44th AIAA Aerospace Sciences Meeting and Exhibit
9 - 12 January 2006, Reno, Nevada

AIAA 2006-708

44rd Aerospace Sciences Meeting, January 9-12,2006, Reno,NV

Aerodynamic Simulation and Shape Optimization for High Speed Flow

Antony Jameson *
Stanford University
Stanford, CA

Luigi Martinelli †
Princeton University
Princeton, NJ

John Vassberg ‡
The Boeing Company
Huntington Beach, CA

Susan Cliff §
NASA Ames
Moffet Field, CA

In this paper we review the development of aerodynamic simulation and shape optimization techniques which had their inception in Princeton during the early eighties, when Seymour M. Bogdonoff was chairman of the department. We focus in particular on the development of simulation algorithms for unstructured meshes, represented by the Airplane Code, and on adjoint based optimization techniques. It seems a particularly opportune moment to review the Airplane Code, because we originally announced it exactly twenty years ago at the AIAA Aerospace Sciences meeting in Reno in January 1986.

I. Introduction

We have prepared this paper as a tribute to the late Seymour M. Bogdonoff ("Boggy", as he was affectionately known). Boggy was a man of tremendous charm and great force of personality, who had a major influence on the development of aeronautical science during the period 1960 - 1990 both through his own research and his pervasive influence throughout the aeronautical community and as an advisor to the Air Force. Boggy's contributions are discussed in more detail in the paper by Lex Smits at this Symposium.

While much of Boggy's research was focused on experimental studies of supersonic shock wave-boundary layer interactions, his first love was hypersonics. U.S. research on hypersonics has had a roller coaster ride during the last four decades. After a period of intense interest in the sixties, exemplified by the notable book of Hayes and Probstein, it was almost entirely disregarded in the seventies. There was another spurt of interest in the eighties driven by the attempt to develop a single stage to orbit (SSTO) launch vehicle, during which there were important developments in the computational simulation of hypersonic flow by MacCormack, Candler and NASA researchers. After the abandonment of the SSTO project, hypersonics again receded into the background. There appears, however, to be a resurgence of interest at the present time sparked by the Air Force's new emphasis on the importance of reduced time to target, and also the emergence of concepts for magneto hydrodynamic flow control. A more effective policy might be to recognize the potential future importance of hypersonics, and to fund long term hypersonic research with a continuing stable level of funding.

In this paper we review the development of aerodynamic simulation and shape optimization techniques which had their inception in Princeton during the decade 1980 - 1990. We focus in particular on the development of simulation algorithms for unstructured meshes, represented by the Airplane Code,[1] and on adjoint based optimization techniques.[4,6,5] It seems a particularly opportune moment to review the Airplane Code, because we originally announced it exactly twenty years ago at the AIAA Aerospace Sciences meeting in Reno in January 1986. We believe that it was the first computer program which could solve the three dimensional Euler equations for arbitrary configurations using unstructured meshes. The discretization

*Thomas V. Jones Professor of Engineering, Stanford University, Fellow AIAA
†Associate Professor, Princeton University, Associate Fellow AIAA
‡Boeing Technical Fellow, The Boeing Company, Associate Fellow AIAA
§Aerospace Engineer, NASA Ames

Copyright © 2006 by the Authors. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

Copyright © 2006 by the Authors. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

scheme can equally well be regarded as a finite volume method or a finite element method using a simplified Galerkin formulation with linear elements, stabilized by upwind biasing via artificial diffusion. The numerical scheme was implemented using what were at that time novel face and edge based data structures, which have since been widely adopted. These aspects are presented in the next section. Anyone with exposure to airplane design will realize that a simulation capability by itself, although extremely useful, does not show the designer how to make an improvement. In section V we review shape optimization methods based on control theory which we began developing in Princeton in 1988.[4] All four authors have been heavily involved in the continuing improvement of these methods and their application to real design problems, and we highlight a few of the more notable successes.

II. Origins of the Airplane Code

While there were important advances during the eighties in the solution of the Euler and Navier-Stokes equations for three dimensional configurations, notably including the work of J. Shang and his associates at the Air Force Laboratory, the first author had become convinced by 1984 that the growing demand for simulations of increasingly complex configurations could only be satisfied by resorting to unstructured meshes, because the labor involved in mesh generation would otherwise become prohibitive. We had already carried out a pilot experiment in the calculation of transonic potential flow on unstructured meshes, embodied in the thesis of Richard Pelz[2] (who sadly died prematurely in 2004). In 1984 the first author tested several alternative algorithms to solve the two dimensional Euler equations on triangular meshes, and by the beginning of 1985 was confident that these algorithms were robust, sufficiently accurate, and also fast enough for practical use. He was also aware, as results both of a presentation by Charles Peskin at the Courant Institute and telephone conversations with Brian McCartin (a former student) at UTRC that the Delaunay triangulation algorithm might be a viable method for connecting an arbitrary cloud of points to form a tetrahedral mesh. Nigel Weatherill was visiting Princeton for six months and he agreed to embark on the development of a Delaunay triangulator, subsequently completed by Tim Baker, who also tackled the problem of how to distribute a suitable cloud of points. In the meanwhile the first author of this paper focused on the formulation of a three dimensional discretization scheme. We faced a serious issue of computer resources. Our departmental computer, an IBM 4341, had a computing speed of about .15 megaflops, and had recently been upgraded from 2 to 4 megabytes of memory. Its components were distributed in separate cabinets which filled a large room. It was clearly inadequate for the task at hand. We could access the University IBM 3080 computer, but its cost of more than $1,000 per CPU hour was prohibitive.

Fortunately we had developed a good working relationship with Cray Research and at a meeting with Mr. Rollwagen he agreed to give us access to Cray's in-house XMP computer to support the development of the code. In order to take advantage of this offer we had to fly to Minnesota, where we were given access to the computer (between midnight and 6 am during weekends.) Equally important to us was access to Cray's Evan and Sutherland graphics terminals, since we had no visualization capability in Princeton. This was all before the emergence of X Windows as a widely available tool.

A. Computational Methodology and Finite Element Approximation

The Euler equations in integral form can be written as

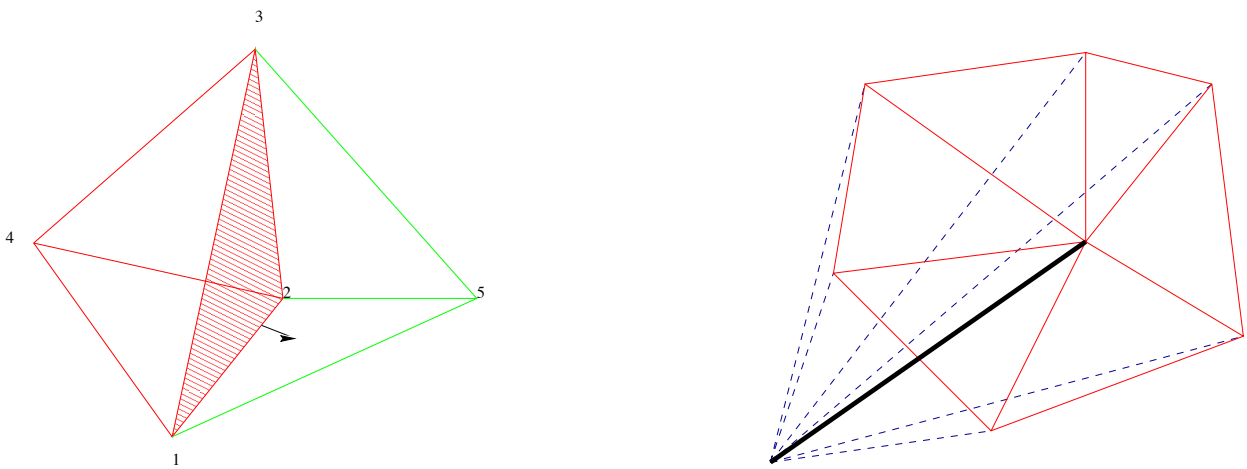
$$\frac{d}{dt}\int_V w dV + \int_S F \cdot dS = 0 \tag{1}$$

Equation (1) can be approximated on a tetrahedral mesh by first writing the flux balance for each tetrahedron assuming the fluxes (F) to vary linearly over each face. Then at any given mesh point one considers the rate of change of w for a control volume consisting of the union of the tetrahedra meeting at a common vertex. This gives

$$\frac{d}{dt}\left(\sum_k V_k w\right) + \sum_k R_k = 0. \tag{2}$$

where V_k is the volume of the kth tetrahedron meeting at a given mesh point and R_k is the flux of that tetrahedron.

When the flux balances of the neighboring tetrahedra are summed, all contributions across interior faces cancel. Referring to Figure 1(a), which illustrates a portion of a three dimensional mesh, it may be seen that with a tetrahedral mesh, each face is a common external boundary of exactly two control volumes. Therefore each internal face can be associated with a set of 5 mesh points consisting of its corners 1, 2 and 3, and the vertices 4 and 5 of the two control volumes on either side of the common face. It is now possible to generate the approximation in equation (2), by pre-setting the flux balance at each mesh point to zero, and then performing a single loop over the faces. For each face, one first calculates the fluxes of mass, momentum and energy across each face, and then one assigns these contributions of the vertices 4 and 5 with positive and negative signs respectively. Since every contribution is transferred from one control volume into another, all quantities are perfectly conserved. Mesh points on the inner and outer boundaries lie on the surface of their own control volumes, and the accumulation of the flux balance in these volumes has to be correspondingly modified. At a solid surface it is also necessary to enforce the boundary condition that there is no convective flux through the faces contained in the surface.



(a) Evaluation of fluxes - Face Based

(b) Evaluation of fluxes - Edge Based

Figure 1. Finite Volume Discretization on a Tetrahedron

While the original formulation of this method used a face-based loop to accumulate the fluxes, the first author in 1987[3] modified the integration to be edge based. Independently, the third author also discovered this technique in 1987.[18] The motivations for this change are two fold. First, the number of edges in a tetrahedral unstructured mesh is typically about two-thirds the number of triangular faces. Hence, the flux balance of equation (2) can be performed using the edge data structure in less than two-thirds the time as that of the face based data structure. The second benefit is that most of the face based data structure is no longer required by the flow solver, therefore a significant reduction in memory requirements can be realized. Figure 1(b) illustrates an interior edge surrounded by the group of tetrahedra surrounding that edge. In this figure, note the umbrella like surface depicted by the solid-red triangles coincident to one vertex of the interior edge. Now accumulate the outward pointing areas of the triangles of this umbrella surface, and associate this directed area with the edge. Notice that the other umbrella surface formed by the dashed-blue triangles yield the same directed area, but opposite in sign. While not immediately obvious, one can transform the face-based evaluation of the fluxes into an edge-based algorithm, which only requires special treatment at the domain boundary. The implementation of this edge-based algorithm is provided in the fortran code listed below. Note that the primary internal loops have been grouped in such a way that vectorization and fine-grain parallel procesing can be exploited on applicable computing platforms.

```
C
C     SET THE FLUX BALANCES IN EACH CONTROL VOLUME TO ZERO
C
      DO N=1,5
      DO I=ND1,ND2
         DW(N,I)   = 0.
```

```
      END DO
      END DO
C
C     CALCULATE THE CONVECTIVE FLUXES ACROSS EACH EDGE
C     AND ACCUMULATE THE FLUX BALANCES FOR EACH POLYHEDRAL SUBDOMAIN
C
      DO L=NGRPG1,NGRPG2
         I1        = IGRPG(L)
         I2        = IGRPG(L+1)  -1
C$DIR NO_RECURRENCE,FORCE_VECTOR,FORCE_PARALLEL_EXT
CDIR$ IVDEP
      DO I=I1,I2

         N1        = NDG(1,I)
         N2        = NDG(2,I)

         QS1       = (W(2,N1)*SG(1,I)
     .               +W(3,N1)*SG(2,I)
     .               +W(4,N1)*SG(3,I))/W(1,N1)
         QS2       = (W(2,N2)*SG(1,I)
     .               +W(3,N2)*SG(2,I)
     .               +W(4,N2)*SG(3,I))/W(1,N2)
         PA        = P(N1)  +P(N2)

         FS1       = QS1*W(1,N1)  +QS2*W(1,N2)
         FS2       = QS1*W(2,N1)  +QS2*W(2,N2)  +PA*SG(1,I)
         FS3       = QS1*W(3,N1)  +QS2*W(3,N2)  +PA*SG(2,I)
         FS4       = QS1*W(4,N1)  +QS2*W(4,N2)  +PA*SG(3,I)
         FS5       = QS1*(W(5,N1)  +P(N1))  +QS2*(W(5,N2)  +P(N2))

         DW(1,N1)  = DW(1,N1)  +FS1
         DW(2,N1)  = DW(2,N1)  +FS2
         DW(3,N1)  = DW(3,N1)  +FS3
         DW(4,N1)  = DW(4,N1)  +FS4
         DW(5,N1)  = DW(5,N1)  +FS5

         DW(1,N2)  = DW(1,N2)  -FS1
         DW(2,N2)  = DW(2,N2)  -FS2
         DW(3,N2)  = DW(3,N2)  -FS3
         DW(4,N2)  = DW(4,N2)  -FS4
         DW(5,N2)  = DW(5,N2)  -FS5

      END DO
      END DO
```

While the following second loop correct the boundary cells:

```
C
C     CORRECT THE FLUX BALANCES OF BOUNDARY CELLS
C
      L1        = NGRPF1
      L2        = NGRPFB
      S         = 1.
      IF (BCM.EQ.0.) S = 0.
   11 DO L=L1,L2
         I1       = IGRPF(L)
         I2       = IGRPF(L+1)  -1
C$DIR NO_RECURRENCE,FORCE_VECTOR,FORCE_PARALLEL_EXT
CDIR$ IVDEP
      DO I=I1,I2
```

```
      N1        = NDF(1,I)
      N2        = NDF(2,I)
      N3        = NDF(3,I)
      N4        = NDF(4,I)

      SX        = X(2,N1)*(X(3,N3)  -X(3,N2))
     .           +X(2,N2)*(X(3,N1)  -X(3,N3))
     .           +X(2,N3)*(X(3,N2)  -X(3,N1))
      SY        = X(3,N1)*(X(1,N3)  -X(1,N2))
     .           +X(3,N2)*(X(1,N1)  -X(1,N3))
     .           +X(3,N3)*(X(1,N2)  -X(1,N1))
      SZ        = X(1,N1)*(X(2,N3)  -X(2,N2))
     .           +X(1,N2)*(X(2,N1)  -X(2,N3))
     .           +X(1,N3)*(X(2,N2)  -X(2,N1))

      PA        = P(N1)  +P(N2)  +P(N3)
      QS        = (W(2,N1)*SX  +W(3,N1)*SY  +W(4,N1)*SZ)/W(1,N1)
      FS1       = QS*W(1,N1)
      FS2       = QS*W(2,N1)  +P(N1)*SX
      FS3       = QS*W(3,N1)  +P(N1)*SY
      FS4       = QS*W(4,N1)  +P(N1)*SZ
      FS5       = QS*(W(5,N1)  +P(N1))

      QS        = (W(2,N2)*SX  +W(3,N2)*SY  +W(4,N2)*SZ)/W(1,N2)
      GS1       = QS*W(1,N2)
      GS2       = QS*W(2,N2)  +P(N2)*SX
      GS3       = QS*W(3,N2)  +P(N2)*SY
      GS4       = QS*W(4,N2)  +P(N2)*SZ
      GS5       = QS*(W(5,N2)  +P(N2))

      QS        = (W(2,N3)*SX  +W(3,N3)*SY  +W(4,N3)*SZ)/W(1,N3)
      HS1       = QS*W(1,N3)
      HS2       = QS*W(2,N3)  +P(N3)*SX
      HS3       = QS*W(3,N3)  +P(N3)*SY
      HS4       = QS*W(4,N3)  +P(N3)*SZ
      HS5       = QS*(W(5,N3)  +P(N3))

      Q1        = FS1  +GS1  +HS1
      Q2        = FS2  +GS2  +HS2
      Q3        = FS3  +GS3  +HS3
      Q4        = FS4  +GS4  +HS4
      Q5        = FS5  +GS5  +HS5

      R1        = S*Q1
      R2        = S*(Q2  -PA*SX)
      R3        = S*(Q3  -PA*SY)
      R4        = S*(Q4  -PA*SZ)
      R5        = S*Q5

      DW(1,N1)  = DW(1,N1)  +.5*Q1  +2.5*FS1  -R1
      DW(2,N1)  = DW(2,N1)  +.5*Q2  +2.5*FS2  -R2
      DW(3,N1)  = DW(3,N1)  +.5*Q3  +2.5*FS3  -R3
      DW(4,N1)  = DW(4,N1)  +.5*Q4  +2.5*FS4  -R4
      DW(5,N1)  = DW(5,N1)  +.5*Q5  +2.5*FS5  -R5

      DW(1,N2)  = DW(1,N2)  +.5*Q1  +2.5*GS1  -R1
      DW(2,N2)  = DW(2,N2)  +.5*Q2  +2.5*GS2  -R2
      DW(3,N2)  = DW(3,N2)  +.5*Q3  +2.5*GS3  -R3
      DW(4,N2)  = DW(4,N2)  +.5*Q4  +2.5*GS4  -R4
      DW(5,N2)  = DW(5,N2)  +.5*Q5  +2.5*GS5  -R5
```

```
      DW(1,N3)  = DW(1,N3)  +.5*Q1  +2.5*HS1  -R1
      DW(2,N3)  = DW(2,N3)  +.5*Q2  +2.5*HS2  -R2
      DW(3,N3)  = DW(3,N3)  +.5*Q3  +2.5*HS3  -R3
      DW(4,N3)  = DW(4,N3)  +.5*Q4  +2.5*HS4  -R4
      DW(5,N3)  = DW(5,N3)  +.5*Q5  +2.5*HS5  -R5

      DW(1,N4)  = DW(1,N4)  -R1
      DW(2,N4)  = DW(2,N4)  -R2
      DW(3,N4)  = DW(3,N4)  -R3
      DW(4,N4)  = DW(4,N4)  -R4
      DW(5,N4)  = DW(5,N4)  -R5

    END DO
    END DO
```

B. Dissipation

A simple way to introduce dissipation is to add a term generated from the difference between the value at a given node and its nearest neighbors. That is, at node 0, we add a term

$$D_o = \sum_k \epsilon_{ko}^{(1)} (w_k - w_o) \tag{3}$$

where the sum is over the nearest neighbors. This contribution is balanced by a corresponding contribution at node k, with the result that the scheme remains conservative. The coefficients $\epsilon_{ko}^{(1)}$ may incorporate metric information depending on local cell volumes and face areas, and can also be adapted to gradients of the solution. As equation (3) is only first-order accurate (unless the coefficients are proportional to the mesh spacing), a more accurate scheme is obtained by recycling the edge differencing procedure. After setting

$$E_o = \sum_k (w_k - w_o) \tag{4}$$

at every mesh point, one then sets

$$D_o = -\sum_k \epsilon_{ok}^{(2)} (E_k - E_o) \tag{5}$$

An effective scheme is produced by blending equation (3) and (5), and adapting $\epsilon_{ko}^{(1)}$ to the local pressure gradient. This scheme has been found to have good shock capturing properties and the required sums can be efficiently assembled by loops over the edges.

Other shock capturing schemes that satisfy the LED property have also been implemented, and have been found to work equally efficiently. However, due to the robust nature of the simple scalar dissipation model described above, we have used it for all the computations in this study.

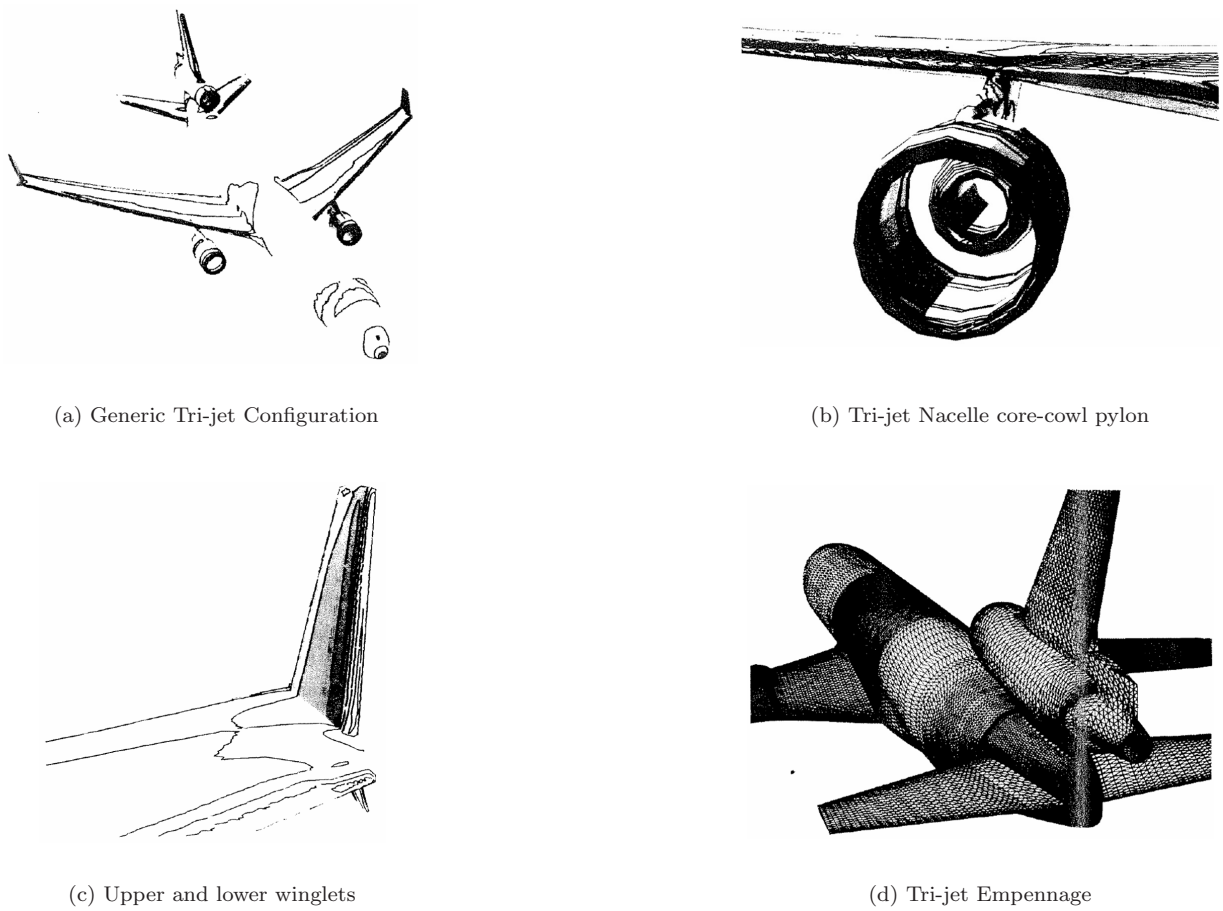
C. Integration to Steady State and Convergence Acceleration Techniques

The resulting spatial discretizations yield a set of coupled ordinary differential equations that can be integrated in time to obtain steady state solutions of the Euler equations. To maximize the allowable time step, the same multistage schemes that have proven to be efficient in rectilinear meshes[9] have been used on unstructured meshes. These schemes bear close resemblance to Runge-Kutta schemes [7] with modifications to the evaluation of the dissipation terms that enlarge the stability limit of the scheme along the imaginary axis, thereby allowing convective waves to be resolved.

Convergence to steady state is accelerated by using a variable time step close to the stability limit of each mesh point. The scheme is accelerated further by the introduction of residual averaging.[10]

III. Aerodynamic Analysis in the Transonic Regime using Airplane

The possibility to address transonic flows about complex geometries prompted the Douglas Aircraft Company (DAC), in August 1989, to request a challenging demonstration of the Airplane Code on a complete tri-jet commercial transport aircraft configuration. The generic tri-jet geometry for this demonstration was comprised of a fuselage, wing, empennage, under-wing engine group, and winglet system; see Figure 2(a). The engine group included a nacelle, core-cowl, and pylon; see Figure 2(b). The winglet system was comprised



(a) Generic Tri-jet Configuration

(b) Tri-jet Nacelle core-cowl pylon

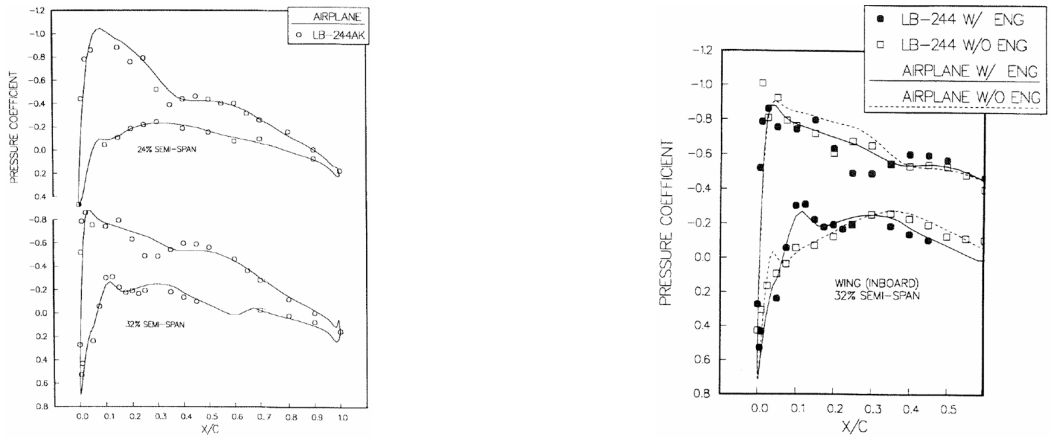
(c) Upper and lower winglets

(d) Tri-jet Empennage

Figure 2. Tri-Jet Configuration

of both upper and lower winglets; see Figure 2(c). The empennage included a horizontal tail, a vertical tail, a nacelle, a core-cowl, and a boundary-layer diverter; see Figure 2(d). The internal flows of the nacelles and core-cowls corresponded to natural (unforced) conditions to match data from an available wind-tunnel test. The elapsed turnaround time of this initial benchmark was a single week. The resulting unstructured mesh contained 384,914 nodes and 2,332,022 tetrahedra. Figure 3(a) provides a comparison of computed pressure distributions with wind-tunnel data at two wing stations on the inboard side of the pylon. The quality of these results and the quick turnaround time prompted DAC to license Airplane before the end of the year. References[16,17] documents this and other DAC benchmarks of the Airplane Code.

DAC utilized Airplane heavily during the early 1990's to study various component-to-component interference effects, to design specialty fairings for juncture-flow regions, and to develop new winglet concepts. The method was applied to commercial and military transonic transport aircraft as well as to the supersonic High-Speed Civil Transport (HSCT) aircraft studied during the High-Speed Research (HSR) cooperative program with NASA. For example, Figure 3(b) illustrates the engine-installation effect on the wing pressure distribution, just inboard of the pylon station. Here, the pressures with the engine are depicted with the solid line and solid symbols, for computed and experimental data, respectively. The pressures without the engine are provided with the dashed line and open symbols. Note that the absolute match between computed and



(a) Pressure Distribution

(b) Effects of Engine Installation

Figure 3. Comparison of Computed and Measured Pressure Distribution

experimental pressures on the lower surface is very good, and although the comparison on the upper surface is not as close, the trend is captured.

The above validation study provided sufficient confidence in Airplane to allow Vassberg to design an MD-11 pylon fairing to improve the juncture flow in the wing-pylon intersection; this is identified as Region 1 in Figure 4.

This design effort was completed in less than three weeks of elapsed time. The resulting pylon fairing design was the single largest improvement item of the five-year MD-11 Cruise Performance Improvement Program (CPIP). The significance of this accomplishment is further emphasized by the fact that a previous attempt by a veteran aerodynamicist to design a pylon fairing was determined to be essentially unsuccessful by MD-11 CPIP flight tests. Figure 5(a) illustrates the Airplane computed pressure distributions near the leading edge on the outboard side of the pylon-wing intersection. This Figure also includes a comparison of the intersection-line geometries. Note that the baseline MD-11 pressure peak occurs at a local Mach number of about 1.8, the previous attempt (OCT'90) had a peak local Mach number of 1.6, while Vassberg's bulbous design (JUNE'91) completely suppressed the pressure peak.

With a maximum local Mach number of 1.06 and a relatively flat pressure distribution, this improved pylon fairing design allowed the flowfield in this juncture region to remain fully attached as it accelerates around the leading edge and progresses downstream on the outboard side of the pylon. As measured by the accurate flight tests of the MD-11 CPIP, this pylon fairing yielded a 0.8% reduction in drag at nominal cruise, and has even larger benefits at higher lifting conditions. The fairing was immediately added to the MD-11 production geometry and retrofitted to aircraft that had already been placed into airline service. A more recent Navier-Stokes solution of this improvement is depicted in Figure 5(b) which shows the separated flow region of the baseline geometry and the attached flow of the pylon fairing design. A conservative estimate is that the performance improvements associated with this pylon fairing design saved DAC, Pratt-Whitney, and General Electric tens of millions of dollars in penalty payments to the airline customers.

Some contributions to the Airplane Code from DAC research include faster grid generation (30X) , better edge coloring algorithms to improve parallel processing on vector-processor-class supercomputers, better element ordering algorithms to improve throughput on cache-based computers(2X), and a coupled boundary-layer capability to capture viscous effects in the flow computations. Additional advancements developed for unstructured-mesh methods include automated multigrid acceleration and time-accurate simulations based on the pseudo-time stepping technique; see References.[18,19,20]

IV. Aerodynamic Analysis in the Supersonic Regime using Airplane

Airplane has been used extensively in the supersonic regime at NASA Ames Research Center. The following sections describe the results of some representative applications.

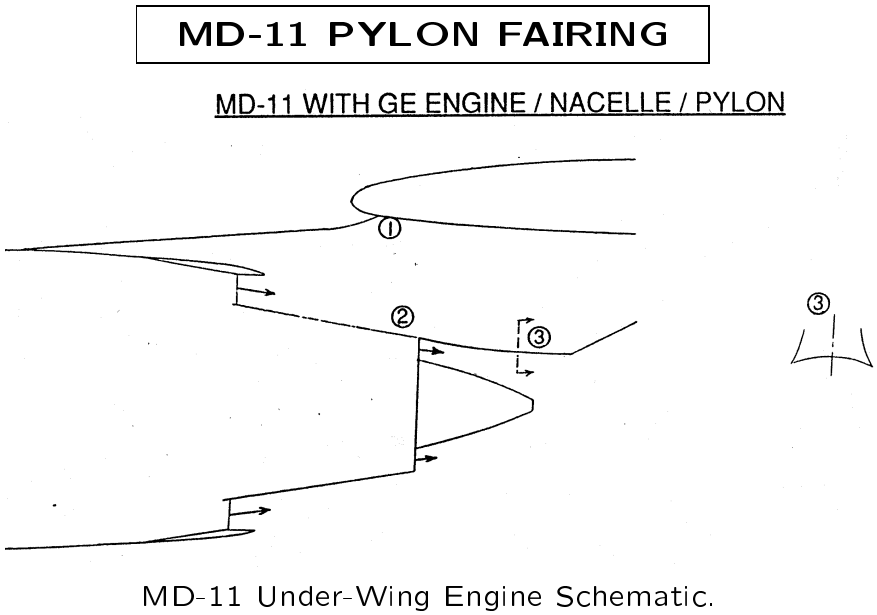
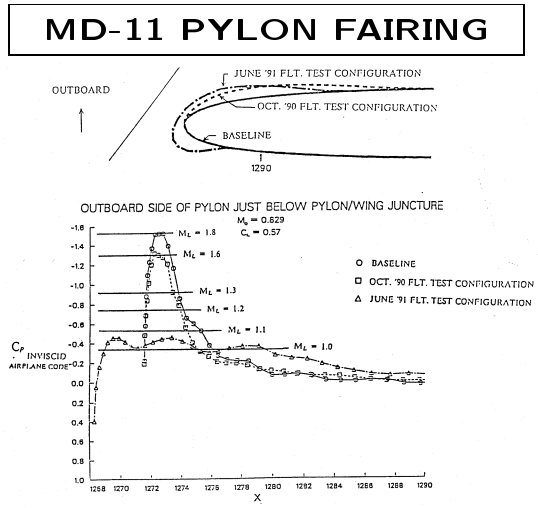


Figure 4. Wing/Pylon Fairing

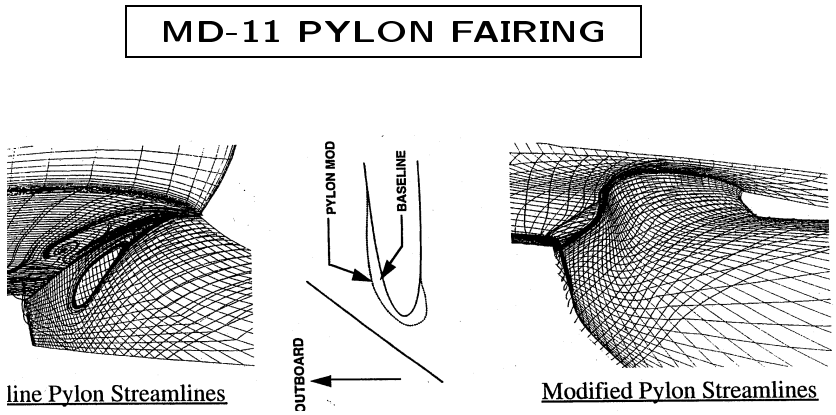
A. HSR Sonic Boom work with Airplane

The HSR program originally was centered on developing configurations with low sonic boom loudness levels. The next generation supersonic transport configuration must meet performance criteria as well as environmental constraints. The design mission for the aircraft assumed a range of 5,500 nautical miles with 300 passengers. A ground signature with a PLdB of 95 or less was used as a goal for the design, assuming flight was restricted to designated corridors. The design was limited to conventional shapes so that the aircraft would be acceptable to the aviation community. Airplane and other inviscid CFD codes combined with planar extrapolation methods provided accurate sonic boom pressure signatures at distances greater than one body length from supersonic configurations. This was the first application of CFD codes to sonic boom predictions. Airplane's unstructured grid method provided very dense and smoothly varying off body computational grids that provided an ideal tool (prediction was best from this tool in a blind study) for sonic boom predictions in the nearfield. Airplane used a staging of meshes that increase in density from the outer boundaries to the configuration surface by use of a sequence of nested boxes. The dense grid near the surface does not propagate to the outer boundaries, resulting in a more efficient use of points than the structured grid methods. Airplane was used as an analysis code because of its demonstrated ability to give accurate flow field predictions and handle highly complex configurations. Systematic changes in geometric shape were handled with HFLO4 (a structured code utilizing an H-Mesh topology that was also developed at Princeton by Jameson and Baker) to design new supersonic transport configurations with low sonic boom characteristics; at the time Airplane was not a design tool. An aircraft synthesis code, in combination with the CFD and extrapolation code, was used to close the design. The resulting Ames LBWT (Low Boom Wing Tail) configuration that had the lowest sonic boom loudness levels and was a closed vehicle design with acceptable take off weight and efficient cruise performance. The Ames LBWT has a highly swept cranked arrow wing with conventional tails, and was designed to accommodate either 3 or 4 engines. The complete configuration, including nacelles and boundary layer diverters, was evaluated using the Airplane Code. Figure 6 shows the configuration and an Airplane solution of the symmetry plane with the near field pressure signature superimposed.



MD-11 Pylon/Wing Intersection Pressure Distributions.

(a) Computed Pressure Distribution Outboard of Pylon-Wing Intersection



MD-11 Pylon Computed Streamlines.

(b) Navier-Stokes Simulation of Pylon-Wing Intersection

Figure 5. MD-11 Pylon Fairing Design

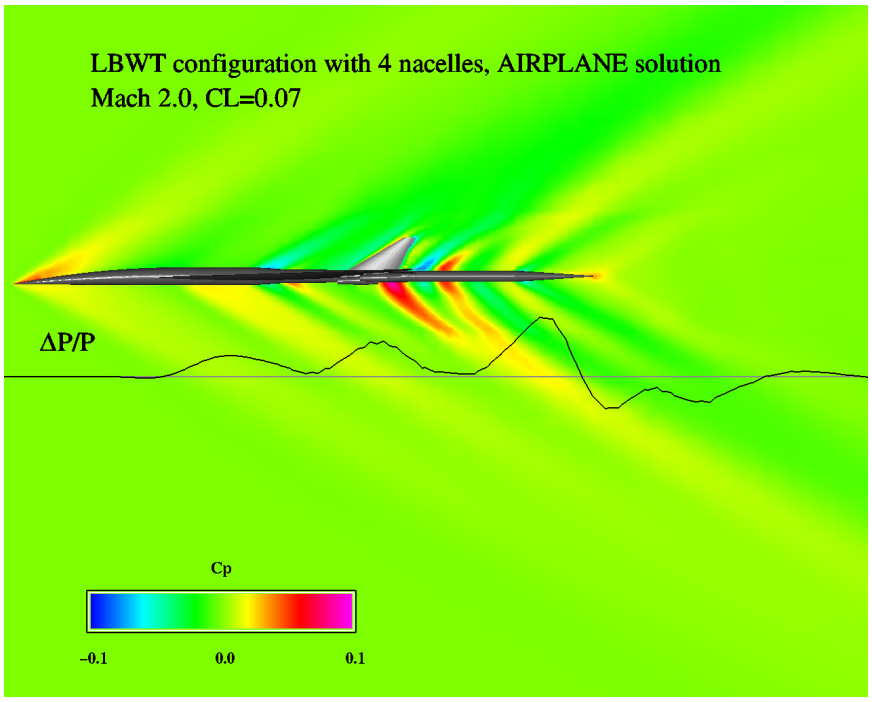


Figure 6. LBWT Configuration

B. HSR Performance Design with Airplane

The Airplane Code was the main workhorse analysis code in the performance-based design phase of the HSR program (1995-1998). Airplane provided analysis for essentially every configuration during the program. In addition, it was an integral component to the design process in that it provided the*pseudo*-nacelle effects in the design code. Following a great success in optimization at Ames on a Douglas configuration (Wing 4), optimization was tackled in earnest on more realistic configurations that had been extensively optimized using linear design tools.

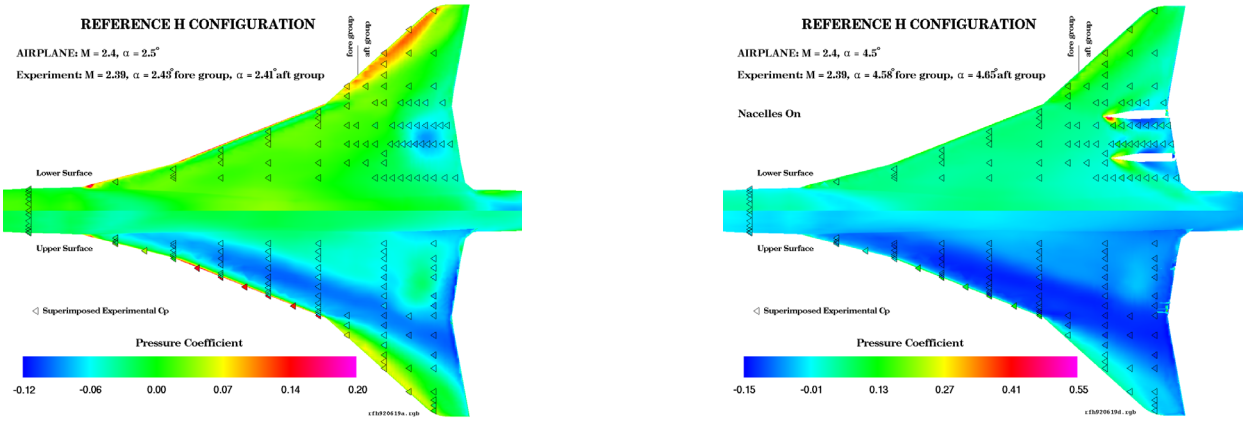
1. Reference H Analysis with Airplane

The first was the Boeing Reference H configuration. Airplane was purely used for analysis of the Ref H configuration, and wing body shape optimization was done without nacelles and diverters using structured grid methods. The baseline Reference H wind tunnel model was instrumented with a large number of pressure taps in order to accurately measure surface pressures and to assess Airplane and other computational methods. Figure 7(a) and 7(b) show the computed surface pressures from Airplane on the upper and lower surfaces on the wing. Superimposed onto the image are the experimental pressures from the wind tunnel. The lack of change in color from the triangular experimentally derived pressures and Airplanes computations show how well Airplane captures the pressures on both upper and lower surfaces.

Figure 8 shows an isometric view of a complete solution using Airplane, while Figure 9 shows force and moment computations of the baseline and optimized configurations with Airplane and experiment.

2. Inlet Unstart Using Airplane

The Boeing Reference H configuration was tested in the NASA Ames 9x7 Supersonic Wind Tunnel. This experiment was devised to simulate an unstarted inlet as well as determine the aerodynamic performance of the configuration with and without nacelle and diverter components. This very detailed inlet unstart experiment was done since unstart can cause drastic changes in the aerodynamic forces and moments acting on an aircraft, often resulting in severe controllability problems. Inlet unstart can arise from a rapid maneuver, a sudden change in atmospheric temperature (Mach number), or an engine induced disturbance. Any of these conditions can alter the shock system in a mixed compression inlet and cause the normal shock to



(a) Reference H Configuration - Top

(b) Reference H Configuration - Bottom

Figure 7. Comparison of Measured and Computed Pressure Distributions

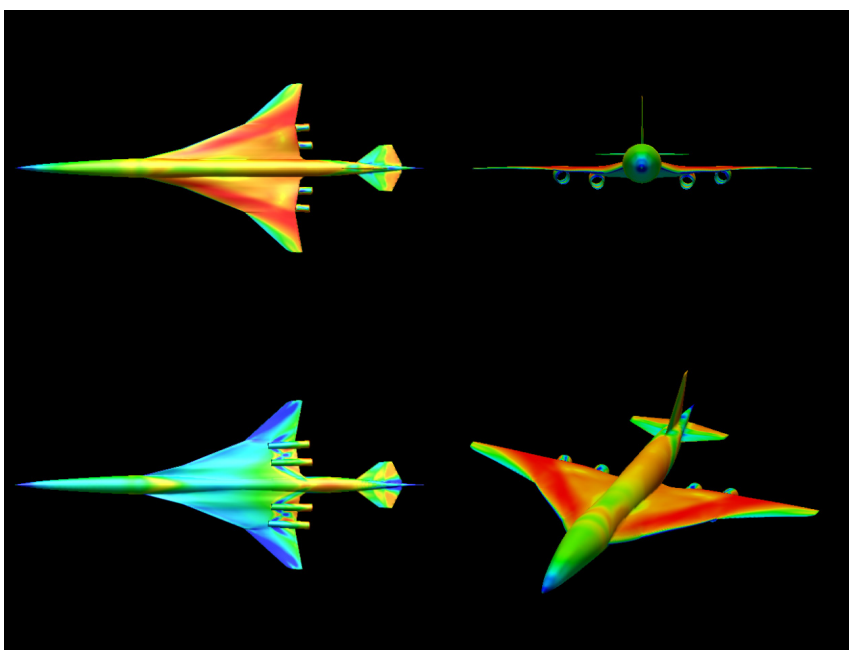


Figure 8. Reference H Configuration- Isometric View

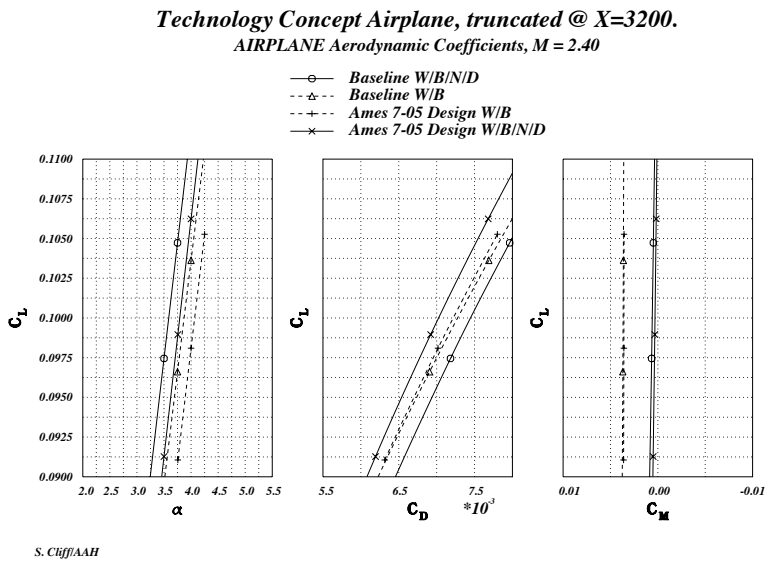
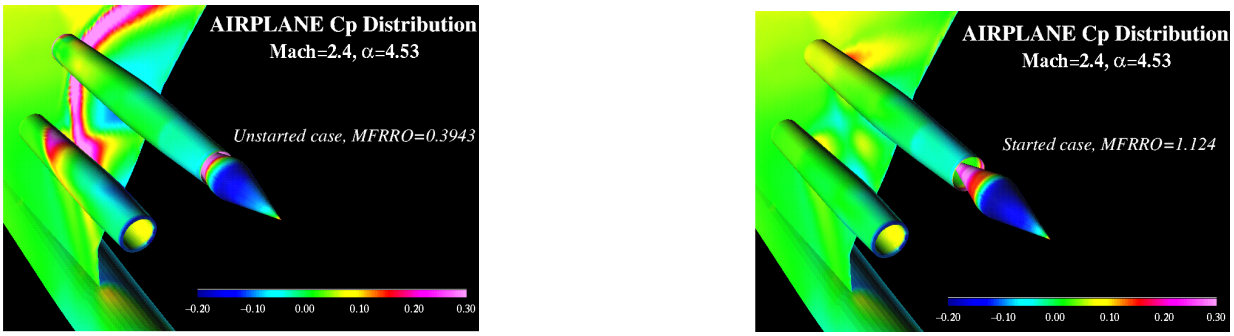


Figure 9. Computed Aerodynamic Forces - Baseline and Optimized Configurations

propagate forward of the nacelle. Wind tunnel tests are usually performed to determine the severity of the controllability problem associated with an inlet unstart by measuring changes in the forces and moments with one or more unstarted inlets. Such tests are expensive and time consuming; consequently an efficient computational method to predict changes in force and moment coefficients due to inlet unstart would be a great benefit to the aircraft designer. Airplane computational predictions were compared with experimental results for the Boeing Reference H configuration obtained during a test in the NASA Ames 9x7 Supersonic Wind Tunnel. Airplane and Wind tunnel comparisons of the configuration without nacelles, with fully started captive nacelles, fully started non-captive nacelles, as well as unstarted non-captive nacelles were done. The study showed how the computer simulations with Airplane provided considerable insight into the aerodynamic effects of an inlet unstart. For the unstart cases, the mass-flow within the nacelle was controlled by the use of a mass-flow plug. The plug was translated along the nacelle axis into the aft portion of the nacelle to change the nacelle exit area. The aft portion of the outboard nacelle was extended nearly two diameters downstream to ensure that the interference from the mass-flow plug would not alter the pressure distributions on the forward part of the nacelles. This modified outboard nacelle was treated as a separate component and replaced the original outboard nacelle in the data set. The axisymmetric mass-flow plug was defined by attaching two 30 degree cones to a smooth transitionary surface which imposed tangency along the base of each cone. The small included cone angles and the smooth transitionary surface on the plug were designed to produce an attached bow shock wave for started cases, and also to limit the expansion on the aft portion of the plug. For each new plug position, the translated plug definition simply replaced the previous plug data set and a complete mesh was automatically generated. The computational plug was subsequently moved to 5 other positions to provide a range of MFR data from 0.385 to 1.124. The Airplane computations were obtained at the same Mach number and angle of attack as experiment (M = 2.41, AOA = 4.53^o). However, the experimental model could not be positioned at precisely zero degrees yaw and the experimental data was, in fact, taken at a yaw angle of -0.56. This primarily affected the yawing moment and had little impact on the other forces and moments. A view of the computational surface pressure coefficient contours for the unstarted case with MFR of 0.39 and the started case with the plug moved aft (MFR of 1.124) are shown in Figure 10 using the same color map. The dramatic bow shock generated by the unstarted nacelle is readily apparent on the wing lower surface. The inboard nacelle shocks are benign in comparison, but are evident by the increased shock strength on the wing between the nacelles.

The comparisons between computational and experimental results were good, and demonstrated that the Euler method is capable of efficiently and accurately predicting the changes in the aerodynamic coefficients associated with inlet unstart and the effects of the nacelle and diverter components.



(a) (b)

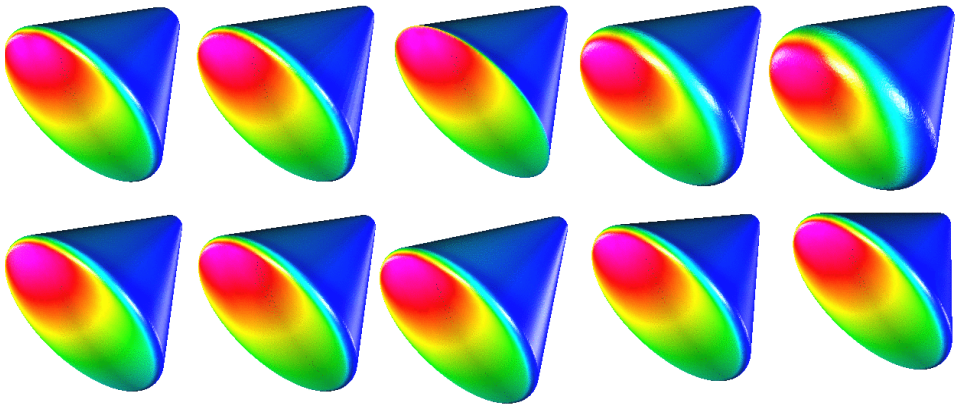
Figure 10. Unstarted Nacelle Simulation

C. Apollo Capsule Optimization using Airplane

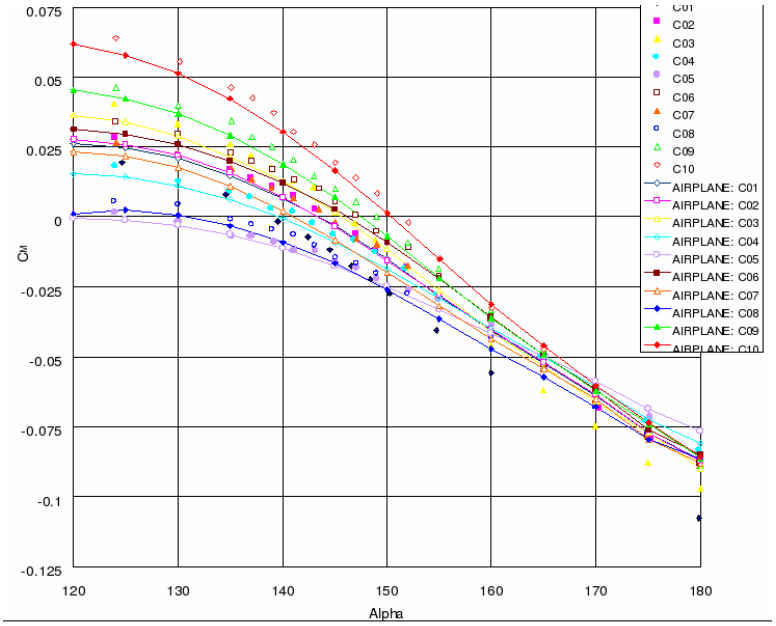
Following the HSR work Airplane has undergone extensive validation with experimental data on numerous space vehicles with blunt-based aft bodies. The validation is over subsonic to hypersonic Mach numbers and includes high angles of attack. The accuracy of the method is well understood, and the method is superior to preliminary design methods in the subsonic to low supersonic Mach number range. (The data and geometries are proprietary). Airplane is now also coupled to a constrained gradient-based optimization algorithm and is used for aerodynamic shape optimization (ASO). Performance increments/improvements over baseline configurations held through wind tunnel tests and Navier-Stokes comparisons on a previous crew transfer vehicle (CTV) design. The method has been successfully used for multipoint complete configuration optimization with performance and stability/trim objectives on Lockheed Martin CTV configurations. With the recent NASA mandate for the development of a manned crew exploration vehicle, Airplane was evaluated by comparison with the extensive wind tunnel data obtained during the Apollo Space Program. During this investigation the baseline Apollo CM (employed on all flights) was found to have an undesirable characteristic in that it is both stable and trimmed in an apex-forward position that poses a safety risk if the CM separates from the launch tower during abort. The Euler-based optimization program that was used successfully on winged configurations seemed ideally suited to perform optimization of the CM to remedy this undesirable characteristic. Alterations and additional components that were developed to eliminate the dual trim point of the Apollo capsule were built during the Apollo program, and the aerodynamic quantities of these vehicles were obtained through extensive wind tunnel tests. Numerical optimization was employed on the Apollo Command Module to modify its external shape. The Apollo Command Module (CM) that was used on all NASA human spaceflights during the Apollo Space Program is stable and trimmed in an apex-forward (alpha of approximately 40 to 80 degrees) position. This poses a safety risk if the CM separates from the launch tower during abort. Optimization was employed on the Apollo CM to remedy the undesirable stability characteristics of the configuration. Geometric shape changes were limited to axisymmetric modifications that altered the radius of the apex (RA), base radius (RO), corner radius (RC), and the cone half angle while the maximum diameter of the CM was held constant.

The results of multipoint optimization on the CM indicated that the cross-range performance can be improved while maintaining robust apex-aft stability with a single trim point. There exits a wealth of experimental test data from the 1960s that is significant to the calibration of CFD methods and was therefore used to compare with the Euler code on a series of 10 parametric designs of the capsule with shape changes that altered the radius of the apex (RA), base radius (RO), corner radius (RC), and the cone half angle while holding the maximum diameter of the CM constant. Solutions of the 10 shapes are shown in Figure 11(a).

Airplane analysis of ten alternative CM vehicles with different values of the above four parameters were compared with the published experimental results of numerous wind tunnel tests during the late 1960s. These comparisons cover a wide Mach number range and a full 180-degree pitch range and show that the Euler methods are capable of fairly accurate force and moment computations and can separate the vehicle characteristics of these ten alternative configurations. The Figure 11(b) is significant because it shows that Airplane is capable of predicting the differences in moment coefficient data between geometrically similar



(a) Pressure Contours



(b) Computed and Measured Coefficient of Moment

Figure 11. Parametric analysis of the Apollo Capsule

objects; thus the increments between the experimental data of pairs of very similar models compare well with the increments computed by the Euler predictions. The actual values of the pitching moment also compare well.

V. Overview of the Design Process using Adjoint Based Methods

Although an accurate analysis of the flowfield about a complex aerodynamic geometry is of significant importance, it does not directly provide the designer with sufficient information on how to improve the performance of the existing configuration. Acquiring such data can be achieved using optimization techniques. Unfortunately, the standard methods of computing sensitivities or gradients can be prohibitively expensive, or alternatively, the design space can be too restrictive to yield a substantial improvement while providing a reasonable shape that works well at off-design conditions. Furthermore, the expense of traversing the design space to the optimum can also be a limiting factor. Hence, the authors have developed optimization technologies that augment the standard schemes with the result being an affordable high-fidelity aerodynamic shape optimization method.

In order to find optimum aerodynamic shapes with reasonable computational costs, it pays to embed the flow physics within the optimization process. In fact, one may regard a wing as a device to control the flow in order to produce lift with minimum drag. As a result, one can draw on concepts which have been developed in the mathematical theory of control of systems governed by partial differential equations. In particular, an acceptable aerodynamic design must have characteristics that smoothly vary with small changes in shape and flow conditions. Consequently, gradient-based procedures are appropriate for aerodynamic shape optimization. Two main issues affect the efficiency of gradient-based procedures; the first is the actual calculation of the gradient, and the second is the construction of an efficient search procedure which utilizes the gradient.

For flow about a wing, or a complete aircraft, 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

$$\delta I = \frac{\partial I^T}{\partial w}\delta w + \frac{\partial I^T}{\partial \mathcal{F}}\delta \mathcal{F}, \tag{6}$$

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 (6). 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

$$R(w, \mathcal{F}) = 0 \tag{7}$$

Then δw is determined from the equation

$$\delta R = \left[\frac{\partial R}{\partial w}\right]\delta w + \left[\frac{\partial R}{\partial \mathcal{F}}\right]\delta \mathcal{F} = 0 \tag{8}$$

Next, introducing a Lagrange Multiplier ψ, we have

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

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

Choosing ψ to satisfy the adjoint equation

$$\left[\frac{\partial R}{\partial w}\right]^T \psi = \frac{\partial I}{\partial w} \tag{9}$$

the first term is eliminated and we find that

$$\delta I = \mathcal{G}\delta\mathcal{F} \tag{10}$$

where

$$\mathcal{G} = \frac{\partial I^T}{\partial \mathcal{F}} - \psi^T \left[\frac{\partial R}{\partial \mathcal{F}}\right] \tag{11}$$

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.

The implementation of an adjoint based method, requires several steps which are describing by the flow-chart in Figure 12.

A. Gradient formulation for Airplane

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[15,13] 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

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

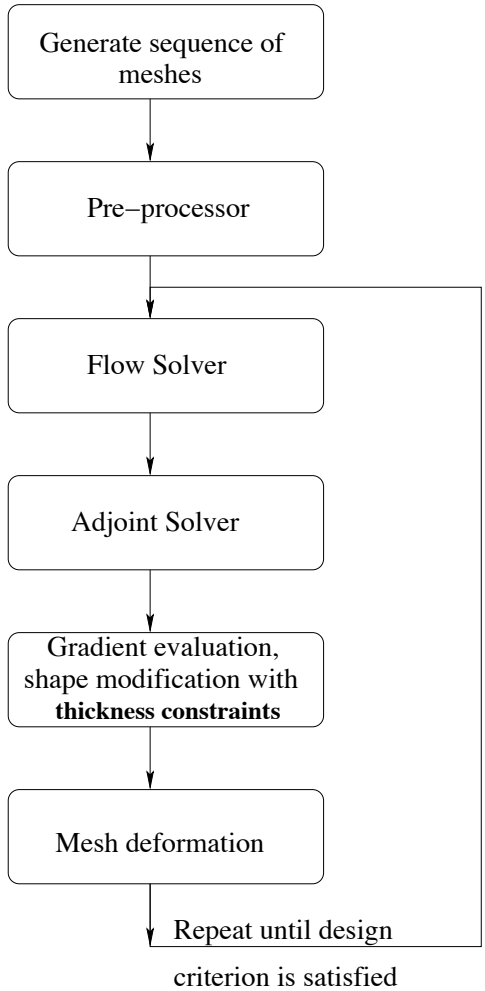


Figure 12. Flow chart of the overall design process

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,

$$\begin{aligned} \int_{\mathcal{D}} \psi^T \delta R d\mathcal{D} &= \int_{\mathcal{D}} \frac{\partial}{\partial \xi_i} C_i(\delta w - \delta w^*) d\mathcal{D} \\ &= \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_i} C_i(\delta w - \delta w^*) \\ &= \int_{\mathcal{B}} \psi^T C_i(\delta w - \delta w^*) d\mathcal{B} \end{aligned} \tag{12}$$

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.

$$\begin{aligned} \delta I = \int_{\mathcal{B}} \psi^T (\delta S_{2j} f_j + C_2 \delta w^*) d\xi_1 d\xi_3 - \\ \int\int_{\mathcal{B}} (\delta S_{21}\psi_2 + \delta S_{22}\psi_3 + \delta S_{23}\psi_4) p d\xi_1 d\xi_3 \end{aligned} \tag{13}$$

B. 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

$$(u, v) = \int_a^b uv dx$$

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,[21] 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[11]

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

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.

C. Modifications to the Airplane Code to Treat the Adjoint Equations

In order to adapt the numerical scheme to treat the adjoint equations three main modifications were required.

First, because the adjoint equation appears in a non-conservative quasi-linear form, the convective terms have to be calculated in a different manner. The derivatives $\frac{\partial \psi}{\partial x_i}$ are calculated by applying the Gauss theorem to the polyhedral control volume consisting of the tetrahedrons that surround each node. Thus the formula

$$\frac{\partial \psi}{\partial x_i} = \frac{1}{V} \int_S \psi dS_{x_i} \tag{14}$$

is replaced by its discrete analog, and the contributions are accumulated by edge and face loops in the same manner as the flux balance of equation (1). The transposed Jacobian matrices are simplified by using a transformation to the symmetrizing variables. Thus the Jacobian for flux in the x direction is expressed as

$$A = M\hat{A}M^{-1}, A^T = M^{-1}\hat{A}M^T$$

where

$$M = \begin{pmatrix} \frac{\rho}{c} & 0 & 0 & 0 & -\frac{1}{c^2} \\ \frac{\rho u}{c} & \rho & 0 & 0 & -\frac{u}{c^2} \\ \frac{\rho v}{c} & 0 & \rho & 0 & -\frac{v}{c^2} \\ \frac{\rho w}{c} & 0 & 0 & \rho & -\frac{w}{c^2} \\ \frac{\rho H}{c} & \rho u & \rho v & \rho w & -\frac{q^2}{2c^2} \end{pmatrix} \tag{15}$$

and

$$\hat{A} = \begin{pmatrix} Q & S_x c & S_y c & S_y c & 0 \\ S_x c & Q & 0 & 0 & 0 \\ S_y c & 0 & Q & 0 & 0 \\ S_z c & 0 & 0 & Q & 0 \\ 0 & 0 & 0 & 0 & Q \end{pmatrix} \tag{16}$$

Second, the direction of time integration to a steady state is reversed because the directions of wave propagation are reversed. Third, while the artificial diffusion terms are calculated by the same subroutines that are used for the flow solution, they are subtracted instead of added to the convective terms to give a downwind instead of an upwind bias. Because of the reversed sign of the time derivatives the diffusive terms in the time dependent equation correspond to the diffusion equation with the proper sign.

D. 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; more research is required to develop general techniques in this regard. In our current approach we introduce cutting-planes at various span-wise locations along the wing and transform 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.

E. Mesh Deformation

The modifications to the shape of the boundary are transferred to the volume mesh using a 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.

F. Parallel Implementation of the Flow Solver

For computational efficiency a multigrid procedure[12] was implemented for both the flow and adjoint solver in which the coarser grids are either obtained through an independent mesh generator or through an edge-collapsing algorithm. In either approach, transfer coefficients between the various meshes are accumulated in a pre-processing step and recomputed when the meshes are deformed.

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.

G. 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 13(a)-13(d), 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 14(a)-14(d). The drag has been reduced from 235 counts to 215 counts in about 8 design cycles. The lift was constrained 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 15(a)-15(d) shows the pressure distribution before and Figures 16(a)-16(d) after the redesign. The shape optimization procedure resulted in a drag reduction of 11 counts.

H. Shape Optimization of Supersonic Business Jets

The design method has also been applied to several supersonic business jet configurations. 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. An example is presented next. A wing-body configuration was obtained from Dassault and several re-design were carried out using SYN88, an implementation of the design optimization for structured exahedral meshes that preserve both the wing planform and the body shape. Each point on the wing surface is regarded as an independent design parameter. Also, the leading edge camber was introduced as an additional design variable since it was found to further reduce the drag.

Figure 16 shows the computed pressure distribution on the top of the wing surface, and at three spanwise cut of the original wing-body geometry. The incoming Mach number is 1.8 and the coefficient of lift is constrained at the nominal target value of .18. The computed pressure drag is 159 counts.

Figure 17 shows the results of a shape optimization run after 30 design cycle. The incoming Mach number is 1.8 and the coefficient of lift is constrained at the nominal target value of .18. The computed pressure drag is 145 counts, a 8.8% improvement. Since the thickness of the wing was not allowed to decrease, the drag reduction is mainly derived from modification of the camber.

Figures 18 and 19 show the results obtained by allowing the thickness of the wing to decrease below the original value by 10% and 20%. The pressure drag is reduced to 140 counts , and to 135 counts, respectively.

VI. Conclusions

Since their inception twenty years ago, CFD methods using unstructured tetrahedral meshes have been successfully applied in a variety of projects, including the McDonnell-Douglas MD-11 and the NASA HSCT studies described here. These are only representative studies: There are many different examples on how aircraft design teams have utilized the rapidly provided information of aerodynamic analysis and shape optimization to make improvements to their aircraft configurations. The Airplane Code was also licensed to Dornier and subsequently transferred to EADS via a series of company acquisitions; Deutsche Aerospace acquired Dornier, which in turn was later absorbed by EADS. At EADS, the Aerodynamics group developed an enhanced version of the code called Airplane+, written in the C language by Edwin van der Weide, and used on projects like the X-31. Other notable work which was enabled by the unstructured-mesh developments at Princeton include Dimitri Mavriplis' NSU2D and NSU3D methods.

CFD analysis and optimization is not intended to replace the judgment and insight of the aircraft designers. Rather, it should properly be viewed as an enabling tool that allows the designers to focus their efforts on the creative aspects of aircraft design, by relieving them of the need to spend large amounts of time exploring small variations.

However, the existing problems of geometry modelling, CAD repair and grid generation present a major bottleneck in the design process. In order to alleviate this problem, we are currently in the process of implementing an extension of the method to unstructured meshes of arbitrary elements.[22] Our goal is to produce a "mesh-blind" scheme, which does not need to know what kind of cells are contained in the mesh, but which should retain the computational efficiency of structured-mesh methods.

Acknowledgment

Portions of the work performed at Princeton University has been sponsored by the Office of Naval Research under several grants. The support of ONR is gratefully acknowledged. Also, the development of the Airplane Code described in this paper has benefited from the generous support of Cray Research, IBM, and Convex Computers.

References

[1]A. Jameson, T.J. Baker and N.P. Weatherill, Calculation of Inviscid Transonic Flow over a Complete Aircraft *AIAA Paper 86-0103*, 24th AIAA Aerospace Sciences Meeting, Reno, January, 1986.

[2]Richard B. Pelz Transonic Flow Calculations Using Triangular Finite Elements Ph.D Thesis, MAE 1617-T, Princeton University, September 1983.

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

[4]Jameson A., Aerodynamic Design Using Control Theory, Journal of Scientific Computing",1988,3,pp 233–260.

[5]A. Jameson and Luigi Martinelli, Aerodynamic Shape Optimization Techniques Based on Control Theory, *CIME (International Mathematical Summer Center)*, Martina Fran-ca, Italy, June 1999.

[6]Jameson A., Optimum Aerodynamic Design via Boundary Control, *RIAC Technical Report 94.17, Princeton University Report MAE 1996, Proceedings of AGARD FDP/Von Karman Institute Special Course on "Optimum Design Methods in Aerodynamics"*, Brussels, April 1994, pp. 3.1-3.33.

[7]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.

[8]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*, 9th AIAA Symposium on Multidisciplinary Analysis and Optimization, Atlanta, September, 2002.

[9]A. Jameson, Transonic Flow Calculations, *Princeton University, MAE Report No. 1651.*

[10]T. H. Pulliam and J. L. Steger, Implicit Finite Difference Simulations of three dimensional Compressible Flow, *AIAA Journal* Vol. 18, 1980, pp. 159-167.

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

[12]Jameson A., Mavriplis D J and Martinelli L, Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes, *ICASE Report 89-11*, *AIAA Paper 89-0283*, AIAA 27th Aerospace Sciences Meeting, Reno, January, 1989.

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

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

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

[16]JC Vassberg and KB Daily, AIRPLANE: Experiences, Benchmarks and Improvements, *AIAA Paper 90-2998-CP*, AIAA Applied Aerodynamics Conference, Portland, OR, August, 1990.

[17]JC Vassberg, KB Daily and DM Friedman, AIRPLANE: Unstructured-Mesh Applications, *SAE Paper 901857*, Aerospace Technology Conference and Exposition, Long Beach, CA, October, 1990.

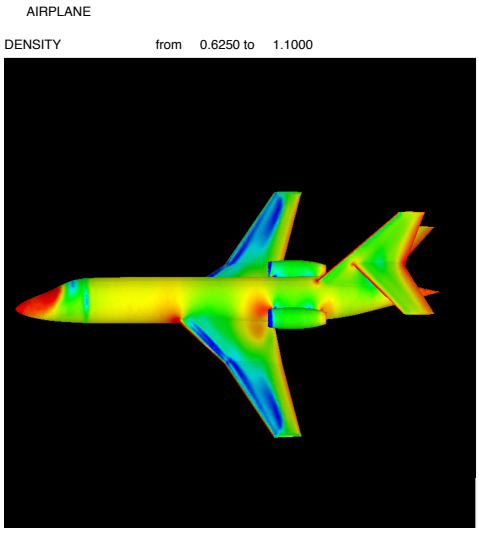
[18]JC Vassberg, Techniques for Efficient, Unstructured-Mesh Calculations, Dissertation, University of Southern California, 1992.

[19]JC Vassberg, A Fast, Implicit Unstructured-Mesh Euler Method, *AIAA Paper 92-2693*, AIAA Applied Aerodynamics Conference, Palo Alto, June, 1992.

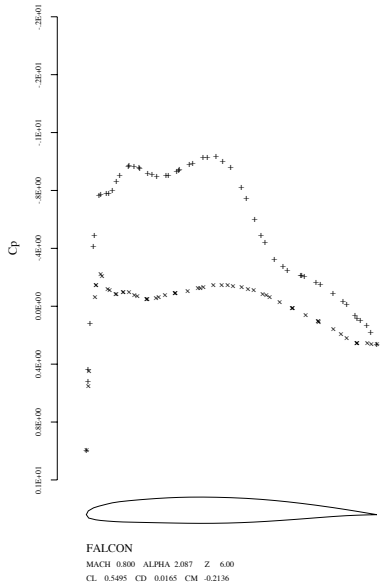
[20]JC Vassberg, An Unstructured-Mesh Euler Method for Multielement Airfoils, *AIAA Paper 90-3051*, AIAA Applied Aerodynamics Conference, Portland, OR, August, 1990.

[21]Jameson A and Vassberg J. C. Studies of alternative numerical optimization methods applied to the brachistochrone problem. *Proc Proceedings of OptiCON'99*, Newport Beach, CA, October Computational Fluid Dynamics Journal, vol. 9, 2000, pp281-296

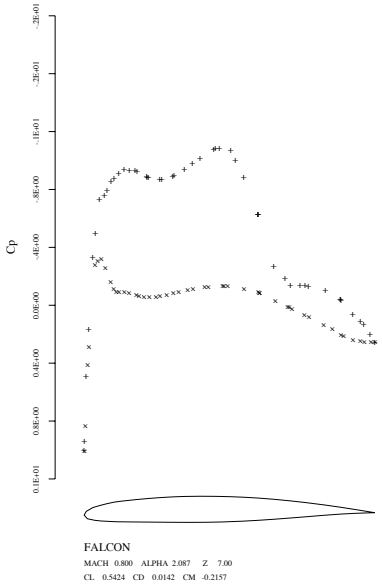
[22]Georg May and Antony Jameson, Unstructured Algorithms for Inviscid and Viscous Flows Embedded in a Unified Solver Architecture *AIAA Paper 2005-0318*, AIAA Aerospace Sciences Meeting & Exhibit, AIAA Paper 2005-0318, Reno, NV, January 10-13, 2005.



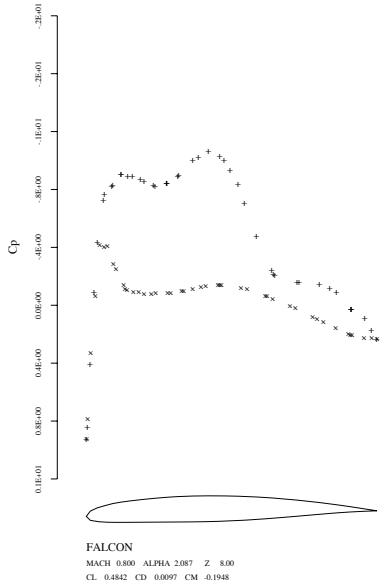
(a) Density contours for a business jet at $M = 0.8$, $\alpha = 2$



(b) Pressure distribution at 66 % wing span

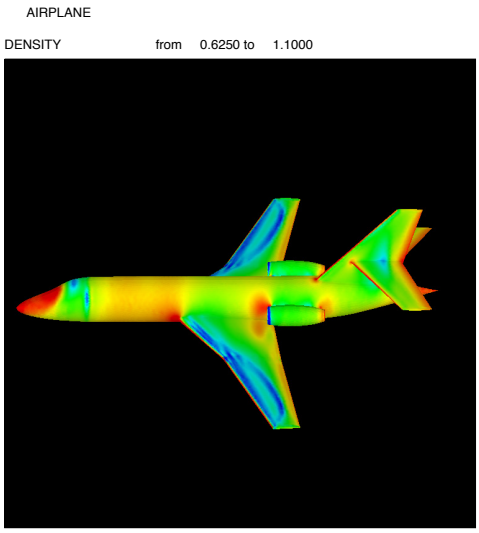


(c) Pressure distribution at 77 % wing span

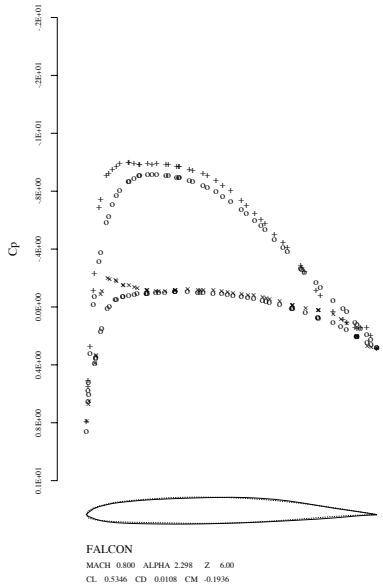


(d) Pressure distribution at 88 % wing span

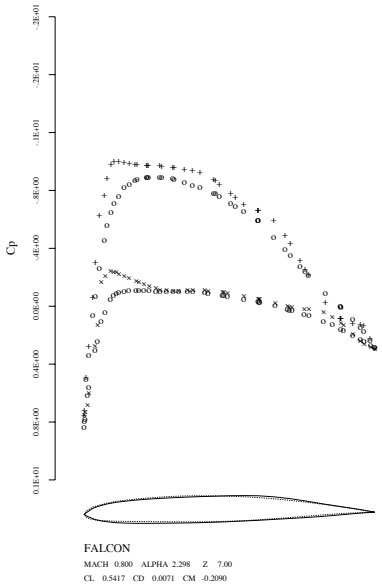
Figure 13. Pressure Distribution on Original Configuration of a Business jet



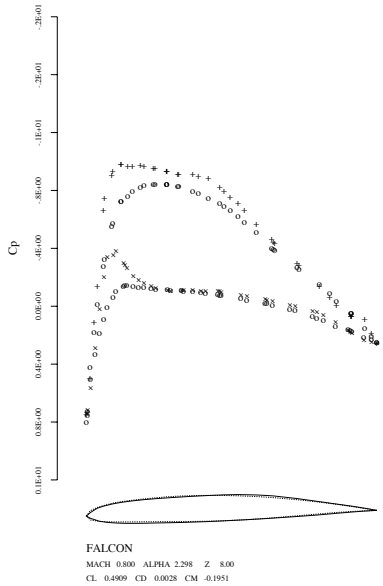
(a) Density contours for a business jet at $M = 0.8$, $\alpha = 2.3$, after redesign



(b) Pressure distribution at 66 % wing span, after redesign, Dashed line: original geometry, solid line: redesigned geometry

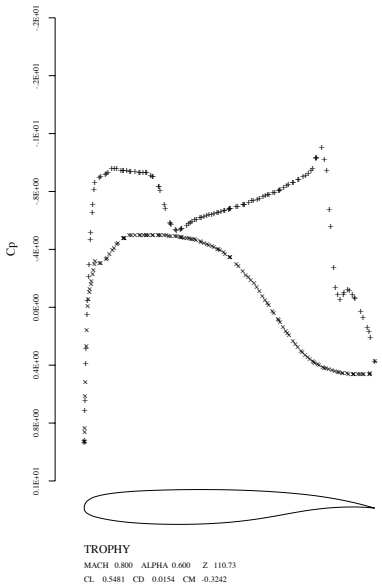


(c) Pressure distribution at 77 % wing span, after redesign, Dashed line: original geometry, solid line: redesigned geometry

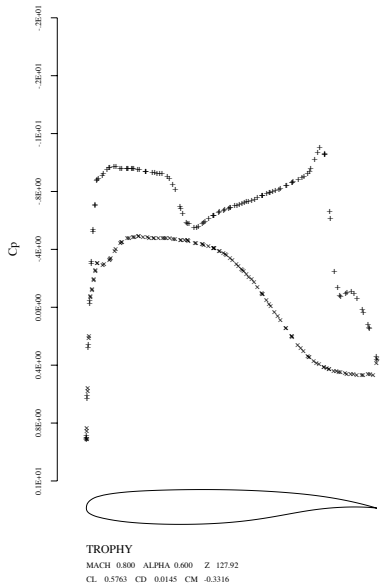


(d) Pressure distribution at 88 % wing span, after redesign, Dashed line: original geometry, solid line: redesigned geometry

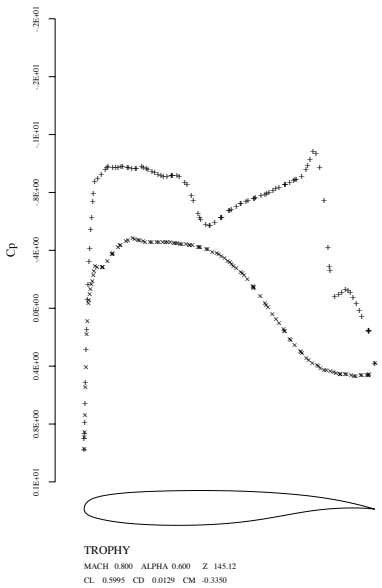
Figure 14. Pressure Distribution on Re-designed Configuration of a Business jet



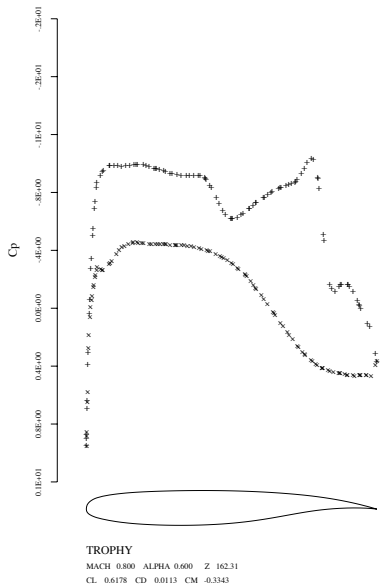
(a) Pressure distribution at 55 % wing span



(b) Pressure distribution at 66 % wing span

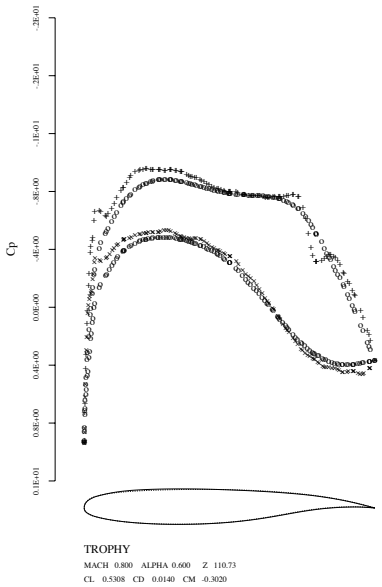


(c) Pressure distribution at 77 % wing span

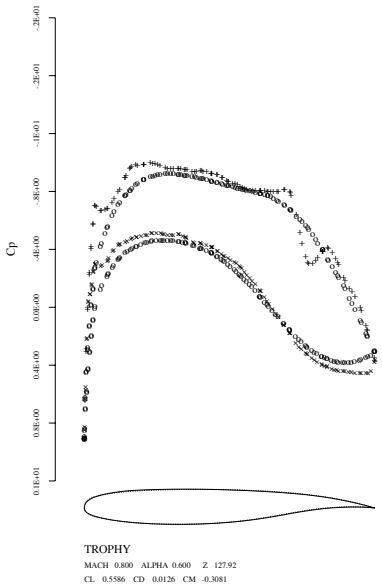


(d) Pressure distribution at 88 % wing span

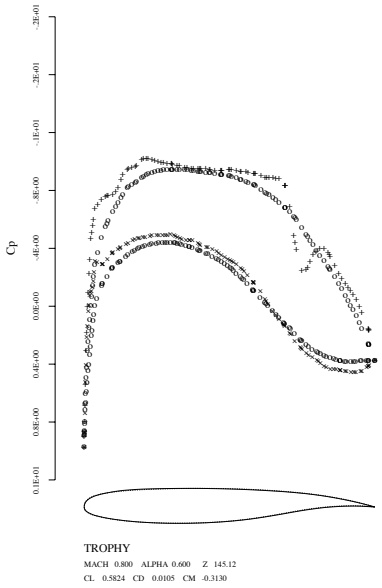
Figure 15. Pressure Distribution on Original Configuration of a Business jet



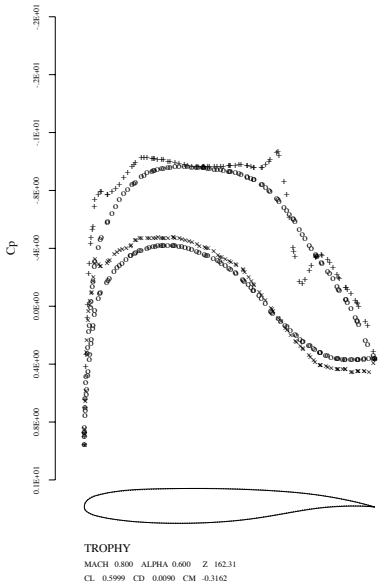
(a) Pressure distribution at 55 % wing span



(b) Pressure distribution at 66 % wing span



(c) Pressure distribution at 77 % wing span



(d) Pressure distribution at 88 % wing span

Figure 16. Pressure Distribution on Re-designed Configuration of a Business jet

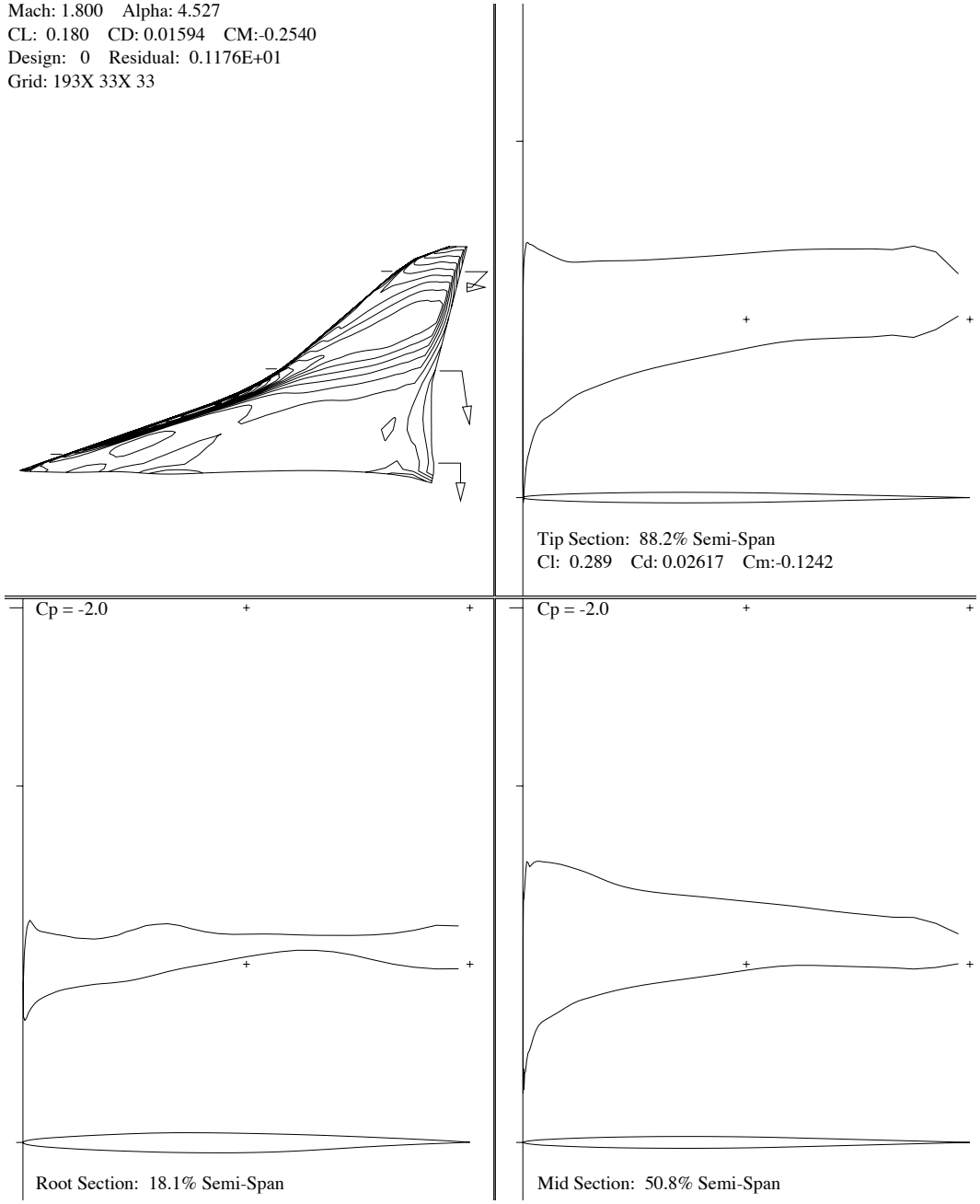


Figure 17. Pressure distribution - Original Supersonic Business Jet

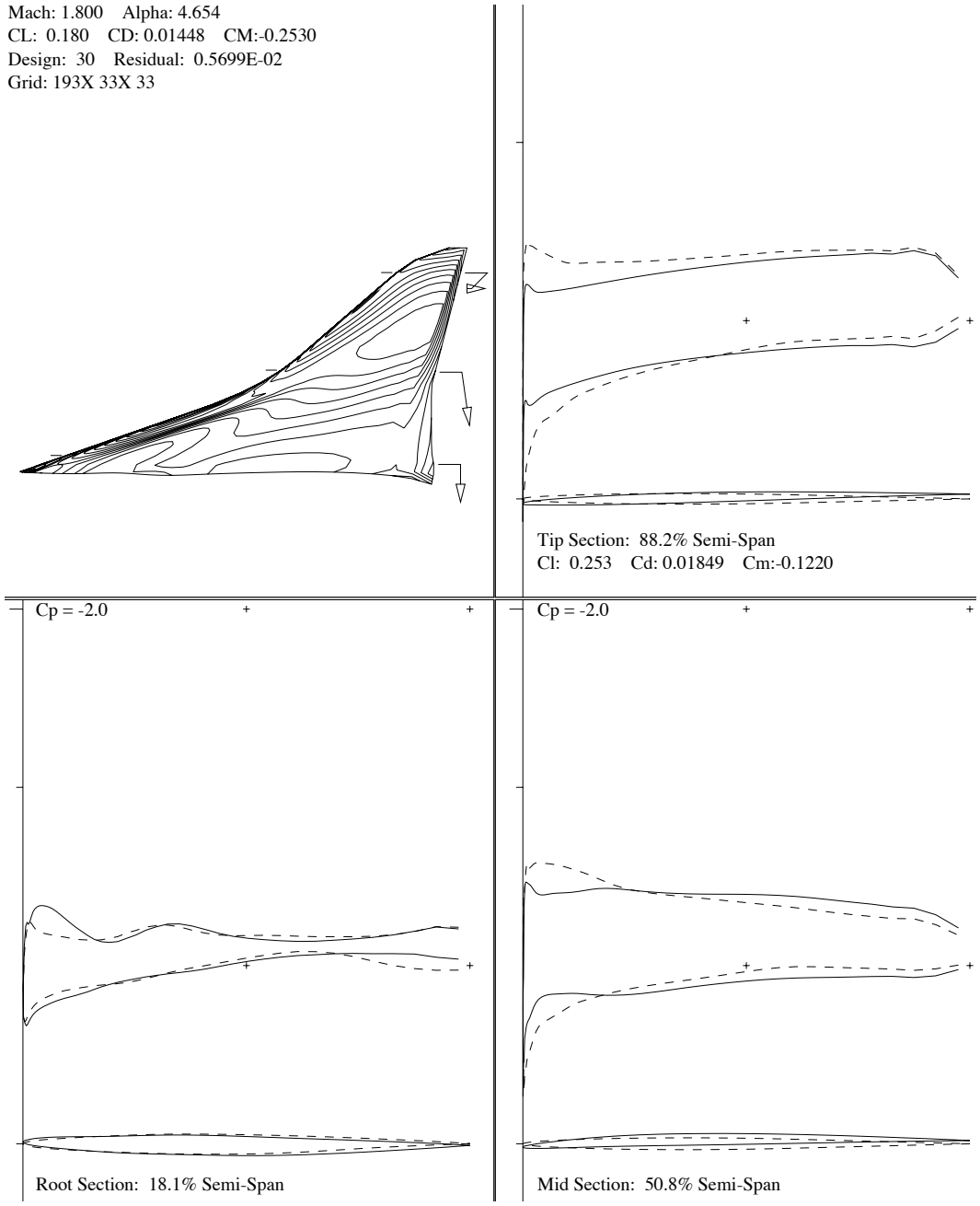


Figure 18. Pressure distribution - Redesigned Supersonic Business Jet - Thickness was not allowed to decrease

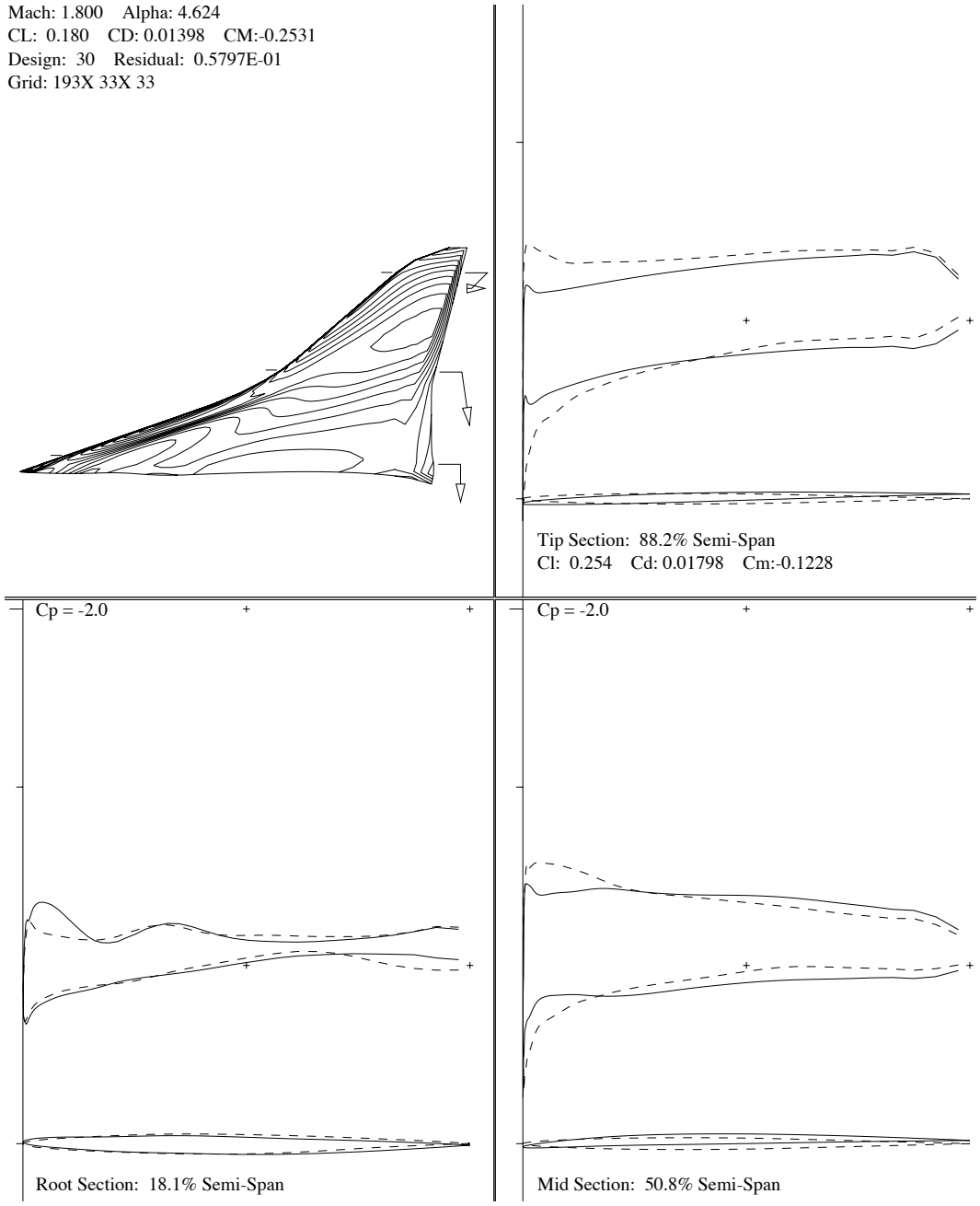


Figure 19. Pressure distribution - Redesigned Supersonic Business Jet - Thickness was allowed to decrease 90%

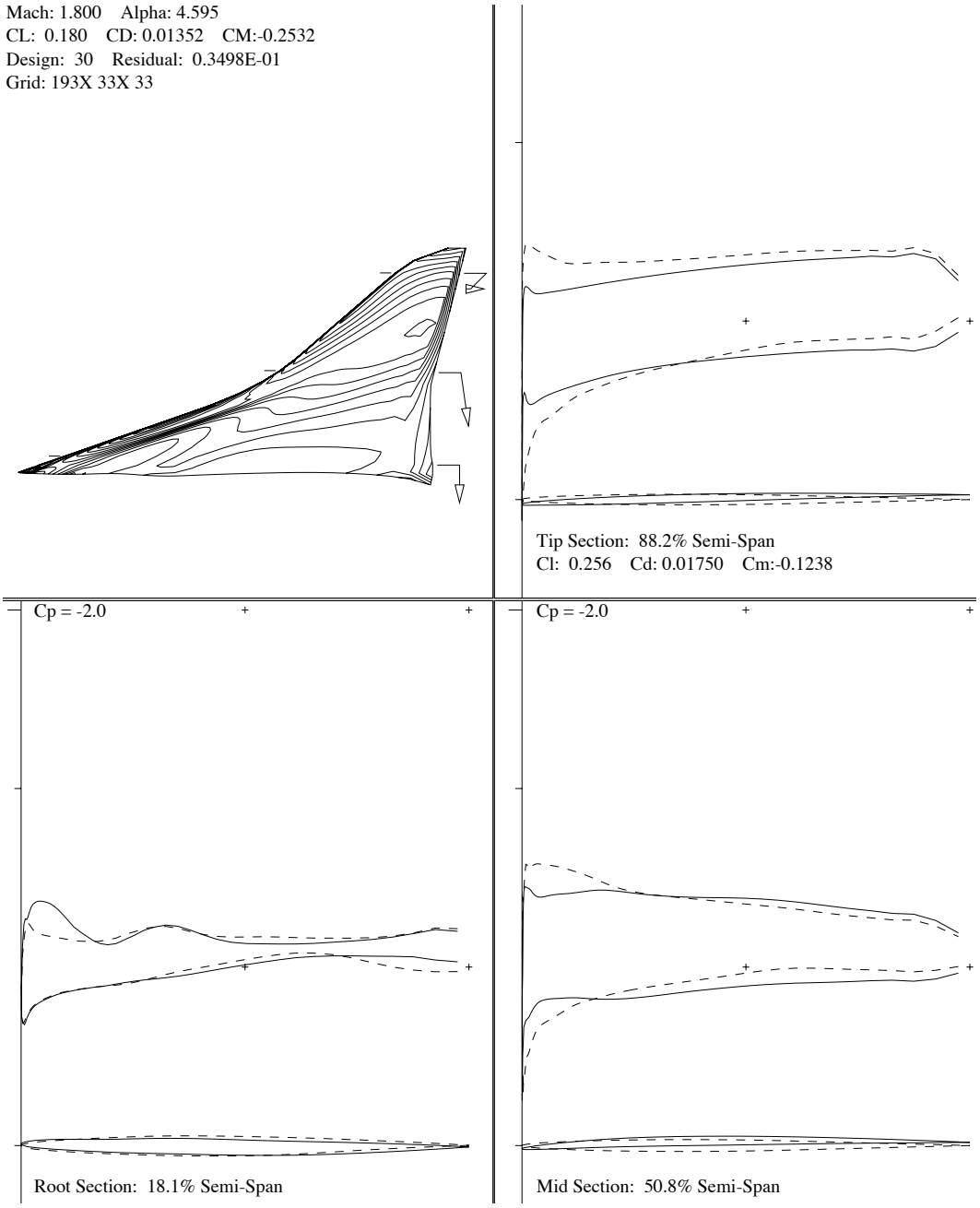


Figure 20. Pressure distribution - Redesigned Supersonic Business Jet - Thickness was allowed to decrease 80%