43rd Aerospace Sciences Meeting, January 10-13,2005, Reno,NV

Aerodynamic Shape Optimization of Transonic and Supersonic Aircraft configurations

Antony Jameson *, Sriram
Stanford University
Stanford, CA

Luigi Martinelli †
Princeton University
Princeton, NJ

Susan Cliff ‡, Scott Thomas §
NASA Ames
Moffet Field, CA

I. Introduction

With the availability of high performance computing platforms and robust numerical methods to simulate fluid flows, it is possible to shift attention to automated design procedures which use CFD combined with gradient-based optimization techniques. Typically, in gradient-based optimization techniques, a control function to be optimized (the wing shape, for example) is parameterized with a set of design variables and a suitable cost function to be minimized is defined. For aerodynamic problems, the cost function is typically lift, drag or a specified target pressure distribution. Then, a constraint, the governing equations, can be introduced in order to express the dependence between the cost function and the control function. The sensitivity derivatives of the cost function with respect to the design variables are calculated in order to get a direction of improvement. Finally, a step is taken in this direction and the procedure is repeated until convergence is achieved. Finding a fast and accurate way of calculating the necessary gradient information is essential to developing an effective design method since this can be the most time consuming portion of the design process. This is particularly true in problems which involve a very large number of design variables as is the case in a typical three dimensional shape optimization.

The control theory approach[8 7 9] has dramatic computational cost advantages over the finite-difference method of calculating gradients. The control theory approach is also called the adjoint method as the necessary gradients are obtained through the solution of an adjoint system of equations of the governing equations of interest. The adjoint method is extremely efficient since the computational expense incurred in the calculation of the complete gradient is effectively independent of the number of design variables.

In this study, a continuous adjoint formulation has been used to derive the adjoint system of equations. Hence, the adjoint equations are derived directly from the governing equations and then discretized. Hence, this approach has the advantage over the discrete adjoint formulation in that the resulting adjoint equations are independent of the form of discretized flow equations. The adjoint system of equations have a similar form to the governing equations of the flow and hence the numerical methods developed for the flow equations[1 2 3] can be reused for the adjoint equations. The gradient is derived directly from the adjoint solution and the surface motion independent of the mesh modification. This is critical for our design methodology to work on unstructured meshes. If the gradient depends on the form of the mesh modification, then the field integral in the gradient calculation has to be recomputed for mesh modifications corresponding to each design variable. To reduce the computational cost with this approach, the number of design variables would have to reduced by parameterizing the geometry[5 4 .6] However, this reduced set of design variables would be incapable of

*Professor, Stanford University
†Professor, Princeton University
‡Aerospace Engineer, NASA Ames
§Programmer/Analyst, Raytheon

recovering all possible shape variations. Using the gradients computed with our new formulation, a steepest descent method is used to improve an existing design.

Viable Supersonic Business Jet (SBJ) designs are currently being actively explored by many aircraft companies within the US and outside. In this paper we plan to use the adjoint based design method to perform shape optimization of SBJ configurations.

II. The general formulation of the Adjoint Approach to Optimal Design

For flow about an airfoil, wing or sail, the aerodynamic properties which define the cost function are functions of the flow-field variables, w, and the physical location of the boundary, which may be represented by the function, $\mathcal{F}$, say. Then

$$I = I(w, \mathcal{F}),$$

and a change in $\mathcal{F}$ results in a change

$$\delta I = \frac{\partial I^T}{\partial w}\delta w + \frac{\partial I^T}{\partial \mathcal{F}}\delta \mathcal{F}, \tag{1}$$

in the cost function. Using control theory, the governing equations of the flow field are introduced as a constraint in such a way that the final expression for the gradient does not require re-evaluation of the flow-field. In order to achieve this, δw must be eliminated from equation 1. Suppose that the governing equation R which expresses the dependence of w and $\mathcal{F}$ within the flow field domain D can be written as

$$R(w, \mathcal{F}) = 0 \tag{2}$$

Then δw is determined from the equation

$$\delta R = \left[\frac{\partial R}{\partial w}\right]\delta w + \left[\frac{\partial R}{\partial \mathcal{F}}\right]\delta \mathcal{F} = 0 \tag{3}$$

Next, introducing a Lagrange Multiplier ψ, we have

$$\delta I = \frac{\partial I^T}{\partial w}\delta w + \frac{\partial I^T}{\partial \mathcal{F}}\delta \mathcal{F} - \psi^T\left(\left[\frac{\partial R}{\partial w}\right]\delta w + \left[\frac{\partial R}{\partial \mathcal{F}}\right]\delta \mathcal{F}\right)$$

$$\delta I = \left(\frac{\partial I^T}{\partial w} - \psi^T\left[\frac{\partial R}{\partial w}\right]\right)\delta w + \left(\frac{\partial I^T}{\partial \mathcal{F}} - \psi^T\left[\frac{\partial R}{\partial \mathcal{F}}\right]\right)\delta \mathcal{F}$$

Choosing ψ to satisfy the adjoint equation

$$\left[\frac{\partial R}{\partial w}\right]^T \psi = \frac{\partial I}{\partial w} \tag{4}$$

the first term is eliminated and we find that

$$\delta I = \mathcal{G}\delta \mathcal{F} \tag{5}$$

where

$$\mathcal{G} = \frac{\partial I^T}{\partial \mathcal{F}} - \psi^T\left[\frac{\partial R}{\partial \mathcal{F}}\right] \tag{6}$$

This process allows for elimination of the terms that depend on the flow solution with the result that the gradient with respect with an arbitrary number of design variables can be determined without the need for additional flow field evaluations.

After taking a step in the negative gradient direction, the gradient is recalculated and the process repeated to follow the path of steepest descent until a minimum is reached. In order to avoid violating constraints, such as the minimum acceptable wing thickness, the gradient can be projected into an allowable subspace within which the constraints are satisfied. In this way one can devise procedures which must necessarily converge at least to a local minimum and which can be accelerated by the use of more sophisticated descent methods such as conjugate gradient or quasi-Newton algorithms. There is a possibility of more than one local minimum, but in any case this method will lead to an improvement over the original design.

III. Gradient formulations

Continuous adjoint formulations have generally used a form of the gradient that depends on the manner in which the mesh is modified for perturbations in each design variable. To represent all possible shapes the control surface should be regarded as a free surface. If the surface mesh points are used to define the surface, this leaves the designer with a thousands of design variables. On an unstructured mesh evaluating the gradient by perturbing each design variable in turn, would be prohibitively expensive because of the need to determine corresponding perturbations of the entire mesh. This would inhibit the use of this design tool in any meaningful design process. Hence an alternate formulation to the gradient calculation is followed in this study. This idea was developed by Jameson[11,9] and was validated for two and three dimensional problems with structured grids. However, as it is possible to devise mesh modification routines that are computationally cheap on structured grids, the major benefit of this alternate gradient formulation is for general three dimensional unstructured grids. To complete the formulation of the control theory approach to shape optimization, the gradient formulations are outlined next. The formulation for the reduced gradients in the continuous limit is presented in the context of transformation between the physical domain and the computational domain, and are easily extended to unstructured grid methods where these transformations are not explicitly used.

Let

$$Q = JK^{-1}$$

where

$$K_{ij} = \frac{\partial x_i}{\partial \xi_j}, J = det(K)$$

then the transformed equations are

$$\frac{\partial F_i}{\partial \xi_i} = \frac{\partial (Q_{ij} f_j)}{\partial \xi_i} = 0$$

A shape modification causes a flux change

$$\delta F_i = \delta Q_{ij} f_j + C_i \delta w$$

where

$$C_i = Q_{ij} \frac{\partial f_j}{\partial w}$$

One can augment the cost variation by

$$\int_{\mathcal{D}} \psi^T \frac{\partial \delta F_i}{\partial \xi_i} d\xi = \int_{\mathcal{B}} n_i \psi^T \delta F_i d\xi_{\mathcal{B}} - \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi} \delta F_i d\xi$$

and choosing ψ to satisfy the adjoin equation the field integral is reduced to

$$\int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi} (\delta Q_{ij} f_j) d\xi$$

The evaluation of this term requires the evaluation of the metric variations δQ_{ij}. The true gradient should not depend on the way the mesh is modified. consider the case of a mesh variation with a fixed boundary. Then

$$\delta I = 0$$

but there is a variation in the transformed flux

$$\delta F_i = \delta Q_{ij} f_j + Q_{ij} \frac{\partial f_j}{\partial w} \delta w$$

Here the true solution is unchanged, so the variation δw is due to the mesh movement δx at fixed ξ. Therefore

$$\delta w = \Delta w . \delta x = \frac{\partial w}{\partial x_j} \delta x_j$$

and since

$$\frac{\partial \delta F_i}{\partial \xi} = 0$$

it follows that

$$\int_{\mathcal{D}} \psi^T \frac{\partial (\delta Q_{ij} f_j)}{\partial \xi_i} d\xi = -\int_{\mathcal{D}} \psi^T Q_{ij} \frac{\partial f_j}{\partial w} \delta w d\xi$$

or

$$\int_{\mathcal{D}} \psi^T \frac{\partial (\delta Q_{ij} f_j)}{\partial \xi_i} d\xi = \int_{\mathcal{D}} C_i \frac{\partial w}{\partial x_j} \delta x_j d\xi$$

A similar relationship can be derived in the general case with boundary movement and the complete derivation will be presented in the final version of the paper.

Now,

$$\begin{aligned} \int_{\mathcal{D}} \psi^T \delta R d\mathcal{D} &= \int_{\mathcal{D}} \frac{\partial}{\partial \xi_i} C_i (\delta w - \delta w^*) d\mathcal{D} \\ &= \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_i} C_i (\delta w - \delta w^*) \\ &= \int_{\mathcal{B}} \psi^T C_i (\delta w - \delta w^*) d\mathcal{B} \end{aligned} \tag{7}$$

Hence on the wall boundary

$$C_2 \delta w = \delta F_2 - \delta S_{2j} f_j$$

Thus by choosing ψ to satisfy the adjoint equation and the adjoint boundary condition, we have the following expression for the reduced gradient.

$$\begin{aligned} \delta I = \int_{\mathcal{B}} \psi^T (\delta S_{2j} f_j + C_2 \delta w^*) d\xi_1 d\xi_3 - \\ \int \int_{\mathcal{B}} (\delta S_{21} \psi_2 + \delta S_{22} \psi_3 + \delta S_{23} \psi_4) p d\xi_1 d\xi_3 \end{aligned} \tag{8}$$

A. The need for a Sobolev inner product in the definition of the gradient

Another key issue for successful implementation of the continuous adjoint method is the choice of an appropriate inner product for the definition of the gradient. It turns out that there is an enormous benefit from the use of a modified Sobolev gradient, which enables the generation of a sequence of smooth shapes. This can be illustrated by considering the simplest case of a problem in calculus of variations.

Choose $y(x)$ to minimize

$$I = \int_a^b F(y, y') dx$$

with fixed end points $y(a)$ and $y(b)$. Under a variation $\delta y(x)$,

$$\begin{aligned} \delta I &= \int_a^b \left(\frac{\partial F}{\partial y} \delta y + \frac{\partial F}{\partial y'} \delta y' \right) dx \\ &= \int_a^b \left(\frac{\partial F}{\partial y} - \frac{d}{dx} \frac{\partial F}{\partial y'} \right) \delta y dx \end{aligned}$$

Thus defining the gradient as

$$g = \frac{\partial F}{\partial y} - \frac{d}{dx} \frac{\partial F}{\partial y'}$$

and the inner product as

$$(u, v) = \int_a^b uvdx$$

we find that

$$\delta I = (g, \delta y)$$

Then if we set

$$\delta y = -\lambda g, \qquad \lambda > 0$$

we obtain a improvement

$$\delta I = -\lambda(g, g) \leq 0$$

unless $g = 0$, the necessary condition for a minimum. Note that g is a function of $y, y^{'}, y^{''}$,

$$g = g(y, y^{'}, y^{''})$$

In the case of the Brachistrone problem, for example

$$g = -\frac{1 + y^{'2} + 2yy^{''}}{2\left(y(1 + y^{'2})\right)^{3/2}}$$

Now each step

$$y^{n+1} = y^n - \lambda^n g^n$$

reduces the smoothness of y by two classes. Thus the computed trajectory becomes less and less smooth, leading to instability.

In order to prevent this we can introduce a modified Sobolev inner product[10]

$$\langle u, v\rangle = \int (uv + \epsilon u^{'} v^{'})dx$$

where ϵ is a parameter that controls the weight of the derivatives. If we define a gradient $\overline{g}$ such that

$$\delta I = \langle \overline{g}, \delta y\rangle$$

Then we have

$$\begin{aligned} \delta I &= \int (\overline{g}\delta y + \epsilon \overline{g}^{'} \delta y^{'})dx \\ &= \int (\overline{g} - \frac{\partial}{\partial x}\epsilon\frac{\partial \overline{g}}{\partial x})\delta y dx \\ &= (g, \delta y) \end{aligned}$$

where

$$\overline{g} - \frac{\partial}{\partial x}\epsilon\frac{\partial \overline{g}}{\partial x} = g$$

and $\overline{g} = 0$at the end points. Thus $\overline{g}$ is obtained from g by a smoothing equation.

Now the step

$$y^{n+1} = y^n - \lambda^n \overline{g}^n$$

gives an improvement

$$\delta I = -\lambda^n \langle \overline{g}^n, \overline{g}^n\rangle$$

but y^{n+1} has the same smoothness as y^n, resulting in a stable process.

In applying control theory for aerodynamic shape optimization, the use of a Sobolev gradient is equally important for the preservation of the smoothness class of the redesigned surface and we have employed it to obtain all the results in this study.

IV. Imposing Thickness Constraints on Unstructured Meshes

In order to perform meaningful drag reduction computations, it is necessary to ensure that constraints such as the thickness of the wing are satisfied during the design process. On an arbitrary unstructured mesh there appears to be no straightforward way to impose thickness constraints. In our approach we introduce cutting-planes at various span-wise locations along the wing and tranform the airfoil sections to shallow bumps by a square root mapping. Then we interpolate the gradients from the nodes on the surface to the airfoil sections on the cutting-planes, and impose the thickness constraints on the mapped sections. The displacements of the points on the surface of the CFD mesh are obtained by interpolation from the mapped airfoil sections, and transformed back to the physical domain by a reverse mapping. These surface displacements are finally used as inputs to a mesh deformation algorithm.

V. Mesh Deformation

The modifications to the shape of the boundary are transferred to the volume mesh using the spring method. This approach has been found to be adequate for the computations performed in this study.

The spring method can be mathematically conceptualized as solving the following equation

$$\frac{\partial \Delta x_i}{\partial t} + \sum_{j=1}^{N} K_{ij}(\Delta x_i - \Delta x_j) = 0$$

where the K_{ij} is the stiffness of the edge connecting node i to node j and its value is inversely proportional to the length of this edge, Δx_i is the displacement of node i and Δx_j is the displacement of node j, the opposite end of the edge. The position of static equilibrium of the mesh is computed using a Jacobi iteration with known initial values for the surface displacements.

VI. Overview of the Design Process

A flow-chart describing the overall design process is shown in figure 1.

VII. Parallel Implementation of the Flow Solver

To exploit the availability of modern parallel computing platforms, the baseline computational program, SYNPLANE, was parallelized. Due to the unstructured nature of the computational grid, a wide variety of possible data structures to implement the underlying numerical algorithms exist. The following sections outline the choice of data structures and algorithms that were made to parallelize the flow solver.

A. Domain Decomposition, Load Balancing

A modified coordinate bisection method was used to recursively divide a given computational mesh into sub-domains (figure 2). The sub-domains were created such that they contained approximately the same number of computational nodes. No effort was made to optimize the domain decomposition process so as to minimize the number of edges that are shared between the sub-domains. The sub-domains are distributed among the available processors in a manner that minimizes a combination of the computational cost associated with the domains in each processor and the cost of communication among the processors for a given distribution. This methodology was found to result in well balanced load distributions. Once the partitions are distributed among the processors, data structures that allow for the exchange of information along processor boundaries are constructed.

As the flow solver uses an edge-based data structure to accumulate the fluxes at each vertex, the edges surrounding the nodes that lie within a partition are accumulated. If an edge connects a pair of nodes that lie across processor boundaries, this edge is duplicated in the two processors and "halo" nodes are constructed for both processors. This idea is illustrated further in figure 3.

B. Parallel implementation of the multigrid algorithm

The use of multigridding techniques for flow analysis necessitates the need to exchange information between a fine and coarse grid point and vice-versa. While the residuals from the fine grid points need to be accumulated at the coarse grid points, the corrections from the coarse grid points need to be transferred back to the fine grid points. In this study a non-nested approach to multigridding has been used and the sequence of meshes are independently generated from a mesh generator. Hence, efficient methods have to be used to generate the interpolation coefficients by identifying the cell in the fine grid or coarse grid that contains a given point.

Our search methodology was implemented using an octree search routine, where the computational domain was recursively subdivided into octants(figure 4). Each octant was allowed to hold a certain number of points. Once an octant contains more than an user specified set of points, then the octant is sub-divided to create 8 new octants. A tree data structure to hold the octants and the its extents is first determined for each mesh. Using this data structure, a given point is identified within an octant and the node closest to a point in this octant is determined. The cells surrounding this point are checked to see if they contain the search point. This check is done by describing circum-spheres around each cell to see if the point lies within it. The search for an octant that contains a given point is very efficient as it recursively eliminates a substantial portion of the computational domain by restricting the seach to a particular octant and its sub-divisions. Octree based search routines are routinely used in computational geometry problems that require point searches, as mesh generation and computer animation. The typical trade-offs in octree based point search algorithms involve the cost involved with creating and storing the octant information which is usually a function of the number of points that a given octant is allowed to contain.

The octree based search routine has been found to be very useful for the implementation of the non-nested multigrid methods as the major cost during the pre-processing step is the associated with the point search routines to compute the interpolation co-efficients between successive meshes. Using the octree data structure, all the interpolation coefficients between a fine grid node and the next coarser mesh in the multigrid cycle are constructed in a pre-processing step. Further, to reduce the cost communication across processors during the transfer of information between the fine and coarse grids, sub-domains on the coarser grids are constructed and distributed so as to conform to the division and distribution of the fine mesh. Typical computational times for the building of the octree and the subsequent point search algorithms are in the range of a few minutes for a mesh containing a million nodes.

VIII. Results

A. Shape optimization for Transonic Jets

The design method has been applied to several complete aircraft configurations, including a transonic business jet. As shown in figures 5, 6, 7, 8, the outboard sections of the wing have a strong shock while flying at cruise conditions. The results of a drag minimization exercise that removes the shocks on the wing are shown in figures 9, 10, 11, 12. The drag has been reduced from 235 counts to 215 counts in about 8 design cycles. The lift was contrained at 0.4 by perturbing the angle of attack. Further, the original thickness of the wing was maintained during the design process ensuring that fuel volume and structural integrity will be maintained by the redesigned shape. The parallel version of SYNPLANE takes under an hour to redesign the wing shape on 8 1.7 Ghz Athlon processors with a communication bandwidth comparable to ethernet.

The computational program has also been tested on other aircraft geometries like the Gulfstream *GIV* and a generic business jet configuration from NASA Ames and found to reduce transonic drag by 10 to 18 counts. Figures(13,14,15,16) shows the pressure distribution before and figures(17,18,19,20) after the redesign. The shape optimization procedure resulted in a drag reduction of 11 counts.

B. Shape Optimization of Supersonic Business Jets

The design method has also been applied to a supersonic business jet configuration. The fine mesh in the computation had around 900000 nodes and three levels of multigrid were used for the computations. It was observed that due to the finer resolution of the triangulation of the wing surface the number of cutting planes had to be increased to decrease the interpolation error associated with imposing the procedure to impose thickness constraints on the unstructured mesh. Additionally, it was also observed that instead of shape changes that were normal to the geometry, shape changes in the vertical direction resulted in improved drag

reduction. Figures (21,22,23,24) show that the initial geometry had a weak shock that extended over the length of the wing planform. Figures (25,26,27,28) show that the result of shape optimization has almost eliminated the shock and resulted in a drag reduction of about 3 drag counts. In addition, the first author has done numerous computations with other supersonic wing-body configurations with structured grid codes and the adjoint procedure has always resulted in reduction in drag. He has also used the leading edge camber as another design variable that was found to further reduce the drag.

IX. Conclusions

In this study we have used an adjoint based method to control the shape of the aircraft using unstructured grids to represent the computational domain. The computational package can currently execute on parallel machines and hence provides a fast design tool that can be used to analyze and design complete aircraft geometries in under an hour. For supersonic configurations we currently obtain modest improvements in drag. In the future we plan to perform some theoretical studies on wings in supersonic flows and ensure that the numerical method produces correct estimates of the lift and drag and also performs meaningful changes to the shape. The current computational package has also been integrated with a boundary layer code with which we plan to perform viscous redesigns of supersonic configurations.

X. Acknowledgements

The authors greatfully acknowledge support for this research from a NASA Ames grant from the System Analysis Group administered by Mary Livingstone. They would also like to thank Georg May for generating the coarser meshes used in the multigrid calculations, Mike Harbeck for incorporating robust cutting plane routines and boundary layer computations and Don Howe from Gulfstream for generating the meshes for the aircraft configurations and Timothy Baker for providing the mesh for the Falcon geometry.

References

[1]A. Jameson, W. Schmidt and E. Turkel Numerical Solution of the Euler equations by finite volume methods using Runger-Kutta time stepping schemes. *AIAA Paper 81-1259*, June, 1981.

[2]A. Jameson and T.J. Baker Improvements to the Aircraft Euler Method. *AIAA Paper 87-0353*, 25^{th} AIAA Aerospace Sciences Meeting, Reno, January, 1987.

[3]T.J. Barth Apects of unstructured grids and finite volume solvers for the Euler and Navier-Stokes equations. *AIAA Paper 91-0237*, 29^{th} AIAA Aerospace Sciences Meeting, Reno, January, 1994.

[4]J. Elliot and J. Peraire Aerodynamic design using unstructured meshes *AIAA Paper 96-1941*, 33^{rd} AIAA Aerospace Sciences Meeting, Reno January, 1996

[5]K. Anderson and V. Venkatakrishnan Aerodynamic Design Optimization on Unstructured grids using a continuous adjoint formulation *AIAA Paper 97-0643*, 34^{th} AIAA Aerospace Sciences Meeting, Reno, January, 1997

[6]S. E. Cliff, S.D. Thomas, T. J. Baker, A. Jameson and R. M. Hicks Aerodynamic Shape optimization using unstructured grid method *AIAA Paper 02-5550*, 9^{th} AIAA Symposium on Multidisciplinary Analysis and Optimization, Atlanta, September, 2002.

[7]O. Pironneau. *Optimal Shape Design for Elliptic Systems*. Springer-Verlag, New York, 1984.

[8]J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, 1971. Translated by S.K. Mitter.

[9]A. Jameson. Optimum Aerodynamic Design Using Control Theory Computational Fluid Dynamics Review 1995 Wiley, 1995.

[10]A. Jameson, L. Martinelli and J. Vassberg Using CFD for Aerodynamics - A critical Assesment *Proceedings of ICASE 2002*, September 8-13, 2002, Toronto, Canada

[11]A. Jameson, Sangho Kim Reduction of the Adjoint Gradient Formula in the Continuous Limit *AIAA Paper*, 41^{st} AIAA Aerospace Sciences Meeting, Reno January, 2003

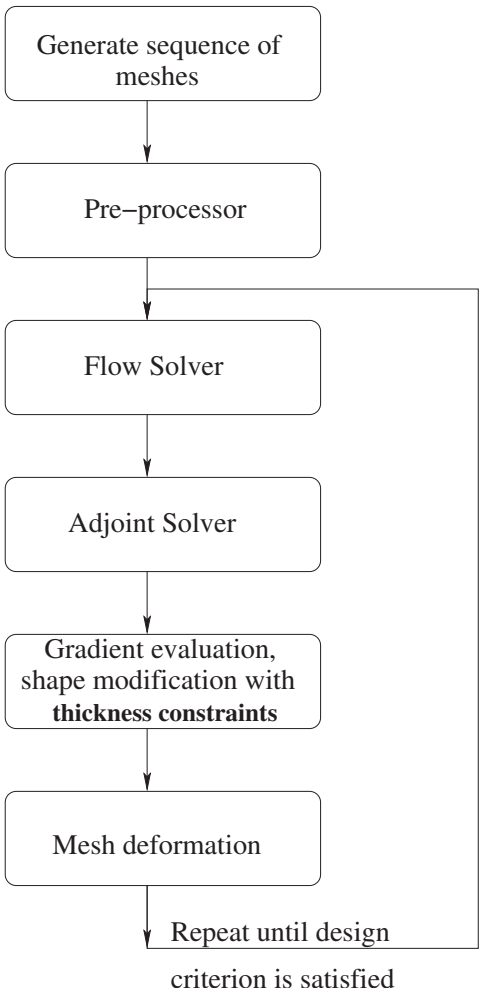

Figure 1. Flow chart of the overall design process

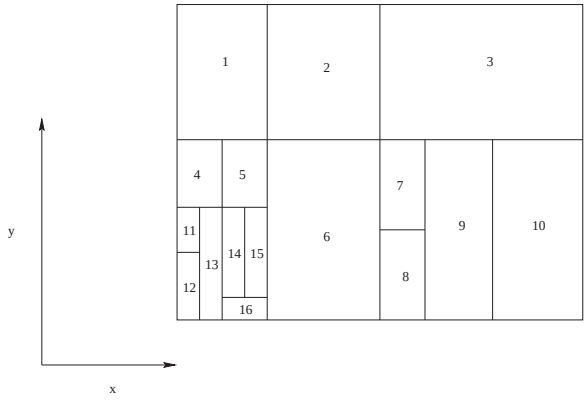

Figure 2. Domain decomposition of a rectangular region using a modified bisection method

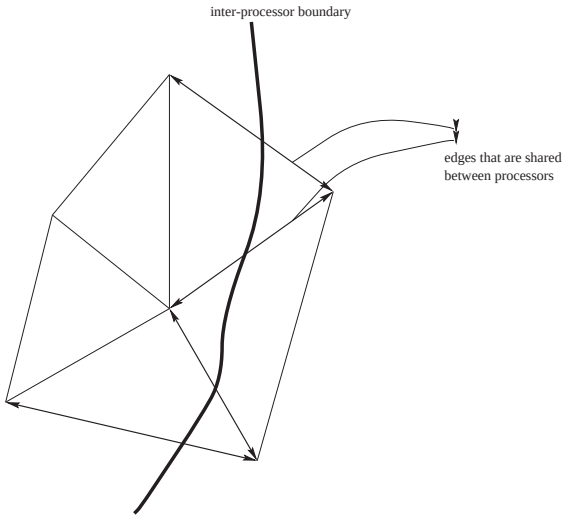

Figure 3. Halo nodes and the distribution of edges along processor boundaries

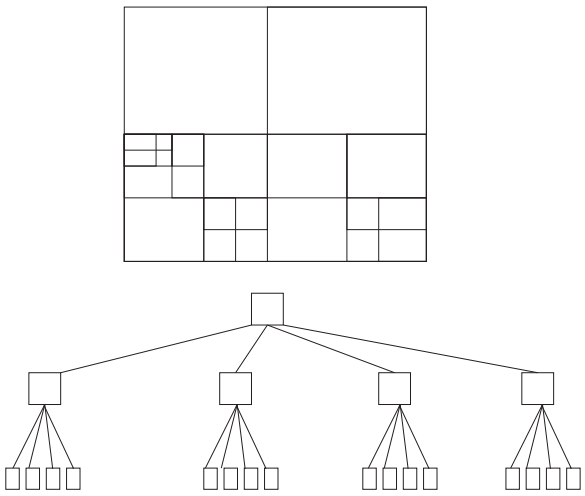

Figure 4. Quadtree partition of a domain and the associated data structure

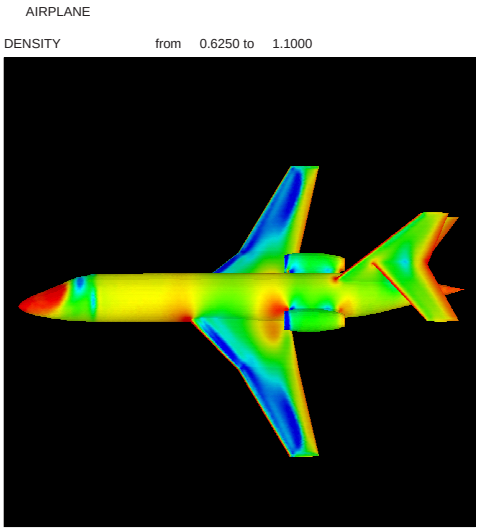

Figure 5. Density contours for a business jet at $M = 0.8$, $\alpha = 2$

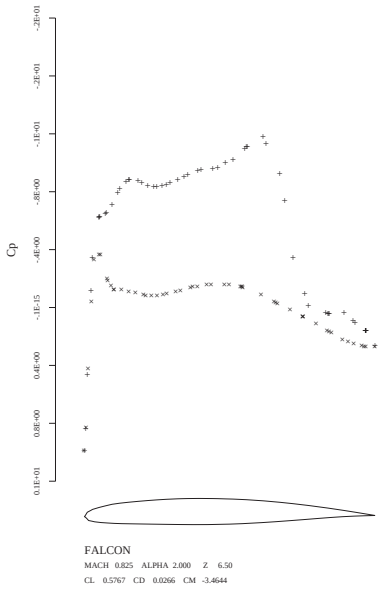

Figure 6. Pressure distribution at 66 % wing span

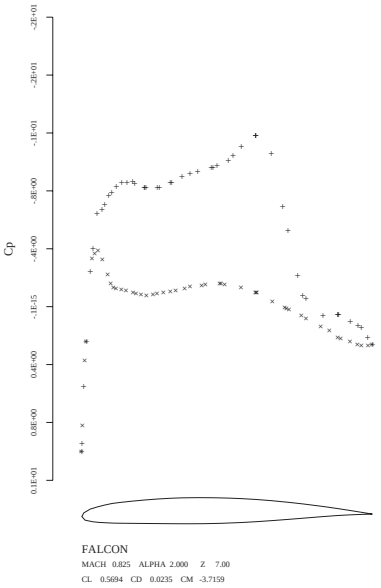

Figure 7. Pressure distribution at 77 % wing span

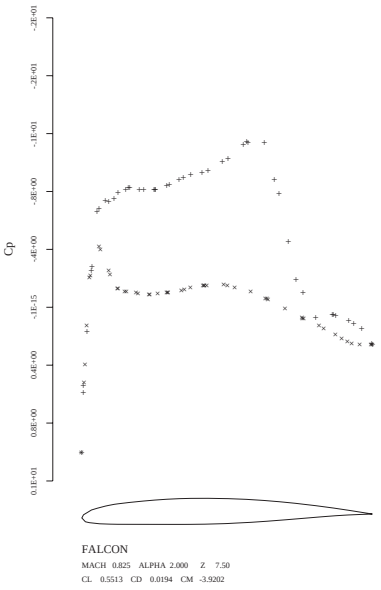

Figure 8. Pressure distribution at 88 % wing span

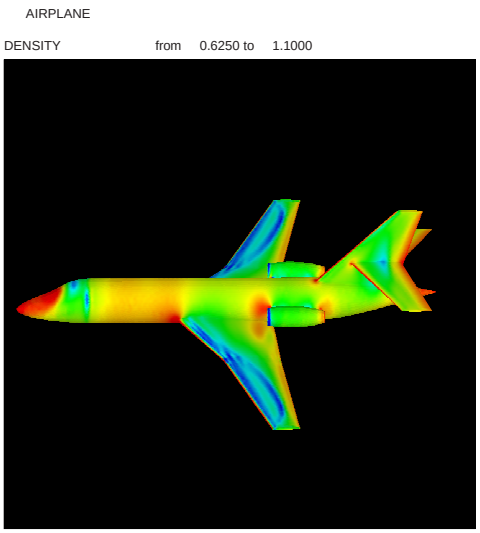

Figure 9. Density contours for a business jet at $M = 0.8$, $\alpha = 2.3$, after redesign

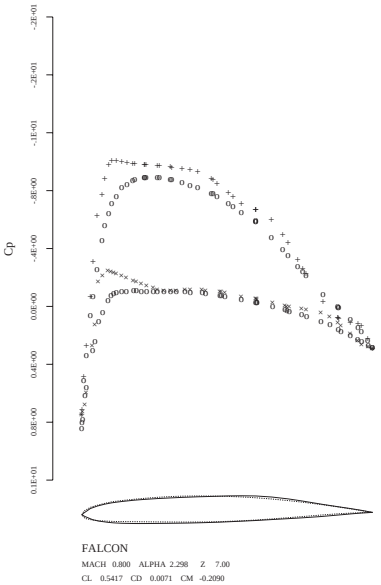

Figure 11. Pressure distribution at 77 % wing span, after redesign, Dashed line: original geometry, solid line: redesigned geometry

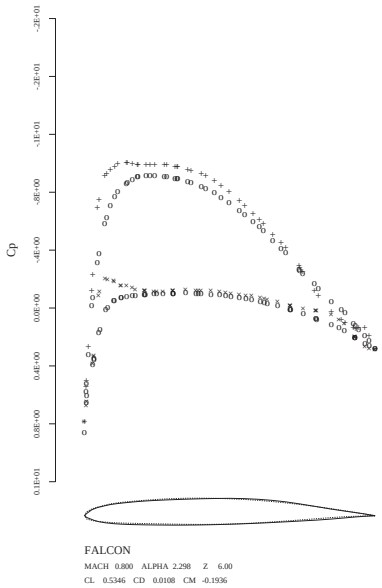

Figure 10. Pressure distribution at 66 % wing span, after redesign, Dashed line: original geometry, solid line: redesigned geometry

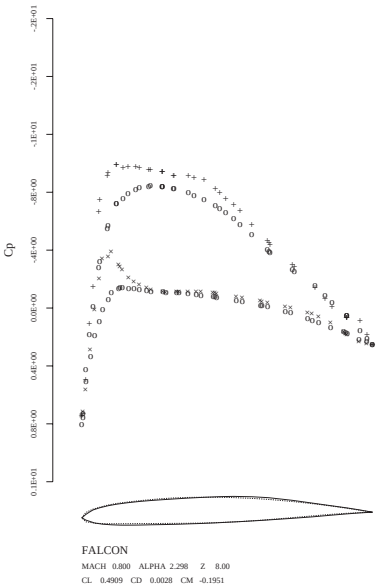

Figure 12. Pressure distribution at 88 % wing span, after redesign, Dashed line: original geometry, solid line: redesigned geometry

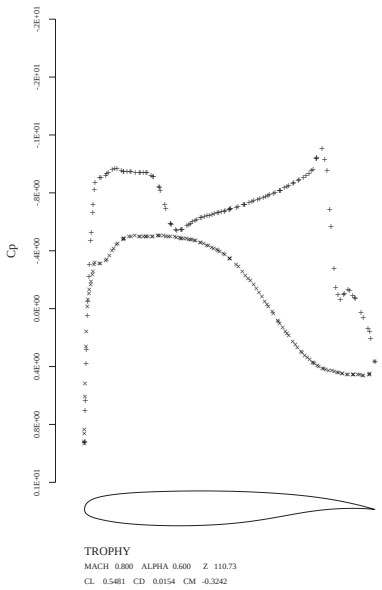

Figure 13. Pressure distribution at 55 % wing span

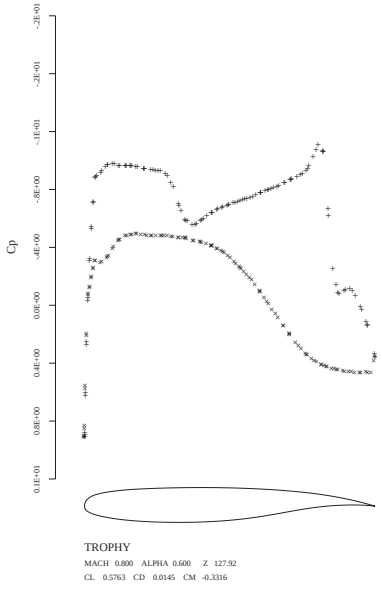

Figure 14. Pressure distribution at 66 % wing span

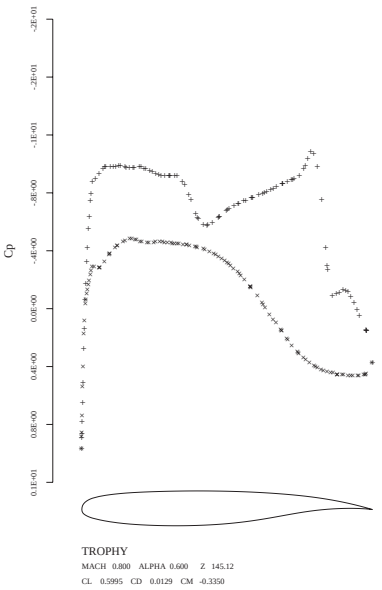

Figure 15. Pressure distribution at 77 % wing span

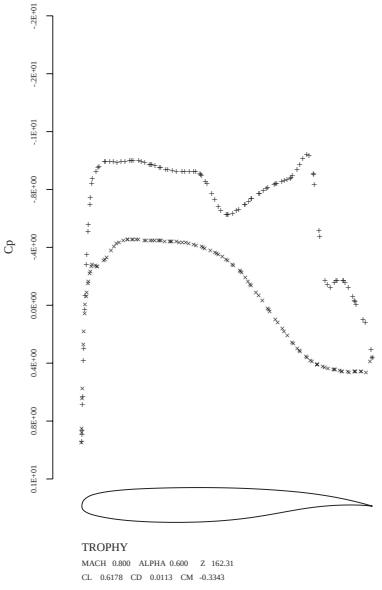

Figure 16. Pressure distribution at 88 % wing span

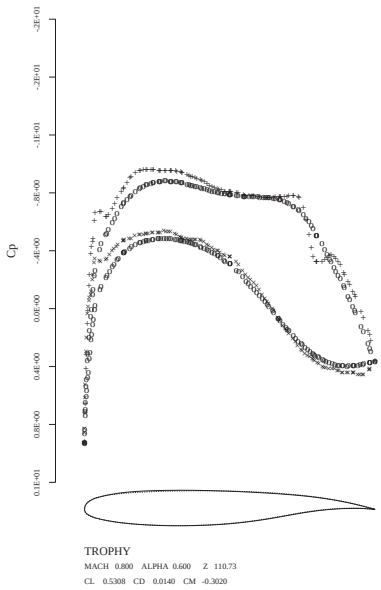

Figure 17. Pressure distribution at 55 % wing span

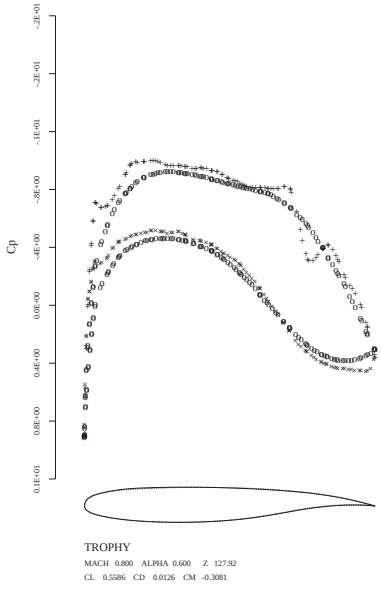

Figure 18. Pressure distribution at 66 % wing span

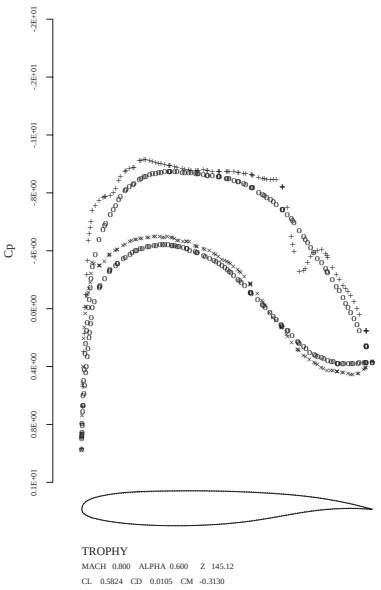

Figure 19. Pressure distribution at 77 % wing span

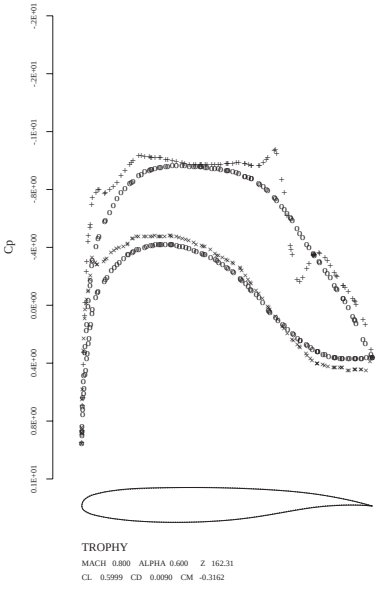

Figure 20. Pressure distribution at 88 % wing span

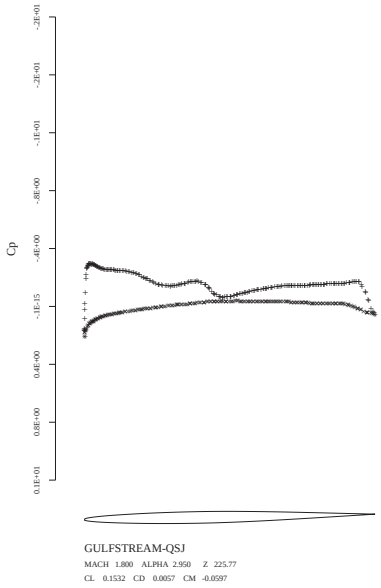

Figure 21. Pressure distribution at 55 % wing span

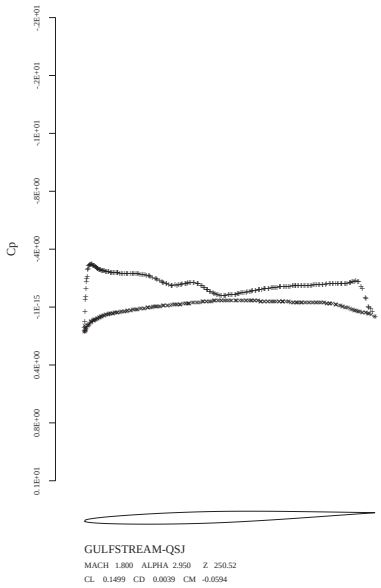

Figure 22. Pressure distribution at 66 % wing span

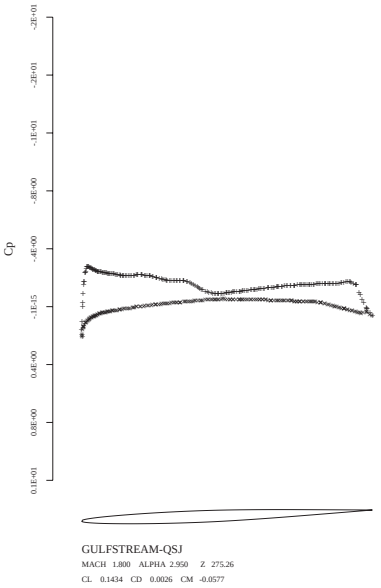

Figure 23. Pressure distribution at 77 % wing span

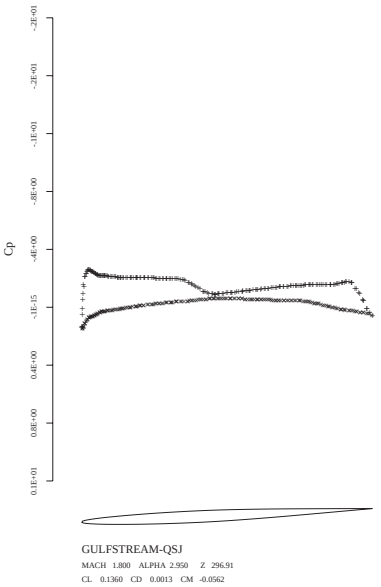

Figure 24. Pressure distribution at 88 % wing span

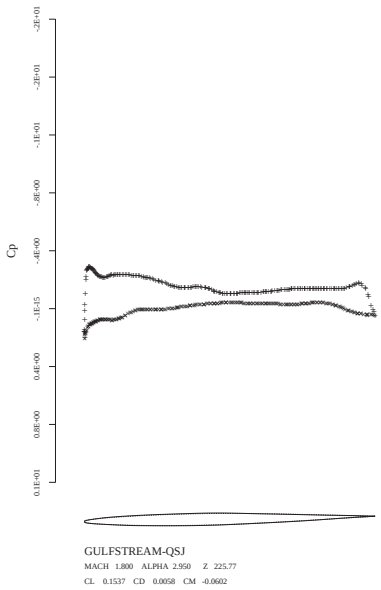

Figure 25. Pressure distribution at 55 % wing span

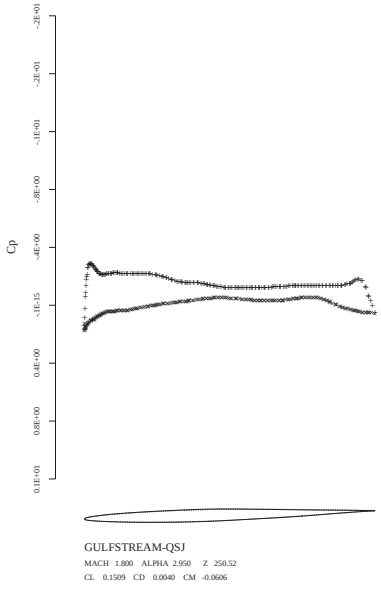

Figure 26. Pressure distribution at 66 % wing span

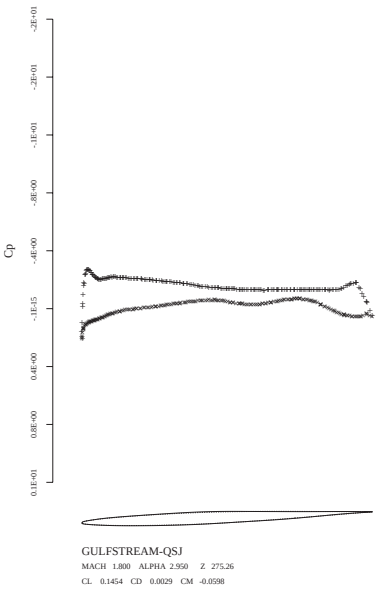

Figure 27. Pressure distribution at 77 % wing span

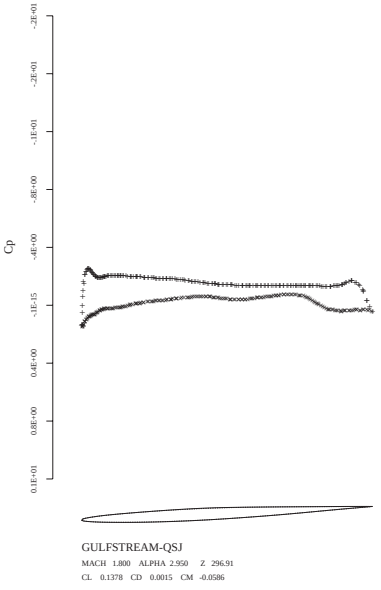

Figure 28. Pressure distribution at 88 % wing span