ONERA

Numerical Optimization in Aerodynamic Design with application to a Supersonic Transport Aircraft

NAL International Workshop on Supersonic Transport Design

16-17 March 1998, TOKYO - JAPAN

Richard Grenon

Applied Aerodynamics Department

ONERA BP 72, 92322 Châtillon Cedex, FRANCE

ONERA

Table of contents

ONERA

Abstract

A numerical optimization code for the aerodynamic design of wing-body configurations has been recently developed at ONERA. This optimization code has been built by coupling the CONMIN code of G.N. Vanderplaats which performs the constrained minimization of a function of N variables with an ONERA Euler code for aerodynamic computations. For this purpose, a geometry generator taking into account a great number of design variables and an automatic grid generator have also been developed.

In the first part of this paper, the general organization of the ONERA optimization code is presented. In a second part, the first results from an application of this code to the design of a supersonic transport aircraft are presented and the strategy of optimization is discussed.

1 Introduction

Numerical optimization has been used at ONERA since several years to improve wing design at transonic cruise [1]. These numerical techniques were successfully applied to wave drag reduction or wing-engine interference minimization on subsonic transport aircraft. But until now, the number of design variables used for wing shape control during optimization runs was rather low (less than 10) and the aerodynamic computations were always performed using potential codes.

In 1995, the necessity appeared of having a powerful design tool to optimize a wing at transonic and supersonic cruise conditions with full control on the wing shape. Following the work of Doherty and Parker on supersonic optimization [2], it has been decided to develop a new optimization code including an Euler code for aerodynamic computations. The first version of our new optimization code was able to control the twist and camber laws of a wing with 25 design variables and to optimize the isolated wing simultaneously at two cruise points. In 1996, this code has been extended to have a wing-body configuration capability. Wing-body geometry control includes now the definition of the wing twist, camber and thickness laws with 45 to 75 design variables, depending on the desired degree of control of the thickness law, the modification of the wing planform with up to 10 design variables, the deflections of the leading edge slats and trailing edge flaps with one design variable for each control surface, and finally the wing location relative to the body with 3 design variables.

In a first part, the general organization of the ONERA optimization code will be presented. In a second part, results of optimization runs performed on the isolated wing of the Aerospatiale ATSF configuration, on the European ESCT wing-body configuration and finally on the Aerospatiale AS4 wing-body configuration will be presented and discussed.

2 ONERA Optimization code for Aerodynamic design

The organization of the ONERA optimization code for the aerodynamic design of a wing-body configuration is shown on the figure 1. This code is built around the CONMIN code of G.N. Vanderplaats [3] which can perform a constrained minimization of a function of N design variables. The user must provide a routine which has to compute the objective function to be minimized and the associated constraint functions. The minimization code CONMIN is used with its COPES interface [4], also written by Vanderplaats, which makes easier the exchange of information between CONMIN and the user's routine.

2.1 CONMIN minimization module

The formulation of the constrained minimization problem in CONMIN is :

- minimize : **F(X)** function of **N** design variables **X(i) i = 1,N**

- subject to **Nc** inequality constraints depending on the design variables :

$$G_j(X) \le 0 \qquad j = 1, Nc$$

- and to some side constraints bounding the design variables :

$$Vmin(i) \le X(i) \le Vmax(i) \qquad i = 1,N$$

Here **F** is the objective function, **X** is the vector of the **N** design variables, **Gj** is the j^{th} of **Nc** constraint functions of the **N** design variables, **Vmin** and **Vmax** are the arrays of the boundary values of the **N** design variables.

ONERA

A constraint function is considered as active by CONMIN only when its value comes near zero. The user has to define a "constraint thickness" parameter ε to recognize active constraint functions.

The constraint function G_j is said to be active when :

$$-\varepsilon \le G_j(X) \le +\varepsilon$$

The constraint function Gj is said to be violated when :

$$G_j(X) > +\varepsilon$$

When $G_j(X) < -\varepsilon$, the constraint function is inactive and ignored by the optimization algorithm.

Starting with an initial design vector **X**, the minimum of the objective function **F** is reached iteratively after a number of cycles where modifications of the design variables are computed. The minimization of a function requires two steps at each cycle : at the beginning of a minimization cycle **q**, the design variables vector obtained at the end of the cycle **q-1** is X^{q-1} and CONMIN first computes a search direction S^q from X^{q-1} in the **N**-dimensional space of the design variables. Then CONMIN computes the amount α of the move in the search direction S^q which gives the greatest reduction of the objective function. The new design variables vector X^q at the end of the cycle **q** is given by :

$$X^q = X^{q-1} + \alpha * S^q$$

Finding the search direction S^q requires the computation of the gradient of the objective function **F** and of its associated active constraint functions G_j. The search direction S^q is found from the gradient information using the Fletcher-Reeves algorithm when the problem is unconstrained or the Zoutendijk's method of Feasible Directions when the problem is constrained. CONMIN uses the Fletcher-Reeves algorithm until one or more constraint functions become active, then it uses the Feasible Directions method to find the minimum of the objective function while trying to maintain the constraint functions active without violation of side constraints. All these algorithms are detailed in the reference [5]. The gradient computation requires **N** evaluations of the objective function and of the constraint functions. The amount α of the move in the search direction S^q is found using polynomial approximations and it requires 2 or 3 more evaluations of the objective function and of the constraint functions. So **N+3** evaluations of the objective function and of the constraint functions should be expected at each cycle of a minimization run.

The minimization code CONMIN is used with its COPES interface [4], also written by Vanderplaats, which makes easier the exchange of information between CONMIN and the user's routine which has to compute from a given set of design variables the objective function to be minimized and the constraint functions. The only programming rule is that the user's routine should be called ANALIZ and should have only one integer argument which specifies the task to be performed.

The main program COPES manages the calls to CONMIN and the calls to the user's routine ANALIZ. The values of the design variables, the objective function and the user's constraint functions are passed through a common array. When the optimization run starts, COPES reads an INPUT file where the user has defined the location of the design variables, of the objective function and of the user's constraint functions in the common array. The INPUT file also contains boundary values and scaling factors for the design variables and for the constraint functions. The sign of a special scaling factor to be applied to the objective function indicates if the function is to be minimized or to be maximized. If the objective function is to be maximized, COPES reverses its sign so that CONMIN has only to consider a minimization problem.

The CONMIN code reaches the minimum more quickly if all gradient components are of the same order, so appropriate scaling factors applied to each design variable and to each constraint function may be used to normalize the gradients. If we make the following design variable change **X'(i) = X(i) / Sc(i)**, the gradient component relative to the new design variable **X'(i)** becomes :

$$\partial F/\partial X'(i) = Sc(i) * \partial F/\partial X(i).$$

In the same way, if a user's constraint function $g_j(X)$ is associated with two boundary values $g_j min$ and $g_j max$ and a scaling factor Sc_j, COPES builds two new constraint functions **Gj1** and **Gj2** that are effectively passed to CONMIN. The scaling factors should be chosen so that CONMIN could use the same "constraint thickness" parameter ε for all constraints :

$$Gj1(X) = (g_j min - g_j(X))/Sc_j \text{ to be compared with } \pm\varepsilon$$

$$Gj2(X) = (g_j(X) - g_j max)/Sc_j \text{ to be compared with } \pm\varepsilon$$

Gradient normalization with scaling on the design variables and redefinition of the constraint functions is performed by the main program COPES and the user's routine ANALIZ knows only unscaled variables and unscaled user's constraint functions.

ONERA

2.2 Aerodynamic problem formulation

2.2.1 User's main routine

The user's routine ANALIZ has to perform three tasks following the value of its integer argument :

- with the argument value 1 at the first call, ANALIZ has to initialize the design variables and to put these initial values in the communication array. At this stage, the geometry, the aerodynamic conditions and all user's variables are also initialized.
- with the argument value 2 at following calls, ANALIZ has to read in the communication array the design variables values each time they have been modified by CONMIN and has to compute and to put back in the communication array the corresponding values of the objective function and of the associated constraint functions.
- with the argument value 3 at the last call, ANALIZ has to output the final values of the design variables, of the objective function and of the constraint functions. The optimized geometry and its aerodynamic performance are also output at this stage. The initialization file necessary to a restart of a new minimization run is also written.

The main task of the routine ANALIZ during a minimization run is the computation of the objective function and of the constraint functions from a given set of design variables. For the formulation of the aerodynamic design problem, ANALIZ has been written as a main routine calling several specialized modules that may be easily changed depending on the specific problem to be solved. These specific aerodynamic modules are :

- ☞ a geometry generator which modifies the wing-body geometry according to a big set of design variables that are returned by CONMIN,
- ☞ a geometric constraints module which defines a set of geometric constraint functions and that must be written especially for a given problem,
- ☞ a grid generator,
- ☞ an Euler code for aerodynamic computations.

After having performed the aerodynamic computation on the new geometry obtained from the set of design variables, the objective function is stored in the communication array : it may be the inviscid drag to be minimized or the inviscid Lift to Drag ratio to be maximized. Two aerodynamic computations may be successively performed on the same geometry with different cruise conditions if the problem is a dual point design, and in this case, the objective function may be a linear function of the drag or of the Lift to Drag ratio at each cruise point.

Aerodynamic constraint functions will be defined inside COPES using the aerodynamic force coefficients. For instance, a single point optimization could be the minimization of the drag with a minimum value imposed on the lift and two boundary values imposed on the pitching moment. A dual point design could be the minimization of the transonic drag with a minimum value imposed on each one of the transonic lift, the supersonic lift and the supersonic Lift to Drag ratio. The Lift to Drag ratio and all force coefficients returned in ANALIZ for each cruise point are stored in the communication array and used by COPES if necessary with proper scaling factors and boundary values to define the aerodynamic constraint functions to be passed to CONMIN.

Additional aerodynamic pressure constraints may also be defined. For instance, we can impose a limit on the minimum value of the local pressure coefficient Cp or on the maximum Mach number normal to the isobars. A specific module is added to the Euler code to compute these pressure functions, but as we shall see later, it is not easy to properly define a pressure constraint because the pressure field obtained with the coarse grid required during optimization runs is very different from that it would be with a fine analysis grid. The values of the pressure functions returned in ANALIZ are also stored in the communication array.

The last aerodynamic constraint to be considered is the maximum value allowed for the body angle of attack. This angle is not a data but is obtained from the design variables controlling the wing location on the body and the angle of attack of the reference plane of the wing. The wing reference plane angle of attack, which may be a decision variable, and the body angle of attack are also stored in the communication array.

2.2.2 Geometry Generator

The geometry generator actually performs only deformations of the wing on a wing-body configuration defined by a large set of points. Possible wing modifications concern the twist, camber and thickness laws as well as the planform.

ONERA

The geometry generator can also modify the wing-body relative location and can simulate the deflection of leading edge slats and trailing edge flaps.

A Twist, camber and thickness

The design variables define the twist, camber and thickness laws of the wing only at a reduced set of design sections, usually five. The twist law is defined by the angle of attack of each design section relative to the wing reference plane z=0. The camber and the thickness laws at each design section are defined using Bezier's curves [6]. A Bezier's curve of degree **n** depends on **n+1** control points, including the two end points. If P_0, P_1, P_2, ... , P_{n-1} and P_n are its **n+1** ordered control points and **t** is a parameter between **0** and **1**, the current point of the Bezier's curve is given by :

$$\vec{P}(t) = \sum_{i=0}^{i=n} f_n^i(t) \cdot \vec{a}_i \quad \text{with} \quad \vec{a}_i = \vec{P_{i-1}P_i}\ ,\ i = 1 \text{ to } n\ , \text{ and } \vec{a}_o = \vec{P_0}$$

The end points P_0 and P_n are obtained respectively with **t=0** and **t=1** :

$$\vec{P}(0) = \vec{P_0} \qquad \vec{P}(1) = \vec{P_n}$$

The tangent direction at the end points is controlled by P_1 and P_{n-1} :

$$\vec{P_0P_1} = \text{tangent direction at } P_0\ ,\ \vec{P_{n-1}P_n} = \text{tangent direction at } P_n$$

The $f_n^i(t)$ coefficients are the Bezier's polynoms of order **i** and degree **n** and are defined as :

$$f_n^i(t) = \frac{(-t)^i}{(i-1)!} \cdot \frac{d^{i-1}\Phi_n(t)}{dt^{i-1}} \quad \text{with } \Phi_n(t) = \frac{(1-t)^n - 1}{t} \text{ and } f_n^0(t) = 1$$

These polynoms may also be written as :

$$f_n^i(t) = \sum_{j=i}^{j=n} (-1)^{j+i} \cdot C_n^j \cdot C_{j-1}^{i-1} \cdot t^j$$

where : $C_n^j = \frac{n!}{j! \cdot (n-j)!}$ and $C_{j-1}^{i-1} = \frac{(j-1)!}{(i-1)! \cdot (j-i)!}$

The twist and camber laws at each design section are actually controlled by **5** design variables according to the sketch shown on the figure 2. The local chord line angle of attack relative to the reference plane of the wing defines the section twist angle and is the first design variable. The four other variables define the camber law of the design section relative to its chord line with one Bezier's curve of degree **3** having **4** control points. The two first camber design variables are the camber line slopes relative to the chord line respectively at the leading edge and at the trailing edge which are defined by the two intermediate control points P_1 and P_2 , and the two last camber design variables are the distances of these two intermediate control points respectively from the leading edge and from the trailing edge. Of course, the number of intermediate control points of the camber law may be increased, but we must remember that adding a new control point on a design section adds two design variables which are its **x** and **z** coordinates.

The thickness law at each design section is controlled by **4** or **10** design variables according to the sketch shown on the figure 3. The minimum control of the thickness law is obtained with **4** main design variables that are the maximum thickness, the longitudinal location of the maximum thickness, the leading edge radius and the trailing edge slope with no camber. To have the maximum thickness and the longitudinal location of the maximum thickness as design variables, the thickness law is defined by two Bezier's curves which have curvature continuity at the common point of maximum thickness where the tangent direction is parallel to the chord line. Each Bezier's curve must have at least two intermediate control points to define the correct tangent direction at its end points and that leads to more than 4 control parameters for the thickness law. We have defined an automatic construction of the intermediate control points from the **4** main thickness design variables, but we were constrained to use Bezier's curves of degree **4** with **3** intermediate control points when we tried to get good approximations of the thickness laws of some typical sections of the European configuration ESCT. If a finer tuning on the thickness law is needed, **6** extra design variables can be added to have a full control on all intermediate control points.

The twist and camber laws defined by the set of design variables are always considered in an incremental form as variations of the twist and camber laws of the initial wing. Each design section of the initial wing is first rotated by an angle equal to the local twist design variable, then the camber deformation is equally added at the upper side and at the lower side of the rotated design section as a function of the chordwise position in a direction normal to the rotated chord line. In this way, if the initial wing has non zero twist and camber, the twist and camber laws defined by the set of design variables are added to the twist and camber laws of the initial wing and do not replace them.

When it is used, the thickness law is always considered in an absolute form and each design section is rebuilt by adding the new thickness law to its initial mean camber line before adding twist and camber deformations.

After the deformations from the initial geometry due to twist, camber and thickness modifications have been computed at each point of the design sections, they are interpolated along the span using cubic splines and are added at each point of the initial wing.

B Planform

The next step is the wing planform modification without changing the projected area, the aspect ratio and the twist, camber and relative thickness laws according to the sketches shown on the figure 4. There is an automatic detection of the leading edge parts and of the trailing edge parts which have a constant sweep angle and one design variable controlling the sweep angle change may be attributed to each one of these parts. For instance, the wing shown on the figure 4 may have two leading edge design variables and three trailing edge design variables. The sweep angle change on the parts with variable sweep angle is interpolated along the span from the sweep angle changes on the parts with constant sweep angle. The sweep angle change design variables define translations of the leading edge and of the trailing edge points in the chordwise X direction of the XY reference plane, then the chord of each section is corrected of the same amount at the trailing edge in the XY plane to maintain a constant projected area. As the wing span does not change, the aspect ratio also remains unchanged. Once the new planform has been determined, each section of the wing is rebuilt with a translation and a scaling to meet the new planform boundaries according to the third sketch shown on the figure 4. This transformation keeps unchanged the section angle of attack and its camber and thickness laws relative to the chord.

C Wing location

After having performed wing deformations according to the twist, camber, thickness and planform design variables, the geometry generator calls a geometry checking module (fig. 1) which computes the wing-body intersection using a Newton method to find the intersection of each spanwise wing definition curve with the body panels. This module may modify the design variables controlling the wing location relative to the body if the geometry is not realistic. A geometry is realistic when the intersection of the wing leading edge and trailing edge curves with the body may be computed. But the leading edge and the trailing edge of the wing-body intersection should not be too close to the symmetry plane : first it would probably lead to aerodynamic problems, especially at the trailing edge, and finally the grid generator will probably fail as it will be seen in a following section. So it has been specified that the leading edge and the trailing edge of the wing-body intersection should always be at a distance from the symmetry plane greater than half the body radius (fig. 5). The design variables controlling the wing location relative to the body are modified until this requirement is satisfied and this modification of the design variables is passed to CONMIN for a correct estimation of the gradient components and of the move in the search direction. This geometric requirement is not considered as a constraint function to be used by CONMIN because it is not easy to bound the wing location design variables : these are unknown combinations of these variables that are forbidden rather than specific values of the variables, and these forbidden combinations strongly depend on the body shape.

D Control surfaces

The last step is the simulation of the deflection of some control surfaces such as leading edge slats or trailing edge flaps. The simulation may be very simple, with only a modification of the **z** coordinate on the control surface, or much more sophisticated with a true rotation of the control surface with smoothing of the shape around the hinge line. The only constraint comes from the grid generator which needs a wing definition with constant **y** sections and we have to maintain the inboard side and the outboard side of each control surface parallel to the symmetry plane. The control surfaces must be defined by their spanwise extension and by the chordwise extension at each side. For each control surface, a design variable defines the rotation in a plane normal to the hinge line.

2.2.3 Geometric Constraints formulation

The geometric constraints module must be written for each peculiar problem with very simple rules. The routine must store the constraint functions values in a specific array and has to return to ANALIZ only the number of constraint functions that have been effectively defined. Most of the geometric constraints to be defined in aerodynamic design are

wing thickness constraints. If **t** is the actual wing thickness at a given point and t_0 is the required minimum thickness at this point, a thickness constraint function **g** may be defined as $g = \max (t_0 - t)$ computed over a set of points, and as **g** is negative when the constraint is satisfied, it is in a form directly usable by CONMIN and its upper boundary value must be zero.

Additional constraints concerning the wing planform or the location of the wing relative to the body may also be defined. Actually, only four geometric constraint functions are defined, three of them being thickness constraint functions defined as above and the last one concerning the wing height relative to the cabin floor.

A Undercarriage bay thickness constraint

This thickness constraint function is computed over some wing points included in a box bounding the undercarriage bay and where a simple minimum thickness law has been defined. Parameters defining this desired minimum thickness law as well as the boundaries of the undercarriage bay are read at the first call of the routine ANALIZ.

B Spar thickness constraint

This thickness constraint function is computed over a set of points taken along the front spar and the rear spar. The coordinates of these points and the required minimum thickness at each point are read at the first call of the routine ANALIZ.

C Required maximum section thickness

This thickness constraint function concerns the minimum value required for the maximum thickness at each wing section. The constraint function is computed at the point of maximum thickness of some wing sections. The **y** location and the required minimum value for the maximum thickness of the concerned sections are read at the first call of the routine ANALIZ.

D Cabin floor constraint

This is the last geometric constraint actually defined. It states that the upper side of the wing-body intersection must be always under the cabin floor. This constraint function is defined as **g = max (z wing upper surface - z cabin floor)** computed over all points of the upper side of the wing body intersection. Here again, the function is negative when the constraint is satisfied. The parameters defining the cabin floor height along the body are read at the first call of the routine ANALIZ.

2.2.4 Grid Generator

The grid generator builds an O type grid around the wing-body configuration. This topology has been retained because it reduces the number of cells while allowing a good level of refinement near the aircraft surface. As it can be seen on the figure 5, it may be difficult to maintain a good mesh quality when the wing is set very low on the body because the upper side of the wing-body intersection may come very close to the symmetry plane and the cells in this area will be too much skewed. The control of the wing location by the geometry checking module is supposed to avoid this problem.

The grid is built using simple fast algebraic methods as it is shown on the figure 6, so it can be wholly rebuilt each time the geometry has been modified. The wing-body intersection has been already computed by the geometry checking module and a surface mesh is built on the wing-body configuration by interpolation of the longitudinal curves and of the transversal curves of the wing and of the body. Then a full O type grid is built from the surface grid using analytical curves and transfinite interpolations [7]. The surface grid generation including the computation of the wing-body intersection is the most CPU time consuming part of the grid generation.

The outer boundary of the grid is spherical and put at a distance from the aircraft which depends on the aerodynamic conditions. At supersonic Mach numbers, the outer boundary is as close as possible to the aircraft and has some sweep angle depending on the free stream Mach number in such a manner that a Mach line starting from the nose of the aircraft always goes behind the aircraft after it has been reflected on the outer boundary. The figures 7 to 10 show some typical surfaces for a supersonic type fine grid to be used for fine analysis at Mach 2. The wing planform surface around the aircraft (fig. 7), which is the common boundary between upper surface blocks and lower surface blocks, has a heart shape due to the added sweep angle. The surface grid of the wing-body configuration is shown on the figure 8 and it has been chosen to give the body surface grid an H type topology relative to the wing-body intersection. To understand how the grid is built, let imagine that the aircraft skin is swelling up to the outer boundary with a number of layers between the aircraft and the outer boundary. The figure 9 shows an intermediate layer and the figure 10 shows the final layer which is the swept outer boundary.

In the transonic case, the topology remains identical but the outer boundary is put far from the aircraft at a distance which is read in the initialization files and that can be easily changed, and the shape of the boundary does no longer depend on the Mach number. Only a little sweep angle is added so that the boundary points corresponding to the wing tip section are at a good position which gives a regular grid.

2.2.5 Aerodynamic module

The aerodynamic module is an ONERA finite volumes code solving the full unsteady Euler equations using a cell-centered numerical scheme [8] including an explicit stage with a Runge-Kutta scheme followed by an implicit stage with a Lerat-Sides residual smoothing scheme. The stability is ensured by adding artificial viscosity following the Turkel-Jameson method. This code is enough robust to work with high CFL values allowing good rates of convergence and the cell-centered method makes easier the definition of the boundary conditions.

With a coarse optimization grid of 28800 cells and high CFL parameter values, full convergence of aerodynamic computations for a wing-body configuration is obtained within 150 iterations at supersonic Mach numbers and 500 iterations at transonic Mach numbers. The corresponding CPU times on the ONERA CRAY C98 computer are respectively 61 seconds and 175 seconds. This good rate of convergence in the transonic case has been obtained by putting the outer boundary rather close to the aircraft at a distance of about 2 root chords only and of course, the drag value given in the coarse transonic grid is rather false, but we may expect that it varies in the same way as the true inviscid drag.

No convergence test is made during an aerodynamic computation. Instead we prefer to impose a number of iterations for the numerical scheme, so we have to perform some tests before the optimization runs to find the good number of iterations and the good CFL value. As the deformations of the geometry are rather small during an optimization run, we may expect that the convergence rate of the aerodynamic code would not change very much.

The body angle of attack which is passed to the aerodynamic module is computed from the wing reference plane angle of attack and the design variables controlling the wing root section twist angle and the wing root section - body axis angle (fig. 11) :

$$\alpha_{body} = \alpha_{ref} + \Theta - \Phi$$

α_{ref}= Wing reference plane angle of attack

Θ= Wing root Section Twist angle variation

Φ= Wing root Section - Body axis angle variation

The wing reference plane angle of attack may be a design variable, and the body angle of attack may be constrained.

3 Optimization results on Supersonic Transport Design

3.1 Aerospatiale ATSF Wing

3.1.1 Design strategy

The first version of the ONERA optimization code was written for an isolated wing [9], so the first optimization tests were performed on the isolated wing of the Aerospatiale ATSF configuration (fig. 12). The purpose of these tests was to improve the performance of the wing by optimization of its twist and camber laws. Three optimization runs were performed starting from the same initial twisted and cambered ATSF wing :

* twist and camber single point optimization at supersonic cruise conditions,
* twist and camber single point optimization at transonic cruise conditions,
* twist and camber dual point optimization.

During the single point optimization runs, the objective function was the inviscid drag to be minimized. A lower boundary value was imposed on the lift coefficient and two boundary values were imposed on the pitching moment coefficient. The wing reference plane angle of attack was always set to 4° and the wing shape was optimized using simultaneously 25 design variables to control the twist angles and the camber laws of 5 design sections, so 27 or 28 aerodynamic computations (25 + 2 or 3) were required at each minimization cycle.

During the dual point optimization run, the objective function to be minimized was a linear combination of the inviscid drag at supersonic cruise (80%) and of the inviscid drag at transonic cruise (20%). At each cruise point, a

lower boundary value is imposed on the lift coefficient and two boundary values are imposed on the pitching moment coefficient. The wing reference plane angle of attack at the supersonic cruise point was set again to 4° but the wing reference plane angle of attack at the transonic cruise point was a design variable and the optimization was performed using simultaneously 26 design variables, so 56 or 58 aerodynamic computations (2 x { 26 + 2 or 3 }) were required at each minimization cycle.

In addition, a fourth optimization run was performed at supersonic cruise conditions starting from an untwisted and uncambered wing having the same planform and the same thickness law as the ATSF wing.

The lower surface and the upper surface of the wing were both given by 49 sections of 49 points. During optimization runs, the lower surface and the upper surface of the wing were described by only 25 sections of 25 points and only 13 layers were used from the wing surface towards the outer boundary to build the coarse O-type optimization grids. For the isolated wing, full convergence of an aerodynamic computation in the coarse optimization grids could be obtained within 100 iterations at supersonic cruise conditions and within 300 iterations at transonic cruise conditions.

The first supersonic design seemed to have converged after 24 minimization cycles with about 680 aerodynamic computations. The drag reduction estimated from the coarse grid computations was about 4% while the lift at each cycle's end remained well constrained to the initial value. The pitching moment at the end of the run has almost recovered its initial value. Half of the drag reduction was obtained after 7 minimization cycles only with about 200 aerodynamic computations. But the initial version of our grid generator produced grids with poor quality and the same supersonic design with an improved version of the grid generator has converged after 12 minimization cycles only giving approximately the same amount of drag reduction. After these first tests, all following optimization runs have been performed with the improved version of the grid generator.

The transonic design has converged after 19 minimization cycles with about 550 aerodynamic computations. Here the drag reduction was about 17% and the pitching moment at the end of the run has almost recovered again its initial value.

It has been shown in the supersonic design that single precision or double precision aerodynamic computations gave almost the same convergence history. So, due to the number of required aerodynamic computations, it has been decided to perform the dual point design in single precision only. The dual point design has converged after 17 minimization cycles with 500 aerodynamic computations at each cruise point. Starting from 4°, the transonic cruise angle of attack converged towards 3.65°. Here the drag reduction is about 10% at the transonic cruise point and 2% at the supersonic cruise point.

The final optimization test was a supersonic design starting from an untwisted, uncambered wing having the same planform and thickness law as the ATSF wing. The convergence of this design was very low. The angle of attack was set again to 4° and the lift of the flat wing was greater than the desired lift of the ATSF wing. About 2 minimization cycles were necessary to reach the desired lift and the optimization run has converged after 59 minimization cycles. As the performance of the final wing was not as good as the performance of the optimized wing obtained starting from the ATSF wing, the final design variables have been modified to force the camber of the leading edge and the optimization run has been restarted from these modified values for some minimization cycles. This restart procedure has been used a second time and finally, after 70 minimization cycles and 2000 aerodynamic computations, we have obtained a wing having almost the same performance as the optimized wing obtained from the ATSF wing.

3.1.2 Fine analysis of the optimized wings

The fine analysis of each optimized wing has been performed with a refined grid having 25 layers from the wing surface towards the outer boundary, the wing lower surface and the wing upper surface being described by 49 sections of 49 points. This grid is certainly not refined enough but it has the same surface density as the grids used for previous computations on the ATSF wing and on the full ATSF wing-body configuration. The optimization effects on the ATSF wing performance are shown by the bar charts on the figures 13, 15 and 17 where we have plotted the variations of the inviscid drag (first group of bars), of the lift (second group of bars) and of the Lift to Drag ratio (third group of bars) relative to the corresponding values for the initial ATSF wing.

If we consider the performance at the supersonic cruise point M=2 and at 4° angle of attack (fig. 13), we can see that the single point supersonic design has reduced the drag by 4% at constant lift coefficient, so the Lift to Drag ratio has been increased by 4.18%. But if the single point transonic design has reduced the drag by 3.46% at constant angle of attack, the lift could not be maintained and has been reduced by 6.58%, so the Lift to Drag ratio has been reduced by 3.24%. For the dual point design, the lift could be maintained at this angle of attack and the performance improvement is more than half of what has been obtained from the single point supersonic design, the drag being reduced by 2.5% and the Lift to Drag ratio being increased by 2.56%. The pressure distributions obtained at the supersonic cruise point on the basic wing and the three optimized wings are shown on the figure 14. The supersonic design reduces the maximum velocity at the upper surface especially on the trailing part of the low pressure area which follows the highly swept leading edge. The transonic design reduces too much the load on the leading edge of the external part of the wing where

the flow may be not well adapted. The pressure distributions obtained from the dual point design seem intermediate between those obtained with the two single point designs.

If we now consider the performance at the transonic cruise point M=0.95 at 4° angle of attack (fig. 15), we can see that the drag reduction obtained from the single point transonic design at constant lift is 17.46%, which increases the Lift to Drag ratio by 21.20%. The single point supersonic design reduces the drag by 5.41% and the lift by 1.58% at constant angle of attack, and this small lift reduction allows to improve the Lift to Drag ratio by 4.05%. The dual point design has given an optimized angle of attack of 3.65° at M=0.95, and at this angle of attack the drag is reduced by 9.76% and the lift is increased by 1.24%, so the transonic Lift to Drag ratio is increased by 12.30%. Here again the performance improvement obtained from the dual point design is slightly greater than half of what has been obtained from the single point transonic design. The pressure distributions obtained at the transonic cruise point on the basic wing and its three optimized versions are shown on the figure 16. The supersonic design seems to have only a weak influence on the transonic pressure distributions. The transonic design almost completely eliminates the high velocity peak along the external part of the leading edge, but the Mach number before the rear shock wave parallel to the trailing edge seems higher and a constraint on its maximum admissible value should be taken into account for future optimization. Here again, the pressure distributions obtained from the dual point design seem intermediate between those obtained with the two single point designs.

Let us now consider the performance at the supersonic cruise point of the wings obtained from a single point supersonic design. The optimization results obtained starting from the initial ATSF wing and those obtained starting from a flat untwisted, uncambered wing having the same planform and the same thickness law are compared on the figure 17. We can see that the drag reduction and the Lift to Drag ratio improvement relative to the performance of the initial ATSF wing does not depend too much on the starting geometry. However, the design obtained from the initial ATSF wing seems to be slightly better, but maybe the convergence of the optimization run starting from the flat wing could have gone further with more restarts with forced camber at the leading edge. The figure 18 shows the upper surface pressure distributions for the initial wing and the two single point supersonic designs. As we have already seen, the optimization starting from the ATSF wing leads to a quite similar pressure distribution, but starting from a flat wing leads to a very different result : the high velocity area remains very close to the leading edge when we start from the flat wing and it remains aft of the leading edge when we start from the exact ATSF wing.

3.2 European ESCT wing-body configuration

3.2.1 Starting geometry definition

The last version of the ONERA wing-body optimization code has been checked on a simplified model of the European ESCT configuration (fig. 19). The body has a constant circular section along the major part of the wing-body intersection and the wing has been given with no twist and no camber with sections having the same airfoil of constant relative thickness. The wing planform is almost the exact one, the rounded tip being replaced by an extension of the straight leading edge of the external part of the wing in order to simplify the grid generation.

The geometry of the datum configuration is given by a large set of points. The body is given by 166 transversal semi-circular sections of 19 points and the wing is given by 107 longitudinal airfoil sections of 201 points.

3.2.2 Example of supersonic design

For the single point supersonic design, the first stage was the optimization of the thickness law at M=2 and zero angle of attack starting from the flat wing. The geometry generator was used with 5 design sections and 4 main thickness design variables for each section (maximum thickness, longitudinal location of maximum thickness, leading edge radius and trailing edge slope), so the thickness law was entirely controlled by 20 design variables and each optimization cycle requires 22 or 23 evaluations of the objective function. As the thickness law of the geometry generator works in an absolute way, we had first to determine an initial set of thickness design variables to reproduce the airfoil of the datum configuration with a reasonable approximation. This initial set of the thickness design variables has been found after few iterations using an interactive graphic code and then, starting from this initial thickness law, a first thickness optimization run has been performed and has converged within 12 minimization cycles (fig. 20). The objective function was the inviscid drag to be minimized without aerodynamic constraint but the 4 geometric constraints previously defined for the wing-body configuration were taken into account.

Starting now from the untwisted, uncambered wing with this improved thickness law, we had to optimize twist, camber and wing setting on the body with the wing longitudinal location being fixed. During the first runs, the wing was always set at the lowest position allowed by the geometry checking module to eliminate problems with the cabin floor constraint which seemed to be always critical, so the wing setting was controlled only by the wing root section-body axis angle. The twist and camber laws were again defined on 5 design sections with 5 design variables for each section, and the geometry was fully controlled by 26 design variables. The wing reference plane angle of attack was not

a design variable and was always set to 4°. Each minimization cycle requires 28 or 29 evaluations of the objective function.

During these twist and camber optimization runs, the objective function was again the inviscid drag to be minimized with a lower boundary value imposed on the lift coefficient and an upper boundary value imposed on the body angle of attack. Only the three geometric constraints defined inside the body were taken into account : these were the cabin floor constraint, the undercarriage bay constraint and the required maximum section thickness constraint. The convergence history is shown on the figure 21 for the objective function and the figure 22 for the aerodynamic constraints. The strategy of optimization is discussed below.

We first have optimized only the twist law with 5 design variables. The convergence is obtained within 24 cycles, but it seems almost be reached within 14 cycles only. Then we have optimized twist and camber together, but no significative drag reduction could be obtained until the wing thickness inside the body and the sets of points to be taken into account for the geometric constraints have been redefined at cycle 28. Finally a new drag reduction could be obtained, but the pitching moment which have been increased during the twist optimization is now decreased with twist + camber optimization. So an aerodynamic constraint was added at cycle 34 as a lower boundary value on the pitching moment. Convergence of the twist + camber optimization process is reached at cycle 58.

To go further, we have tried several restarts of the twist + camber optimization process, but in a different way of what has been done for the ATSF wing : as the twist + camber laws defined by the design variables are used in an incremental form, we can reset the design variables for zero twist and zero camber deformation and restart a new optimization run from the geometry obtained at the end of a previous optimization run. This restart strategy has been used twice at cycle 56 and cycle 66 and at each time we have obtained a restart of the drag reduction for some minimization cycles (fig. 21). To understand how this restart process works, let us consider what happens on a particular design section with its camber law being defined as a Bezier's curve with 2 intermediate control points : restarting the process from the geometry obtained with a first camber law gives a second camber law depending on 2 other control points and to be added to the first camber law. So the geometry obtained after several restarts has a camber law which may be considered as a linear combination of several camber laws with 2 control points or as a unique camber law depending on more than 2 control points. After two restarts of the twist + camber optimization process, the convergence seems to be reached within 76 cycles.

After these first runs with thickness, twist and camber optimization, some spanwise cuts of the new wing have shown that the thickness law could be further improved. So new thickness optimization runs were started on the geometry obtained at the end of the twist + camber optimization runs. As the thickness law defined by the design variables is used in an absolute form, it was very difficult to improve the first optimized law. The major problem comes from the section thickness constraint and from the spar thickness constraint which were always violated somewhere every time we tried to modify the thickness design variables. The reason is that the thickness constraints are very sensitive to the spanwise interpolation method used between the design sections and it may lead to a wrong computation of their gradient components as it is shown on the figure 23. For instance, when we want to compute the thickness gradient component relative to a design variable linked to a given design section Y2, we have to change only this design variable and look at the new spanwise thickness distribution : in the example shown on the figure 23, if the thickness has increased around the design section Y2, a thickness reduction may occur elsewhere when using cubic spanwise interpolation. The thickness constraint **g** being defined as the maximum value of the difference between the required thickness $\mathbf{t_0}$ and the actual thickness **t**, its value is obtained between the design sections Y3 and Y4. Increasing the thickness at the design section Y2 does not change the maximum value of $\mathbf{g=t_0-t}$ when linear spanwise interpolation is used and it will give a zero gradient component, but using cubic spanwise interpolation will give a non-zero gradient component since the maximum value of $\mathbf{g=t_0-t}$ becomes positive between the design sections Y3 and Y4 where the constraint is violated.

To overcome this problem and further optimize the thickness law, we tried to use linear spanwise interpolation and various optimization techniques : instead of optimizing the whole wing thickness law with 5 design sections and 20 design variables, we first tried to optimize together only the two outer sections using 8 design variables, and finally we tried to optimize individually each design section using 4 design variables each time. In this last case, reduced sets of points were used for the thickness constraints which were checked only on a small area around the design section being optimized.

While improving the thickness law, we tried also to improve the first twist + camber design while keeping the first optimized thickness law. During the first twist + camber optimization runs, the lower boundary value imposed on the lift coefficient CL was somewhat high. From comparisons between coarse grid and fine grid computations on the datum configuration, we have estimated the lift coefficient that should be aimed in the coarse grid to obtain the desired value in the fine grid. So we have restarted the twist + camber optimization with zero deformation design variables from the first optimized geometry with the first optimized thickness law using cubic splines for spanwise interpolation. We have run 20 more minimization cycles with the 25 twist + camber design variables and the wing root section - body axis wing setting variable, an finally we have run 4 last minimization cycles adding the wing height as a 27^{th} design variable.

ONERA

The final geometry combines all the improved results for thickness, twist and camber while using cubic splines for spanwise interpolation. Re-using cubic splines after having used linear spanwise interpolation during the last thickness optimization runs slightly violates some geometric constraints :

- ☆ under-carriage thickness constraint is violated by about 10 mm.
- ☆ spar thickness constraint is violated by 1.3 mm.
- ☆ cabin floor constraint is violated by 2.8 mm.

A fine grid analysis has been performed on the DATUM configuration and on the final supersonic design for the same lift coefficient close to the aimed value 0.115 at M=2.00 and the results are shown with the pressure distributions on the figure 24. The Lift to Drag ratio of the final geometry is 14.20 at CL=0.1146 while it is about 12.04 for the datum configuration at the same CL, so the full optimization of the flat wing configuration has improved its supersonic Lift to Drag ratio by more than 17%. The computed upper surface pressure distributions on the DATUM configuration and on the final supersonic design show that the numerical optimization has greatly reduced the low pressure area along the highly swept leading edge and its extension on the external part of the wing. In addition, it also reduces the flow velocity along the leading edge of the external part of the wing.

This supersonic design may certainly be improved, but we still have to find a better formulation of the thickness law before going further.

3.2.3 Example of leading edge slats optimization at transonic speeds

Starting from the supersonic design discussed above, we have tried to optimize the deflection of some leading edge slats at the transonic cruise point. The leading edge has been split into three contiguous control surfaces, the first one lying on the highly swept part, the second one on the curved middle part and the third one on the straight external part. The control surfaces hinge lines and lateral sides projected on the wing planform are shown on the figure 25. The hinge lines lies along the front spar and the lateral sides are always taken parallel to the symmetry plane for an easier grid generation.

Here the simplified model of control surface deflection has been used : when a control surface is deflected, each point of the geometry lying between the lateral sides and the hinge line is translated along the **z** axis according to its distance from the hinge line. The hinge line end points are projected on the wing mean surface and that gives the hinge line angle Ψxy relative to the **XY** horizontal plane. Then, for each point of the control surface, its distance **Dxy** from a vertical plane passing through the hinge line is computed. Finally, if the control surface deflection angle in a plane normal to the hinge line is Θ positive for the leading edge down, the point translation **dz** along the **z** axis is given by :

$$\mathbf{dz = -Dxy * \tan(\Theta) / \cos(\Psi xy)}$$

Of course, this simplified leading edge droop model does not keep the true shape of the control surface after its rotation, but it gives a good approximation for moderate deflection angles and it does not need to know exactly the **z** position of the hinge line end points.

The optimization test at the transonic cruise point has been made using 4 design variables which were the 3 deflection angles and the aircraft angle of attack. The deflection angles were bounded between 0 and 25°. The objective function to be maximized was the Lift to Drag ratio and the lift was constrained to be at least equal to the value obtained without control surface deflection at cruise conditions. The starting aircraft angle of attack was 3.88° and the aircraft angle of attack was constrained to be less than 4.25°.

In addition, two constraints were imposed on the pressure distributions as upper boundary values for the Mach number normal to the isobars : a first maximum boundary value was imposed in front of the mid-chord line and another maximum boundary value was imposed aft of the mid-chord line. The isobars were assumed to be parallel to the spanwise grid lines. These upper bounds of the normal Mach number should be low enough to avoid flow separation at the shock waves in real viscous flow and they should be guessed from 2D viscous computations or from experimental data. The main difficulty is to find the proper boundary values to be used for the computations in the coarse optimization grids where the shock waves may vanish or numerical oscillations may appear.

The convergence of this optimization test was very sensitive to the boundary values of the pressure constraints and to the scaling factors of the design variables. During the first minimization cycles, the second pressure constraint on the rear part of the wing was quickly critical : as a consequence, the inner slat which has the lowest efficiency on the Lift to Drag ratio has reached at the first minimization cycle its maximum deflection angle, which was first set to 20°, while the pressure constraints limit the deflection angle to about 1.2° for the middle slat and to about 7.5° for the external slat. To get out of this situation, we have first slightly released the pressure constraints and modified the scaling factors of the design variables, then we have increased the maximum deflection angle to 25° and tightened again the pressure constraints. Finally a good solution has been obtained after about 60 minimization cycles. The final aircraft angle of attack is very close to the starting value (3.845° instead of 3.844°) and the three slats deflection angles are :

- ☆ inner slat : 20.6°
- ☆ middle slat : 17.7°
- ☆ external slat : 7.4°

The fine analysis at the transonic cruise point M=0.95 and at the same lift coefficient CL=0.2 shows that the deflection of the slats has improved the Lift to Drag ratio by about 14.8%. The pressure distributions without and with slats deflection at constant lift are shown on the figure 26. With slats deflection, the low pressure peaks are concentrated along the hinge line on the external part of the wing. The maximum normal Mach number on the rear part of the wing seems still somewhat high but maybe it could be reduced with simultaneous optimization of the upward deflection of the trailing edge flaps.

The pressure distributions obtained with slats deflection in the fine analysis grid and in the coarse optimization grid are compared on the figure 27. Here we can see the difficulty of finding a good definition of the pressure constraints for the coarse grids. Near the leading edge and the hinge line, the low pressure peaks are much more important in the fine grid. From the mid-chord line to the trailing edge, the pressure decreases up to the rear shock wave in the fine grid while it is always increasing in the coarse grid.

3.3 Aerospatiale AS4 configuration

The Aerospatiale AS4 wing-body configuration is quite similar to the ESCT one (fig. 28). This configuration has been used to check the capability of the code to optimize the planform at the supersonic cruise point. Here again, the rounded wing tip has been replaced by an extension of the straight leading edge of the external part of the wing in order to simplify the grid generation.

The planform has been optimized at the supersonic cruise point M=2 and Alpha=4°. A design variable defining a sweep angle variation has been attributed to each of the three zones with constant sweep angle at the leading edge and each of the three zones with constant sweep angle at the trailing edge, so the planform was entirely controlled by six design variables. The sweep angle variations on each part of the leading edge and of the trailing edge were bounded to ±5°. The aircraft angle of attack was also a design variable and was constrained to be less than 4.1°. In addition, the lift and the pitching moment were constrained to be greater than or equal to the corresponding value for the initial geometry. The planform optimization with 7 design variables has converged very quickly after 13 minimization cycles only and the final value of the aircraft angle of attack is 3.95°.

Starting from the optimized planform, we have tried to optimize the twist law of the wing at the supersonic cruise point without changing the camber law. Five design variables corresponding to the angle of attack variation of five design sections were used. The wing root section angle relative to the body axis was maintained constant so the aircraft angle of attack was directly linked to the first design section and was not used as a design variable. The same aerodynamic constraints as for the planform optimization were used and the upper boundary value of the aircraft angle of attack has been reduced to 4°. The twist law optimization with 5 design variables has converged after 9 minimization cycles and the aircraft angle of attack has almost reached the initial value (3.995° instead of 4°).

The initial planform and the optimized one are compared on the figure 29. We see that the sweep angle has been slightly reduced on the inner part of the leading edge and has been increased elsewhere. The maximum sweep angle variation of +5° has been reached on the external part of the leading edge and on the middle part of the trailing edge. The variations of the lift coefficient CL, the Lift to Drag ratio CL/CD and the pitching moment coefficient Cm obtained from the fine analysis at the supersonic cruise point are summarized below.

- ☞ Variations obtained from planform optimization relative to the performance of the initial geometry :
  - ☆ ΔCL = +0.15%
  - ☆ Δ(CL/CD) = +2.92%
  - ☆ ΔCm = +0.001296 (absolute increase)

- ☞ Variations obtained from the twist optimization relative to the performance of the geometry with the optimized planform :
  - ☆ ΔCL = -0.11%
  - ☆ Δ(CL/CD) = +0.54%
  - ☆ ΔCm = +0.000256 (absolute increase)

☞ Variations obtained from planform + twist optimization relative to the performance of the initial geometry :

- ☆ ΔCL = +0.04%
- ☆ Δ(CL/CD) = +3.47%
- ☆ ΔCm = +0.001553 (absolute increase)

The improvement of the supersonic Lift to Drag ratio obtained from the planform and the twist law optimization seems very interesting, and the resulting small positive increase of the pitching moment might help to reduce the trimmed drag. But the isobar charts on the figure 30 show that the counterpart is a higher load on the external part of the wing and this may lead to flow separation at the transonic cruise point. We have not yet checked whether flow separation at the transonic cruise point can be avoided or not with the deflection of the leading edge slats, but the problem would certainly become less critical if we accept a limited improvement of only 1 or 2% on the supersonic Lift to Drag ratio with variations of the planform sweep angles less than 5°.

4 Conclusion

Since 1995, ONERA has developed a new optimization code for the aerodynamic design of SST wing-body configurations. This optimization code has been built by coupling the CONMIN minimization code of G.N. Vanderplaats with geometric and meshing routines and an ONERA Euler code for aerodynamic computations. The geometry generator of the optimization code takes into account a great number of design variables which control the wing twist, camber and thickness laws, the wing planform, the wing location relative to the body and the deflection of some leading edge and trailing edge control surfaces.

Significant improvements of the Lift to Drag ratio have been obtained for the isolated Aerospatiale ATSF wing from twist and camber optimization at supersonic and transonic cruise from single point designs as well as from a dual point design.

A good level of performance could be obtained at the supersonic cruise point from the full optimization of the twist, camber and thickness laws of the European ESCT wing-body configuration starting from a flat wing while taking into account realistic geometric constraints. A good level of performance could also be obtained at the transonic cruise point from the optimization of the deflection of some leading edge slats starting from the supersonic design, but it was very difficult to properly define pressure constraints in the coarse optimization grid.

Finally, a planform optimization on the Aerospatiale AS4 wing-body configuration at the supersonic cruise point has shown that the supersonic Lift to Drag ratio could still be significantly improved with small sweep angles variations.

All these optimization attempts have demonstrated the capability of the ONERA code to deal with many design variables and various constraints for a reasonable cost, but the modelisation of the thickness problem could be improved. Also precision problems remain for a correct evaluation of the drag and of the pressure constraints in the transonic range due to the coarseness of the optimization grids. The next developments of the ONERA optimization code could be the improvement of the geometry generator and the introduction of a multi-grid Euler code allowing to work with more refined grids in the transonic range.

From all these tests, it is also obvious that the strategy of optimization is very important and cannot be fully automated. If the code is able to fully optimize a wing-body configuration starting from a flat wing with constant airfoil, the cost of the optimization will be high, especially in the transonic range, and it is not sure that the solution will be the absolute optimum. So it is more efficient to use such numerical optimization methods starting from geometries not too far from the optimum.

ONERA

5 Bibliography

[1] DESTARAC D., RENEAUX J.
Applications de l'optimisation numérique à l'aérodynamique des avions de transport
La Recherche Aérospatiale, Année 1993, n° 2 (Mars-Avril) p. 39 à 55

[2] DOHERTY J.J., PARKER N.T.
Dual-point design of a supersonic transport wing using a constrained optimisation method
Defence Research Agency, Farnborough, United Kingdom

[3] VANDERPLAATS Garret N.
CONMIN - A Fortran Program for Constrained Function Minimization - User's Manual
NASA TM X. 62,282 - August 1973

[4] VANDERPLAATS Garret N.
COPES - A Fortran **C**ontrol **P**rogram for **E**ngineering **S**ynthesis

[5] VANDERPLAATS Garret N.
Numerical optimization techniques for engineering design - With applications
McGraw-Hill Book Company - 1984

[6] BEZIER P.
Procédé de définition numérique des courbes et des surfaces non mathématiques - Système UNISURF
Mécanique-électricité - Août/Septembre 1968

[7] ERIKSSON L.E.
Generation of Boundary-Conforming Grids around Wing-Body Configurations using Transfinite Interpolations
AIAA Journal, Vol. 20 n° 10, October 1982

[8] COUAILLIER V.
Adaptation du code SESAME à une approche "Cell centered" et validation sur un calcul d'aile isolée
ONERA R.T.S. 40/1285 AY - Juillet 1993

[9] GRENON R.
Définition d'une méthode d'optimisation de voilure - Application à un avion de transport supersonique
ONERA RTS n° 7/3733 AY - Décembre 1996

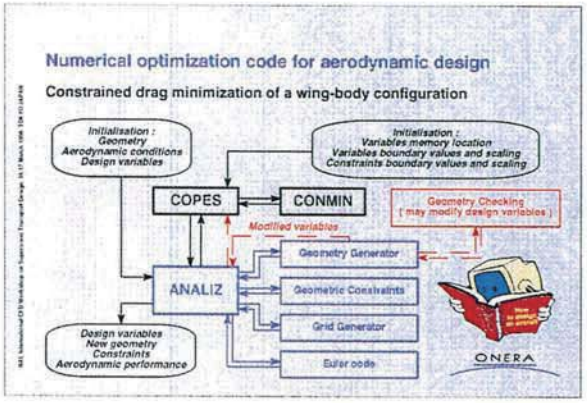

Fig. 1 : Organization of the ONERA wing-body optimization code

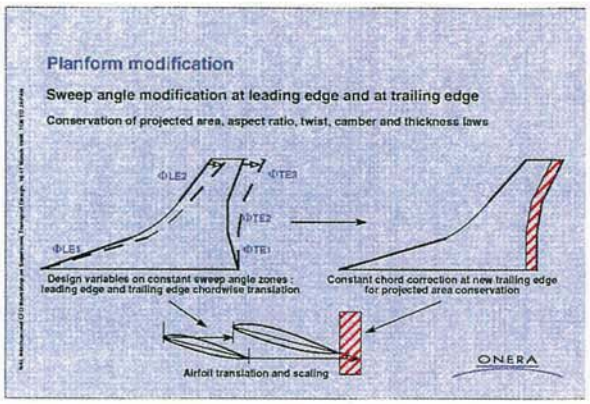

Fig. 4 : Planform control

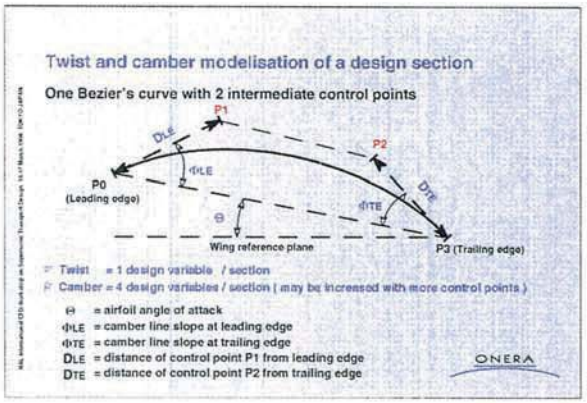

Fig. 2 : Twist and camber control

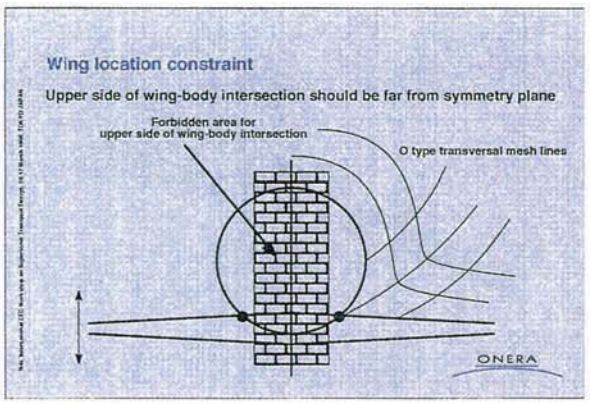

Fig. 5 : Wing location constraint

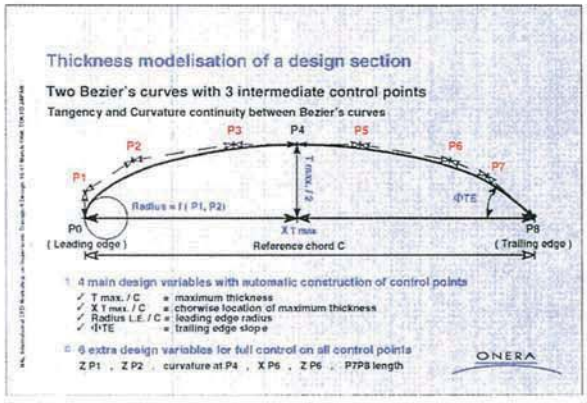

Fig. 3 : Thickness control

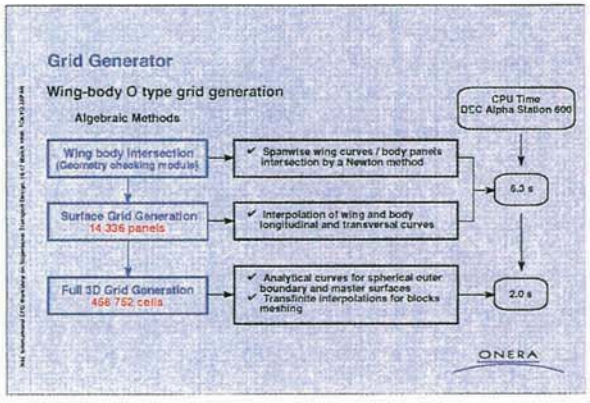

Fig. 6 : Grid generation method

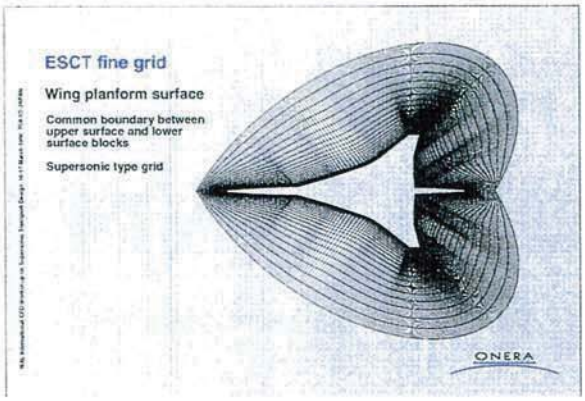

Fig. 7 : ESCT fine grid - Wing planform surface

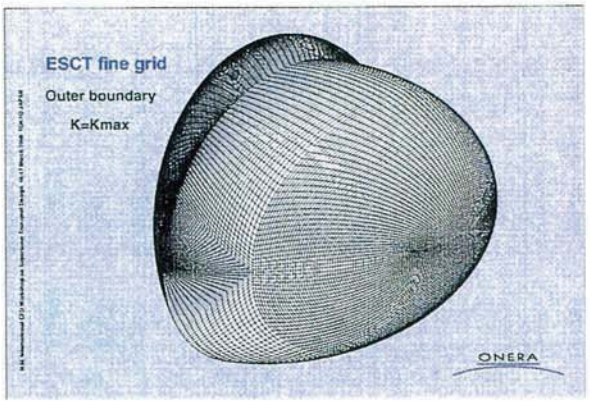

Fig. 10 : ESCT fine grid - Outer boundary K=Kmax

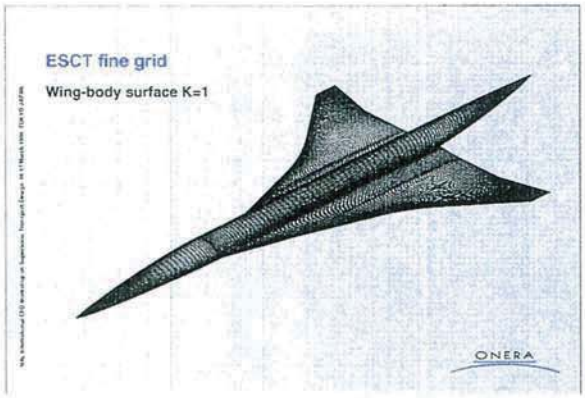

Fig. 8 : ESCT fine grid - Wing-body surface K=1

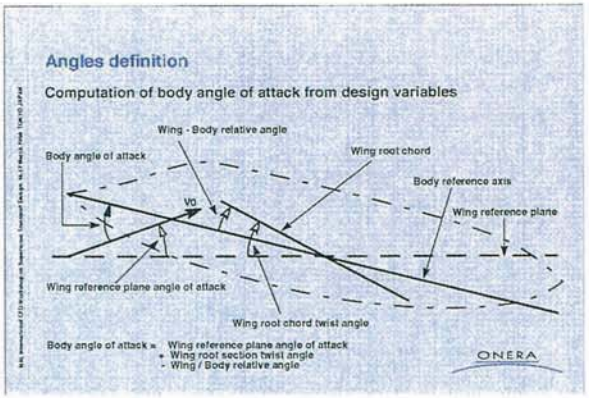

Fig. 11 : Body angle of attack definition

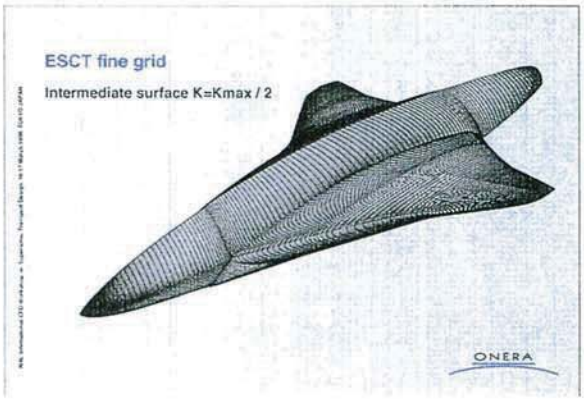

Fig. 9 : ESCT fine grid - Intermediate surface K=Kmax/2

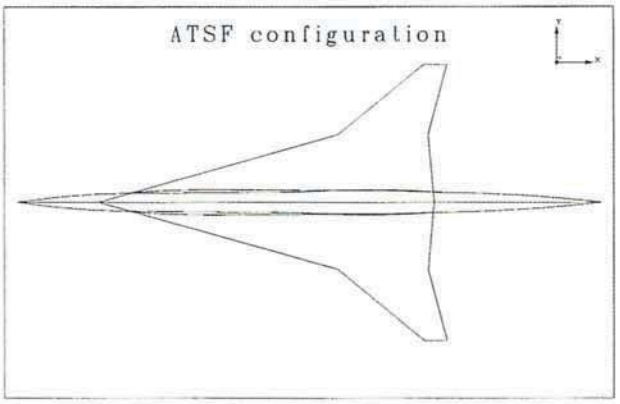

Fig. 12 : Aerospatiale ATSF configuration planform

ONERA

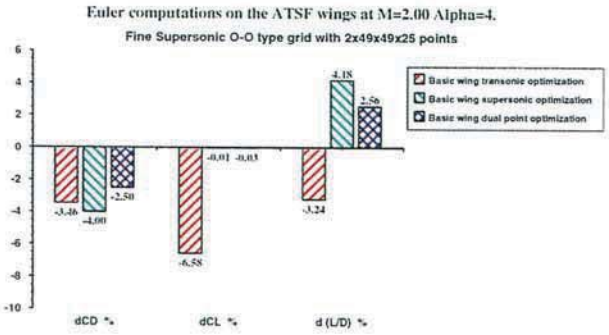

Fig. 13 : Supersonic performance improvement from basic ATSF wing optimization

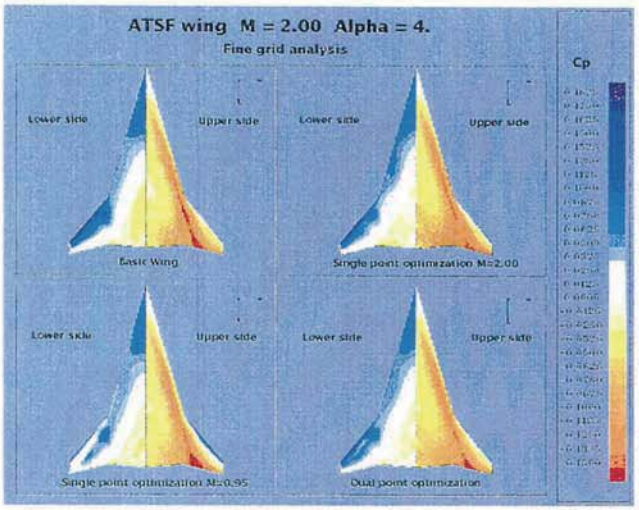

Fig. 14 : Supersonic pressure distributions at M=2.00 on ATSF wings

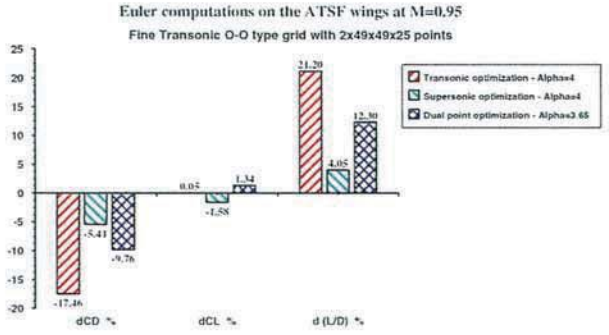

Fig. 15 : Transonic performance improvement from basic ATSF wing optimization

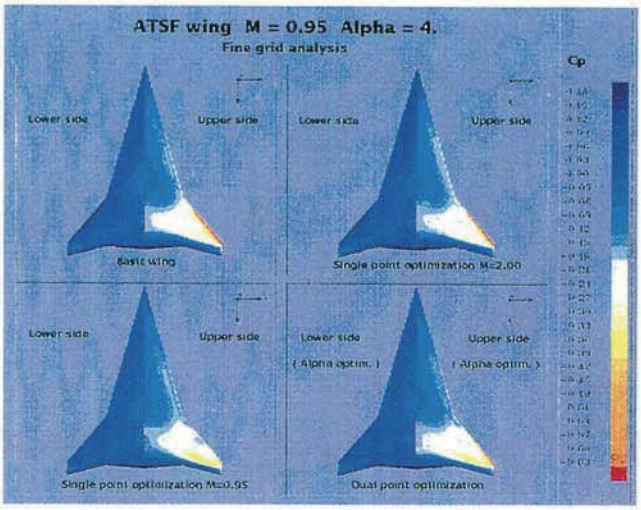

Fig. 16 : Transonic pressure distributions at M=0.95 on ATSF wings

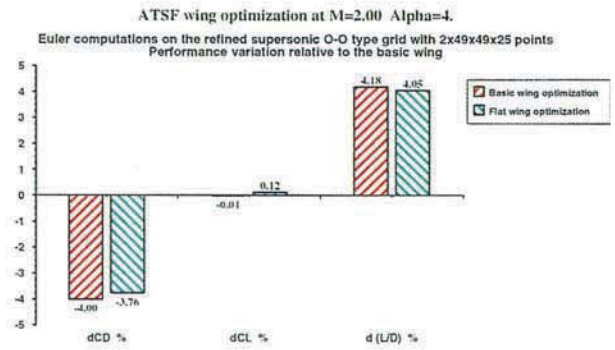

Fig. 17 : Influence of the starting geometry on the supersonic optimization results

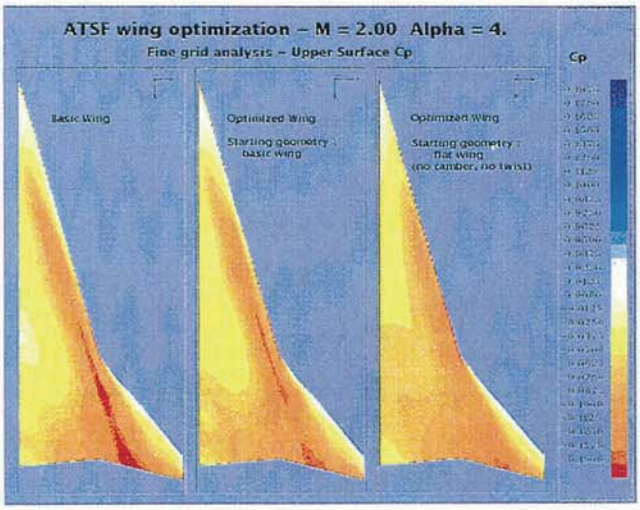

Fig. 18 : Influence of the starting geometry on the final pressure distributions

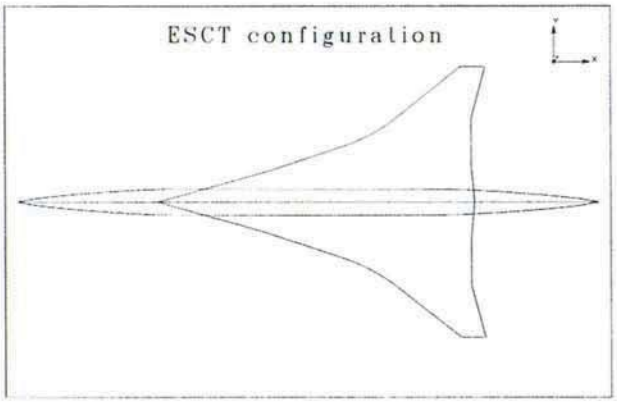

Fig. 19 : Simplified European ESCT configuration planform

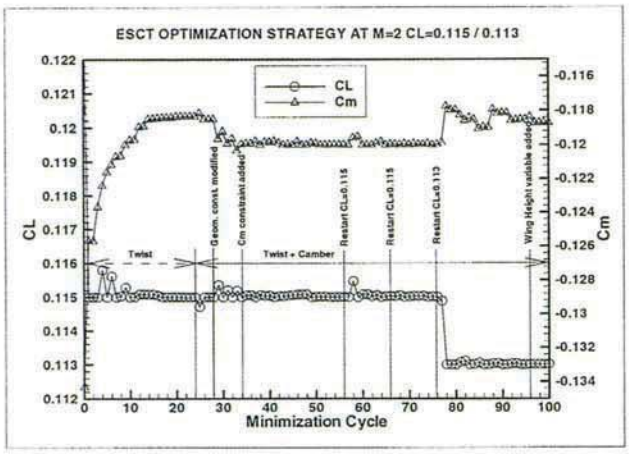

Fig. 22 : History of the aerodynamic constraints for the ESCT supersonic design

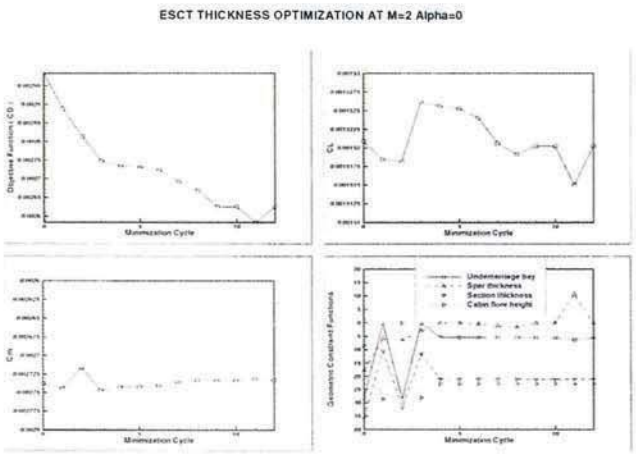

Fig. 20 : History of the ESCT thickness supersonic design at M=2 Alpha=0.

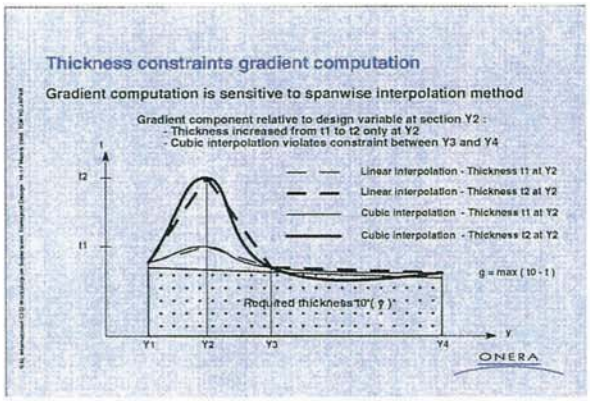

Fig. 23 : Thickness constraints gradient evaluation problems

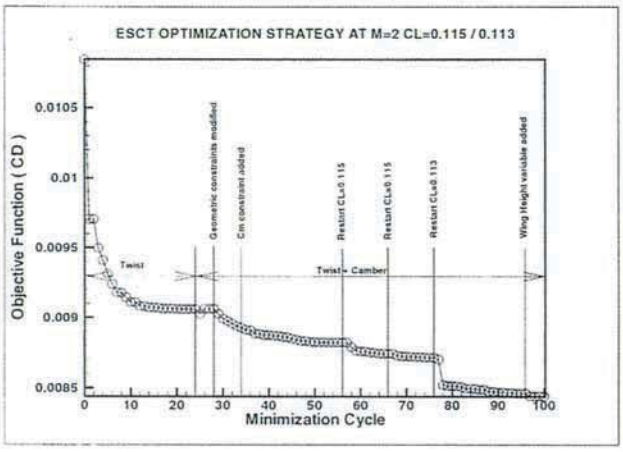

Fig. 21 : History of the ESCT twist+camber supersonic design at M=2

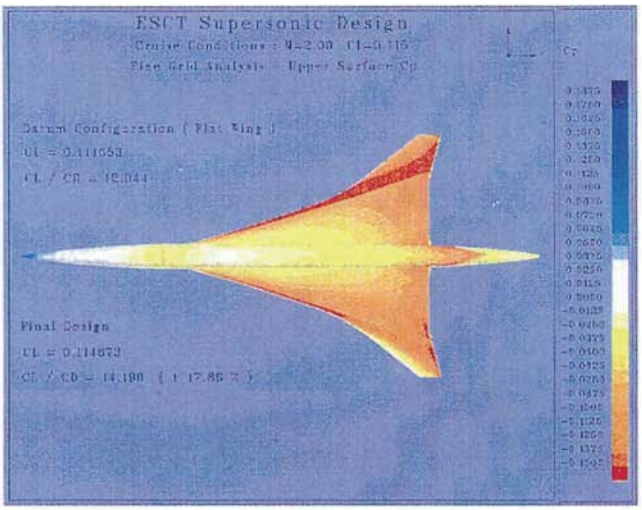

Fig. 24 : ESCT supersonic design at M=2 CL=0.1146 Upper surface pressure distributions

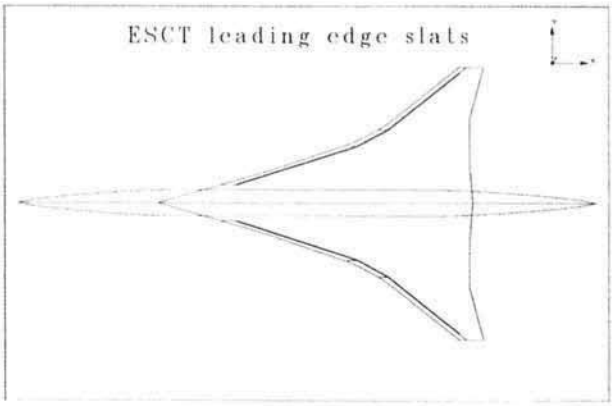

Fig. 25 : ESCT leading edge slats definition

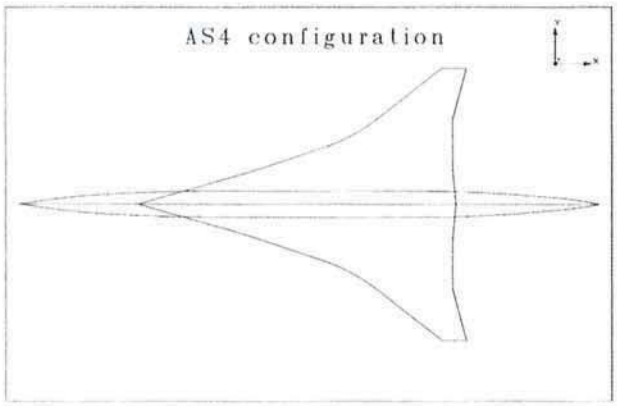

Fig. 28 : Simplified Aerospatiale AS4 configuration planform

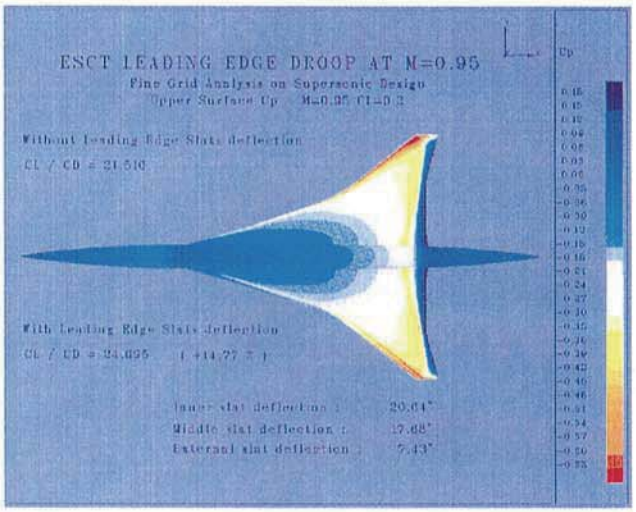

Fig. 26 : ESCT Leading edge droop at M=0.95 CL=0.2 Upper surface pressure distributions

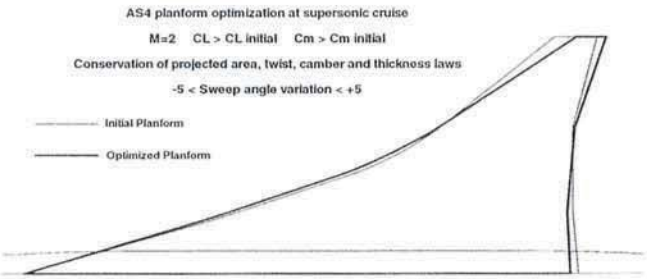

Fig. 29 : Comparison between initial and optimized AS4 planforms

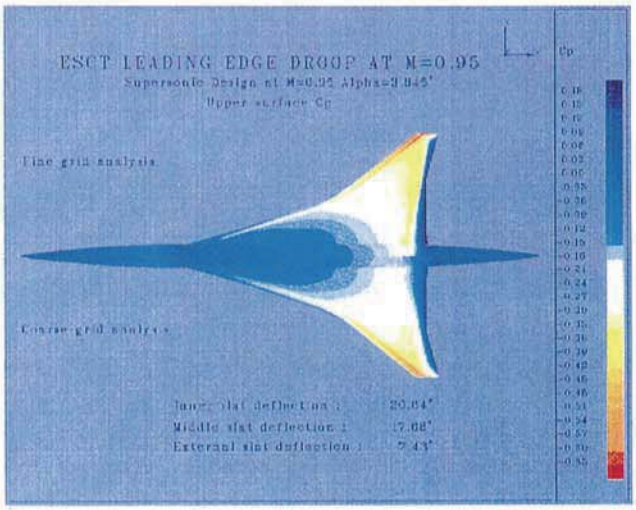

Fig. 27 : Grid coarseness effect on computed transonic pressure distributions

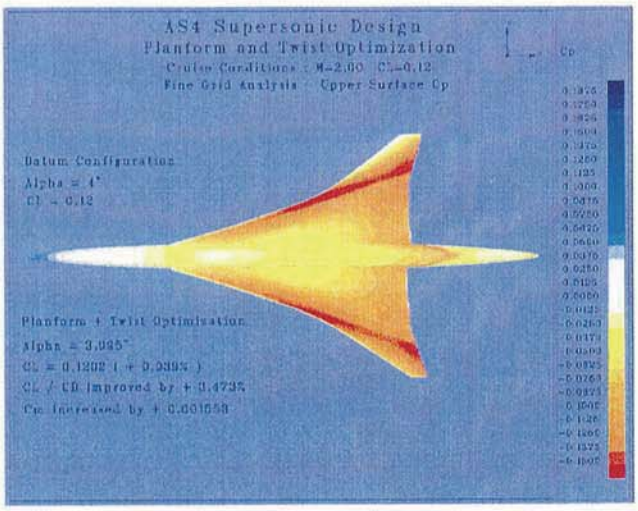

Fig. 30 : AS4 planform and twist optimization at supersonic cruise -Upper surface pressure distributions