoot};
Point(186) = {.4286358656, .0194644275, 0, lcWingRoot};
Point(187) = {.420890792, .0206806733, 0, lcWingRoot};
Point(188) = {.4131383493, .0218490257, 0, lcWingRoot};
Point(189) = {.4053787815, .0229690797, 0, lcWingRoot};
Point(190) = {.397612347, .0240404824, 0, lcWingRoot};
Point(191) = {.3898392976, .0250627889, 0, lcWingRoot};
Point(192) = {.3820599152, .0260357295, 0, lcWingRoot};
Point(193) = {.3742744686, .0269588895, 0, lcWingRoot};
Point(194) = {.3664832375, .0278318834, 0, lcWingRoot};
```

```
Point(195) = {.3586865122, .0286543615, 0, lcWingRoot};
Point(196) = {.350884584, .0294259254, 0, lcWingRoot};
Point(197) = {.3430777694, .0301463691, 0, lcWingRoot};
Point(198) = {.3352663785, .0308153704, 0, lcWingRoot};
Point(199) = {.3274507162, .0314324688, 0, lcWingRoot};
Point(200) = {.319631114, .0319974535, 0, lcWingRoot};
Point(201) = {.3118079125, .0325101931, 0, lcWingRoot};
Point(202) = {.3039814405, .0329703229, 0, lcWingRoot};
Point(203) = {.2961520463, .0333777122, 0, lcWingRoot};
Point(204) = {.2883200732, .0337320806, 0, lcWingRoot};
Point(205) = {.2804858734, .0340332303, 0, lcWingRoot};
Point(206) = {.2726498017, .0342809345, 0, lcWingRoot};
Point(207) = {.2648122223, .0344751656, 0, lcWingRoot};
Point(208) = {.2569734952, .0346156232, 0, lcWingRoot};
Point(209) = {.2491339854, .0347019809, 0, lcWingRoot};
Point(210) = {.2412940657, .0347340596, 0, lcWingRoot};
Point(211) = {.2334541113, .0347120301, 0, lcWingRoot};
Point(212) = {.2256144987, .0346356024, 0, lcWingRoot};
Point(213) = {.2177756092, .0345045374, 0, lcWingRoot};
Point(214) = {.209937825, .03431878, 0, lcWingRoot};
Point(215) = {.2021015232, .0340784901, 0, lcWingRoot};
Point(216) = {.1942671012, .0337832015, 0, lcWingRoot};
Point(217) = {.1864349488, .0334328443, 0, lcWingRoot};
Point(218) = {.1786054624, .0330272525, 0, lcWingRoot};
Point(219) = {.1707790297, .0325664817, 0, lcWingRoot};
Point(220) = {.1629560598, .0320502443, 0, lcWingRoot};
Point(221) = {.1551369499, .0314785125, 0, lcWingRoot};
Point(222) = {.1473220987, .0308512493, 0, lcWingRoot};
Point(223) = {.1395119182, .0301682884, 0, lcWingRoot};
Point(224) = {.1317068454, .0294292424, 0, lcWingRoot};
Point(225) = {.1239072791, .0286341847, 0, lcWingRoot};
Point(226) = {.116113651, .027782885, 0, lcWingRoot};
Point(227) = {.1083263827, .026875251, 0, lcWingRoot};
Point(228) = {.1005459087, .0259111082, 0, lcWingRoot};
Point(229) = {.0927726506, .02489042, 0, lcWingRoot};
Point(230) = {.0850070242, .0238132045, 0, lcWingRoot};
Point(231) = {.0772494077, .0226797425, 0, lcWingRoot};
Point(232) = {.0695001181, .0214906807, 0, lcWingRoot};
Point(233) = {.0617593994, .0202470414, 0, lcWingRoot};
Point(234) = {.0540274533, .0189499735, 0, lcWingRoot};
Point(235) = {.046304518, .0176002887, 0, lcWingRoot};
Point(236) = {.0385910118, .0161977225, 0, lcWingRoot};
Point(237) = {.030887706, .0147401843, 0, lcWingRoot};
Point(238) = {.0231957624, .0132238395, 0, lcWingRoot};
Point(239) = {.015516939, .0116424249, 0, lcWingRoot};
//
// Points around the airfoil on the wing tip, clockwise
Point(240) = {.007603185, .0107885044, HalfSpan, lcWingTip};
Point(241) = {.0153118257, .0126211632, HalfSpan, lcWingTip};
Point(242) = {.0229994072, .0142043724, HalfSpan, lcWingTip};
Point(243) = {.0306990611, .0157222297, HalfSpan, lcWingTip};
Point(244) = {.0384095473, .01718112, HalfSpan, lcWingTip};
Point(245) = {.0461298561, .0185849172, HalfSpan, lcWingTip};
Point(246) = {.0538594913, .0199357669, HalfSpan, lcWingTip};
```

```
Point(247) = {.0615981863, .021233961, HalfSpan, lcWingTip};
Point(248) = {.0693458201, .0224787051, HalfSpan, lcWingTip};
Point(249) = {.0771021557, .0236688415, HalfSpan, lcWingTip};
Point(250) = {.0848669251, .024803342, HalfSpan, lcWingTip};
Point(251) = {.0926397546, .02588155, HalfSpan, lcWingTip};
Point(252) = {.1004202284, .026903179, HalfSpan, lcWingTip};
Point(253) = {.1082079061, .0278682078, HalfSpan, lcWingTip};
Point(254) = {.1160023786, .028776675, HalfSpan, lcWingTip};
Point(255) = {.1238031778, .0296287514, HalfSpan, lcWingTip};
Point(256) = {.1316098973, .0304245318, HalfSpan, lcWingTip};
Point(257) = {.1394221276, .0311642491, HalfSpan, lcWingTip};
Point(258) = {.1472394371, .031847827, HalfSpan, lcWingTip};
Point(259) = {.1550613616, .0324756516, HalfSpan, lcWingTip};
Point(260) = {.1628875651, .0330478958, HalfSpan, lcWingTip};
Point(261) = {.1707176012, .0335645932, HalfSpan, lcWingTip};
Point(262) = {.1785510977, .0340257737, HalfSpan, lcWingTip};
Point(263) = {.1863876182, .0344317236, HalfSpan, lcWingTip};
Point(264) = {.1942268078, .0347823894, HalfSpan, lcWingTip};
Point(265) = {.2020682545, .0350779366, HalfSpan, lcWingTip};
Point(266) = {.2099115172, .0353184339, HalfSpan, lcWingTip};
Point(267) = {.217756277, .0355043505, HalfSpan, lcWingTip};
Point(268) = {.2256021368, .035635526, HalfSpan, lcWingTip};
Point(269) = {.2334487074, .0357120155, HalfSpan, lcWingTip};
Point(270) = {.2412955629, .0357340585, HalfSpan, lcWingTip};
Point(271) = {.2491424103, .0357019454, HalfSpan, lcWingTip};
Point(272) = {.2569888189, .0356155058, HalfSpan, lcWingTip};
Point(273) = {.2648344425, .0354749187, HalfSpan, lcWingTip};
Point(274) = {.272678837, .0352805129, HalfSpan, lcWingTip};
Point(275) = {.2805217341, .0350325871, HalfSpan, lcWingTip};
Point(276) = {.2883627302, .0347311704, HalfSpan, lcWingTip};
Point(277) = {.2962014753, .0343764898, HalfSpan, lcWingTip};
Point(278) = {.30403761, .0339687441, HalfSpan, lcWingTip};
Point(279) = {.3118708035, .0335082135, HalfSpan, lcWingTip};
Point(280) = {.3197006823, .0329950307, HalfSpan, lcWingTip};
Point(281) = {.3275269382, .0324295597, HalfSpan, lcWingTip};
Point(282) = {.3353492382, .0318119316, HalfSpan, lcWingTip};
Point(283) = {.3431672102, .0311423613, HalfSpan, lcWingTip};
Point(284) = {.3509805555, .0304213095, HalfSpan, lcWingTip};
Point(285) = {.358788988, .029649097, HalfSpan, lcWingTip};
Point(286) = {.3665921818, .0288259313, HalfSpan, lcWingTip};
Point(287) = {.3743898285, .0279522132, HalfSpan, lcWingTip};
Point(288) = {.3821816386, .0270282936, HalfSpan, lcWingTip};
Point(289) = {.3899673387, .0260545578, HalfSpan, lcWingTip};
Point(290) = {.3977466608, .0250314212, HalfSpan, lcWingTip};
Point(291) = {.4055193305, .0239591534, HalfSpan, lcWingTip};
Point(292) = {.4132850704, .0228382036, HalfSpan, lcWingTip};
Point(293) = {.4210436523, .0216689211, HalfSpan, lcWingTip};
Point(294) = {.4287947896, .0204517184, HalfSpan, lcWingTip};
Point(295) = {.4365382809, .0191870183, HalfSpan, lcWingTip};
Point(296) = {.4442738875, .0178751933, HalfSpan, lcWingTip};
Point(297) = {.452001371, .0165167843, HalfSpan, lcWingTip};
Point(298) = {.459720574, .0151122739, HalfSpan, lcWingTip};
Point(299) = {.4674312551, .0136620935, HalfSpan, lcWingTip};
Point(300) = {.4751332518, .0121667949, HalfSpan, lcWingTip};
```

```
Point(301) = {.4828264042, .0106268697, HalfSpan, lcWingTip};
Point(302) = {.4905105258, .0090427256, HalfSpan, lcWingTip};
Point(303) = {.4981854148, .0074148042, HalfSpan, lcWingTip};
Point(304) = {.5058509699, .0057437852, HalfSpan, lcWingTip};
Point(305) = {.5135070563, .0040302453, HalfSpan, lcWingTip};
Point(306) = {.5211535623, .0022746569, HalfSpan, lcWingTip};
Point(307) = {.5287903462, .0004774903, HalfSpan, lcWingTip};
Point(308) = {.5364172833, -.0013607063, HalfSpan, lcWingTip};
Point(309) = {.5440342939, -.0032393553, HalfSpan, lcWingTip};
Point(310) = {.5516412693, -.00515791, HalfSpan, lcWingTip};
Point(311) = {.5592381658, -.007115729, HalfSpan, lcWingTip};
Point(312) = {.5668249072, -.0091122876, HalfSpan, lcWingTip};
Point(313) = {.5744014293, -.011147056, HalfSpan, lcWingTip};
Point(314) = {.5819676559, -.0132194398, HalfSpan, lcWingTip};
Point(315) = {.5895236025, -.0153287899, HalfSpan, lcWingTip};
Point(316) = {.5970692039, -.017474547, HalfSpan, lcWingTip};
Point(317) = {.6046044589, -.0196561667, HalfSpan, lcWingTip};
Point(318) = {.6121293287, -.0218730238, HalfSpan, lcWingTip};
Point(319) = {.619643834, -.0241245048, HalfSpan, lcWingTip};
Point(320) = {.6271480084, -.0264099405, HalfSpan, lcWingTip};
Point(321) = {.6346418418, -.0287288136, HalfSpan, lcWingTip};
Point(322) = {.6421253686, -.0310804244, HalfSpan, lcWingTip};
Point(323) = {.6495986684, -.0334641374, HalfSpan, lcWingTip};
Point(324) = {.6570617448, -.0358793701, HalfSpan, lcWingTip};
Point(325) = {.6645146859, -.0383254728, HalfSpan, lcWingTip};
Point(326) = {.6719575466, -.0408018205, HalfSpan, lcWingTip};
Point(327) = {.6793904136, -.0433077817, HalfSpan, lcWingTip};
Point(328) = {.6868133421, -.0458427793, HalfSpan, lcWingTip};
Point(329) = {.6942264273, -.0484061367, HalfSpan, lcWingTip};
Point(330) = {.7016298081, -.050997162, HalfSpan, lcWingTip};
Point(331) = {.7090235598, -.0536153026, HalfSpan, lcWingTip};
Point(332) = {.71640781, -.0562598948, HalfSpan, lcWingTip};
Point(333) = {.7237826665, -.0589302795, HalfSpan, lcWingTip};
Point(334) = {.7311483065, -.0616257679, HalfSpan, lcWingTip};
Point(335) = {.7385048399, -.0643457909, HalfSpan, lcWingTip};
Point(336) = {.7458523981, -.0670897132, HalfSpan, lcWingTip};
Point(337) = {.7531911496, -.0698568517, HalfSpan, lcWingTip};
Point(338) = {.7605212689, -.0726465196, HalfSpan, lcWingTip};
Point(339) = {.7678429438, -.0754580557, HalfSpan, lcWingTip};
Point(340) = {.7751563371, -.0782908153, HalfSpan, lcWingTip};
Point(341) = {.7824616394, -.0811441158, HalfSpan, lcWingTip};
Point(342) = {.789759068, -.0840172445, HalfSpan, lcWingTip};
Point(343) = {.7970488365, -.0869095143, HalfSpan, lcWingTip};
Point(344) = {.8043311511, -.0898202408, HalfSpan, lcWingTip};
Point(345) = {.8116062409, -.0927487789, HalfSpan, lcWingTip};
Point(346) = {.8188742848, -.0956944788, HalfSpan, lcWingTip};
Point(347) = {.8261355551, -.0986565714, HalfSpan, lcWingTip};
Point(348) = {.8333903201, -.1016342661, HalfSpan, lcWingTip};
Point(349) = {.840638879, -.104626769, HalfSpan, lcWingTip};
Point(350) = {.8478815203, -.1076332972, HalfSpan, lcWingTip};
Point(351) = {.8551185101, -.110653124, HalfSpan, lcWingTip};
Point(352) = {.8623501727, -.1136853722, HalfSpan, lcWingTip};
Point(353) = {.8695768275, -.1167292296, HalfSpan, lcWingTip};
Point(354) = {.8767988464, -.1197836849, HalfSpan, lcWingTip};
```

```
Point(355) = {.8840166785, -.1228476564, HalfSpan, lcWingTip};
Point(356) = {.8912307408, -.1259200813, HalfSpan, lcWingTip};
Point(357) = {.8984415224, -.1289996928, HalfSpan, lcWingTip};
Point(358) = {.9056496334, -.1320849352, HalfSpan, lcWingTip};
Point(359) = {.9128558017, -.1351737867, HalfSpan, lcWingTip};
Point(360) = {.912461712, -.1360928587, HalfSpan, lcWingTip};
Point(361) = {.9052557938, -.1330041144, HalfSpan, lcWingTip};
Point(362) = {.8980482683, -.1299191226, HalfSpan, lcWingTip};
Point(363) = {.890838295, -.1268398564, HalfSpan, lcWingTip};
Point(364) = {.8836252239, -.1237678538, HalfSpan, lcWingTip};
Point(365) = {.8764085322, -.1207043666, HalfSpan, lcWingTip};
Point(366) = {.8691877856, -.1176504496, HalfSpan, lcWingTip};
Point(367) = {.8619625423, -.1146071871, HalfSpan, lcWingTip};
Point(368) = {.8547324019, -.1115755775, HalfSpan, lcWingTip};
Point(369) = {.8474970474, -.1085564334, HalfSpan, lcWingTip};
Point(370) = {.8402561423, -.1055506265, HalfSpan, lcWingTip};
Point(371) = {.8330094163, -.1025588807, HalfSpan, lcWingTip};
Point(372) = {.8257565802, -.0995819784, HalfSpan, lcWingTip};
Point(373) = {.8184973444, -.0966207164, HalfSpan, lcWingTip};
Point(374) = {.8112314299, -.0936758801, HalfSpan, lcWingTip};
Point(375) = {.8039585696, -.0907482403, HalfSpan, lcWingTip};
Point(376) = {.7966785559, -.0878384343, HalfSpan, lcWingTip};
Point(377) = {.7893911805, -.0849471148, HalfSpan, lcWingTip};
Point(378) = {.7820962297, -.0820749626, HalfSpan, lcWingTip};
Point(379) = {.7747934931, -.0792226652, HalfSpan, lcWingTip};
Point(380) = {.7674827565, -.0763909357, HalfSpan, lcWingTip};
Point(381) = {.7601638254, -.0735804543, HalfSpan, lcWingTip};
Point(382) = {.7528365269, -.0707918612, HalfSpan, lcWingTip};
Point(383) = {.745500685, -.0680258211, HalfSpan, lcWingTip};
Point(384) = {.7381561233, -.0652830191, HalfSpan, lcWingTip};
Point(385) = {.730802673, -.0625641375, HalfSpan, lcWingTip};
Point(386) = {.723440192, -.0598698067, HalfSpan, lcWingTip};
Point(387) = {.7160685676, -.0572005939, HalfSpan, lcWingTip};
Point(388) = {.7086876472, -.0545571958, HalfSpan, lcWingTip};
Point(389) = {.7012973021, -.0519402631, HalfSpan, lcWingTip};
Point(390) = {.6938974068, -.0493504595, HalfSpan, lcWingTip};
Point(391) = {.6864878825, -.0467883352, HalfSpan, lcWingTip};
Point(392) = {.679068607, -.0442545871, HalfSpan, lcWingTip};
Point(393) = {.6716394749, -.0417498872, HalfSpan, lcWingTip};
Point(394) = {.6642004221, -.0392748084, HalfSpan, lcWingTip};
Point(395) = {.6567513713, -.0368299848, HalfSpan, lcWingTip};
Point(396) = {.649292258, -.0344160369, HalfSpan, lcWingTip};
Point(397) = {.6418230142, -.03203362, HalfSpan, lcWingTip};
Point(398) = {.6343436069, -.0296833061, HalfSpan, lcWingTip};
Point(399) = {.6268539879, -.0273657397, HalfSpan, lcWingTip};
Point(400) = {.6193541034, -.025081613, HalfSpan, lcWingTip};
Point(401) = {.6118439615, -.022831442, HalfSpan, lcWingTip};
Point(402) = {.6043235325, -.020615896, HalfSpan, lcWingTip};
Point(403) = {.596792811, -.0184355917, HalfSpan, lcWingTip};
Point(404) = {.5892518011, -.0162911433, HalfSpan, lcWingTip};
Point(405) = {.581700533, -.0141831022, HalfSpan, lcWingTip};
Point(406) = {.5741390451, -.0121120195, HalfSpan, lcWingTip};
Point(407) = {.5665673551, -.010078552, HalfSpan, lcWingTip};
Point(408) = {.5589855093, -.008083285, HalfSpan, lcWingTip};
```

```
Point(409) = {.5513935768, -.0061267487, HalfSpan, lcWingTip};
Point(410) = {.5437916361, -.0042094673, HalfSpan, lcWingTip};
Point(411) = {.5361797441, -.0023320843, HalfSpan, lcWingTip};
Point(412) = {.5285579922, -.0004951409, HalfSpan, lcWingTip};
Point(413) = {.5209264711, .0013007834, HalfSpan, lcWingTip};
Point(414) = {.5132852887, .0030551458, HalfSpan, lcWingTip};
Point(415) = {.5056345782, .0047674785, HalfSpan, lcWingTip};
Point(416) = {.4979744783, .0064373044, HalfSpan, lcWingTip};
Point(417) = {.4903051142, .0080640499, HalfSpan, lcWingTip};
Point(418) = {.4826265985, .0096470342, HalfSpan, lcWingTip};
Point(419) = {.4749391041, .0111858225, HalfSpan, lcWingTip};
Point(420) = {.4672428164, .0126800085, HalfSpan, lcWingTip};
Point(421) = {.4595379106, .0141290984, HalfSpan, lcWingTip};
Point(422) = {.4518245597, .0155325395, HalfSpan, lcWingTip};
Point(423) = {.4441029681, .0168899083, HalfSpan, lcWingTip};
Point(424) = {.4363733363, .0182007155, HalfSpan, lcWingTip};
Point(425) = {.4286358656, .0194644275, HalfSpan, lcWingTip};
Point(426) = {.420890792, .0206806733, HalfSpan, lcWingTip};
Point(427) = {.4131383493, .0218490257, HalfSpan, lcWingTip};
Point(428) = {.4053787815, .0229690797, HalfSpan, lcWingTip};
Point(429) = {.397612347, .0240404824, HalfSpan, lcWingTip};
Point(430) = {.3898392976, .0250627889, HalfSpan, lcWingTip};
Point(431) = {.3820599152, .0260357295, HalfSpan, lcWingTip};
Point(432) = {.3742744686, .0269588895, HalfSpan, lcWingTip};
Point(433) = {.3664832375, .0278318834, HalfSpan, lcWingTip};
Point(434) = {.3586865122, .0286543615, HalfSpan, lcWingTip};
Point(435) = {.350884584, .0294259254, HalfSpan, lcWingTip};
Point(436) = {.3430777694, .0301463691, HalfSpan, lcWingTip};
Point(437) = {.3352663785, .0308153704, HalfSpan, lcWingTip};
Point(438) = {.3274507162, .0314324688, HalfSpan, lcWingTip};
Point(439) = {.319631114, .0319974535, HalfSpan, lcWingTip};
Point(440) = {.3118079125, .0325101931, HalfSpan, lcWingTip};
Point(441) = {.3039814405, .0329703229, HalfSpan, lcWingTip};
Point(442) = {.2961520463, .0333777122, HalfSpan, lcWingTip};
Point(443) = {.2883200732, .0337320806, HalfSpan, lcWingTip};
Point(444) = {.2804858734, .0340332303, HalfSpan, lcWingTip};
Point(445) = {.2726498017, .0342809345, HalfSpan, lcWingTip};
Point(446) = {.2648122223, .0344751656, HalfSpan, lcWingTip};
Point(447) = {.2569734952, .0346156232, HalfSpan, lcWingTip};
Point(448) = {.2491339854, .0347019809, HalfSpan, lcWingTip};
Point(449) = {.2412940657, .0347340596, HalfSpan, lcWingTip};
Point(450) = {.2334541113, .0347120301, HalfSpan, lcWingTip};
Point(451) = {.2256144987, .0346356024, HalfSpan, lcWingTip};
Point(452) = {.2177756092, .0345045374, HalfSpan, lcWingTip};
Point(453) = {.209937825, .03431878, HalfSpan, lcWingTip};
Point(454) = {.2021015232, .0340784901, HalfSpan, lcWingTip};
Point(455) = {.1942671012, .0337832015, HalfSpan, lcWingTip};
Point(456) = {.1864349488, .0334328443, HalfSpan, lcWingTip};
Point(457) = {.1786054624, .0330272525, HalfSpan, lcWingTip};
Point(458) = {.1707790297, .0325664817, HalfSpan, lcWingTip};
Point(459) = {.1629560598, .0320502443, HalfSpan, lcWingTip};
Point(460) = {.1551369499, .0314785125, HalfSpan, lcWingTip};
Point(461) = {.1473220987, .0308512493, HalfSpan, lcWingTip};
Point(462) = {.1395119182, .0301682884, HalfSpan, lcWingTip};
```

```
Point(463) = {.1317068454, .0294292424, HalfSpan, lcWingTip};
Point(464) = {.1239072791, .0286341847, HalfSpan, lcWingTip};
Point(465) = {.116113651, .027782885, HalfSpan, lcWingTip};
Point(466) = {.1083263827, .026875251, HalfSpan, lcWingTip};
Point(467) = {.1005459087, .0259111082, HalfSpan, lcWingTip};
Point(468) = {.0927726506, .02489042, HalfSpan, lcWingTip};
Point(469) = {.0850070242, .0238132045, HalfSpan, lcWingTip};
Point(470) = {.0772494077, .0226797425, HalfSpan, lcWingTip};
Point(471) = {.0695001181, .0214906807, HalfSpan, lcWingTip};
Point(472) = {.0617593994, .0202470414, HalfSpan, lcWingTip};
Point(473) = {.0540274533, .0189499735, HalfSpan, lcWingTip};
Point(474) = {.046304518, .0176002887, HalfSpan, lcWingTip};
Point(475) = {.0385910118, .0161977225, HalfSpan, lcWingTip};
Point(476) = {.030887706, .0147401843, HalfSpan, lcWingTip};
Point(477) = {.0231957624, .0132238395, HalfSpan, lcWingTip};
Point(478) = {.015516939, .0116424249, HalfSpan, lcWingTip};
//
// Lines around the wing's profile on the wing root, clockwise
Line(5001) = {1, 2};
Line(5002) = {2, 3};
Line(5003) = {3, 4};
Line(5004) = {4, 5};
Line(5005) = {5, 6};
Line(5006) = {6, 7};
Line(5007) = {7, 8};
Line(5008) = {8, 9};
Line(5009) = {9, 10};
Line(5010) = {10, 11};
Line(5011) = {11, 12};
Line(5012) = {12, 13};
Line(5013) = {13, 14};
Line(5014) = {14, 15};
Line(5015) = {15, 16};
Line(5016) = {16, 17};
Line(5017) = {17, 18};
Line(5018) = {18, 19};
Line(5019) = {19, 20};
Line(5020) = {20, 21};
Line(5021) = {21, 22};
Line(5022) = {22, 23};
Line(5023) = {23, 24};
Line(5024) = {24, 25};
Line(5025) = {25, 26};
Line(5026) = {26, 27};
Line(5027) = {27, 28};
Line(5028) = {28, 29};
Line(5029) = {29, 30};
Line(5030) = {30, 31};
Line(5031) = {31, 32};
Line(5032) = {32, 33};
Line(5033) = {33, 34};
Line(5034) = {34, 35};
Line(5035) = {35, 36};
Line(5036) = {36, 37};
```

```
Line(5037) = {37, 38};
Line(5038) = {38, 39};
Line(5039) = {39, 40};
Line(5040) = {40, 41};
Line(5041) = {41, 42};
Line(5042) = {42, 43};
Line(5043) = {43, 44};
Line(5044) = {44, 45};
Line(5045) = {45, 46};
Line(5046) = {46, 47};
Line(5047) = {47, 48};
Line(5048) = {48, 49};
Line(5049) = {49, 50};
Line(5050) = {50, 51};
Line(5051) = {51, 52};
Line(5052) = {52, 53};
Line(5053) = {53, 54};
Line(5054) = {54, 55};
Line(5055) = {55, 56};
Line(5056) = {56, 57};
Line(5057) = {57, 58};
Line(5058) = {58, 59};
Line(5059) = {59, 60};
Line(5060) = {60, 61};
Line(5061) = {61, 62};
Line(5062) = {62, 63};
Line(5063) = {63, 64};
Line(5064) = {64, 65};
Line(5065) = {65, 66};
Line(5066) = {66, 67};
Line(5067) = {67, 68};
Line(5068) = {68, 69};
Line(5069) = {69, 70};
Line(5070) = {70, 71};
Line(5071) = {71, 72};
Line(5072) = {72, 73};
Line(5073) = {73, 74};
Line(5074) = {74, 75};
Line(5075) = {75, 76};
Line(5076) = {76, 77};
Line(5077) = {77, 78};
Line(5078) = {78, 79};
Line(5079) = {79, 80};
Line(5080) = {80, 81};
Line(5081) = {81, 82};
Line(5082) = {82, 83};
Line(5083) = {83, 84};
Line(5084) = {84, 85};
Line(5085) = {85, 86};
Line(5086) = {86, 87};
Line(5087) = {87, 88};
Line(5088) = {88, 89};
Line(5089) = {89, 90};
Line(5090) = {90, 91};
```

```
Line(5091) = {91, 92};
Line(5092) = {92, 93};
Line(5093) = {93, 94};
Line(5094) = {94, 95};
Line(5095) = {95, 96};
Line(5096) = {96, 97};
Line(5097) = {97, 98};
Line(5098) = {98, 99};
Line(5099) = {99, 100};
Line(5100) = {100, 101};
Line(5101) = {101, 102};
Line(5102) = {102, 103};
Line(5103) = {103, 104};
Line(5104) = {104, 105};
Line(5105) = {105, 106};
Line(5106) = {106, 107};
Line(5107) = {107, 108};
Line(5108) = {108, 109};
Line(5109) = {109, 110};
Line(5110) = {110, 111};
Line(5111) = {111, 112};
Line(5112) = {112, 113};
Line(5113) = {113, 114};
Line(5114) = {114, 115};
Line(5115) = {115, 116};
Line(5116) = {116, 117};
Line(5117) = {117, 118};
Line(5118) = {118, 119};
Line(5119) = {119, 120};
Line(5120) = {120, 121};
Line(5121) = {121, 122};
Line(5122) = {122, 123};
Line(5123) = {123, 124};
Line(5124) = {124, 125};
Line(5125) = {125, 126};
Line(5126) = {126, 127};
Line(5127) = {127, 128};
Line(5128) = {128, 129};
Line(5129) = {129, 130};
Line(5130) = {130, 131};
Line(5131) = {131, 132};
Line(5132) = {132, 133};
Line(5133) = {133, 134};
Line(5134) = {134, 135};
Line(5135) = {135, 136};
Line(5136) = {136, 137};
Line(5137) = {137, 138};
Line(5138) = {138, 139};
Line(5139) = {139, 140};
Line(5140) = {140, 141};
Line(5141) = {141, 142};
Line(5142) = {142, 143};
Line(5143) = {143, 144};
Line(5144) = {144, 145};
```

```
Line(5145) = {145, 146};
Line(5146) = {146, 147};
Line(5147) = {147, 148};
Line(5148) = {148, 149};
Line(5149) = {149, 150};
Line(5150) = {150, 151};
Line(5151) = {151, 152};
Line(5152) = {152, 153};
Line(5153) = {153, 154};
Line(5154) = {154, 155};
Line(5155) = {155, 156};
Line(5156) = {156, 157};
Line(5157) = {157, 158};
Line(5158) = {158, 159};
Line(5159) = {159, 160};
Line(5160) = {160, 161};
Line(5161) = {161, 162};
Line(5162) = {162, 163};
Line(5163) = {163, 164};
Line(5164) = {164, 165};
Line(5165) = {165, 166};
Line(5166) = {166, 167};
Line(5167) = {167, 168};
Line(5168) = {168, 169};
Line(5169) = {169, 170};
Line(5170) = {170, 171};
Line(5171) = {171, 172};
Line(5172) = {172, 173};
Line(5173) = {173, 174};
Line(5174) = {174, 175};
Line(5175) = {175, 176};
Line(5176) = {176, 177};
Line(5177) = {177, 178};
Line(5178) = {178, 179};
Line(5179) = {179, 180};
Line(5180) = {180, 181};
Line(5181) = {181, 182};
Line(5182) = {182, 183};
Line(5183) = {183, 184};
Line(5184) = {184, 185};
Line(5185) = {185, 186};
Line(5186) = {186, 187};
Line(5187) = {187, 188};
Line(5188) = {188, 189};
Line(5189) = {189, 190};
Line(5190) = {190, 191};
Line(5191) = {191, 192};
Line(5192) = {192, 193};
Line(5193) = {193, 194};
Line(5194) = {194, 195};
Line(5195) = {195, 196};
Line(5196) = {196, 197};
Line(5197) = {197, 198};
Line(5198) = {198, 199};
```

```
Line(5199) = {199, 200};
Line(5200) = {200, 201};
Line(5201) = {201, 202};
Line(5202) = {202, 203};
Line(5203) = {203, 204};
Line(5204) = {204, 205};
Line(5205) = {205, 206};
Line(5206) = {206, 207};
Line(5207) = {207, 208};
Line(5208) = {208, 209};
Line(5209) = {209, 210};
Line(5210) = {210, 211};
Line(5211) = {211, 212};
Line(5212) = {212, 213};
Line(5213) = {213, 214};
Line(5214) = {214, 215};
Line(5215) = {215, 216};
Line(5216) = {216, 217};
Line(5217) = {217, 218};
Line(5218) = {218, 219};
Line(5219) = {219, 220};
Line(5220) = {220, 221};
Line(5221) = {221, 222};
Line(5222) = {222, 223};
Line(5223) = {223, 224};
Line(5224) = {224, 225};
Line(5225) = {225, 226};
Line(5226) = {226, 227};
Line(5227) = {227, 228};
Line(5228) = {228, 229};
Line(5229) = {229, 230};
Line(5230) = {230, 231};
Line(5231) = {231, 232};
Line(5232) = {232, 233};
Line(5233) = {233, 234};
Line(5234) = {234, 235};
Line(5235) = {235, 236};
Line(5236) = {236, 237};
Line(5237) = {237, 238};
Line(5238) = {238, 239};
Line(5239) = {239, 1};
//
// Lines around the wing's profile on the wing tip, clockwise
Line(5240) = {240, 241};
Line(5241) = {241, 242};
Line(5242) = {242, 243};
Line(5243) = {243, 244};
Line(5244) = {244, 245};
Line(5245) = {245, 246};
Line(5246) = {246, 247};
Line(5247) = {247, 248};
Line(5248) = {248, 249};
Line(5249) = {249, 250};
Line(5250) = {250, 251};
```

```
Line(5251) = {251, 252};
Line(5252) = {252, 253};
Line(5253) = {253, 254};
Line(5254) = {254, 255};
Line(5255) = {255, 256};
Line(5256) = {256, 257};
Line(5257) = {257, 258};
Line(5258) = {258, 259};
Line(5259) = {259, 260};
Line(5260) = {260, 261};
Line(5261) = {261, 262};
Line(5262) = {262, 263};
Line(5263) = {263, 264};
Line(5264) = {264, 265};
Line(5265) = {265, 266};
Line(5266) = {266, 267};
Line(5267) = {267, 268};
Line(5268) = {268, 269};
Line(5269) = {269, 270};
Line(5270) = {270, 271};
Line(5271) = {271, 272};
Line(5272) = {272, 273};
Line(5273) = {273, 274};
Line(5274) = {274, 275};
Line(5275) = {275, 276};
Line(5276) = {276, 277};
Line(5277) = {277, 278};
Line(5278) = {278, 279};
Line(5279) = {279, 280};
Line(5280) = {280, 281};
Line(5281) = {281, 282};
Line(5282) = {282, 283};
Line(5283) = {283, 284};
Line(5284) = {284, 285};
Line(5285) = {285, 286};
Line(5286) = {286, 287};
Line(5287) = {287, 288};
Line(5288) = {288, 289};
Line(5289) = {289, 290};
Line(5290) = {290, 291};
Line(5291) = {291, 292};
Line(5292) = {292, 293};
Line(5293) = {293, 294};
Line(5294) = {294, 295};
Line(5295) = {295, 296};
Line(5296) = {296, 297};
Line(5297) = {297, 298};
Line(5298) = {298, 299};
Line(5299) = {299, 300};
Line(5300) = {300, 301};
Line(5301) = {301, 302};
Line(5302) = {302, 303};
Line(5303) = {303, 304};
Line(5304) = {304, 305};
```

```
Line(5305) = {305, 306};
Line(5306) = {306, 307};
Line(5307) = {307, 308};
Line(5308) = {308, 309};
Line(5309) = {309, 310};
Line(5310) = {310, 311};
Line(5311) = {311, 312};
Line(5312) = {312, 313};
Line(5313) = {313, 314};
Line(5314) = {314, 315};
Line(5315) = {315, 316};
Line(5316) = {316, 317};
Line(5317) = {317, 318};
Line(5318) = {318, 319};
Line(5319) = {319, 320};
Line(5320) = {320, 321};
Line(5321) = {321, 322};
Line(5322) = {322, 323};
Line(5323) = {323, 324};
Line(5324) = {324, 325};
Line(5325) = {325, 326};
Line(5326) = {326, 327};
Line(5327) = {327, 328};
Line(5328) = {328, 329};
Line(5329) = {329, 330};
Line(5330) = {330, 331};
Line(5331) = {331, 332};
Line(5332) = {332, 333};
Line(5333) = {333, 334};
Line(5334) = {334, 335};
Line(5335) = {335, 336};
Line(5336) = {336, 337};
Line(5337) = {337, 338};
Line(5338) = {338, 339};
Line(5339) = {339, 340};
Line(5340) = {340, 341};
Line(5341) = {341, 342};
Line(5342) = {342, 343};
Line(5343) = {343, 344};
Line(5344) = {344, 345};
Line(5345) = {345, 346};
Line(5346) = {346, 347};
Line(5347) = {347, 348};
Line(5348) = {348, 349};
Line(5349) = {349, 350};
Line(5350) = {350, 351};
Line(5351) = {351, 352};
Line(5352) = {352, 353};
Line(5353) = {353, 354};
Line(5354) = {354, 355};
Line(5355) = {355, 356};
Line(5356) = {356, 357};
Line(5357) = {357, 358};
Line(5358) = {358, 359};
```

```
Line(5359) = {359, 360};
Line(5360) = {360, 361};
Line(5361) = {361, 362};
Line(5362) = {362, 363};
Line(5363) = {363, 364};
Line(5364) = {364, 365};
Line(5365) = {365, 366};
Line(5366) = {366, 367};
Line(5367) = {367, 368};
Line(5368) = {368, 369};
Line(5369) = {369, 370};
Line(5370) = {370, 371};
Line(5371) = {371, 372};
Line(5372) = {372, 373};
Line(5373) = {373, 374};
Line(5374) = {374, 375};
Line(5375) = {375, 376};
Line(5376) = {376, 377};
Line(5377) = {377, 378};
Line(5378) = {378, 379};
Line(5379) = {379, 380};
Line(5380) = {380, 381};
Line(5381) = {381, 382};
Line(5382) = {382, 383};
Line(5383) = {383, 384};
Line(5384) = {384, 385};
Line(5385) = {385, 386};
Line(5386) = {386, 387};
Line(5387) = {387, 388};
Line(5388) = {388, 389};
Line(5389) = {389, 390};
Line(5390) = {390, 391};
Line(5391) = {391, 392};
Line(5392) = {392, 393};
Line(5393) = {393, 394};
Line(5394) = {394, 395};
Line(5395) = {395, 396};
Line(5396) = {396, 397};
Line(5397) = {397, 398};
Line(5398) = {398, 399};
Line(5399) = {399, 400};
Line(5400) = {400, 401};
Line(5401) = {401, 402};
Line(5402) = {402, 403};
Line(5403) = {403, 404};
Line(5404) = {404, 405};
Line(5405) = {405, 406};
Line(5406) = {406, 407};
Line(5407) = {407, 408};
Line(5408) = {408, 409};
Line(5409) = {409, 410};
Line(5410) = {410, 411};
Line(5411) = {411, 412};
Line(5412) = {412, 413};
```

```
Line(5413) = {413, 414};
Line(5414) = {414, 415};
Line(5415) = {415, 416};
Line(5416) = {416, 417};
Line(5417) = {417, 418};
Line(5418) = {418, 419};
Line(5419) = {419, 420};
Line(5420) = {420, 421};
Line(5421) = {421, 422};
Line(5422) = {422, 423};
Line(5423) = {423, 424};
Line(5424) = {424, 425};
Line(5425) = {425, 426};
Line(5426) = {426, 427};
Line(5427) = {427, 428};
Line(5428) = {428, 429};
Line(5429) = {429, 430};
Line(5430) = {430, 431};
Line(5431) = {431, 432};
Line(5432) = {432, 433};
Line(5433) = {433, 434};
Line(5434) = {434, 435};
Line(5435) = {435, 436};
Line(5436) = {436, 437};
Line(5437) = {437, 438};
Line(5438) = {438, 439};
Line(5439) = {439, 440};
Line(5440) = {440, 441};
Line(5441) = {441, 442};
Line(5442) = {442, 443};
Line(5443) = {443, 444};
Line(5444) = {444, 445};
Line(5445) = {445, 446};
Line(5446) = {446, 447};
Line(5447) = {447, 448};
Line(5448) = {448, 449};
Line(5449) = {449, 450};
Line(5450) = {450, 451};
Line(5451) = {451, 452};
Line(5452) = {452, 453};
Line(5453) = {453, 454};
Line(5454) = {454, 455};
Line(5455) = {455, 456};
Line(5456) = {456, 457};
Line(5457) = {457, 458};
Line(5458) = {458, 459};
Line(5459) = {459, 460};
Line(5460) = {460, 461};
Line(5461) = {461, 462};
Line(5462) = {462, 463};
Line(5463) = {463, 464};
Line(5464) = {464, 465};
Line(5465) = {465, 466};
Line(5466) = {466, 467};
```

```
Line(5467) = {467, 468};
Line(5468) = {468, 469};
Line(5469) = {469, 470};
Line(5470) = {470, 471};
Line(5471) = {471, 472};
Line(5472) = {472, 473};
Line(5473) = {473, 474};
Line(5474) = {474, 475};
Line(5475) = {475, 476};
Line(5476) = {476, 477};
Line(5477) = {477, 478};
Line(5478) = {478, 240};
//
// Spanwise lines across the wing, from tip to root
Line(5479) = {240, 1};
Line(5480) = {241, 2};
Line(5481) = {242, 3};
Line(5482) = {243, 4};
Line(5483) = {244, 5};
Line(5484) = {245, 6};
Line(5485) = {246, 7};
Line(5486) = {247, 8};
Line(5487) = {248, 9};
Line(5488) = {249, 10};
Line(5489) = {250, 11};
Line(5490) = {251, 12};
Line(5491) = {252, 13};
Line(5492) = {253, 14};
Line(5493) = {254, 15};
Line(5494) = {255, 16};
Line(5495) = {256, 17};
Line(5496) = {257, 18};
Line(5497) = {258, 19};
Line(5498) = {259, 20};
Line(5499) = {260, 21};
Line(5500) = {261, 22};
Line(5501) = {262, 23};
Line(5502) = {263, 24};
Line(5503) = {264, 25};
Line(5504) = {265, 26};
Line(5505) = {266, 27};
Line(5506) = {267, 28};
Line(5507) = {268, 29};
Line(5508) = {269, 30};
Line(5509) = {270, 31};
Line(5510) = {271, 32};
Line(5511) = {272, 33};
Line(5512) = {273, 34};
Line(5513) = {274, 35};
Line(5514) = {275, 36};
Line(5515) = {276, 37};
Line(5516) = {277, 38};
Line(5517) = {278, 39};
Line(5518) = {279, 40};
```

```
Line(5519) = {280, 41};
Line(5520) = {281, 42};
Line(5521) = {282, 43};
Line(5522) = {283, 44};
Line(5523) = {284, 45};
Line(5524) = {285, 46};
Line(5525) = {286, 47};
Line(5526) = {287, 48};
Line(5527) = {288, 49};
Line(5528) = {289, 50};
Line(5529) = {290, 51};
Line(5530) = {291, 52};
Line(5531) = {292, 53};
Line(5532) = {293, 54};
Line(5533) = {294, 55};
Line(5534) = {295, 56};
Line(5535) = {296, 57};
Line(5536) = {297, 58};
Line(5537) = {298, 59};
Line(5538) = {299, 60};
Line(5539) = {300, 61};
Line(5540) = {301, 62};
Line(5541) = {302, 63};
Line(5542) = {303, 64};
Line(5543) = {304, 65};
Line(5544) = {305, 66};
Line(5545) = {306, 67};
Line(5546) = {307, 68};
Line(5547) = {308, 69};
Line(5548) = {309, 70};
Line(5549) = {310, 71};
Line(5550) = {311, 72};
Line(5551) = {312, 73};
Line(5552) = {313, 74};
Line(5553) = {314, 75};
Line(5554) = {315, 76};
Line(5555) = {316, 77};
Line(5556) = {317, 78};
Line(5557) = {318, 79};
Line(5558) = {319, 80};
Line(5559) = {320, 81};
Line(5560) = {321, 82};
Line(5561) = {322, 83};
Line(5562) = {323, 84};
Line(5563) = {324, 85};
Line(5564) = {325, 86};
Line(5565) = {326, 87};
Line(5566) = {327, 88};
Line(5567) = {328, 89};
Line(5568) = {329, 90};
Line(5569) = {330, 91};
Line(5570) = {331, 92};
Line(5571) = {332, 93};
Line(5572) = {333, 94};
```

```
Line(5573) = {334, 95};
Line(5574) = {335, 96};
Line(5575) = {336, 97};
Line(5576) = {337, 98};
Line(5577) = {338, 99};
Line(5578) = {339, 100};
Line(5579) = {340, 101};
Line(5580) = {341, 102};
Line(5581) = {342, 103};
Line(5582) = {343, 104};
Line(5583) = {344, 105};
Line(5584) = {345, 106};
Line(5585) = {346, 107};
Line(5586) = {347, 108};
Line(5587) = {348, 109};
Line(5588) = {349, 110};
Line(5589) = {350, 111};
Line(5590) = {351, 112};
Line(5591) = {352, 113};
Line(5592) = {353, 114};
Line(5593) = {354, 115};
Line(5594) = {355, 116};
Line(5595) = {356, 117};
Line(5596) = {357, 118};
Line(5597) = {358, 119};
Line(5598) = {359, 120};
Line(5599) = {360, 121};
Line(5600) = {361, 122};
Line(5601) = {362, 123};
Line(5602) = {363, 124};
Line(5603) = {364, 125};
Line(5604) = {365, 126};
Line(5605) = {366, 127};
Line(5606) = {367, 128};
Line(5607) = {368, 129};
Line(5608) = {369, 130};
Line(5609) = {370, 131};
Line(5610) = {371, 132};
Line(5611) = {372, 133};
Line(5612) = {373, 134};
Line(5613) = {374, 135};
Line(5614) = {375, 136};
Line(5615) = {376, 137};
Line(5616) = {377, 138};
Line(5617) = {378, 139};
Line(5618) = {379, 140};
Line(5619) = {380, 141};
Line(5620) = {381, 142};
Line(5621) = {382, 143};
Line(5622) = {383, 144};
Line(5623) = {384, 145};
Line(5624) = {385, 146};
Line(5625) = {386, 147};
Line(5626) = {387, 148};
```

```
Line(5627) = {388, 149};
Line(5628) = {389, 150};
Line(5629) = {390, 151};
Line(5630) = {391, 152};
Line(5631) = {392, 153};
Line(5632) = {393, 154};
Line(5633) = {394, 155};
Line(5634) = {395, 156};
Line(5635) = {396, 157};
Line(5636) = {397, 158};
Line(5637) = {398, 159};
Line(5638) = {399, 160};
Line(5639) = {400, 161};
Line(5640) = {401, 162};
Line(5641) = {402, 163};
Line(5642) = {403, 164};
Line(5643) = {404, 165};
Line(5644) = {405, 166};
Line(5645) = {406, 167};
Line(5646) = {407, 168};
Line(5647) = {408, 169};
Line(5648) = {409, 170};
Line(5649) = {410, 171};
Line(5650) = {411, 172};
Line(5651) = {412, 173};
Line(5652) = {413, 174};
Line(5653) = {414, 175};
Line(5654) = {415, 176};
Line(5655) = {416, 177};
Line(5656) = {417, 178};
Line(5657) = {418, 179};
Line(5658) = {419, 180};
Line(5659) = {420, 181};
Line(5660) = {421, 182};
Line(5661) = {422, 183};
Line(5662) = {423, 184};
Line(5663) = {424, 185};
Line(5664) = {425, 186};
Line(5665) = {426, 187};
Line(5666) = {427, 188};
Line(5667) = {428, 189};
Line(5668) = {429, 190};
Line(5669) = {430, 191};
Line(5670) = {431, 192};
Line(5671) = {432, 193};
Line(5672) = {433, 194};
Line(5673) = {434, 195};
Line(5674) = {435, 196};
Line(5675) = {436, 197};
Line(5676) = {437, 198};
Line(5677) = {438, 199};
Line(5678) = {439, 200};
Line(5679) = {440, 201};
Line(5680) = {441, 202};
```

```
Line(5681) = {442, 203};
Line(5682) = {443, 204};
Line(5683) = {444, 205};
Line(5684) = {445, 206};
Line(5685) = {446, 207};
Line(5686) = {447, 208};
Line(5687) = {448, 209};
Line(5688) = {449, 210};
Line(5689) = {450, 211};
Line(5690) = {451, 212};
Line(5691) = {452, 213};
Line(5692) = {453, 214};
Line(5693) = {454, 215};
Line(5694) = {455, 216};
Line(5695) = {456, 217};
Line(5696) = {457, 218};
Line(5697) = {458, 219};
Line(5698) = {459, 220};
Line(5699) = {460, 221};
Line(5700) = {461, 222};
Line(5701) = {462, 223};
Line(5702) = {463, 224};
Line(5703) = {464, 225};
Line(5704) = {465, 226};
Line(5705) = {466, 227};
Line(5706) = {467, 228};
Line(5707) = {468, 229};
Line(5708) = {469, 230};
Line(5709) = {470, 231};
Line(5710) = {471, 232};
Line(5711) = {472, 233};
Line(5712) = {473, 234};
Line(5713) = {474, 235};
Line(5714) = {475, 236};
Line(5715) = {476, 237};
Line(5716) = {477, 238};
Line(5717) = {478, 239};
//
// Line Loops on curved surface of wing
Line Loop(1001) = {5240, 5480, -5001, -5479};
Line Loop(1002) = {5241, 5481, -5002, -5480};
Line Loop(1003) = {5242, 5482, -5003, -5481};
Line Loop(1004) = {5243, 5483, -5004, -5482};
Line Loop(1005) = {5244, 5484, -5005, -5483};
Line Loop(1006) = {5245, 5485, -5006, -5484};
Line Loop(1007) = {5246, 5486, -5007, -5485};
Line Loop(1008) = {5247, 5487, -5008, -5486};
Line Loop(1009) = {5248, 5488, -5009, -5487};
Line Loop(1010) = {5249, 5489, -5010, -5488};
Line Loop(1011) = {5250, 5490, -5011, -5489};
Line Loop(1012) = {5251, 5491, -5012, -5490};
Line Loop(1013) = {5252, 5492, -5013, -5491};
Line Loop(1014) = {5253, 5493, -5014, -5492};
Line Loop(1015) = {5254, 5494, -5015, -5493};
```

```
Line Loop(1016) = {5255, 5495, -5016, -5494};
Line Loop(1017) = {5256, 5496, -5017, -5495};
Line Loop(1018) = {5257, 5497, -5018, -5496};
Line Loop(1019) = {5258, 5498, -5019, -5497};
Line Loop(1020) = {5259, 5499, -5020, -5498};
Line Loop(1021) = {5260, 5500, -5021, -5499};
Line Loop(1022) = {5261, 5501, -5022, -5500};
Line Loop(1023) = {5262, 5502, -5023, -5501};
Line Loop(1024) = {5263, 5503, -5024, -5502};
Line Loop(1025) = {5264, 5504, -5025, -5503};
Line Loop(1026) = {5265, 5505, -5026, -5504};
Line Loop(1027) = {5266, 5506, -5027, -5505};
Line Loop(1028) = {5267, 5507, -5028, -5506};
Line Loop(1029) = {5268, 5508, -5029, -5507};
Line Loop(1030) = {5269, 5509, -5030, -5508};
Line Loop(1031) = {5270, 5510, -5031, -5509};
Line Loop(1032) = {5271, 5511, -5032, -5510};
Line Loop(1033) = {5272, 5512, -5033, -5511};
Line Loop(1034) = {5273, 5513, -5034, -5512};
Line Loop(1035) = {5274, 5514, -5035, -5513};
Line Loop(1036) = {5275, 5515, -5036, -5514};
Line Loop(1037) = {5276, 5516, -5037, -5515};
Line Loop(1038) = {5277, 5517, -5038, -5516};
Line Loop(1039) = {5278, 5518, -5039, -5517};
Line Loop(1040) = {5279, 5519, -5040, -5518};
Line Loop(1041) = {5280, 5520, -5041, -5519};
Line Loop(1042) = {5281, 5521, -5042, -5520};
Line Loop(1043) = {5282, 5522, -5043, -5521};
Line Loop(1044) = {5283, 5523, -5044, -5522};
Line Loop(1045) = {5284, 5524, -5045, -5523};
Line Loop(1046) = {5285, 5525, -5046, -5524};
Line Loop(1047) = {5286, 5526, -5047, -5525};
Line Loop(1048) = {5287, 5527, -5048, -5526};
Line Loop(1049) = {5288, 5528, -5049, -5527};
Line Loop(1050) = {5289, 5529, -5050, -5528};
Line Loop(1051) = {5290, 5530, -5051, -5529};
Line Loop(1052) = {5291, 5531, -5052, -5530};
Line Loop(1053) = {5292, 5532, -5053, -5531};
Line Loop(1054) = {5293, 5533, -5054, -5532};
Line Loop(1055) = {5294, 5534, -5055, -5533};
Line Loop(1056) = {5295, 5535, -5056, -5534};
Line Loop(1057) = {5296, 5536, -5057, -5535};
Line Loop(1058) = {5297, 5537, -5058, -5536};
Line Loop(1059) = {5298, 5538, -5059, -5537};
Line Loop(1060) = {5299, 5539, -5060, -5538};
Line Loop(1061) = {5300, 5540, -5061, -5539};
Line Loop(1062) = {5301, 5541, -5062, -5540};
Line Loop(1063) = {5302, 5542, -5063, -5541};
Line Loop(1064) = {5303, 5543, -5064, -5542};
Line Loop(1065) = {5304, 5544, -5065, -5543};
Line Loop(1066) = {5305, 5545, -5066, -5544};
Line Loop(1067) = {5306, 5546, -5067, -5545};
Line Loop(1068) = {5307, 5547, -5068, -5546};
Line Loop(1069) = {5308, 5548, -5069, -5547};
```

```
Line Loop(1070) = {5309, 5549, -5070, -5548};
Line Loop(1071) = {5310, 5550, -5071, -5549};
Line Loop(1072) = {5311, 5551, -5072, -5550};
Line Loop(1073) = {5312, 5552, -5073, -5551};
Line Loop(1074) = {5313, 5553, -5074, -5552};
Line Loop(1075) = {5314, 5554, -5075, -5553};
Line Loop(1076) = {5315, 5555, -5076, -5554};
Line Loop(1077) = {5316, 5556, -5077, -5555};
Line Loop(1078) = {5317, 5557, -5078, -5556};
Line Loop(1079) = {5318, 5558, -5079, -5557};
Line Loop(1080) = {5319, 5559, -5080, -5558};
Line Loop(1081) = {5320, 5560, -5081, -5559};
Line Loop(1082) = {5321, 5561, -5082, -5560};
Line Loop(1083) = {5322, 5562, -5083, -5561};
Line Loop(1084) = {5323, 5563, -5084, -5562};
Line Loop(1085) = {5324, 5564, -5085, -5563};
Line Loop(1086) = {5325, 5565, -5086, -5564};
Line Loop(1087) = {5326, 5566, -5087, -5565};
Line Loop(1088) = {5327, 5567, -5088, -5566};
Line Loop(1089) = {5328, 5568, -5089, -5567};
Line Loop(1090) = {5329, 5569, -5090, -5568};
Line Loop(1091) = {5330, 5570, -5091, -5569};
Line Loop(1092) = {5331, 5571, -5092, -5570};
Line Loop(1093) = {5332, 5572, -5093, -5571};
Line Loop(1094) = {5333, 5573, -5094, -5572};
Line Loop(1095) = {5334, 5574, -5095, -5573};
Line Loop(1096) = {5335, 5575, -5096, -5574};
Line Loop(1097) = {5336, 5576, -5097, -5575};
Line Loop(1098) = {5337, 5577, -5098, -5576};
Line Loop(1099) = {5338, 5578, -5099, -5577};
Line Loop(1100) = {5339, 5579, -5100, -5578};
Line Loop(1101) = {5340, 5580, -5101, -5579};
Line Loop(1102) = {5341, 5581, -5102, -5580};
Line Loop(1103) = {5342, 5582, -5103, -5581};
Line Loop(1104) = {5343, 5583, -5104, -5582};
Line Loop(1105) = {5344, 5584, -5105, -5583};
Line Loop(1106) = {5345, 5585, -5106, -5584};
Line Loop(1107) = {5346, 5586, -5107, -5585};
Line Loop(1108) = {5347, 5587, -5108, -5586};
Line Loop(1109) = {5348, 5588, -5109, -5587};
Line Loop(1110) = {5349, 5589, -5110, -5588};
Line Loop(1111) = {5350, 5590, -5111, -5589};
Line Loop(1112) = {5351, 5591, -5112, -5590};
Line Loop(1113) = {5352, 5592, -5113, -5591};
Line Loop(1114) = {5353, 5593, -5114, -5592};
Line Loop(1115) = {5354, 5594, -5115, -5593};
Line Loop(1116) = {5355, 5595, -5116, -5594};
Line Loop(1117) = {5356, 5596, -5117, -5595};
Line Loop(1118) = {5357, 5597, -5118, -5596};
Line Loop(1119) = {5358, 5598, -5119, -5597};
Line Loop(1120) = {5359, 5599, -5120, -5598};
Line Loop(1121) = {5360, 5600, -5121, -5599};
Line Loop(1122) = {5361, 5601, -5122, -5600};
Line Loop(1123) = {5362, 5602, -5123, -5601};
```

```
Line Loop(1124) = {5363, 5603, -5124, -5602};
Line Loop(1125) = {5364, 5604, -5125, -5603};
Line Loop(1126) = {5365, 5605, -5126, -5604};
Line Loop(1127) = {5366, 5606, -5127, -5605};
Line Loop(1128) = {5367, 5607, -5128, -5606};
Line Loop(1129) = {5368, 5608, -5129, -5607};
Line Loop(1130) = {5369, 5609, -5130, -5608};
Line Loop(1131) = {5370, 5610, -5131, -5609};
Line Loop(1132) = {5371, 5611, -5132, -5610};
Line Loop(1133) = {5372, 5612, -5133, -5611};
Line Loop(1134) = {5373, 5613, -5134, -5612};
Line Loop(1135) = {5374, 5614, -5135, -5613};
Line Loop(1136) = {5375, 5615, -5136, -5614};
Line Loop(1137) = {5376, 5616, -5137, -5615};
Line Loop(1138) = {5377, 5617, -5138, -5616};
Line Loop(1139) = {5378, 5618, -5139, -5617};
Line Loop(1140) = {5379, 5619, -5140, -5618};
Line Loop(1141) = {5380, 5620, -5141, -5619};
Line Loop(1142) = {5381, 5621, -5142, -5620};
Line Loop(1143) = {5382, 5622, -5143, -5621};
Line Loop(1144) = {5383, 5623, -5144, -5622};
Line Loop(1145) = {5384, 5624, -5145, -5623};
Line Loop(1146) = {5385, 5625, -5146, -5624};
Line Loop(1147) = {5386, 5626, -5147, -5625};
Line Loop(1148) = {5387, 5627, -5148, -5626};
Line Loop(1149) = {5388, 5628, -5149, -5627};
Line Loop(1150) = {5389, 5629, -5150, -5628};
Line Loop(1151) = {5390, 5630, -5151, -5629};
Line Loop(1152) = {5391, 5631, -5152, -5630};
Line Loop(1153) = {5392, 5632, -5153, -5631};
Line Loop(1154) = {5393, 5633, -5154, -5632};
Line Loop(1155) = {5394, 5634, -5155, -5633};
Line Loop(1156) = {5395, 5635, -5156, -5634};
Line Loop(1157) = {5396, 5636, -5157, -5635};
Line Loop(1158) = {5397, 5637, -5158, -5636};
Line Loop(1159) = {5398, 5638, -5159, -5637};
Line Loop(1160) = {5399, 5639, -5160, -5638};
Line Loop(1161) = {5400, 5640, -5161, -5639};
Line Loop(1162) = {5401, 5641, -5162, -5640};
Line Loop(1163) = {5402, 5642, -5163, -5641};
Line Loop(1164) = {5403, 5643, -5164, -5642};
Line Loop(1165) = {5404, 5644, -5165, -5643};
Line Loop(1166) = {5405, 5645, -5166, -5644};
Line Loop(1167) = {5406, 5646, -5167, -5645};
Line Loop(1168) = {5407, 5647, -5168, -5646};
Line Loop(1169) = {5408, 5648, -5169, -5647};
Line Loop(1170) = {5409, 5649, -5170, -5648};
Line Loop(1171) = {5410, 5650, -5171, -5649};
Line Loop(1172) = {5411, 5651, -5172, -5650};
Line Loop(1173) = {5412, 5652, -5173, -5651};
Line Loop(1174) = {5413, 5653, -5174, -5652};
Line Loop(1175) = {5414, 5654, -5175, -5653};
Line Loop(1176) = {5415, 5655, -5176, -5654};
Line Loop(1177) = {5416, 5656, -5177, -5655};
```

```
Line Loop(1178) = {5417, 5657, -5178, -5656};
Line Loop(1179) = {5418, 5658, -5179, -5657};
Line Loop(1180) = {5419, 5659, -5180, -5658};
Line Loop(1181) = {5420, 5660, -5181, -5659};
Line Loop(1182) = {5421, 5661, -5182, -5660};
Line Loop(1183) = {5422, 5662, -5183, -5661};
Line Loop(1184) = {5423, 5663, -5184, -5662};
Line Loop(1185) = {5424, 5664, -5185, -5663};
Line Loop(1186) = {5425, 5665, -5186, -5664};
Line Loop(1187) = {5426, 5666, -5187, -5665};
Line Loop(1188) = {5427, 5667, -5188, -5666};
Line Loop(1189) = {5428, 5668, -5189, -5667};
Line Loop(1190) = {5429, 5669, -5190, -5668};
Line Loop(1191) = {5430, 5670, -5191, -5669};
Line Loop(1192) = {5431, 5671, -5192, -5670};
Line Loop(1193) = {5432, 5672, -5193, -5671};
Line Loop(1194) = {5433, 5673, -5194, -5672};
Line Loop(1195) = {5434, 5674, -5195, -5673};
Line Loop(1196) = {5435, 5675, -5196, -5674};
Line Loop(1197) = {5436, 5676, -5197, -5675};
Line Loop(1198) = {5437, 5677, -5198, -5676};
Line Loop(1199) = {5438, 5678, -5199, -5677};
Line Loop(1200) = {5439, 5679, -5200, -5678};
Line Loop(1201) = {5440, 5680, -5201, -5679};
Line Loop(1202) = {5441, 5681, -5202, -5680};
Line Loop(1203) = {5442, 5682, -5203, -5681};
Line Loop(1204) = {5443, 5683, -5204, -5682};
Line Loop(1205) = {5444, 5684, -5205, -5683};
Line Loop(1206) = {5445, 5685, -5206, -5684};
Line Loop(1207) = {5446, 5686, -5207, -5685};
Line Loop(1208) = {5447, 5687, -5208, -5686};
Line Loop(1209) = {5448, 5688, -5209, -5687};
Line Loop(1210) = {5449, 5689, -5210, -5688};
Line Loop(1211) = {5450, 5690, -5211, -5689};
Line Loop(1212) = {5451, 5691, -5212, -5690};
Line Loop(1213) = {5452, 5692, -5213, -5691};
Line Loop(1214) = {5453, 5693, -5214, -5692};
Line Loop(1215) = {5454, 5694, -5215, -5693};
Line Loop(1216) = {5455, 5695, -5216, -5694};
Line Loop(1217) = {5456, 5696, -5217, -5695};
Line Loop(1218) = {5457, 5697, -5218, -5696};
Line Loop(1219) = {5458, 5698, -5219, -5697};
Line Loop(1220) = {5459, 5699, -5220, -5698};
Line Loop(1221) = {5460, 5700, -5221, -5699};
Line Loop(1222) = {5461, 5701, -5222, -5700};
Line Loop(1223) = {5462, 5702, -5223, -5701};
Line Loop(1224) = {5463, 5703, -5224, -5702};
Line Loop(1225) = {5464, 5704, -5225, -5703};
Line Loop(1226) = {5465, 5705, -5226, -5704};
Line Loop(1227) = {5466, 5706, -5227, -5705};
Line Loop(1228) = {5467, 5707, -5228, -5706};
Line Loop(1229) = {5468, 5708, -5229, -5707};
Line Loop(1230) = {5469, 5709, -5230, -5708};
Line Loop(1231) = {5470, 5710, -5231, -5709};
```

```
Line Loop(1232) = {5471, 5711, -5232, -5710};
Line Loop(1233) = {5472, 5712, -5233, -5711};
Line Loop(1234) = {5473, 5713, -5234, -5712};
Line Loop(1235) = {5474, 5714, -5235, -5713};
Line Loop(1236) = {5475, 5715, -5236, -5714};
Line Loop(1237) = {5476, 5716, -5237, -5715};
Line Loop(1238) = {5477, 5717, -5238, -5716};
Line Loop(1239) = {5478, 5479, -5239, -5717};
//
// Line Loop on wing root, directed outwards
Line Loop(1240) = {
5001,
5002,
5003,
5004,
5005,
5006,
5007,
5008,
5009,
5010,
5011,
5012,
5013,
5014,
5015,
5016,
5017,
5018,
5019,
5020,
5021,
5022,
5023,
5024,
5025,
5026,
5027,
5028,
5029,
5030,
5031,
5032,
5033,
5034,
5035,
5036,
5037,
5038,
5039,
5040,
5041,
5042,
5043,
```

5044,
5045,
5046,
5047,
5048,
5049,
5050,
5051,
5052,
5053,
5054,
5055,
5056,
5057,
5058,
5059,
5060,
5061,
5062,
5063,
5064,
5065,
5066,
5067,
5068,
5069,
5070,
5071,
5072,
5073,
5074,
5075,
5076,
5077,
5078,
5079,
5080,
5081,
5082,
5083,
5084,
5085,
5086,
5087,
5088,
5089,
5090,
5091,
5092,
5093,
5094,
5095,
5096,
5097,

5098,
5099,
5100,
5101,
5102,
5103,
5104,
5105,
5106,
5107,
5108,
5109,
5110,
5111,
5112,
5113,
5114,
5115,
5116,
5117,
5118,
5119,
5120,
5121,
5122,
5123,
5124,
5125,
5126,
5127,
5128,
5129,
5130,
5131,
5132,
5133,
5134,
5135,
5136,
5137,
5138,
5139,
5140,
5141,
5142,
5143,
5144,
5145,
5146,
5147,
5148,
5149,
5150,
5151,

5152,
5153,
5154,
5155,
5156,
5157,
5158,
5159,
5160,
5161,
5162,
5163,
5164,
5165,
5166,
5167,
5168,
5169,
5170,
5171,
5172,
5173,
5174,
5175,
5176,
5177,
5178,
5179,
5180,
5181,
5182,
5183,
5184,
5185,
5186,
5187,
5188,
5189,
5190,
5191,
5192,
5193,
5194,
5195,
5196,
5197,
5198,
5199,
5200,
5201,
5202,
5203,
5204,
5205,

```
5206,
5207,
5208,
5209,
5210,
5211,
5212,
5213,
5214,
5215,
5216,
5217,
5218,
5219,
5220,
5221,
5222,
5223,
5224,
5225,
5226,
5227,
5228,
5229,
5230,
5231,
5232,
5233,
5234,
5235,
5236,
5237,
5238,
5239};
//
// Line Loop on wing tip, directed inwards
Line Loop(1241) = {
5240,
5241,
5242,
5243,
5244,
5245,
5246,
5247,
5248,
5249,
5250,
5251,
5252,
5253,
5254,
5255,
5256,
```

5257,
5258,
5259,
5260,
5261,
5262,
5263,
5264,
5265,
5266,
5267,
5268,
5269,
5270,
5271,
5272,
5273,
5274,
5275,
5276,
5277,
5278,
5279,
5280,
5281,
5282,
5283,
5284,
5285,
5286,
5287,
5288,
5289,
5290,
5291,
5292,
5293,
5294,
5295,
5296,
5297,
5298,
5299,
5300,
5301,
5302,
5303,
5304,
5305,
5306,
5307,
5308,
5309,
5310,

5311,
5312,
5313,
5314,
5315,
5316,
5317,
5318,
5319,
5320,
5321,
5322,
5323,
5324,
5325,
5326,
5327,
5328,
5329,
5330,
5331,
5332,
5333,
5334,
5335,
5336,
5337,
5338,
5339,
5340,
5341,
5342,
5343,
5344,
5345,
5346,
5347,
5348,
5349,
5350,
5351,
5352,
5353,
5354,
5355,
5356,
5357,
5358,
5359,
5360,
5361,
5362,
5363,
5364,

5365,
5366,
5367,
5368,
5369,
5370,
5371,
5372,
5373,
5374,
5375,
5376,
5377,
5378,
5379,
5380,
5381,
5382,
5383,
5384,
5385,
5386,
5387,
5388,
5389,
5390,
5391,
5392,
5393,
5394,
5395,
5396,
5397,
5398,
5399,
5400,
5401,
5402,
5403,
5404,
5405,
5406,
5407,
5408,
5409,
5410,
5411,
5412,
5413,
5414,
5415,
5416,
5417,
5418,

5419,
5420,
5421,
5422,
5423,
5424,
5425,
5426,
5427,
5428,
5429,
5430,
5431,
5432,
5433,
5434,
5435,
5436,
5437,
5438,
5439,
5440,
5441,
5442,
5443,
5444,
5445,
5446,
5447,
5448,
5449,
5450,
5451,
5452,
5453,
5454,
5455,
5456,
5457,
5458,
5459,
5460,
5461,
5462,
5463,
5464,
5465,
5466,
5467,
5468,
5469,
5470,
5471,
5472,

```
5473,
5474,
5475,
5476,
5477,
5478};
//
// Plane Surfaces on curved surface of wing
Plane Surface(15001) = {1001};
Plane Surface(15002) = {1002};
Plane Surface(15003) = {1003};
Plane Surface(15004) = {1004};
Plane Surface(15005) = {1005};
Plane Surface(15006) = {1006};
Plane Surface(15007) = {1007};
Plane Surface(15008) = {1008};
Plane Surface(15009) = {1009};
Plane Surface(15010) = {1010};
Plane Surface(15011) = {1011};
Plane Surface(15012) = {1012};
Plane Surface(15013) = {1013};
Plane Surface(15014) = {1014};
Plane Surface(15015) = {1015};
Plane Surface(15016) = {1016};
Plane Surface(15017) = {1017};
Plane Surface(15018) = {1018};
Plane Surface(15019) = {1019};
Plane Surface(15020) = {1020};
Plane Surface(15021) = {1021};
Plane Surface(15022) = {1022};
Plane Surface(15023) = {1023};
Plane Surface(15024) = {1024};
Plane Surface(15025) = {1025};
Plane Surface(15026) = {1026};
Plane Surface(15027) = {1027};
Plane Surface(15028) = {1028};
Plane Surface(15029) = {1029};
Plane Surface(15030) = {1030};
Plane Surface(15031) = {1031};
Plane Surface(15032) = {1032};
Plane Surface(15033) = {1033};
Plane Surface(15034) = {1034};
Plane Surface(15035) = {1035};
Plane Surface(15036) = {1036};
Plane Surface(15037) = {1037};
Plane Surface(15038) = {1038};
Plane Surface(15039) = {1039};
Plane Surface(15040) = {1040};
Plane Surface(15041) = {1041};
Plane Surface(15042) = {1042};
Plane Surface(15043) = {1043};
Plane Surface(15044) = {1044};
Plane Surface(15045) = {1045};
Plane Surface(15046) = {1046};
```

```
Plane Surface(15047) = {1047};
Plane Surface(15048) = {1048};
Plane Surface(15049) = {1049};
Plane Surface(15050) = {1050};
Plane Surface(15051) = {1051};
Plane Surface(15052) = {1052};
Plane Surface(15053) = {1053};
Plane Surface(15054) = {1054};
Plane Surface(15055) = {1055};
Plane Surface(15056) = {1056};
Plane Surface(15057) = {1057};
Plane Surface(15058) = {1058};
Plane Surface(15059) = {1059};
Plane Surface(15060) = {1060};
Plane Surface(15061) = {1061};
Plane Surface(15062) = {1062};
Plane Surface(15063) = {1063};
Plane Surface(15064) = {1064};
Plane Surface(15065) = {1065};
Plane Surface(15066) = {1066};
Plane Surface(15067) = {1067};
Plane Surface(15068) = {1068};
Plane Surface(15069) = {1069};
Plane Surface(15070) = {1070};
Plane Surface(15071) = {1071};
Plane Surface(15072) = {1072};
Plane Surface(15073) = {1073};
Plane Surface(15074) = {1074};
Plane Surface(15075) = {1075};
Plane Surface(15076) = {1076};
Plane Surface(15077) = {1077};
Plane Surface(15078) = {1078};
Plane Surface(15079) = {1079};
Plane Surface(15080) = {1080};
Plane Surface(15081) = {1081};
Plane Surface(15082) = {1082};
Plane Surface(15083) = {1083};
Plane Surface(15084) = {1084};
Plane Surface(15085) = {1085};
Plane Surface(15086) = {1086};
Plane Surface(15087) = {1087};
Plane Surface(15088) = {1088};
Plane Surface(15089) = {1089};
Plane Surface(15090) = {1090};
Plane Surface(15091) = {1091};
Plane Surface(15092) = {1092};
Plane Surface(15093) = {1093};
Plane Surface(15094) = {1094};
Plane Surface(15095) = {1095};
Plane Surface(15096) = {1096};
Plane Surface(15097) = {1097};
Plane Surface(15098) = {1098};
Plane Surface(15099) = {1099};
Plane Surface(15100) = {1100};
```

```
Plane Surface(15101) = {1101};
Plane Surface(15102) = {1102};
Plane Surface(15103) = {1103};
Plane Surface(15104) = {1104};
Plane Surface(15105) = {1105};
Plane Surface(15106) = {1106};
Plane Surface(15107) = {1107};
Plane Surface(15108) = {1108};
Plane Surface(15109) = {1109};
Plane Surface(15110) = {1110};
Plane Surface(15111) = {1111};
Plane Surface(15112) = {1112};
Plane Surface(15113) = {1113};
Plane Surface(15114) = {1114};
Plane Surface(15115) = {1115};
Plane Surface(15116) = {1116};
Plane Surface(15117) = {1117};
Plane Surface(15118) = {1118};
Plane Surface(15119) = {1119};
Plane Surface(15120) = {1120};
Plane Surface(15121) = {1121};
Plane Surface(15122) = {1122};
Plane Surface(15123) = {1123};
Plane Surface(15124) = {1124};
Plane Surface(15125) = {1125};
Plane Surface(15126) = {1126};
Plane Surface(15127) = {1127};
Plane Surface(15128) = {1128};
Plane Surface(15129) = {1129};
Plane Surface(15130) = {1130};
Plane Surface(15131) = {1131};
Plane Surface(15132) = {1132};
Plane Surface(15133) = {1133};
Plane Surface(15134) = {1134};
Plane Surface(15135) = {1135};
Plane Surface(15136) = {1136};
Plane Surface(15137) = {1137};
Plane Surface(15138) = {1138};
Plane Surface(15139) = {1139};
Plane Surface(15140) = {1140};
Plane Surface(15141) = {1141};
Plane Surface(15142) = {1142};
Plane Surface(15143) = {1143};
Plane Surface(15144) = {1144};
Plane Surface(15145) = {1145};
Plane Surface(15146) = {1146};
Plane Surface(15147) = {1147};
Plane Surface(15148) = {1148};
Plane Surface(15149) = {1149};
Plane Surface(15150) = {1150};
Plane Surface(15151) = {1151};
Plane Surface(15152) = {1152};
Plane Surface(15153) = {1153};
Plane Surface(15154) = {1154};
```

```
Plane Surface(15155) = {1155};
Plane Surface(15156) = {1156};
Plane Surface(15157) = {1157};
Plane Surface(15158) = {1158};
Plane Surface(15159) = {1159};
Plane Surface(15160) = {1160};
Plane Surface(15161) = {1161};
Plane Surface(15162) = {1162};
Plane Surface(15163) = {1163};
Plane Surface(15164) = {1164};
Plane Surface(15165) = {1165};
Plane Surface(15166) = {1166};
Plane Surface(15167) = {1167};
Plane Surface(15168) = {1168};
Plane Surface(15169) = {1169};
Plane Surface(15170) = {1170};
Plane Surface(15171) = {1171};
Plane Surface(15172) = {1172};
Plane Surface(15173) = {1173};
Plane Surface(15174) = {1174};
Plane Surface(15175) = {1175};
Plane Surface(15176) = {1176};
Plane Surface(15177) = {1177};
Plane Surface(15178) = {1178};
Plane Surface(15179) = {1179};
Plane Surface(15180) = {1180};
Plane Surface(15181) = {1181};
Plane Surface(15182) = {1182};
Plane Surface(15183) = {1183};
Plane Surface(15184) = {1184};
Plane Surface(15185) = {1185};
Plane Surface(15186) = {1186};
Plane Surface(15187) = {1187};
Plane Surface(15188) = {1188};
Plane Surface(15189) = {1189};
Plane Surface(15190) = {1190};
Plane Surface(15191) = {1191};
Plane Surface(15192) = {1192};
Plane Surface(15193) = {1193};
Plane Surface(15194) = {1194};
Plane Surface(15195) = {1195};
Plane Surface(15196) = {1196};
Plane Surface(15197) = {1197};
Plane Surface(15198) = {1198};
Plane Surface(15199) = {1199};
Plane Surface(15200) = {1200};
Plane Surface(15201) = {1201};
Plane Surface(15202) = {1202};
Plane Surface(15203) = {1203};
Plane Surface(15204) = {1204};
Plane Surface(15205) = {1205};
Plane Surface(15206) = {1206};
Plane Surface(15207) = {1207};
Plane Surface(15208) = {1208};
```

```
Plane Surface(15209) = {1209};
Plane Surface(15210) = {1210};
Plane Surface(15211) = {1211};
Plane Surface(15212) = {1212};
Plane Surface(15213) = {1213};
Plane Surface(15214) = {1214};
Plane Surface(15215) = {1215};
Plane Surface(15216) = {1216};
Plane Surface(15217) = {1217};
Plane Surface(15218) = {1218};
Plane Surface(15219) = {1219};
Plane Surface(15220) = {1220};
Plane Surface(15221) = {1221};
Plane Surface(15222) = {1222};
Plane Surface(15223) = {1223};
Plane Surface(15224) = {1224};
Plane Surface(15225) = {1225};
Plane Surface(15226) = {1226};
Plane Surface(15227) = {1227};
Plane Surface(15228) = {1228};
Plane Surface(15229) = {1229};
Plane Surface(15230) = {1230};
Plane Surface(15231) = {1231};
Plane Surface(15232) = {1232};
Plane Surface(15233) = {1233};
Plane Surface(15234) = {1234};
Plane Surface(15235) = {1235};
Plane Surface(15236) = {1236};
Plane Surface(15237) = {1237};
Plane Surface(15238) = {1238};
Plane Surface(15239) = {1239};
//
// Plane Surface on wing tip
Plane Surface(15240) = {-1241};
//
// Points at the corners of the wind tunnel
Point(479) = {-WTAhead, -WTBelow, 0, lcWT};
Point(480) = {-WTAhead, -WTBelow, WTWidth, lcWT};
Point(481) = {WTAstern, -WTBelow, WTWidth, lcWT};
Point(482) = {WTAstern, -WTBelow, 0, lcWT};
Point(483) = {-WTAhead, WTAbove, 0, lcWT};
Point(484) = {-WTAhead, WTAbove, WTWidth, lcWT};
Point(485) = {WTAstern, WTAbove, WTWidth, lcWT};
Point(486) = {WTAstern, WTAbove, 0, lcWT};
//
// Structures relating to the Inlet of the wind tunnel
Line(5718) = {479, 480};
Line(5719) = {480, 484};
Line(5720) = {484, 483};
Line(5721) = {483, 479};
Line Loop(1242) = {-5718, -5721, -5720, -5719};
Plane Surface(15241) = {1242};
//
// Structures relating to Outlet of the wind tunnel
```

```
Line(5722) = {482, 481};
Line(5723) = {481, 485};
Line(5724) = {485, 486};
Line(5725) = {486, 482};
Line Loop(1243) = {5722, 5723, 5724, 5725};
Plane Surface(15242) = {1243};
//
// Structures relating to the LeftWall of the wind tunnel
Line(5726) = {480, 481};
Line(5727) = {484, 485};
Line Loop(1244) = {-5726, 5719, 5727, -5723};
Plane Surface(15243) = {1244};
//
// Structures relating to the Top of the wind tunnel
Line(5728) = {483, 486};
Line Loop(1245) = {-5727, 5720, 5728, -5724};
Plane Surface(15244) = {1245};
//
// Structures relating to the Bottom of the wind tunnel
Line(5729) = {479, 482};
Line Loop(1246) = {5726, -5722, -5729, 5718};
Plane Surface(15245) = {1246};
//
// Structures relating to the right wall (MidPlane) of wind tunnel
Line Loop(1247) = {5729, -5725, -5728, 5721};
Plane Surface(15246) = {1247, 1240};
//
// Surface Loop, directed inwards
Surface Loop(20000) = {
15241,
15242,
15243,
15244,
15245,
15246,
15240,
15001,
15002,
15003,
15004,
15005,
15006,
15007,
15008,
15009,
15010,
15011,
15012,
15013,
15014,
15015,
15016,
15017,
15018,
```

15019,
15020,
15021,
15022,
15023,
15024,
15025,
15026,
15027,
15028,
15029,
15030,
15031,
15032,
15033,
15034,
15035,
15036,
15037,
15038,
15039,
15040,
15041,
15042,
15043,
15044,
15045,
15046,
15047,
15048,
15049,
15050,
15051,
15052,
15053,
15054,
15055,
15056,
15057,
15058,
15059,
15060,
15061,
15062,
15063,
15064,
15065,
15066,
15067,
15068,
15069,
15070,
15071,
15072,

15073,
15074,
15075,
15076,
15077,
15078,
15079,
15080,
15081,
15082,
15083,
15084,
15085,
15086,
15087,
15088,
15089,
15090,
15091,
15092,
15093,
15094,
15095,
15096,
15097,
15098,
15099,
15100,
15101,
15102,
15103,
15104,
15105,
15106,
15107,
15108,
15109,
15110,
15111,
15112,
15113,
15114,
15115,
15116,
15117,
15118,
15119,
15120,
15121,
15122,
15123,
15124,
15125,
15126,

15127,
15128,
15129,
15130,
15131,
15132,
15133,
15134,
15135,
15136,
15137,
15138,
15139,
15140,
15141,
15142,
15143,
15144,
15145,
15146,
15147,
15148,
15149,
15150,
15151,
15152,
15153,
15154,
15155,
15156,
15157,
15158,
15159,
15160,
15161,
15162,
15163,
15164,
15165,
15166,
15167,
15168,
15169,
15170,
15171,
15172,
15173,
15174,
15175,
15176,
15177,
15178,
15179,
15180,

15181,
15182,
15183,
15184,
15185,
15186,
15187,
15188,
15189,
15190,
15191,
15192,
15193,
15194,
15195,
15196,
15197,
15198,
15199,
15200,
15201,
15202,
15203,
15204,
15205,
15206,
15207,
15208,
15209,
15210,
15211,
15212,
15213,
15214,
15215,
15216,
15217,
15218,
15219,
15220,
15221,
15222,
15223,
15224,
15225,
15226,
15227,
15228,
15229,
15230,
15231,
15232,
15233,
15234,

```
15235,
15236,
15237,
15238,
15239};
//
// Volume occupied by fluid inside the wind tunnel
Volume(20001) = {20000};
//
// Physical Surfaces on the wing
Physical Surface("Wing.1") = {15001};
Physical Surface("Wing.2") = {15002};
Physical Surface("Wing.3") = {15003};
Physical Surface("Wing.4") = {15004};
Physical Surface("Wing.5") = {15005};
Physical Surface("Wing.6") = {15006};
Physical Surface("Wing.7") = {15007};
Physical Surface("Wing.8") = {15008};
Physical Surface("Wing.9") = {15009};
Physical Surface("Wing.10") = {15010};
Physical Surface("Wing.11") = {15011};
Physical Surface("Wing.12") = {15012};
Physical Surface("Wing.13") = {15013};
Physical Surface("Wing.14") = {15014};
Physical Surface("Wing.15") = {15015};
Physical Surface("Wing.16") = {15016};
Physical Surface("Wing.17") = {15017};
Physical Surface("Wing.18") = {15018};
Physical Surface("Wing.19") = {15019};
Physical Surface("Wing.20") = {15020};
Physical Surface("Wing.21") = {15021};
Physical Surface("Wing.22") = {15022};
Physical Surface("Wing.23") = {15023};
Physical Surface("Wing.24") = {15024};
Physical Surface("Wing.25") = {15025};
Physical Surface("Wing.26") = {15026};
Physical Surface("Wing.27") = {15027};
Physical Surface("Wing.28") = {15028};
Physical Surface("Wing.29") = {15029};
Physical Surface("Wing.30") = {15030};
Physical Surface("Wing.31") = {15031};
Physical Surface("Wing.32") = {15032};
Physical Surface("Wing.33") = {15033};
Physical Surface("Wing.34") = {15034};
Physical Surface("Wing.35") = {15035};
Physical Surface("Wing.36") = {15036};
Physical Surface("Wing.37") = {15037};
Physical Surface("Wing.38") = {15038};
Physical Surface("Wing.39") = {15039};
Physical Surface("Wing.40") = {15040};
Physical Surface("Wing.41") = {15041};
Physical Surface("Wing.42") = {15042};
Physical Surface("Wing.43") = {15043};
Physical Surface("Wing.44") = {15044};
```

```
Physical Surface("Wing.45") = {15045};
Physical Surface("Wing.46") = {15046};
Physical Surface("Wing.47") = {15047};
Physical Surface("Wing.48") = {15048};
Physical Surface("Wing.49") = {15049};
Physical Surface("Wing.50") = {15050};
Physical Surface("Wing.51") = {15051};
Physical Surface("Wing.52") = {15052};
Physical Surface("Wing.53") = {15053};
Physical Surface("Wing.54") = {15054};
Physical Surface("Wing.55") = {15055};
Physical Surface("Wing.56") = {15056};
Physical Surface("Wing.57") = {15057};
Physical Surface("Wing.58") = {15058};
Physical Surface("Wing.59") = {15059};
Physical Surface("Wing.60") = {15060};
Physical Surface("Wing.61") = {15061};
Physical Surface("Wing.62") = {15062};
Physical Surface("Wing.63") = {15063};
Physical Surface("Wing.64") = {15064};
Physical Surface("Wing.65") = {15065};
Physical Surface("Wing.66") = {15066};
Physical Surface("Wing.67") = {15067};
Physical Surface("Wing.68") = {15068};
Physical Surface("Wing.69") = {15069};
Physical Surface("Wing.70") = {15070};
Physical Surface("Wing.71") = {15071};
Physical Surface("Wing.72") = {15072};
Physical Surface("Wing.73") = {15073};
Physical Surface("Wing.74") = {15074};
Physical Surface("Wing.75") = {15075};
Physical Surface("Wing.76") = {15076};
Physical Surface("Wing.77") = {15077};
Physical Surface("Wing.78") = {15078};
Physical Surface("Wing.79") = {15079};
Physical Surface("Wing.80") = {15080};
Physical Surface("Wing.81") = {15081};
Physical Surface("Wing.82") = {15082};
Physical Surface("Wing.83") = {15083};
Physical Surface("Wing.84") = {15084};
Physical Surface("Wing.85") = {15085};
Physical Surface("Wing.86") = {15086};
Physical Surface("Wing.87") = {15087};
Physical Surface("Wing.88") = {15088};
Physical Surface("Wing.89") = {15089};
Physical Surface("Wing.90") = {15090};
Physical Surface("Wing.91") = {15091};
Physical Surface("Wing.92") = {15092};
Physical Surface("Wing.93") = {15093};
Physical Surface("Wing.94") = {15094};
Physical Surface("Wing.95") = {15095};
Physical Surface("Wing.96") = {15096};
Physical Surface("Wing.97") = {15097};
Physical Surface("Wing.98") = {15098};
```

```
Physical Surface("Wing.99") = {15099};
Physical Surface("Wing.100") = {15100};
Physical Surface("Wing.101") = {15101};
Physical Surface("Wing.102") = {15102};
Physical Surface("Wing.103") = {15103};
Physical Surface("Wing.104") = {15104};
Physical Surface("Wing.105") = {15105};
Physical Surface("Wing.106") = {15106};
Physical Surface("Wing.107") = {15107};
Physical Surface("Wing.108") = {15108};
Physical Surface("Wing.109") = {15109};
Physical Surface("Wing.110") = {15110};
Physical Surface("Wing.111") = {15111};
Physical Surface("Wing.112") = {15112};
Physical Surface("Wing.113") = {15113};
Physical Surface("Wing.114") = {15114};
Physical Surface("Wing.115") = {15115};
Physical Surface("Wing.116") = {15116};
Physical Surface("Wing.117") = {15117};
Physical Surface("Wing.118") = {15118};
Physical Surface("Wing.119") = {15119};
Physical Surface("Wing.120") = {15120};
Physical Surface("Wing.121") = {15121};
Physical Surface("Wing.122") = {15122};
Physical Surface("Wing.123") = {15123};
Physical Surface("Wing.124") = {15124};
Physical Surface("Wing.125") = {15125};
Physical Surface("Wing.126") = {15126};
Physical Surface("Wing.127") = {15127};
Physical Surface("Wing.128") = {15128};
Physical Surface("Wing.129") = {15129};
Physical Surface("Wing.130") = {15130};
Physical Surface("Wing.131") = {15131};
Physical Surface("Wing.132") = {15132};
Physical Surface("Wing.133") = {15133};
Physical Surface("Wing.134") = {15134};
Physical Surface("Wing.135") = {15135};
Physical Surface("Wing.136") = {15136};
Physical Surface("Wing.137") = {15137};
Physical Surface("Wing.138") = {15138};
Physical Surface("Wing.139") = {15139};
Physical Surface("Wing.140") = {15140};
Physical Surface("Wing.141") = {15141};
Physical Surface("Wing.142") = {15142};
Physical Surface("Wing.143") = {15143};
Physical Surface("Wing.144") = {15144};
Physical Surface("Wing.145") = {15145};
Physical Surface("Wing.146") = {15146};
Physical Surface("Wing.147") = {15147};
Physical Surface("Wing.148") = {15148};
Physical Surface("Wing.149") = {15149};
Physical Surface("Wing.150") = {15150};
Physical Surface("Wing.151") = {15151};
Physical Surface("Wing.152") = {15152};
```

```
Physical Surface("Wing.153") = {15153};
Physical Surface("Wing.154") = {15154};
Physical Surface("Wing.155") = {15155};
Physical Surface("Wing.156") = {15156};
Physical Surface("Wing.157") = {15157};
Physical Surface("Wing.158") = {15158};
Physical Surface("Wing.159") = {15159};
Physical Surface("Wing.160") = {15160};
Physical Surface("Wing.161") = {15161};
Physical Surface("Wing.162") = {15162};
Physical Surface("Wing.163") = {15163};
Physical Surface("Wing.164") = {15164};
Physical Surface("Wing.165") = {15165};
Physical Surface("Wing.166") = {15166};
Physical Surface("Wing.167") = {15167};
Physical Surface("Wing.168") = {15168};
Physical Surface("Wing.169") = {15169};
Physical Surface("Wing.170") = {15170};
Physical Surface("Wing.171") = {15171};
Physical Surface("Wing.172") = {15172};
Physical Surface("Wing.173") = {15173};
Physical Surface("Wing.174") = {15174};
Physical Surface("Wing.175") = {15175};
Physical Surface("Wing.176") = {15176};
Physical Surface("Wing.177") = {15177};
Physical Surface("Wing.178") = {15178};
Physical Surface("Wing.179") = {15179};
Physical Surface("Wing.180") = {15180};
Physical Surface("Wing.181") = {15181};
Physical Surface("Wing.182") = {15182};
Physical Surface("Wing.183") = {15183};
Physical Surface("Wing.184") = {15184};
Physical Surface("Wing.185") = {15185};
Physical Surface("Wing.186") = {15186};
Physical Surface("Wing.187") = {15187};
Physical Surface("Wing.188") = {15188};
Physical Surface("Wing.189") = {15189};
Physical Surface("Wing.190") = {15190};
Physical Surface("Wing.191") = {15191};
Physical Surface("Wing.192") = {15192};
Physical Surface("Wing.193") = {15193};
Physical Surface("Wing.194") = {15194};
Physical Surface("Wing.195") = {15195};
Physical Surface("Wing.196") = {15196};
Physical Surface("Wing.197") = {15197};
Physical Surface("Wing.198") = {15198};
Physical Surface("Wing.199") = {15199};
Physical Surface("Wing.200") = {15200};
Physical Surface("Wing.201") = {15201};
Physical Surface("Wing.202") = {15202};
Physical Surface("Wing.203") = {15203};
Physical Surface("Wing.204") = {15204};
Physical Surface("Wing.205") = {15205};
Physical Surface("Wing.206") = {15206};
```

```
Physical Surface("Wing.207") = {15207};
Physical Surface("Wing.208") = {15208};
Physical Surface("Wing.209") = {15209};
Physical Surface("Wing.210") = {15210};
Physical Surface("Wing.211") = {15211};
Physical Surface("Wing.212") = {15212};
Physical Surface("Wing.213") = {15213};
Physical Surface("Wing.214") = {15214};
Physical Surface("Wing.215") = {15215};
Physical Surface("Wing.216") = {15216};
Physical Surface("Wing.217") = {15217};
Physical Surface("Wing.218") = {15218};
Physical Surface("Wing.219") = {15219};
Physical Surface("Wing.220") = {15220};
Physical Surface("Wing.221") = {15221};
Physical Surface("Wing.222") = {15222};
Physical Surface("Wing.223") = {15223};
Physical Surface("Wing.224") = {15224};
Physical Surface("Wing.225") = {15225};
Physical Surface("Wing.226") = {15226};
Physical Surface("Wing.227") = {15227};
Physical Surface("Wing.228") = {15228};
Physical Surface("Wing.229") = {15229};
Physical Surface("Wing.230") = {15230};
Physical Surface("Wing.231") = {15231};
Physical Surface("Wing.232") = {15232};
Physical Surface("Wing.233") = {15233};
Physical Surface("Wing.234") = {15234};
Physical Surface("Wing.235") = {15235};
Physical Surface("Wing.236") = {15236};
Physical Surface("Wing.237") = {15237};
Physical Surface("Wing.238") = {15238};
Physical Surface("Wing.239") = {15239};
//
// Physical Surfaces on the wing tip
Physical Surface("Wing.240") = {15240};
//
// Physical Surfaces on the wind tunnel
Physical Surface("Inlet") = {15241};
Physical Surface("Outlet") = {15242};
Physical Surface("LeftWall") = {15243};
Physical Surface("Top") = {15244};
Physical Surface("Bottom") = {15245};
Physical Surface("MidPlane") = {15246};
Physical Volume("Internal") = {20001};
```

Appendix "B"

Listing of 13 files from the case directory of the base case OpenFoam run

Listing of file `constant/polyMesh/boundary`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       polyBoundaryMesh;
    location    "constant/polyMesh";
    object      boundary;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

246
(
    Wing.1
    {
        type            wall;
        nFaces          3940;
        startFace       10280679;
    }
    Wing.2
    {
        type            wall;
        nFaces          3912;
        startFace       10284619;
    }
```

For the sake of brevity, I have not listed the lines which describe plane surfaces #3 through #238 on the membrane.

```
    Wing.239
    {
        type            wall;
        nFaces          3958;
        startFace       11208451;
    }
    Wing.240
    {
        type            wall;
        nFaces          1901;
        startFace       11212409;
    }
    Inlet
    {
        type            patch;
        nFaces          2122;
```

```
        startFace       11214310;
    }
    Outlet
    {
        type            patch;
        nFaces          2122;
        startFace       11216432;
    }
    LeftWall
    {
        type            symmetryPlane;
        nFaces          2466;
        startFace       11218554;
    }
    Top
    {
        type            symmetryPlane;
        nFaces          2462;
        startFace       11221020;
    }
    Bottom
    {
        type            symmetryPlane;
        nFaces          2470;
        startFace       11223482;
    }
    MidPlane
    {
        type            symmetryPlane;
        nFaces          20385;
        startFace       11225952;
    }
)
```

Listing of file `constant/RASProperties`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      RASProperties;
}
RASModel          kOmegaSST;
turbulence          on;
printCoeffs         on;

kOmegaSSTCoeffs
{
    alphaK1     0.85034;
```

```
    alphaK2     1.0;
    alphaOmega1 0.5;
    alphaOmega2 0.85616;
    beta1       0.075;
    beta2       0.0828;
    betaStar    0.09;
    gamma1      0.5532;
    gamma2      0.4403;
    a1          0.31;
    c1          10.0;
}
```

Listing of file `constant/transportProperties`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      transportProperties;
}

// U.S. Standard Atmosphere
// Altitude---   Density-----  Dynamic visc---  Kinematic visc-
//      0 feet   1.225 kg/m^3  1.789E-5 Ns/m^2  1.4604E-5 m^2/s
//  5,000        0.7364        1.628E-5         2.2108E-5
// 10,000        0.4135        1.458E-5         3.5260E-5
// 15,000        0.1948        1.422E-5         7.2998E-5

transportModel  Newtonian;
nu              nu [0 2 -1 0 0 0 0] 7.2998E-5;
rho             rho [ 1 -3 0 0 0 0 0 ] 0.1948;
```

Listing of file `constant/turbulenceProperties`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      turbulenceProperties;
}
simulationType  RASModel;
```

Listing of file `system/controlDict`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      controlDict;
}
application     simpleFoam;
startFrom       latestTime;
startTime       0;
stopAt          endTime;
endTime         10000;
deltaT          1;
writeControl    timeStep;
writeInterval   100;
purgeWrite      3;
writeFormat     ascii;
writePrecision  8;
writeCompression off;
timeFormat      general;
timePrecision   6;
runTimeModifiable true;
functions
{
    forcesWing
    {
        type                forces;
        functionObjectLibs  ( "libforces.so" );
        patches             (
                              "Wing.*"
                            );
        rhoName             rhoInf;
        pName               p;
        UName               U;
        log                 true;
        rhoInf              0.1948;
        CofR                ( 0 0 0 );
        outputControl       timeStep;
        outputInterval       1;
    }
};
```

Listing of file `system/decomposeParDict`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      decomposeParDict;
}
numberOfSubdomains   8;
method               scotch;
scotchCoeffs
                     {}
distributed          no;
roots
                     ();

// To run a case in parallel, do this:
// 1.  <prompt> decomposePar
// 2.  <prompt> mpirun -np 8 simpleFoam -parallel | tee -a ofLog.txt
// 3.  When done, <prompt> reconstructPar
```

Listing of file `system/fvSchemes`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
ddtSchemes
{
    default         steadyState;
}
gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         Gauss linear;
}
divSchemes
{
```

```
    default                         none;
    div(phi,U)                      Gauss upwind;
    div(phi,k)                      Gauss upwind;
    div(phi,epsilon)                Gauss upwind;
    div(phi,R)                      Gauss upwind;
    div(R)                                 Gauss linear;
    div(phi,nuTilda)                       Gauss upwind;
    div((nuEff*dev(T(grad(U)))))    Gauss linear;
    div(phi,omega)                  Gauss upwind;
}
laplacianSchemes
{
    default                         none;
    laplacian(nuEff,U)                     Gauss linear corrected;
    laplacian((1|A(U)),p)           Gauss linear corrected;
    laplacian(DkEff,k)                     Gauss linear corrected;
    laplacian(DepsilonEff,epsilon)         Gauss linear corrected;
    laplacian(DREff,R)                     Gauss linear corrected;
    laplacian(DnuTildaEff,nuTilda)         Gauss linear corrected;
    laplacian(DomegaEff,omega)             Gauss linear corrected;
}
interpolationSchemes
{
    default         linear;
    interpolate(U)  linear;
}
snGradSchemes
{
    default         corrected;
}
fluxRequired
{
    default         no;
    p               ;
}
```

Listing of file `system/fvSolution`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSolution;
}
solvers
{
    p
    {
        solver           GAMG;
        tolerance        1e-20;
```

```
        relTol              0.01;
        maxIter             20;
        smoother            GaussSeidel;
        nPreSweeps          1;
        nPostSweeps         3;
        nFinestSweeps       3;
        scaleCorrection     true;
        directSolveCoarsest       false;
        cacheAgglomeration  on;
        nCellsInCoarsestLevel     6;
        agglomerator        faceAreaPair;
        mergeLevels         1;
    }
    U
    {
        solver              PBiCG;
        preconditioner      DILU;
        tolerance           1e-20;
        relTol              0.05;
       maxIter                    20;
    }
    k
    {
        solver              PBiCG;
        preconditioner      DILU;
        tolerance           1e-20;
        relTol              0.05;
       maxIter                    20;
    }
    omega
    {
        solver              PBiCG;
        preconditioner      DILU;
        tolerance           1e-20;
        relTol              0.05;
       maxIter                    20;
    }
}
SIMPLE
{
// Start with 2 NonOrthogonalCorrectors
    nNonOrthogonalCorrectors      2;
    residualControl
    {
        p                   1e-5;
        U                   1e-5;
        k                   1e-5;
        epsilon             1e-5;
    }
}
// Start kOmegaSST with p=0.5 and U=k=omega=0.75
// At Iteration #72, reduce pRelax to 0.45 and kRelax to 0.6
// At Iteration #203, reduce URelax to 0.65
relaxationFactors
{
    p                       0.45;
    U                       0.65;
    k                       0.6;
```

```
    omega                   0.75;
}
```

<u>Listing of file `0/p`</u>

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}
dimensions      [0 2 -2 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    Inlet
    {
        type         zeroGradient;
    }
    Outlet
    {
        type         fixedValue;
        value        uniform 0;
    }
    LeftWall
    {
        type         symmetryPlane;
    }
    MidPlane
    {
        type         symmetryPlane;
    }
    Top
    {
        type         symmetryPlane;
    }
    Bottom
    {
        type         symmetryPlane;
    }
    "Wing.*"
    {
        type         zeroGradient;
    }
}
```

Listing of file `0/U`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    location    "0";
    object      U;
}
// 10mph = 4.4704 m/s
// 20mph = 8.9408 m/s
// 30mph = 13.4112 m/s

dimensions      [0 1 -1 0 0 0 0];
internalField   uniform (8.9408 0 0);
boundaryField
{
    Inlet
    {
       type   fixedValue;
       value  uniform (8.9408 0 0);
    }
    Outlet
    {
       type   zeroGradient;
    }
    LeftWall
    {
        type  symmetryPlane;
    }
    MidPlane
    {
        type  symmetryPlane;
    }
    Top
    {
        type  symmetryPlane;
    }
    Bottom
    {
       type   symmetryPlane;
    }
    "Wing.*"
    {
        type  fixedValue;
       value  uniform (0 0 0);
    }
}
```

Listing of file `0/nut`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      nut;
}

dimensions      [0 2 -1 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    Inlet
    {
        type            calculated;
        value           uniform 0;
    }
    Outlet
    {
        type            calculated;
        value           uniform 0;
    }
    LeftWall
    {
        type            symmetryPlane;
    }
    MidPlane
    {
        type            symmetryPlane;
    }
    Top
    {
        type            symmetryPlane;
    }
    Bottom
    {
        type            symmetryPlane;
    }
    "Wing.*"
    {
        type            nutkWallFunction;
        Cmu             0.09;
        kappa           0.41;
        E               9.8;
        value           uniform 0;
    }
```

```
}
```

Listing of file `0/k`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      k;
}

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0.0120;

boundaryField
{
    Inlet
    {
        type         fixedValue;
        value        uniform 0.0120;
    }
    Outlet
    {
        type         zeroGradient;
    }
    LeftWall
    {
        type         symmetryPlane;
    }
    MidPlane
    {
        type         symmetryPlane;
    }
    Top
    {
        type         symmetryPlane;
    }
    Bottom
    {
        type         symmetryPlane;
    }
    "Wing.*"
    {
        type        zeroGradient;
    }
}
```

Listing of file `0/omega`

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.1.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      omega;
}

dimensions      [0 0 -1 0 0 0 0];

internalField   uniform 10.95;

boundaryField
{
    Inlet
    {
        type         fixedValue;
        value        uniform 10.95;
    }
    Outlet
    {
        type         zeroGradient;
    }
    LeftWall
    {
        type         symmetryPlane;
    }
    MidPlane
    {
        type         symmetryPlane;
    }
    Top
    {
        type         symmetryPlane;
    }
    Bottom
    {
        type         symmetryPlane;
    }
    "Wing.*"
    {
        type         zeroGradient;
    }
}
```