

## AE 8900 Special Project

---

### Validation of APAS Aerodynamic Predictions with a Navier-Stokes CFD Analysis of a Hankey-Wedge Forebody

Kirk Sorensen  
December 9, 1999

## Table of Contents

---

List of Figures . . . . .	3
Abstract . . . . .	4
Introduction . . . . .	5
Benchmarking APAS using GASP . . . . .	6
Implementation . . . . .	7
Results and Analysis . . . . .	7
Mach 1.5 Cases . . . . .	8
Mach 6 Cases . . . . .	11
Mach 10 Cases . . . . .	14
Conclusions . . . . .	17
Can CFD be Integrated into an MDO Environment? . . . . .	17
Recommendation . . . . .	19
Summary . . . . .	20
References . . . . .	21
Appendix A: Detailed Description of Analysis Procedure using GASP	
Appendix B: Detailed Description of Analysis Procedure using APAS	
Appendix C: Geometric Equations for Wing Definition	
Appendix D: Zonal Boundaries and Visualization	

## List of Figures

---

Figure 1: Stargazer vehicle in flight, deploying its upper stage.

Figure 2: Pressure distribution and flowfield contours of the Stargazer forebody at Mach 1.5, AOA 6°

Figure 3: Lift and Drag Coefficient Results from GASP and HABP for Mach 1.5

Figure 4: Lift and Drag Coefficient Data from GASP and HABP for Mach 1.5

Figure 5: Lift and Drag Coefficient Results from GASP and HABP for Mach 2.0

Figure 6: Lift and Drag Coefficient Results from GASP and HABP for Mach 3.0

Figure 7: Pressure distribution and flowfield contours of the Stargazer forebody at Mach 6, AOA 4°

Figure 8: Lift and Drag Coefficient Results from GASP and HABP for Mach 6.0

Figure 9: Lift and Drag Coefficient Data from GASP and HABP for Mach 6.0

Figure 10: Shock structure and leading-edge detail of Stargazer forebody at Mach 10, AOA 6°

Figure 11: Lift and Drag Coefficient Results from GASP and HABP for Mach 10.0

Figure 12: Lift and Drag Coefficient Data from GASP and HABP for Mach 10.0

Figure 13: Drag Coefficient at AOA=0, 6 from GASP and HABP for all Mach numbers

Figure 14: Computational Time required for analyses

## Validation of APAS Aerodynamic Predictions with a Navier-Stokes CFD Analysis of a Hankey-Wedge Forebody

Kirk F. Sorensen

*Georgia Institute of Technology, Atlanta, Georgia 30332*

While in the conceptual design phase of launch vehicles, aerodynamic data is often obtained through the use of a simple analytic program called APAS (Aerodynamic Preliminary Analysis System). While suffering from an archaic and temperamental interface, APAS yields results swiftly for simple geometries at a variety of angles of attack and Mach numbers. The results from APAS are compared to those obtained through the analysis of the same vehicle shape using a sophisticated CFD program called GASP (General Aerodynamic Simulation Program). The comparison is made on the forebody for the Stargazer Bantam launch vehicle, which is based on a Hankey-wedge design. Significant differences are noted and techniques to improve the accuracy of APAS output data are suggested.

## Introduction

In the conceptual design environment, there is a constant tradeoff between the accuracy of analysis and the speed of analysis. Few disciplines exemplify this tradeoff more dramatically than aerodynamic analysis.

At first glance, the requirements on aerodynamic analysis seem rather simple. Typically, the end product of the analysis is a table of lift and drag coefficients at a series of Mach numbers corresponding to a series of altitudes. This information is fed into a trajectory program, such as POST (Program to Optimize Simulated Trajectories), to determine the optimal ascent (or descent) trajectory a vehicle should fly. If a trimmed solution is desired, then moment coefficients at different flap angles are included as well. But how the aerodynamics designer arrives at the lift and drag coefficients is a source of uncertainty and difficulty.

Since lift and drag coefficients are derived from the local pressures and shear stresses on a body surface, most simple analyses are local in nature; in other words, they determine only the pressure coefficient at a particular body location and inclination to the freestream flow. They do not take into account the complexities of shock interactions, body shielding, or a host of other phenomena. For forces that are based on viscous effects, such as shear stress on the body surface, empirical equations or other simplifications are used.

A more complex analysis would solve the entire flowfield around a vehicle, or in the case of a supersonic analysis, the entire disturbed flowfield around the vehicle. A flowfield analysis would capture many of the effects that elude a local surface inclination method, such as shock and expansion wave interactions, curving streamlines, pressure gradients, and so forth. But flowfield methods nearly always rely on computational fluid dynamics (CFD) to solve a discretized flowfield, and require extremely fast computers and a great deal of processing time to solve even one case.

Hence, conceptual designers have tended towards computationally inexpensive local surface inclination methods in their analyses. At the Georgia Tech Space Systems Design Laboratory (SSDL), a NASA code originally developed by Rockwell International is extensively used in conceptual analysis. APAS<sup>1</sup> (Aerodynamic Preliminary Analysis System) is an interface that allows the definition of geometries and flight conditions; this information is then solved by UDP<sup>2</sup> (Unified Distributed Panel), which is a vortex-panel code suitable for subsonic and supersonic analysis, or HABP<sup>3</sup> (Hypersonic Arbitrary Body Program), which uses a variety of techniques to locally solve for the pressure on each body region.

How accurate are these solutions? To determine the answer, they could be compared to actual flight data, or wind tunnel analysis, or a detailed flowfield analysis (i.e. CFD). Unfortunately, all but the last technique falls outside the realm of feasibility for the resources of the SSDL. Hence, a CFD comparison of Stargazer to APAS was undertaken to gain a better understanding of just how accurate these conceptual tools are; to benchmark them against a much more detailed (and time-consuming) analysis, and suggest improvements or changes where applicable.

## Benchmarking APAS using GASP

The configuration chosen for the analysis is the Stargazer, a vehicle that has been developed by SSDL under contract to NASA's Marshall Space Flight Center. Stargazer's mission is to deploy small satellites (<300 lbs) for universities and students, with low operational costs and a rapid turnaround. To accomplish this mission, Stargazer takes off horizontally from a runway, and uses rocket-based combined cycle (RBCC) propulsion to attain a maximum velocity of Mach 15 and 250,000 ft altitude. The payload, attached to a small upper stage, is deployed from the Stargazer payload bay, and the upper stage engine fires to insert the payload into a low Earth orbit.

Several points along the Stargazer trajectory were chosen as points of interest and worthy of detailed analysis. Therefore, it was decided to run CFD solutions of the Stargazer vehicle at Mach 1.5, 6, and 10. Mach 1.5 represented a low supersonic Mach number where hypersonic simplifications are increasingly inaccurate. Mach 6 is the velocity where Stargazer transitions from ramjet to scramjet mode, and Mach 10 is the velocity where Stargazer transitions from scramjet to fully rocket mode.



**Figure 1:** Stargazer vehicle in flight, deploying its upper stage.

A variety of angles-of-attack (AOA or  $\alpha$ ) were chosen at each Mach number to identify trends in the aerodynamic data. Angles of  $0^\circ$ ,  $2^\circ$ ,  $4^\circ$ , and  $6^\circ$  were chosen, resulting in 12 CFD runs to be solved overall.

The CFD program that was utilized for this analysis was GASP<sup>4</sup> (General Aerodynamic Simulation Program), which is a structured grid solver developed by AeroSoft. GASP is capable of 3D, Navier-Stokes simulations with a variety of flowfield and gas conditions. For this analysis, the assumption of perfect gases was made, and a solution of the thin-layer Navier-Stokes equations (with adiabatic wall conditions) was chosen due to its computational simplicity.

## **Implementation**

The project began with a three-dimensional solid model of the Stargazer vehicle. Since the vehicle has undergone many iterations, it was necessary to choose a baseline vehicle for the analysis. The decision was made to use the LH2/LOX configuration with an engine thrust-to-weight (T/W) ratio of 15. This design was developed and converged at SSDL in January 1999.

After the geometry was created in SDRC I-DEAS, it was exported in IGES form, suitable for use by GRIDGEN 13, the grid definition program utilized. Sharae McKay of the Aerothermodynamics Research Technology Lab (ARTlab) at Georgia Tech created the structured grids that were used in this analysis.

Problems completing a full vehicle grid led to a reduction in the scope of the project. It was decided to solve only the forebody of the vehicle for each of the cases. Grids for the forebody had been successfully developed, and the reduced computational load made it possible to solve the problems on the computers in the SSDL in the time remaining in the semester.

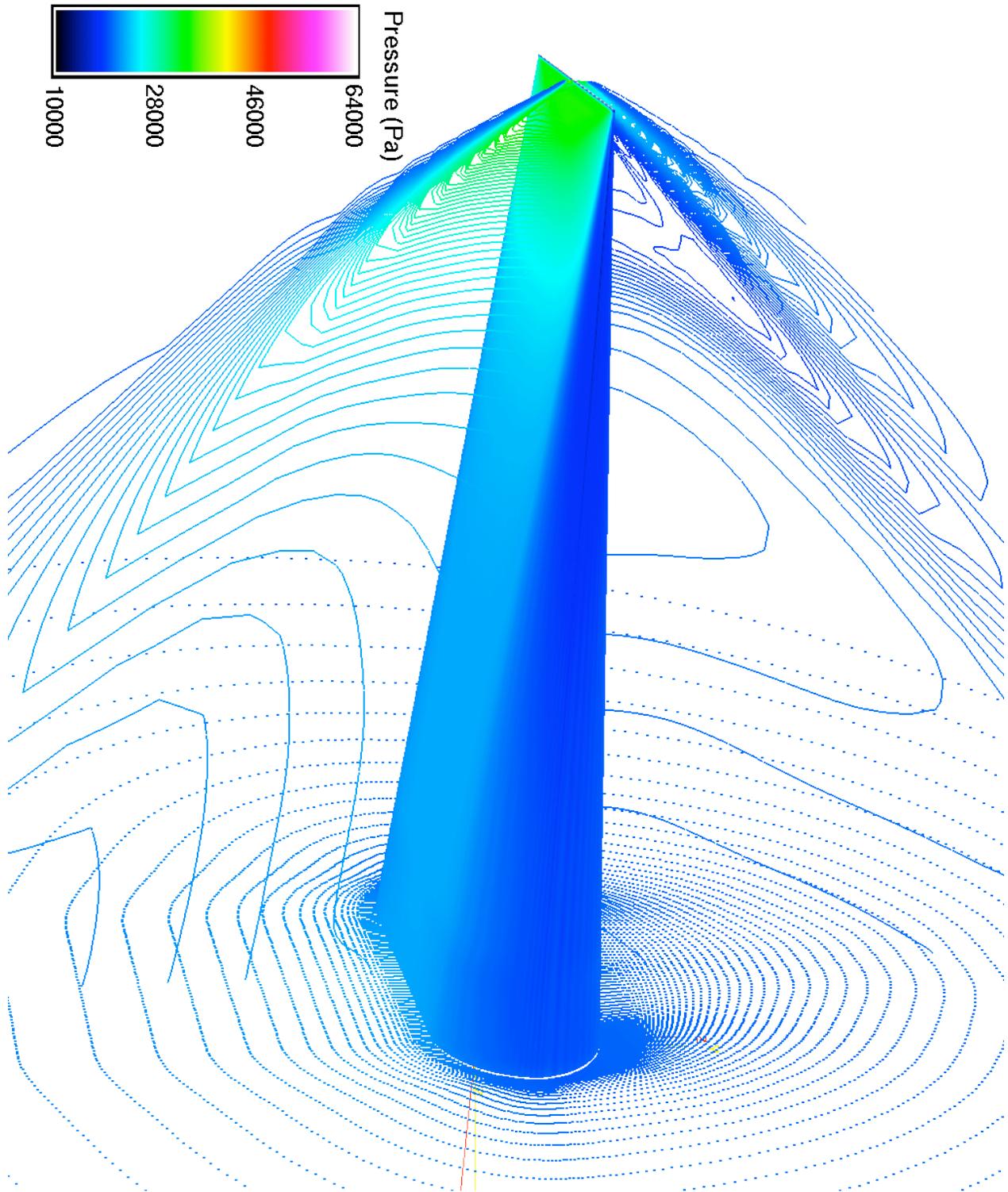
The analysis was conducted using GASP version 3.1.6, and the actual analyses were executed on two SGI Octane computers: Atlas (with an R12000 processor) and Titan (with an R10000 processor). Post-processing was done using FAST (Flow Analysis Software Toolkit), a visualization code created by the NASA Ames Research Center.

Analyses were initially conducted for Mach numbers of 1.5, 6.0, and 10.0, at angles-of-attack of  $0^\circ$ ,  $2^\circ$ ,  $4^\circ$ ,  $6^\circ$ . Questions in the results led to further analyses at Mach 2.0 and 3.0; however, only the first set of Mach numbers will be presented in-depth in this report, although the other two results are used to validate trends and results.

## **Results and Analysis**

The results of the CFD analyses of the Stargazer forebody were compared to similar analyses utilizing APAS/HABP. Since the shape of the Stargazer forebody can accurately be described as a wedge with cone sides, it was decided to compare the CFD results to HABP using both the tangent wedge and tangent cone options of HABP.

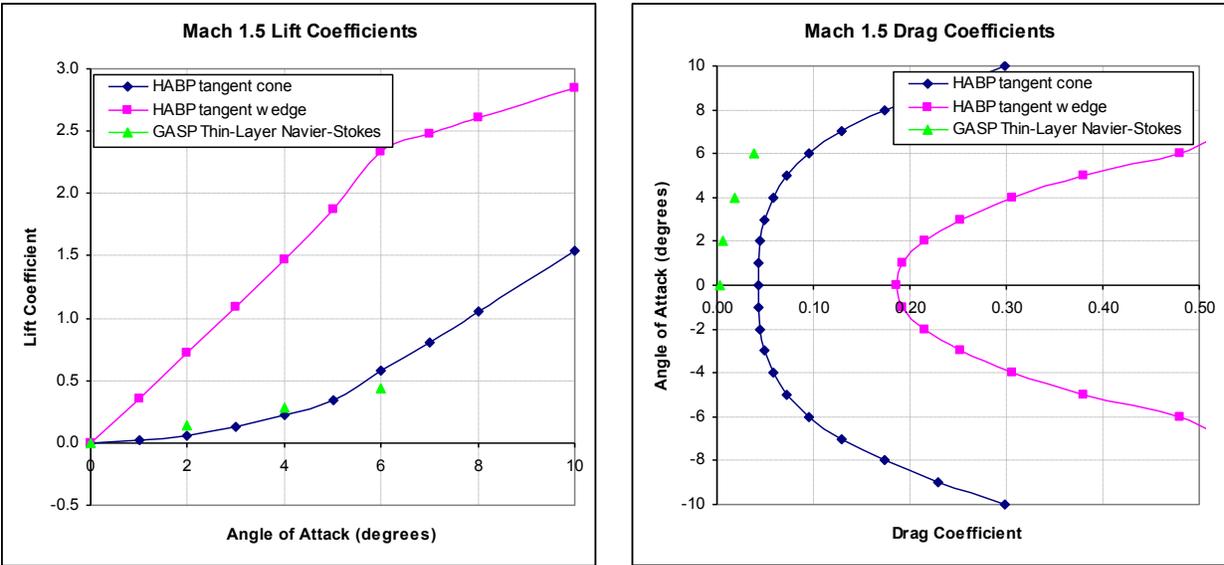
## Mach 1.5 Results



**Figure 2:** Pressure distribution and flowfield contours of the Stargazer forebody at Mach 1.5, AOA 6°

Figure 1 is a visualization of the flowfield around the Stargazer forebody (created using FAST) at Mach 1.5 at 6° AOA. The colors are indications of higher and lower pressures in the flowfield. Several flow structures are obvious. One is the pressure contours visible above and below the forebody indicating the position of the oblique shocks. Although the vehicle is symmetric, they are at different angles because the forebody is inclined to the freestream flow. The points at the backplane are the grid point locations and their color indicates their pressures.

As can be seen from the color distribution of Figure 2, the Mach 1.5 flight regime is characterized by relatively benign shocks and smaller pressure gradients than the other cases examined. The low Mach number also makes the hypersonic assumptions of HARP quite inaccurate. Figure 3 is a series of plots of lift and drag coefficients for both the GASP and HARP results. For lift coefficients, GASP results correlate strongly to the HARP tangent-cone solution. On the graph of drag, however, GASP predicted far lower drag on the forebody than either of the APAS analyses. Figure 4 is a sampling of the raw data used to generate the plots.



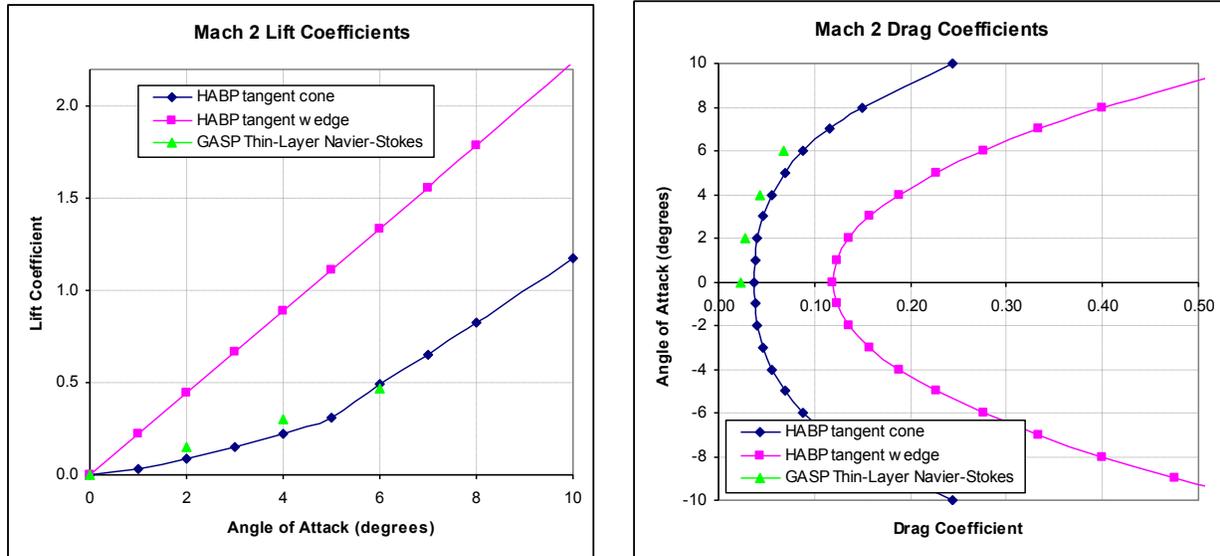
**Figure 3:** Lift and Drag Coefficient Results from GASP and HARP for Mach 1.5

Mach 1.5										
<i>Lift Coefficient</i>				<i>Drag Coefficient</i>				<i>lift/drag coefficient</i>		
AOA	HARP TW	HARP TC	GASP TLNS	HARP TW	HARP TC	GASP TLNS	HARP TW	HARP TC	GASP TLNS	
0.0	0.0000	0.0000	-0.0066	0.1855	0.0435	0.0030	0.0000	0.0000	-2.1877	
1.0	0.3582	0.0175		0.1928	0.0435		1.8579	0.4023		
2.0	0.7195	0.0604	0.1396	0.2148	0.0450	0.0069	3.3496	1.3422	20.2416	
3.0	1.0881	0.1285		0.2523	0.0496		4.3127	2.5907		
4.0	1.4747	0.2218	0.2880	0.3063	0.0584	0.0187	4.8146	3.7979	15.3737	
5.0	1.8766	0.3401		0.3800	0.0729		4.9384	4.6653		
6.0	2.3368	0.5788	0.4397	0.4803	0.0956	0.0388	4.8653	6.0544	11.3441	

**Figure 4:** Lift and Drag Coefficient Data from GASP and HARP for Mach 1.5

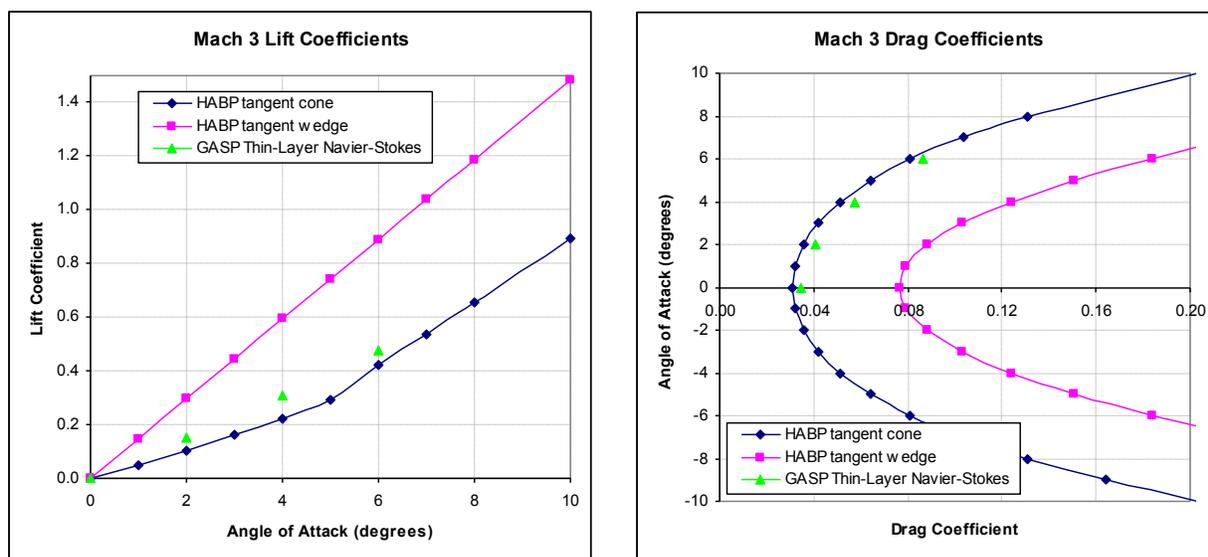
The extremely low values of the drag coefficients is unsettling and counterintuitive. The procedures and derivations used to arrive at these values examined again and again, but no

apparent fault was uncovered. Barring a discovery of error, a prudent course of action would be to compute lift and drag values at Mach numbers close to 1.5, and see if the data correlates closely to the Mach 1.5 data. This was implemented, and the plots of lift and drag for Mach 2 and 3 are shown in figures 5 and 6:



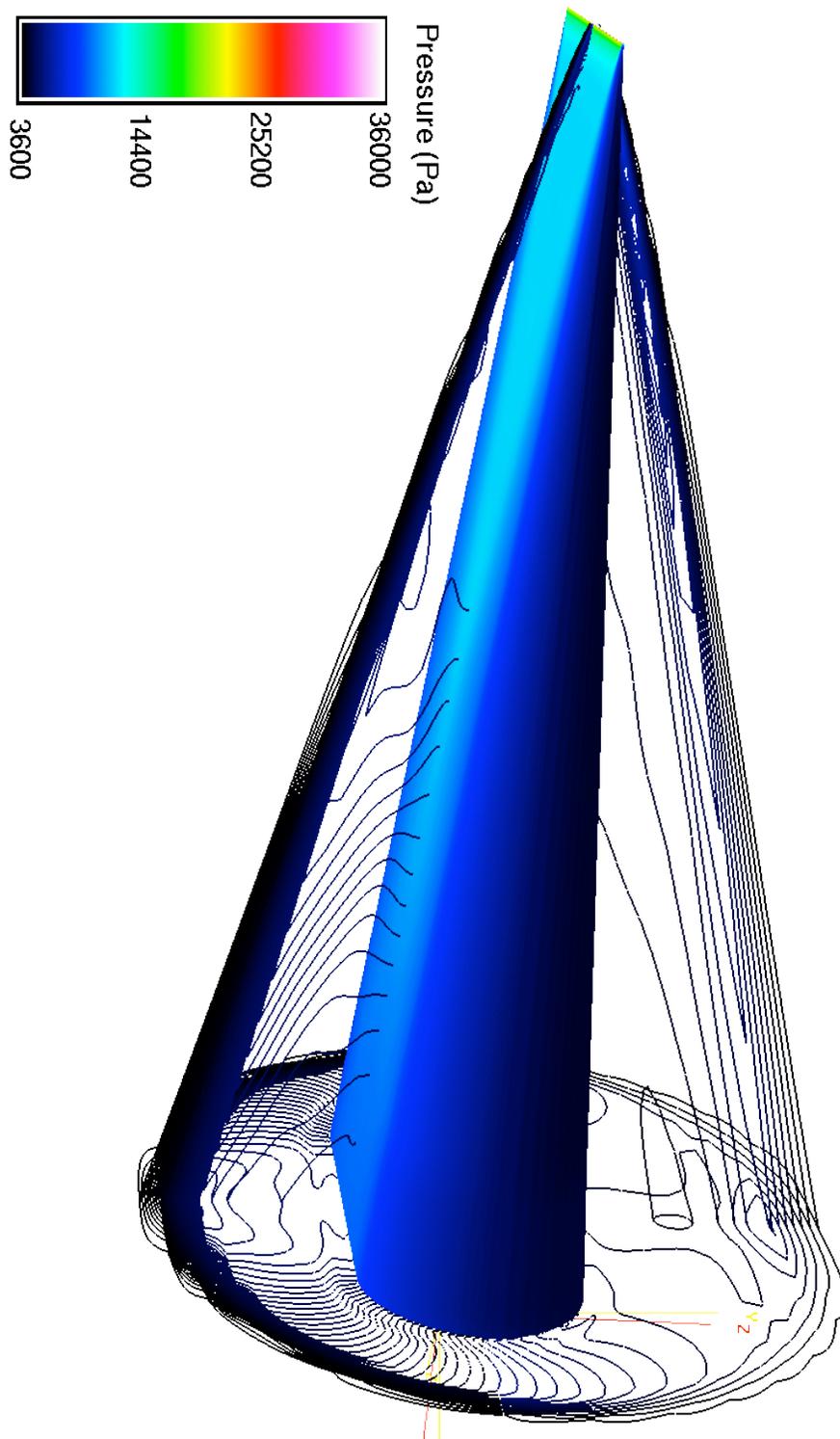
**Figure 5:** Lift and Drag Coefficient Results from GASP and HABP for Mach 2.0

The Mach 2 results again show a close correlation to the HABP tangent-cone solution; and the drag results are below the values predicted by HABP tangent-cone, but no longer are they nearly zero. Observations of the Mach 3 data begin to show a trend in the data: drag coefficients are now greater than the HABP tangent-cone values. The drag coefficients seem to be increasing more rapidly than HABP tangent-cone would predict. Further analysis of this effect will be examined after all results have been presented.



**Figure 6:** Lift and Drag Coefficient Results from GASP and HABP for Mach 3.0

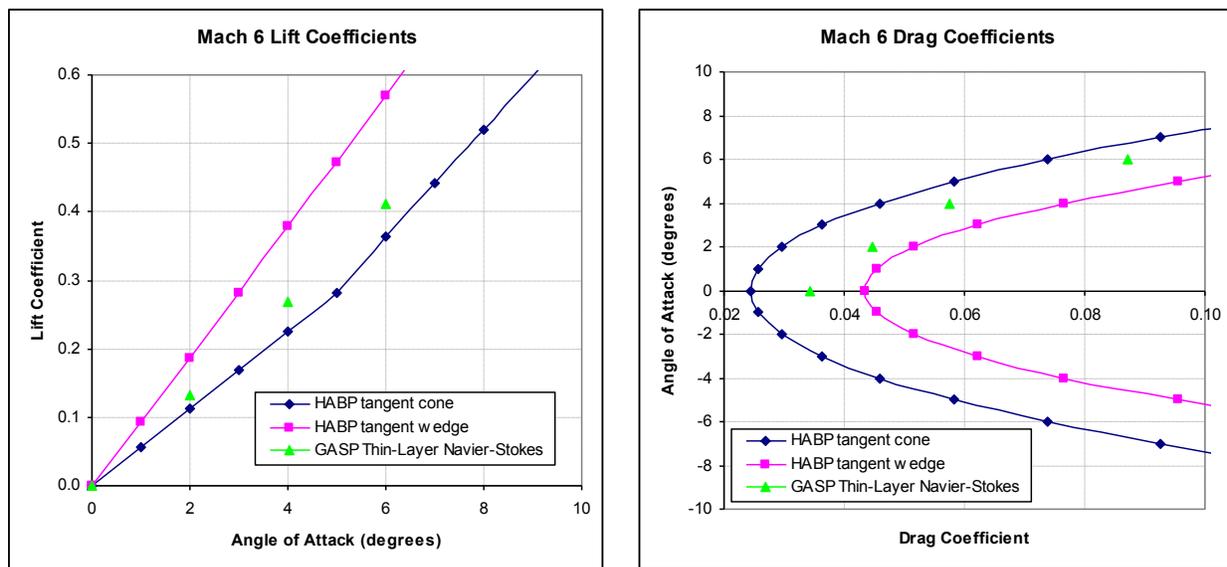
## Mach 6 Results



**Figure 7:** Pressure distribution and flowfield contours of the Stargazer forebody at Mach 6, AOA 4°

A beautiful visualization of the Mach 6 results is presented in Figure 7. Again, pressure distribution is represented, but this time the backplane of the analysis is depicted by contour lines instead of colored grid locations. The shock is significantly more inclined, as would be expected at a higher Mach number, and the most significant variations in pressure occur at the nose region, where a mixed subsonic/supersonic region exists. Another interesting flow feature is the pressure distribution on the lower surface of the forebody; the differential in pressure in the center of the forebody becomes less and less pronounced as the flow proceeds along the forebody. This is due to flow from the lower surface to the upper surface, alleviating the differences in pressure, an important three-dimensional effect that is not captured by the simple methods used in HABP.

As depicted in figures 8 and 9, the results of the Mach 6 analyses were excellent. They fit almost exactly what engineering intuition would suspect; that the lift and drag of a Hankey-wedge forebody would fall somewhere between that predicted by tangent-wedge and tangent-cone solutions, because the geometry itself is a hybrid of wedge and cone surfaces.



**Figure 8:** Lift and Drag Coefficient Results from GASP and HABP for Mach 6.0

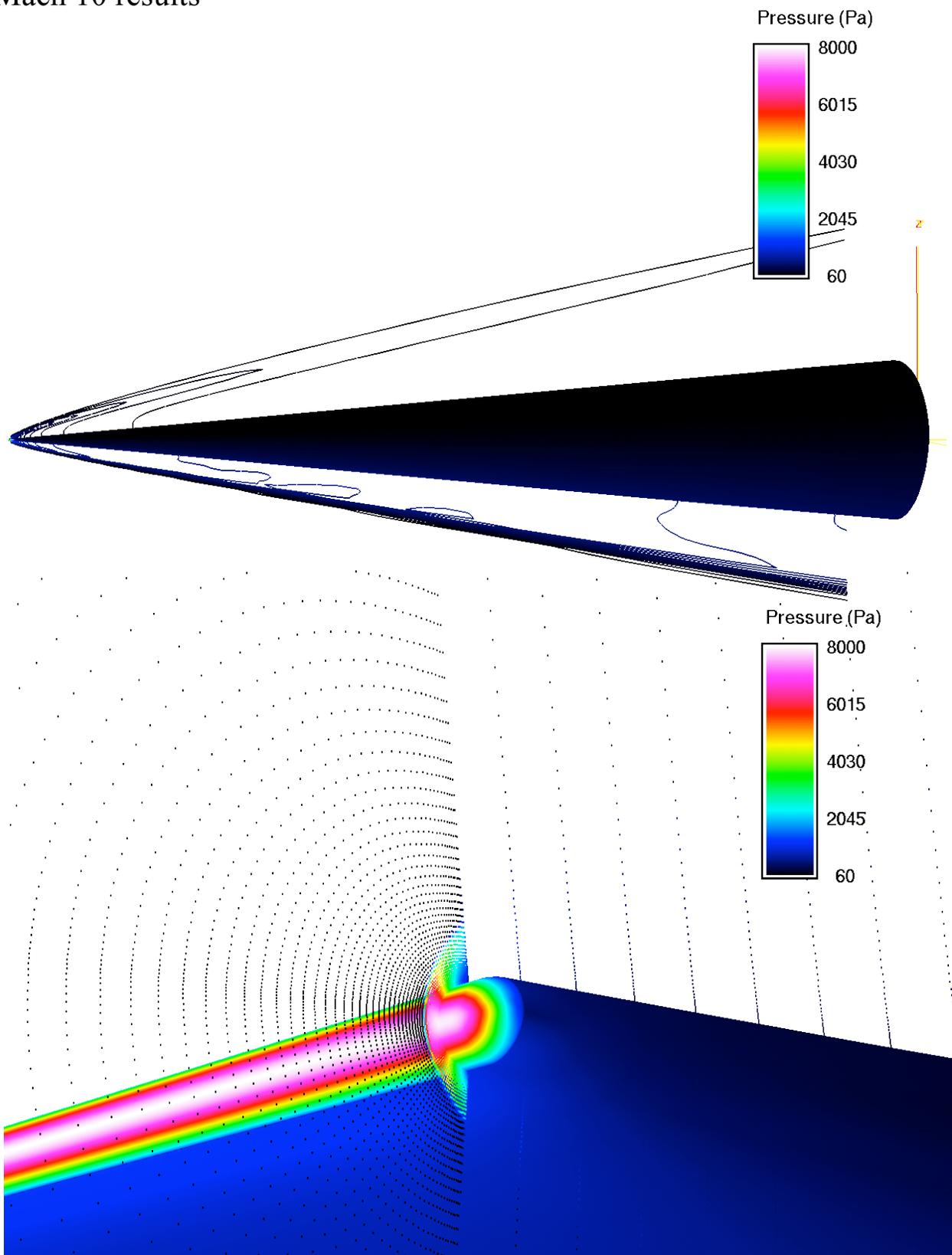
Mach 6										
<i>Lift Coefficient</i>				<i>Drag Coefficient</i>				<i>lift/drag coefficient</i>		
AOA	HABP	TW	HABP TC	GASP TLNS	HABP TW	HABP TC	GASP TLNS	HABP TW	HABP TC	GASP TLNS
0.0	0.0000	0.0000	-0.0005	0.0434	0.0244	0.0342	0.0000	0.0000	-0.0159	
1.0	0.0935	0.0562		0.0455	0.0256		2.0549	2.1953		
2.0	0.1872	0.1125	0.1311	0.0517	0.0296	0.0447	3.6209	3.8007	2.9304	
3.0	0.2814	0.1688		0.0621	0.0364		4.5314	4.6374		
4.0	0.3789	0.2248	0.2689	0.0766	0.0459	0.0575	4.9465	4.8976	4.6759	
5.0	0.4725	0.2807		0.0955	0.0583		4.9476	4.8148		
6.0	0.5696	0.3645	0.4121	0.1187	0.0738	0.0873	4.7987	4.9390	4.7209	

**Figure 9:** Lift and Drag Coefficient Data from GASP and HABP for Mach 6.0

Although the forebody has greater lift than the HABP tangent-cone solution, it still seems to be closer to the tangent-cone values rather than the tangent-wedge values. For the drag coefficients,

the trend is just the opposite. This is the beginning of a trend that will manifest itself more dramatically in the Mach 10 case, and might lead a vehicle designer to consider a different forebody geometry.

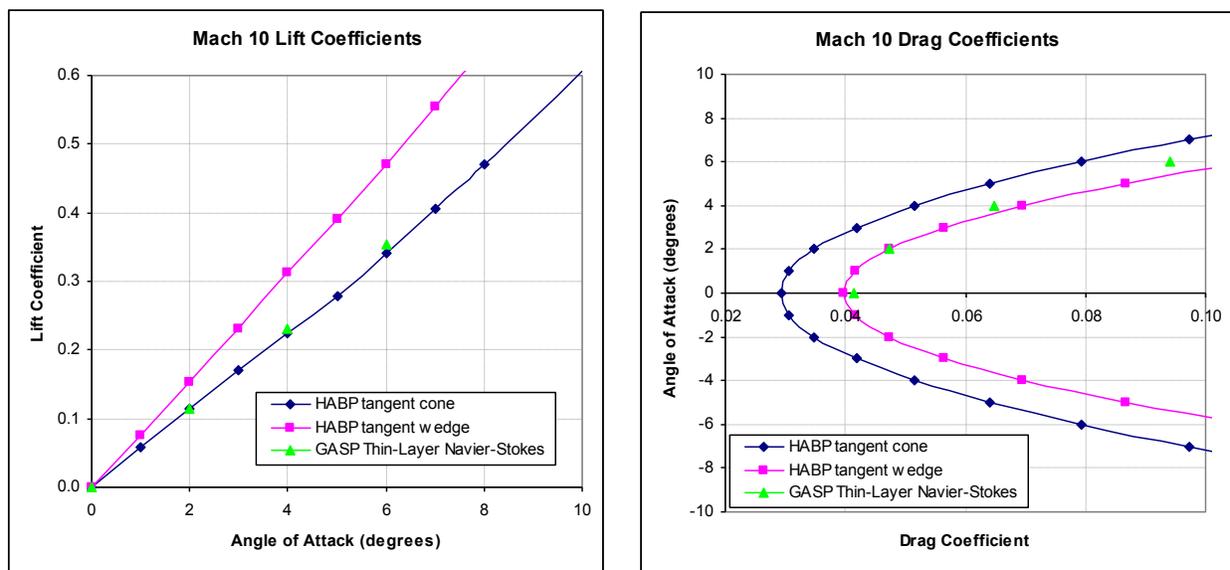
## Mach 10 results



**Figure 10:** Shock structure and leading-edge detail of Stargazer forebody at Mach 10, AOA 6°

Figure 10 presents both a view of the forebody and a detail of the leading edge to demonstrate that at Mach 10, the differentials in pressure on the body surface are insignificant compared the pressure gradients on the leading edge. The leading edge is a location of normal and almost normal shock waves, and at high Mach numbers, is a zone of intense temperature. The grid points surrounding the surface are shown in the visualization to gain an appreciation for how small the subsonic/supersonic region is, yet how intense the pressure differential is there. The leading edge itself is only 0.5 inches in radius.

Figures 11 and 12 show the results for Mach 10 analyses. Lift coefficients are nearly exactly the same as the tangent-cone results, but drag coefficients that tend to be more like the tangent-wedge results. Unfortunately for a vehicle designer, a forebody that lifts like a cone and drags like a wedge is the worst of both worlds!



**Figure 11:** Lift and Drag Coefficient Results from GASP and HABP for Mach 10.0

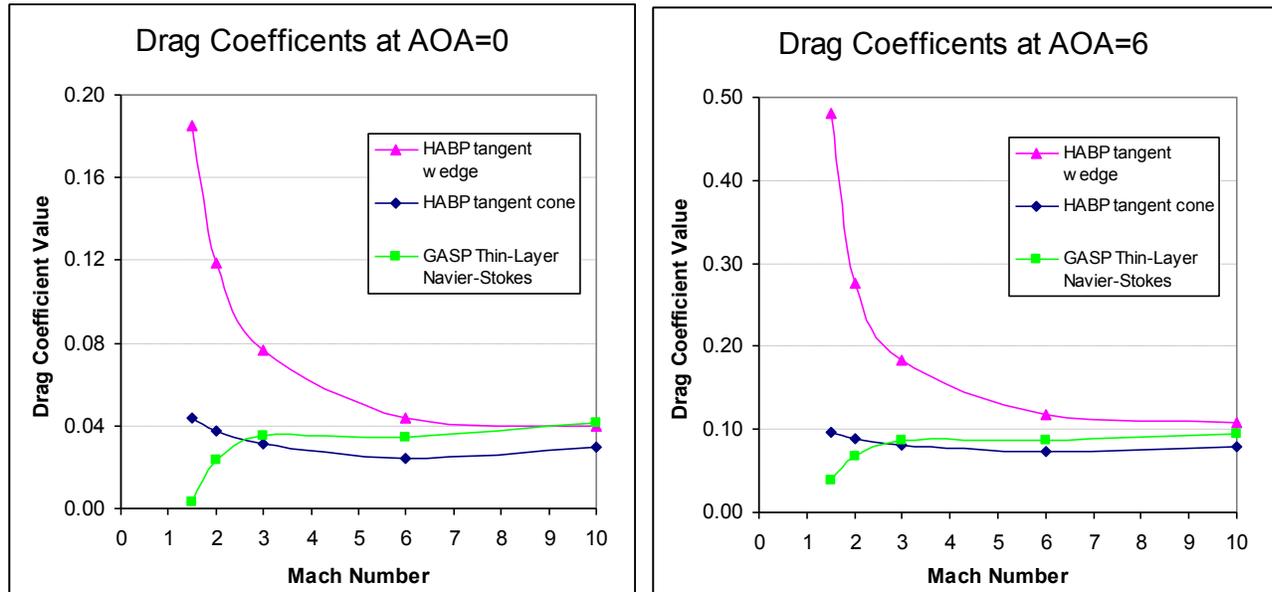
Mach 10									
AOA	Lift Coefficient			Drag Coefficient			lift/drag coefficient		
	HABP TW	HABP TC	GASP TLNS	HABP TW	HABP TC	GASP TLNS	HABP TW	HABP TC	GASP TLNS
0.0	0.0000	0.0000	0.0006	0.0398	0.0294	0.0414	0.0000	0.0000	0.0135
1.0	0.0766	0.0577		0.0417	0.0307		1.8369	1.8795	
2.0	0.1535	0.1145	0.1135	0.0472	0.0348	0.0474	3.2521	3.2902	2.3944
3.0	0.2310	0.1700		0.0565	0.0418		4.0885	4.0670	
4.0	0.3121	0.2244	0.2304	0.0696	0.0515	0.0648	4.4842	4.3573	3.5554
5.0	0.3896	0.2774		0.0866	0.0640		4.4988	4.3344	
6.0	0.4710	0.3419	0.3549	0.1074	0.0792	0.0941	4.3855	4.3169	3.7703

**Figure 12:** Lift and Drag Coefficient Data from GASP and HABP for Mach 10.0

The Mach 10 results begin to clearly show a trend that began to appear at Mach 6, namely, that the three-dimensional effects on a geometry such as the Hankey-wedge cannot be neglected.

The wedge is not wide enough to prevent the higher pressures on the lower surface from redistributing to the upper surface, reducing the lift capability of the configuration significantly.

The question posed by the Mach 1.5 results is revisited, namely, are the ultra-low drag coefficients predicted part of a trend, or an anomaly? In figure 13, the entire span of results, from Mach 1.5 to Mach 10, is plotted, with the drag coefficients at  $0^\circ$  and  $6^\circ$  AOA plotted against Mach number.



**Figure 13:** Drag Coefficient at AOA=0, 6 from GASP and HABP for all Mach numbers

The plot seems to indicate that in the GASP data, drag falls off dramatically at Mach numbers less than 2.5, and that the anomalously low drag values of the Mach 1.5 case correlate with those seen at 2.0 and 3.0. Obviously, drag does not plunge to 0 at Mach 1.4, but the results of GASP seem to indicate that they do!

Perhaps a better observation from the data is that the GASP results, at least using the techniques put forth in this paper, should be considered highly suspect at Mach numbers below 2.5.

## Conclusions

As an initial conclusion for an aerodynamics designer, it would seem that the tangent cone is superior to the tangent-wedge in accurately estimating the lift coefficients of the Stargazer forebody. This would seem to indicate that the three-dimensional effects of the wedge forebody are significant; this is probably related to the high fineness ratio (length/width) of the forebody. This is especially significant because the default solution in APAS for this configuration is tangent-wedge, and a switch to tangent-cone would have to be made by the user.

Since the analysis was done only for the forebody and not for the entire vehicle, one should hesitate to draw broad conclusions for the entire vehicle, but these analyses seem to indicate that the motivations for choosing the Hankey-wedge geometry for the forebody (i.e. significant contribution to lift) might not be borne out by the detailed CFD analysis. It seems that the actual lift of the forebody is much lower than previously expected from the tangent-wedge HABP analysis, and is actually much closer to the tangent-cone values. As mentioned previously, to lift like a cone and drag like a wedge is doubly bad.

Drag coefficients increase from an especially low value at the low Mach numbers to the higher values predicted by the tangent-wedge approximation at higher Mach numbers. For a vehicle designer, it would seem to indicate that the drag numbers generated by HABP should be downward adjusted significantly for low Mach numbers, and the tangent-wedge approximation used as an upper bound on drag for the higher Mach numbers.

This entire exercise demonstrates the utility of an accurate flowfield solution during vehicle configuration selection, since simply running APAS on the forebody would not have indicated which solution method results (tangent-wedge or tangent cone) fit more closely to the actual flowfield results.

## Can CFD be integrated into an MDO environment?

In the broader sense, can advanced CFD analyses be integrated into a multidisciplinary design environment? An answer for this question for the SSDL design environment will be pursued.

Typically the aero designer is given approximately one week to create an aerodynamic deck, in APAS, for a given geometry. After creation of the aero deck, the configuration is scaled photographically so as to preserve the aerodynamic characteristics of the configuration, and eliminate the need to redo the aero deck for each vehicle iteration. This is unfortunate from a design perspective, since it essentially means that variables such as taper ratio, wing sweep angles, and fineness ratio must be “guessed” right on the first attempt, since they will not be changed later.

An ideal scenario would allow aerodynamic variables to be altered automatically, by a global optimizer, and APAS would run in a scripted mode, automatically generating configurations, solution sets, and aerodynamic tables. This goal, so far, has eluded us because of the numerous user inputs and interfaces necessary in APAS.

But, for a moment, let us assume it is possible. Now, given that APAS has serious inaccuracies in its analyses, could we script a GASP/CFD solution to do the same thing?

Let us examine a typical APAS aero deck for the Stargazer.

- 13 Mach numbers (2 subsonic, 4 supersonic, 7 hypersonic)
- Mach 0.3, 0.6, 1.2, 1.5, 2.0, 3.0, 4.0, 6.0, 8.0, 10.0, 12.0, 15.0, 20.0
- 17 angles-of-attack per Mach number
- AOA: -30°, -25°, -20°, -15°, -10°, -5°, -3°, 0°, 3°, 5°, 10°, 15°, 20°, 25°, 30°, 35°, 40°
- 221 solution points

Assuming a simplified CFD solution time of 6 hours per configuration, this leads to a total CPU time of 1326 hours, or roughly 55 days of continuous CPU time to generate a single aero deck. Obviously, executing such a solution, much less integrating it into an optimizer, is beyond the sensible realm of feasibility.

Some of the assumptions of this data set might bear examination. For instance, if the vehicle has two planes of symmetry (or could be reasonably approximated as such), the AOA data set could be run for only positive AOA. Additionally, most trajectories fly at less than 10° AOA for the majority of their flight; this could lead to another reduction in solutions. Assume as well that CFD is only run on Mach numbers that are too high for reliable wind tunnel data; this dataset would include most hypersonic Mach numbers.

This leads us to extrapolate a reduced analysis set:

- 5 Mach numbers (1 supersonic, 4 hypersonic)
- Mach 2.0, 4.0, 6.0, 10.0, 14.0
- 4 angles-of-attack per Mach number
- AOA: 0°, 3°, 6°, 9°; assuming two planes of symmetry in the vehicle geometry
- 20 solution points

Such an analysis would take 120 hours, or 5 days of continuous CPU time. Such a case might begin to fall in the realm of feasibility for a lab environment that contained a computer dedicated to CFD analysis, and an experienced aerodynamics and CFD specialist who could create grids and input files in a reasonable amount of time. In such a scenario, a simplified geometry would be used to create a grid, which would be exported to GASP and solved.

The disk space requirements for such a system would probably be on the order of 200-400 MB per solution, leading to a total of 5-10 GB of total disk space for a single aerodynamic data set. This is a rather hefty requirement on a design lab the size of SSDL, but might be possible.

Another possibility that might make an integrated CFD aerodynamic solution possible is the aggressive pursuit of automatic unstructured grid generators and solvers. Unstructured grids sacrifice ease in grid generation for more computationally complex solutions, but seem to be ideal for a scripted capability.

A significant benefit that the CFD solutions could offer, however, is the temperature distribution over the surface of the vehicle, suitable for use by an aeroheating specialist in thermal protection design. These analyses conducted in this study assumed an adiabatic wall (no heat transfer) but GASP allows the option of a cold wall with a specified wall temperature. Heat loads could then be extracted from the solution, and input into a TPS sizer, such as TCAT, to determine insulation masses and acreages.

In the context of CFD analyses in an MDO environment, the techniques used in this paper are almost certainly not “ready for prime time”. The creation of grids and debugging of problems took an enormous amount of time and taxed the knowledge of even skilled CFD practitioners like Dr. Ruffin. Even after a huge investment of time and effort in setting up and running solutions, only a handful of Mach numbers and a few angles-of-attack were solved.

Computational Times	
Mach	hh:mm
<b>Mach 1.5</b>	
0°	17:12
2°	17:10
4°	13:11
6°	12:07
<b>Mach 2.0</b>	
0°	15:05
2°	11:21
4°	14:15
6°	10:29
<b>Mach 3.0</b>	
0°	9:14
2°	12:12
4°	11:46
6°	8:45
<b>Mach 6.0</b>	
0°	8:40
2°	7:52
4°	8:03
6°	10:00
<b>Mach 10.0</b>	
0°	11:44
2°	16:25
4°	23:17
6°	17:32
Solution on Atlas (R12000)	
Solution on Titan (R10000)	
Mixed (Atlas/Titan) solutions	
Restarted solutions	

**Figure 14:**  
Computational Time  
required for analyses

In fairness, though, nearly all of the time spent on this project was spent in debugging grids, the GASP program itself, and a myriad of other problems that cropped up. Redoing the entire project from scratch, with a complete and fully debugged grid, would probably take only 10 days. However, a new configuration would of necessity entail new difficulties and challenges, and the time spent to overcome these is usually quite significant.

Perhaps future advances in computer speed, automated grid generation, and scripted solutions can successfully shift the paradigm in favor of CFD integration in the MDO environment.

### Recommendations

Accurate and successful grid definition is key to a successful solution. Splitting up the grid generation and solution tasks was necessary to complete the project, but lack of proximity and interaction, complicated by a steep learning curve, often led to problems. If CFD was to be integrated into the SSDL working environment, having a grid generation specialist in the lab would be very important. However, the task of grid generation is so difficult and time-consuming that it might well be the only task such a specialist would have time to do.

The same version of programs should be used by all parties involved. A great deal of the problems encountered on this project were due to the use of different versions of GASP. In all fairness, no one would have ever thought that there would have been so many serious differences between GASP v3.1.6 and

GASP v3.2. Nevertheless, problems encountered, and their ultimate solutions, complicated the logistics of implementing the project severely.

CFD analyses demand the fastest CPUs and huge memory and disk space requirements. As is listed in Figure 14, the computational requirements on the computers used in this project were quite severe. Had the initial plan to solve the entire vehicle flowfield proceeded, it is doubtful that Titan and Atlas would have been up to the task in any reasonable period of time. An account was procured on the High-Performance Computing cluster, which consists of three multi-processor Origin 2000 computers, but the disk space (300 MB) and memory limitations imposed on such accounts might have made even one solution of the full vehicle impossible.

### **Summary**

An analysis of the forebody of the Stargazer vehicle was undertaken at three different Mach numbers, and four angles-of-attack, using a simple heritage code and an advanced CFD analysis. Results were compared, differences noted, and trends were elucidated. Recommendations for improvement of future analyses were made, with special note taken in regards to the possible integration of CFD analyses into the MDO environment. While not quite ready for such a step, CFD solutions offer important validation and visualization of trends that escape simpler analyses.

## References

1. Bonner, E., Clever, W., and Dunn, K., *Aerodynamic Preliminary Analysis System II, Part I - Theory*, NASA Contractor Report #182076, April 1991.
2. Bonner, E., Clever, W., and Dunn, K., *Aerodynamic Preliminary Analysis System II, Part I - Theory*, NASA Contractor Report #182076, April 1991.
3. Bonner, E., Clever, W., and Dunn, K., *Aerodynamic Preliminary Analysis System II, Part I - Theory*, NASA Contractor Report #182076, April 1991.
4. W. D. McGrory, D. C. Slack, M. P. Applebaum, and R. W. Walters. GASP Version 3.2, *The General Aerodynamic Simulation Program*. AeroSoft, Inc., Blacksburg, Virginia, 1997.

## Appendix A

### Detailed Description of Analysis Procedure using GASP

---

#### Analyzing the Grid

Before beginning the desired analysis, it is necessary to have a correct understanding of the grid on which the analysis is based; namely, a correct understanding of the grid geometry, its zones, zonal boundaries, and its orientation to the freestream flow.

For this project, a code developed by the Ames Research Center was used called FAST (Flow Analysis Software Toolkit). FAST is an excellent post-processing and visualization tool, but in this case, it is used to examine the grid structure before beginning.

The FAST outputs used to determine zonal boundaries and grid geometries are included in appendix D.

#### Creating the Input file using FRONT

GASP uses an ASCII input deck to define flow parameters, zonal boundaries, solution procedures, and so forth for each run. Writing the ASCII input deck manually is possible, but very tedious. Fortunately, GASP comes with an excellent GUI (graphical user interface) that will take the user's parameters and automatically write the input deck. The GUI is called FRONT and is executed by typing

*front* (in the GASP/bin directory)

FRONT brings up a main window, which in turn is divided into four primary subsections. This tutorial proceeds step by step through each one.

#### Loading the Grid into GASP

The logical first step within FRONT is to load the grid into memory. For each input deck, FRONT will write out an input file, a grid file, and a solution file. For instance, for an input file saved as *forebody.inp*, FRONT will write the following files:

*forebody.grd* (the grid file)  
*forebody.inp* (the ASCII input file)  
*forebody.sln* (the solution file)

The size of the input file is typically quite small, but the grid and solution files can be very large depending on the number of points in the grid. As a matter of reference, the Stargazer forebody

grid for Mach 6 that is analyzed in this project contains 1.53 million points in four different zones; its grid file is 38.9 MB and the completed solution file is 74.0 MB.

In order to begin the definition of the input deck, first load the appropriate grid by clicking in the “Zones” radio button in the lower portion of the main FRONT window. Click on “Import Grid Files” under the File menu. This will bring up the Import Grid Files dialog box. Click on “Open Grid File” and another dialog box will appear. Specify the appropriate location of the grid file, and its format. For this analysis, three different grid files were used, with the dimensions given in the i, j, and k directions:

<i>grid zones</i>	<i>Mach 1.5 grid</i>	<i>Mach 6 grid</i>	<i>Mach 10 grid</i>
zone 1: upper nose leading edge	45 x 89 x 73	45 x 89 x 73	45 x 89 x 73
zone 2: upper forebody	73 x 89 x 73	73 x 89 x 73	73 x 89 x 73
zone 3: lower leading edge	45 x 89 x 73	45 x 89 x 73	45 x 89 x 73
zone 4: lower forebody	73 x 89 x 73	73 x 89 x 73	73 x 89 x 73
<b><i>total points</i></b>	<b><i>1,533,292</i></b>	<b><i>1,533,292</i></b>	<b><i>1,533,292</i></b>

Each of these grid files are Plot 3D, binary, single-precision files written from GRIDGEN version 13.

Each of the zones must be loaded individually. Click on zone 1 in the “Opened Grid Files for Export” region of the window and click on zone 1 in the “Current Zone List Including Imported Grid Files” region of the window. The radio button reading “Replace Current Zones with Imported Zones” should become active. Select it. Zone 1 in the latter region should list the new grid zone now.

Repeat the procedure for the other zones, except choose the radio button reading “Add Selected Zones as New Zones” until all four of the grid zones have been transferred. Click on Done.

Now the four zones are listed as “Available Zones”. Before continuing onto sequencing the grids and boundary condition definition, the zonal boundaries should be specified. Select “Close Zone Window” under the Window menu.

### **Defining the Zonal Boundaries**

One of the difficulties of working with a multi-zone grid is the necessity to define correct zonal boundaries. It is easy to make mistakes while doing it, and mapping out the correct boundaries can be quite tedious for a grid with many zones. It typically requires some type of viewer to visualize the grid geometry, and for that reason, it is a task that is best done in concert with the definition of the boundary conditions.

Appendix D is a graphical representation of the maximum and minimum planes in each of the flow directions and can be used to define a series of zonal boundaries. The results of that definition are listed below:

Zonal Boundary 1:	Zone 1 I0 plane to Zone 3 Idim plane	(for all values of j and k)
Zonal Boundary 2:	Zone 1 Idim plane to Zone 2 I0 plane	(for all values of j and k)
Zonal Boundary 3:	Zone 3 I0 plane to Zone 4 I0 plane	(for all values of j and k=1,72 mapping into k=72,1)
Zonal Boundary 4:	Zone 2 J0 plane to Zone 4 J0 plane	(for all values of i and k=1,72 mapping into k=72,1)
Zonal Boundary 5:	Zone 1 J0 plane to Zone 1 J0 plane	(for all values of k and i=1,44 mapping into i=44,1)
Zonal Boundary 6:	Zone 3 J0 plane to Zone 3 J0 plane	(for all values of k and i=1,44 mapping into i=44,1)

Click on the “Zonal Boundaries” radio button in the lower region of the FRONT main window. Select “New Zonal Boundary” under the Edit menu, and define the first zonal boundary by typing 1 and 3 as the zone numbers, and “ $I = i_0$  and “ $I = idim$ ”, respectively, for the zonal boundary faces. Since this boundary is in effect for all values of j and k, no modification is necessary to the directions, and the variable mapping should read “1 2 3 4 5” which indicates that all five flow variables are being passed between the zones.

This zonal boundary represents the boundary between the upper and lower leading edge zones, both of which will use global iteration. Hence, under “Pass Zonal Boundary Information”, “To First and Second Zones” should be selected. If the zonal boundary was a boundary between a global and marching zone, one could safely choose, “To Second Zone Only”, for this selection.

The grid has been defined so that points in one zone correspond directly to points in the next zone. Hence, “Point-to-Point Connectivity” can safely be selected under “Zonal Boundary Type”. If the two grid faces had an arbitrary distribution of points across their faces, “Arbitrary Grid Patching” can be used, and GASP will interpolate flow conditions at the location of the points in the second boundary face and translate them over.

Repeat the process until all six zonal boundaries have been defined. The final two zonal boundaries represent a singular axis decomposition (to be discussed further under “Boundary Conditions”) and the indices on the i-axis must be swapped. Also note that since zone 4 has a k-axis orientation opposite to those of the other zones, any zones that map into zone 4 will have to switch the mapping of k-indices to insure the correct flow of information. This oversight resulted in having to redo an entire series of runs.

Close the Zonal Boundaries window.

### Defining the Boundary Conditions

Next, open up the “Zones” window again and select the “BC’s” radio button for boundary condition definition for each zone. A window comes up with selection boxes for I0, Idim, J0, Jdim, K0, and Kdim. From Appendix D, the proper definition of some faces are obvious. Some are a little more subtle.

Since the solution is a viscous solution, the flow next to the surface of the body should not slip; and the assumption is made that there is no heat transfer to the body. Hence, “No Slip, Adiabatic” should be chosen for the K0 faces of zones 1, 2, and 3 and the Kdim face of zone 4. If the flow had a component of heat transfer, “No Slip,  $T = T_{wall}$ ” would be chosen and FRONT

would prompt for a wall temperature. If the flow were inviscid, “Tangency” would be chosen for these zone faces.

For each face that has a zonal boundary definition (except for the final two that have singular axis decomposition), “Zonal Boundary” should be chosen as the boundary condition. This will prompt GASP to review the zonal boundary definitions for each of these faces. Choose this for the I0 and Idim faces of zone 1; the I0 and J0 faces of zone 2; the I0 and Idim faces of zone 3; and the I0 and J0 faces of zone 4.

The Jdim faces of each zone represent a plane of symmetry (since there is no sideslip in the flow). Select “x-z Plane” for each of these faces.

The Idim faces of zones 2 and 4 represent the outflow of the solution. Choose “1<sup>st</sup> Order from Interior” for each of these faces.

The Kdim faces of zones 1, 2 and 3 and the K0 face of zone 4 are outside of the disturbed flow regions of the grid. Choose “Freestream” for these boundaries.

Finally, the J0 faces of zones 1 and 3 decompose to a line. Choose “x-y Plane, Singular Axis” to decompose the flow properly. This boundary condition will reference the zonal boundaries previously specified to correctly define the flow.

To summarize:

<b>Boundary Condition</b>	<b>zone 1 upper leading edge</b>	<b>zone 2 upper forebody</b>	<b>zone 3 lower leading edge</b>	<b>zone 4 lower forebody</b>
I0	Zonal Boundary (with zone 3 Idim plane)	Zonal Boundary (with zone 1 Idim plane)	Zonal Boundary (with zone 4 I0 plane)	Zonal Boundary (with zone 3 I0 plane)
Idim	Zonal Boundary (with zone 2 I0 plane)	1 <sup>st</sup> Order from Interior	Zonal Boundary (with zone 1 I0 plane)	1 <sup>st</sup> Order from Interior
J0	x-y Plane, Singular Axis	Zonal Boundary (with zone 4 J0 plane)	x-y Plane, Singular Axis	Zonal Boundary (with zone 2 J0 plane)
Jdim	x-z Plane	x-z Plane	x-z Plane	x-z Plane
K0	No Slip, Adiabatic	No Slip, Adiabatic	No Slip, Adiabatic	Freestream
Kdim	Freestream	Freestream	Freestream	No Slip, Adiabatic

Carefully review the Boundary Conditions window for any mistakes and then close the window.

## Sequencing the Grids

Grid sequencing is used to speed the solution of an iterative procedure. By solving first on a grid with coarse resolution, then medium, and ultimately to fine resolution, flow patterns can be defined on a broad scale and then be applied to a fine scale.

It should be noted that in order to properly implement grid sequencing, the grid dimensions must be appropriate. The dimensions used in each zone were specifically altered to be  $(4x)+1$  so that two levels of sequencing were possible.

Grid sequencing is not terribly effective on marching solutions, but for global iterations it is very effective. Hence, it should be used liberally in the mixed subsonic/supersonic global iteration region of the nose leading edge. Highlight “Zone #1” then click on the radio button “New Sequence”. Click on the i, j, and k-increments until each increment is 2. Again click “New Sequence” and increase the increment to 4. Repeat for zone 3. After sequencing, there should be 3 sequences apiece for both zones 1 and 3. Each should have a fine grid (increment 1), and medium grid (increment 2), and a coarse grid (increment 4). Close the Zones window.

### **Defining the Physical Models**

Access the Physical Models window by clicking on “Physical Model Windows” in the upper menu or by hitting the “Physical Models” radio button in the bottom of the main window.

Creating a physical model for GASP means setting a series of parameters, such as the properties of the freestream gases (i.e. density, temperature, pressure) as well as the flow direction. It also involves setting up the solution procedure to be used for inviscid and viscous fluxes. Often different solution techniques and spatial accuracies will be desired for use in the inviscid flow solver, which means that a physical model for each solution technique must be created. For this analysis, three different physical models will be created:

***Physical Model #1: nose global, first order***

a global iteration scheme, using Van Leer flux at first-order accuracy, for use in the mixed subsonic/supersonic region of the nose leading edge.

***Physical Model #2: nose global, third order***

a global iteration scheme, using Van Leer flux at third-order upwind biased accuracy, with the min-mod limiter, also for use in the nose leading edge region.

***Physical Model #3: forebody marching, third order***

a marching scheme, using Van Leer flux at third-order upwind biased accuracy, with the minmod limiter, for the fully supersonic flow on the forebody.

Since the initial and freestream conditions are the same in each case, as well as the viscous and thermochemistry conditions, one physical model will be created and copied twice to create the other two physical models.

### **Physical Models: Freestream Conditions**

In the Freestream Conditions window, parameters are set for the freestream conditions of the flow. Because the assumption is made that the flow is a perfect gas, it is only necessary to input two state variables for each case to determine the others. The inputs derive from an analysis of

the Stargazer trajectory, which gives an altitude corresponding to each Mach number. Freestream temperature and density are derived from the altitude from the US Standard Atmosphere:

	<i>Mach 1.5</i>	<i>Mach 6</i>	<i>Mach 10</i>
Altitude	40,000 ft (12192 m)	75,000 ft (22860 m)	170,000 ft (51816 m)
Pressure	18,823 Pa	3542.6 Pa	63.656 Pa
Temperature	217 K	219 K	271 K
Density	0.3027 kg/m <sup>3</sup>	0.0562 kg/m <sup>3</sup>	0.0008 kg/m <sup>3</sup>

Angle of attack is set by setting the normalized freestream flow direction cosines ( $u/|V|$ ,  $v/|V|$ ,  $w/|V|$ ). This also depends on the orientation of the grid to the upstream flow. From an examination of the grid, it is determined that angle-of-attack ( $\alpha$ ) is modified through the u and w directions. Conversely, sideslip angle ( $\beta$ ) would be modified through the u and v directions for this grid. For this analysis, however, sideslip angles are not considered.

<i>angle-of-attack (<math>\alpha</math>)</i>	<i><math>u/ V </math></i>	<i><math>v/ V </math></i>	<i><math>w/ V </math></i>
0°	$\cos(0^\circ) = 1.000000$	0.0	$\sin(0^\circ) = 0.000000$
2°	$\cos(2^\circ) = 0.999391$	0.0	$\sin(2^\circ) = 0.034900$
4°	$\cos(4^\circ) = 0.997564$	0.0	$\sin(4^\circ) = 0.069756$
6°	$\cos(6^\circ) = 0.994522$	0.0	$\sin(6^\circ) = 0.104528$

Once appropriate flow properties and direction cosines are entered, close the window by selecting “Close Freestream Window” under the Window menu.

### Physical Models: Initial Conditions

Since the initial conditions are the same as the freestream conditions, use the radio button “Copy from Freestream” to automatically update the initial conditions with the freestream conditions. Close the window by selecting “Close Initial Conditions Window” under the Window menu.

### Physical Models: Inviscid Flux

Open the Inviscid Flux Window and select “Global Iteration”. In each direction, set the flux scheme as “Van Leer Flux” and the spatial accuracy as “1<sup>st</sup> Order”. These properties will be modified in the other physical models that will be created from this one. Close the window.

### Physical Models: Viscous Flux

Open the Viscous Flux window and select “Laminar Flow”. This flow is appropriate for this analysis since the Stargazer geometry is slender and smooth, and more importantly, because it greatly simplifies the analysis. De-select the thin-layer terms in the i and j directions so that only thin-layer terms in the k-direction are active. Insure that no cross-derivatives are selected, and change the wall-gradient calculation to 1<sup>st</sup> order accurate.

## Physical Models: Thermo-Chemistry

Open the Thermo-Chemistry window and check to insure that the chemistry model is “Perfect Gas”. Close the Thermo-Chemistry window. Often it is necessary to specify the location of the gas model file, *db1.bin*, for GASP in the main FRONT window.

## Duplicating the Physical Models

Now, using the “Copy Physical Model” and “Paste Physical Model” command under the Edit menu make two copies of the physical model so that there are three physical models defined. Type a description for each in the “Current Physical Model Description” box and hit <enter>.

Physical Model #1: nose global first-order

Physical Model #2: nose global third-order

Physical Model #3: forebody marching third-order

Highlight Physical Model #2 and open the Inviscid Flux window. Set the spatial accuracy as “3<sup>rd</sup> Order Upwind Biased” and choose “Min-Mod” as the flux limiter in each flow direction. Close the Inviscid Flow window.

Highlight Physical Model #3 and open the Inviscid Flux window. Change the global/marching strategy to “March in I-Dir”. The flux scheme in the i-direction will automatically change to Full Flux. In the i-direction, set the spatial accuracy as “2<sup>nd</sup> Order Fully Upwind” and choose “Catastrophic (P and Rho)” as the flux limiter. In the j and k directions, choose Van Leer flux, with 3<sup>rd</sup> order upwind-biased spatial accuracy and the min-mod flux limiter. Close the Inviscid Flow window.

This concludes the definition of the physical models. Close the window.

## Defining the Blocks and Sweeps

Blocks are analogous to stages of the solution. This analysis proceeds from a global iteration of the nose leading edge to a marching solution along the forebody. Within that global iteration, the analysis moves from a first-order solution on the coarse grid, to a first-order solution on the medium grid, to a third-order solution on the medium grid, and finally to a third-order solution on the fine grid. Each of the physical models previously defined will be used to distinguish these blocks from one another.

Sweeps can be thought of as the solution technique within each block. Sweeps are set up for each zone in a block, and specify in which order the zonal grid is to be solved, the maximum and minimum CFL numbers used, the appropriate physical model and sequence to be used, and several other parameters relating to convergence.

To save time and effort, a similar procedure to the definition of the physical models will be used in the definition of the blocks. A generic block (with its attendant sweeps) will be set up and then duplicated and modified several times to create the other blocks. The blocks to be defined are:

- Block #1: nose global, first order, coarse grid
- Block #2: nose global, first order, medium grid
- Block #3: nose global, third order, medium grid
- Block #4: nose global, third order, fine grid
- Block #5: forebody marching, third order, fine grid

Create the first block. Define the name as “nose global, first order, coarse grid”. Set the convergence criteria to 0.0001, since the coarse grid is a computationally inexpensive way to converge quickly. Set the write output increment to 100, and the maximum iterations to 5000.

Now click on the Sweeps button to open the Sweeps window. A sweep must be created for each zone. Define sweep 1 as “upper leading edge”. Set the zone number to 1, physical model to 1, and the sequence to 3. This tells GASP to solve zone 1 using the first-order solutions on the coarse grid. Change the CFL min and max to 1 and 2, respectively. Change the sweep plane to *i* and the direction to “forward”. Change the solution procedure to Jacobi. Copy and paste sweep #1 to create sweep #2. Define sweep 2 as “lower leading edge”. Change the zone number to 3, but leave all other parameters the same. Now there are two sweeps defined for block 1.

In the Block window, copy Block #1 and paste it to create block #2. Define the name as “nose global, first order, medium grid”. For block 2, which is the first-order solution on the medium grid, change the convergence criteria to 0.01, the max iterations to 1000, and the write output increment to 20. In the Sweeps window, change the block number to 2 to view the sweeps for Block #2. Note that when blocks are copied and pasted, their sweeps are copied too. Since most of the parameters of the sweeps of block #2 are very similar to block #1, this saves the user from repetitive effort. Set the sequence number to 2 (the medium grid) for both sweeps.

Copy and paste block #2 to create block #3. Define the name as “nose global, third order, medium grid”. Leave the convergence criteria, max iterations, and write output increment the same. In the Sweeps window, change to physical model in each sweep to 2 (the third-order accurate solution).

Copy and paste block #3 to create block #4. Define the name as “nose global, third order, fine grid”. Leave the convergence criteria and max iterations the same, but change the write output increment to 5, to account for the fact that each iteration is much more computationally expensive, and in the event the iteration is terminated, fewer iterations would be lost. Within the Sweeps window, change the sequence number to 1 (the fine grid) in each sweep.

Note that the global solution to the leading edge converges four orders of magnitude on the coarse grid with the first-order solution; two orders of magnitude for each of the medium grid solutions, and finally two orders of magnitude further on the fine grid. The overall effect is to

converge the fine grid roughly three orders of magnitude, but by staggering the solution through sequencing, the solution will converge far, far more quickly.

Copy and paste block #4 to create block #5. Define the name as “forebody marching, third order, fine grid”. Block 5 is the marching solution over the forebody, and has significantly different parameters. Change the convergence criteria to 0.001, and the write output increment to 50. Marching solutions iterate on each plane until convergence, and iteration typically proceeds extremely quickly. Within the Sweeps window, change sweep #1 from zone 1 to zone 2, and sweep #2 from zone 3 to zone 4. In each sweep, change the physical model to 3 (the marching solution).

For the Mach 10 solution, it was necessary to use a sequenced global iteration solution over the forebody as well. For this case, blocks 5-8 were created with the same parameters as blocks 1-4, except that they were solutions for zones 2 and 4 instead of zones 1 and 3.

Carefully review all the parameters for the sweeps and blocks. It is very easy to make a mistake or an oversight; often this can result in many lost hours of CPU time. If all parameters are accurately set, close the Sweeps and Blocks windows.

### Finalizing the Input Deck

With physical models, zones, zonal boundaries, blocks and sweeps defined, only a few more steps are needed to complete the input deck. Set the reference quantities according the Mach number of the solution. Notice that the reference length is set to 0.3048; since the grid is defined in feet but the solution is in SI, it is necessary to tell GASP that there are 0.3048 model units (feet) in each reference unit (meters).

	<i>Mach 1.5</i>	<i>Mach 6</i>	<i>Mach 10</i>
Rho (kg/m <sup>3</sup> )	0.3027	0.0562	0.0008
Temperature (K)	217	219	271
P gauge (N/m <sup>2</sup> )	0	0	0
U (m/s)	442.844	1779.52	3299.24
Length (m)	0.3048	0.3048	0.3048

On the main FRONT window, “Units” should be set to “Metric”; “Memory Mode” should be set to “Medium”; “Residual Output” should be set to “Block”; and “CPU Output” should be set to “hh:mm:ss”.

Neither of the boxes in “Initialization/Restart” should be highlighted, but should it become necessary to restart the solution at some intermediate point, “Restart Calculation” should be highlighted.

Save the input deck and exit FRONT.

## Beginning and Monitoring the Analysis

With a complete ASCII input file, an appropriate grid file, and a solution file, GASP is ready to start. Begin the solution by typing:

***gasp -i forebody.inp -o forebody.out***

(note: the ***gasp*** binary file must be aliased to the correct directory for this command to work.)

Since GASP runs often take many hours, the shell used to start the GASP run will often time out or be closed by the user. On some systems, this sends a stop command to any processes created in that shell. This problem can be avoided by typing:

***nohup gasp -i forebody.inp -o forebody.out***

The ***nohup*** command tells UNIX not to “hangup”, or send the kill command, if the shell times out or closes down. It also redirects screen output to a file called ***nohup.out***.

Another useful command during long GASP runs is the ***nice*** command, which allows other jobs to use the CPU (at a significantly higher percentage than normal) during processing. During the GASP runs used in this analysis, the ***nice*** and ***nohup*** commands were combined in the following sequence:

***nohup nice -n 20 gasp -i forebody.inp -o forebody.out&***

The ampersand (&) sends the job to the background of the shell.

## Post-processing using the PRINT utility

Once the solution is complete, it is necessary to process it further to create the Plot3D files that can be read and visualized in FAST, as well as to integrate the quantities that will determine lift and drag coefficients over the body. Fortunately, GASP comes with a utility specifically designed to process the results into forms usable by other programs.

Start the PRINT utility by typing:

***print***

Creating an input file with PRINT is similar to creating an input file with FRONT, except instead of defining physical models and solution procedures, PRINT input files define analysis sets and output formats.

In the PRINT main window, it is necessary to define reference lengths and reference areas that PRINT will use to non-dimensionalize forces and moments. Since this analysis is concerned

only with lift and drag coefficients, which in turn are only based on axial and normal coefficients, reference area will be the only quantity that must be defined. If moments and moment coefficients were desired, it would be necessary to define reference lengths and reference locations.

The reference area that will be used for non-dimensionalization is based on the area of the base of the Stargazer forebody. From the configuration, this value is 262.9 ft<sup>2</sup>. Since the GASP analysis only dealt with half the forebody (since it is symmetric about its x-z plane), enter the reference length as 131.5. Note that PRINT assumes that the reference area is in m<sup>2</sup>. It is not necessary to convert 131.5 ft<sup>2</sup> to its equivalent in m<sup>2</sup>; PRINT wants the value in model units, and the grid is defined in feet.

In the box called "GASP Input File" type the name of the input file used to create the solution, such as *forebody2.inp*, and its extension if necessary. Change the "Max Work Space" to 20,000,000 words to insure the PRINT has sufficient memory allocation to create the Plot3D files.

### Creating the Data Sets

Open the Data Sets window by selecting the radio button called "Data Sets". Select "New" under the Option menu. Following the pattern used when creating physical models and blocks, it simplifies things to create a series of data sets for a particular zone and then copy it to the other zones.

Choose New under the file menu to define a solution set. For the creation of Plot3D files, choose cell nodes, SI units, and Fortran Binary as the set parameters.

Create another solution set and define it using integrated quantities, non-dimensional by freestream, and k face as the parameters. Set the i and j slider bars for all i and j, and set the k slider to 1 for zones 1-3, and to kdim for zone 4.

Highlight both solution sets, and choose Copy from the File menu. Change the zone numbers to 2, 3 and 4 in succession, and Paste the zone sets for each.

Close the Data Sets window.

### Printing the Output

Open the Print Output window by clicking on the radio button of the same name. Highlight the eight data sets that have been created, and move them to "Active Data Sets". Click on the PRINT button in the upper left corner of the screen to start the creation of the output files. In the UNIX shell, the output parameters will be displayed for each zone. At the conclusion, close the Print Output window and exit from the Print program.

## Visualization using FAST (Flow Analysis Software Toolkit)

For visualization purposes, the only files that will be necessary will be the Plot3D files created by PRINT, *grid.p3du* and *q.p3du*. Open FAST and the Viewer, Hub, and File\_IO modules should be displayed.

### Loading grid and solution files

Under the File\_IO module, highlight the grid.p3du file. Select “Grid” under solution type, “Multi-Zoned”, “Unformatted”, and Structured. Select “Read File” to begin reading in the grid. In the UNIX shell, the dimensions of each zone should be displayed as the zones are loaded into memory.

Next, highlight the q.p3du file. In CFD calculations, the **q** vector represents density, density times momentum in each flow direction, and energy density for each grid point. Again select the same properties as before, except this time select “Solution” instead of “Grid”. Select “Read file”; again in the UNIX shell a maximum and minimum of each flow property for each zone will be displayed.

Close the File\_IO module by selecting “Quit” under the File menu.

### Calculating flow properties

Next, select “Calculator” under the Modules menu of the Hub module. The Calculator module will open, and the grid and solution zones should already be highlighted in the appropriate fashion. Select the property that is desired for visualization. If the property is a scalar quantity like pressure, Mach number, or temperature, it will be found under Scalar Quantities. After it is highlighted, select “Calc CFD Scalar” and the UNIX shell will display maximum and minimum values of that quantity for each zone.

If, however, the quantity selected is a vector value, such as velocity, repeat a similar process but select “Calc CFD Vector” and

Close the Calculator module by selecting Exit under the File menu.

### Creating surfaces

Finally, open the Surfer module by selecting “Surfer” under the “Modules” menu of the Viewer. Inside the Surfer module, surfaces can be created and viewed according to their locations in the i, j, or k planes.

### Exporting flow images

Once a view has been created that is satisfactory, select the File menu in the Viewer module. Highlight Save-Restore and then “Dump Screen”. Another window will open that will prompt for a file name and directory location. Once this is specified, choose “Write” to copy the

contents of the Visualization window to a .rgb file. The UNIX program XView can then be used to convert RGB images to GIF or JPG format.

### Final processing of lift and drag coefficients

The integrated values for forces in the x and z-directions provided by the PRINT utility are the basis for the calculation of the final lift and drag coefficients. First, however, the individual zonal values must be combined into axial and normal forces for the entire forebody. This is accomplished by adding the values of  $C_x$  and  $C_z$  from each zone; note that zone 4 had its forces calculated on the kdim plane. This will require using the negative of each of the values from this zone in the calculations.

After compiling total values for axial ( $C_x$ ) and normal ( $C_z$ ) force, the axial force must be subtracted by a reduction coefficient. This coefficient accounts for the base region of the forebody wedge, and is calculated by the following expression:

$$C_{reduction} = \frac{p_{\infty}(A_{ref})}{\frac{1}{2}\rho_{\infty}V_{\infty}^2(A_{ref})}$$

	<i>Mach 1.5</i>	<i>Mach 3</i>	<i>Mach 6</i>	<i>Mach 10</i>
Freestream Pressure (N/m <sup>2</sup> )	18845.3	11597	3531.11	62.2
Freestream Density (kg/m <sup>3</sup> )	0.3027	0.1865	0.0562	0.0008
Freestream Velocity (m/s)	442.844	885.687	1779.52	3299.24
Dynamic Pressure (N/m <sup>2</sup> )	29680	73149	88980	4354
Reduction Coefficient	0.63494	0.15854	0.03968	0.01428

After this reduction, the final and accurate values of axial and normal force remain. Let axial force be designated by  $C_a$  and normal force be designated by  $C_n$ . Then lift ( $C_l$ ) and drag ( $C_d$ ) coefficients can be calculated according to the formulae:

$$C_l = C_n \cos \alpha - C_a \sin \alpha$$

$$C_d = C_n \sin \alpha + C_a \cos \alpha$$

where  $\alpha$  represents angle of attack.

## Appendix B

### Detailed Description of Analysis Procedure using APAS

---

#### Starting APAS

APAS requires a component file to be present before it can be executed for the first time. Hence, it is useful to create a new directory and copy the *apas.comp* file into it. APAS can then be started by typing the command *apas*.

The screen will fill with garbage. Hit <enter> several times until APAS prompts for a terminal type. If *Versaterm Pro* is being used as the terminal emulator, choose selection 3. Otherwise choose selection 0.

Using *Versaterm Pro* is highly desirable for APAS use, since a standard UNIX *xterm* window will not allow the user to utilize the mouse or view graphics. Switching to a Tek window will allow graphics to be seen, but no use of the mouse, which is necessary for editing the geometry of the wing and fuselage.

Next, APAS asks for the location of geometry and solution files. Simply hit <enter> to choose the defaults for these options. Then it will ask for baud rate. Since no file transfer will be taking place across modems, the value chosen for the baud rate is unimportant; choose 1920 baud.

Continue to hit <enter> until this appears on the screen:

```
** open file 4 for plotting & run data direct access file new **
** open file 8 for scratch analysis data sequential file scr **
** open file 10 for scratch wave drag data sequential file scr **
** open file 11 for scratch analysis data sequential file scr **
*** this directory does not contain the required ***
*** wave drag matrix file named (lord49) ***
*** one is required for wave drag to be done ***
*** one will be generated and written out. ***
*** created new wave drag matrix file (lord49) ***
*** to check numbers enter n0 of rows or zero for no ***

**ok**
```

Now APAS is ready for input.

APAS stores its geometries in the permanent, local, and component files. In this case, geometry will be created in the permanent file, and then exported to the local file for analysis. All analysis is based on the contents of the local file.

The very first thing that should be done in APAS is to set the correct units for the geometry that is to be defined. Type *attr* (for attributes) and select option 3. There will be an option to choose either meters, inches, or feet as the basic unit. *Stargazer's* geometry is given in feet, so this is selected. If this is not done before any geometry is defined it can interfere with a proper solution.

Again type *attr* and choose option 1. Select the permanent file as the default file.

### Entering the component geometry

Component geometries are defined using the *term* command. The fuselage is defined using the “full ellipse” option, whereas the wing is defined using the “full surface” option. “Half ellipse” is used for engine nacelles and such geometries, and “half surface” is used for tails, winglets and other such control surfaces.

Type the *term* command. APAS will prompt for yaw, pitch, and roll of the component. These are used if an item is defined at a certain angle to the freestream flow, such as a engine nacelle. Hit <enter> to select the default of 0, 0, 0.

Next select “full ellipse” – option 3. APAS will ask for a component number. Fuselage geometries must be numbered between 1 – 100. Type *10* and for a component name type *fuselage*.

Next APAS will ask for the position of the component. This is especially useful for positioning engine components and nacelles, but for the fuselage hit <enter> to select the default of 0, 0, 0.

Next APAS asks for the number of cross sections. This is essentially how many “slices” APAS creates through the vehicle. Later on these cross-sections can be edited, repositioned, and even more sections can be added, so it is not essential to have all the exact positions and data on these sections. Type *4*.

APAS then wants the location along the x-axis, the cross-sectional area, and the elliptical eccentricity of each section. Since *Stargazer*’s cross-sections are neither circles nor ellipses, it will be necessary to edit them later, so the values entered here are not terribly important. We simply need to create the component.

Type:

<i>0.0, 0.0, 1</i>	(nose tip – zero cross-sectional area)
<i>60.1, 100.0, 1</i>	(start of midbody – arbitrary area)
<i>87.7, 100.0, 1</i>	(end of midbody – arbitrary area)
<i>102.8, 0.0, 1</i>	(aft tip – zero cross-sectional area)

At the conclusion of this data entry, APAS will list the maximum height above and below the x-axis. The value is unimportant, but the two values should be equal. Then APAS will prompt for the number of points per cross-section. This is essentially a choice of the designer, with more points giving better accuracy but taking more time to solve. This number cannot be altered later, so the choice must be accurate. In this case, choose *11*.

Strangely enough, APAS does not immediately retain a created component until the user chooses to do. Type *cata*, for catalog. Since the default file location is the permanent file, component #10, named *fuselage*, will be saved in the permanent file.

The components of the permanent file can be examined by typing *file p*, or *fi p* for short.

index	compn	name	type	symm
1	10.00	fuselage	1	2

The local file can be examined by typing *fi l*, or the component file by typing *fi c*.

The next geometry component to be created is the wing. Type *term* and select “full surface”. Surface components must be numbered between 200 – 299. Type *210* for the component number and *wing* for the component name.

Next APAS asks for the location of the tip of the root chord of the wing. From configuration:

x0	30.0 ft	
y0	0.0 ft	(symmetric wing)
z0	0.0 ft	(midbody wing)

APAS uses four geometric properties to determine the wing shape, and also prompts for the wing dihedral angle. Using the solution from the Stargazer configuration, the inputs are:

reference area (s)	610.9 ft <sup>2</sup>
aspect ratio (ar)	1.78
taper ratio (taper)	0.3
leading edge sweep	65°
dihedral	0°

APAS then prompts for an airfoil type. Stargazer’s airfoils are biconvex, with a chord to thickness ratio of 0.04. When APAS prompts for (t/c), it is asking for chord-to-thickness ratio. Type *0.04, 0.04*.

At the conclusion of wing definition, type *cata* to save the component in the permanent file.

If hypersonic analysis through HABP is all that is desired, component definition is complete. However, if a subsonic/supersonic analysis is desired through UDP, then there are still two more components to be created, the slender body model and the interference shell. Creation of those components should be deferred until the wing and fuselage have been edited into their final form.

### Editing the geometry

Editing of components in APAS is done using the *edit* command and the component number. Type *edit 10* to enter the edit environment and alter the fuselage geometry.

Since the Stargazer fuselage cross-sections are neither circles nor ellipses, but a stretched circle, it will be necessary to manually input the locations of each cross-sectional point.

In this simplified example, there are only 4 cross-sections to the Stargazer vehicle, and two of these (the front and the back) have zero-cross sectional area, but still have width.

We bring up the second cross-section by typing **2,l**. The cross section will be displayed graphically on the screen. Enter the following commands to redefine the shape of the cross-section:

```

2,y=0.00,1<enter>      2,z=5.54,1<enter>
2,y=7.59,2<enter>      2,z=5.54,2<enter>
2,y=9.71,3<enter>      2,z=5.11,3<enter>
2,y=11.51,4<enter>     2,z=3.91,4<enter>
s<enter>               (save command)
2,l<enter>             (refresh the window)
2,y=12.71,5<enter>     2,z=2.12,5<enter>
2,y=13.13,6<enter>     2,z=0.00,6<enter>
2,y=12.71,7<enter>     2,z=-2.12,7<enter>
2,y=11.51,8<enter>     2,z=-3.91,8<enter>
s<enter>               (save command)
2,l<enter>             (refresh the window)
2,y=9.71,9<enter>      2,z=-5.11,9<enter>
2,y=7.59,10<enter>     2,z=-5.54,10<enter>
2,y=0.00,11<enter>     2,z=-5.54,11<enter>
s<enter>               (save command)
2,l<enter>             (refresh the window)

```

In these instructions, 2 specifies the cross-section to be modified, the y or z-component value desired, and the point whose y or z value is to be changed. It is very important to specify the point number, since if it is not included, APAS will assume that the user wishes to change all points to that y or z value. However, this can be advantageous, especially if the user wishes to modify the x-location of the cross section. Assume it was desired to place cross-section 2 at an x-location of 62 feet. The command would be:

**2,x=62.0**

It is easy to make a mistake in APAS, or to key a value in incorrectly. If this occurs, simply finish specifying the value, even if it is incorrect, and then reenter the point with the correct value. Type the command **s**, for save, often. It is very easy to make mistakes or to crash the program, but saving the edited changes often will help minimize the losses.

Since section 3 (the aft end of the midbody) has the same cross-sectional dimensions as section 2 (the front end of the midbody), type **3,l** and repeat the commands to define the cross-section point locations. Be sure to substitute 3 for 2 in the beginning of the command.

A description of the editing of the wing and the creation of the slender body and interference shell was omitted since only a hypersonic solution (i.e. HABP) was conducted on the vehicle forebody.

### Setting up the analysis

Now that the geometry is completely defined and edited, it is time to copy it from the permanent file to the local file to begin analysis. This is done using the attach command.

```
atta 10 210 610 611
```

**fil** (list contents of the local file)

```
set
configuration in work file is configuration number: 1
```

Once the set command is invoked, APAS recognizes that there is a new configuration in the local file. It calls it configuration 1, and prompts the user for several aerodynamic quantities related to it.

APAS prompts for wing reference area, wingspan, and average chord length. These are quantities APAS and UDP/HABP will use to non-dimensionalize the aerodynamic forces and moments calculated. As such, it is important that they be the *actual* reference area and average chord, not the theoretical values we initially used to input the wing geometry.

```
enter sref,cbar,span:
100,10,10
```

The vehicle's center of mass location should have been determined in a weights and sizing spreadsheet previously. The LOX/LH2 Stargazer used in this analysis will have the following CG values at takeoff:

```
enter xcg ,ycg ,zcg :
40,0,0
```

```
default configuration is : 1
```

```
enter the following parameters in the indicated units:
  altitude(alt)      : feet
or stag pressure(pr) : lbs/ft2
& stag temperature(temp): rankine
  roughness (ks)    : inches
```

```
** set **
```

Now we are ready to begin setting up the runs that UDP/HABP will solve. Since we will be running a series of angle-of-attack values for each Mach number, it simplifies matters significantly to set up a set of values.

```
set (command to create a new set)
enter list of run variables for set1
```

```
a(-30 -25 -20 -15 -10 -5 -3 0 3 5 10 15 20 25 30 35 40) ks(0.00025)
```

```
** set **
```

*ls* (command for list sets)

```
set1: a(-30 -25 -20 -15 -10 -5 -3 0 3 5 10 15 20 25 30 35 40) ks(0.00025)
```

```
enter list of run variables for set2
```

```
** set **
```

These entries specify a sweep of angle-of-attack values, from  $-30$  to  $40$  degrees at various increments. They also specify a skin friction coefficient of  $0.00025$ , which is used for viscous drag analysis in UDP. When the list sets command is typed, not only will APAS list the sets, it will prompt for a new set of values. Simply press <enter> to bypass the creation of a new set.

Runs are added by typing:

```
add1 run1 m(1.5) alt(40000) hyper
```

```
add1 run1 m(6) alt(75000)
```

```
add1 run1 m(10) alt(170000)
```

```
hypersonic analysis run
```

```
run: 1 reference values:
```

```
mach = 6.000 pressure= 393.117 temp = 389.970 altitude= 40000.000
```

```
beta = 0.0000 p = 0.0000 q = 0.0000 r = 0.0000
```

```
d1 = 0.0000 d2 = 0.0000 d3 = 0.0000 d4 = 0.0000
```

```
d5 = 0.0000 d6 = 0.0000 ct = -1.0000
```

```
independent variable: alpha 1 points
```

```
0.000
```

```
*****
*
*          **** habp analysis option menu ****          *
*
* 0 = accept default options/else: completely reset      *
* 1 = do not include component shielding effects          *
*      (default: include component shielding)             *
* 2 = set model scale                                    *
*      (default: model scale = 1.0 is for input geometry) *
* 3 = do not include skin friction effects in analysis    *
*      (default: include skin friction effects)           *
* 4 = do not add hemispherical nose to body panels        *
*      (default: add hemispherical nose cap between 1st and *
*              2nd input geometry sections if input body *
*              nose half cone angle is greater than 30 deg)*
*
*          eg: 2 4                                         *
*****
```

A typical Stargazer analysis would choose options 1 and 4, but for the APAS-GASP comparison analysis, option 0 was chosen so that HABP would model a hemispherical nose cap between the first and second sections. This was done so as to be comparable to the analysis done on the leading edge by GASP.

```
*****
*
*          **** specify skin friction option ****          *
*
*
*****
```

```

*   enter: 0 - turbulent                                     *
*                                                                 *
*           1 - laminar                                     *
*                                                                 *
*           2 - laminar to turbulent transition            *
*             nasa - wilhite                               *
*           3 - laminar to turbulent transition            *
*             based on momentum thickness                  *
*                                                                 *
*****

```

At this point, APAS is prompting for the type of flow assumptions that HABP should make for viscous analysis. For a typical Stargazer analysis, the laminar-to-turbulent transition (option 3) is chosen, but for the GASP-APAS comparison analysis, laminar flow (option 2) was chosen so as to be consistent with the selections in GASP.

When run definition is complete, type e to exit from the set environment. Upon exiting, if runs are set for analysis, APAS will prompt the user to checkout the runs, or in other words, to examine or change the options that UDP and HABP will use for analysis.

The HABP checkout procedure proceeds as follows:

```

***** configuration 1 hypersonic solution *****
           enter "y" for checkout , "n" to skip

y
do you need to reset habp analysis methods
for any of the components in this configuration ?
y
fuselage           body           :
recommended impact method: tangent cone emp
enter : number : of different method
       return : for recommended method
       1      : to list methods

fuselage           body           :
recommended shadow method: prandtl-meyer emp.
enter : number : of different method
       return : for recommended method
       1      : to list methods

enter value of q/qinf (default: 1.0000) :

fuselage           blunt end      :
recommended impact method: modified newtonian
enter : number : of different method
       return : for recommended method
       1      : to list methods

impact analysis method:modified newtonian
enter value for k (default: 1.8300)
-1.0 will provide stagnation k

fuselage           blunt end      :
recommended shadow method: high mach number base pressure
enter : number : of different method
       return : for recommended method
       1      : to list methods

```

```

enter value of q/qinf (default:      1.0000) :

do you wish to see configuration 1
yes or no :

```

## Executing the analysis

Now that the runs have been set, exit APAS. Subsonic/supersonic analyses are conducted using the UDP (Unified Distributed Panel) program. Enter UDP by typing:

### *udp*

```

input the complete name for permanent geometry data direct access file
<enter>
if file status of **( old )** is correct hit return - if not enter no

opendr - unit 1, file "apas.geom"
        , access=direct, form=unformatted, status=old, recl = 6292

input the complete name for plotting & run data direct access file
<enter>

```

UDP will then begin its solution procedure; the length of time it takes to solve will depend on the number of subsonic/supersonic runs and the number of points they contain. Be warned, however: UDP is notoriously unstable, and oftentimes a solution, once examined, will turn out to be garbage. In such cases, double and triple-check slender bodies and interference shells. Make sure the interference shell touches the center of the root chord. Check the paneling on the wing and re-panel if necessary. Check the slender body and make sure that its first section is a point, but that its last section *is not* one (change this even if it cause the slender body geometry to deviate significantly from the fuselage geometry). Try to ensure that the slender body matches the fuselage closely, if this is possible given the last consideration. Often the only solution to such problems is to erase and completely redo these components.

Next we turn to hypersonic analysis. Fortunately HABP (Hypersonic Arbitrary Body Program) is far more stable and simple than UDP. Indeed, if hypersonic analyses are the only ones desired, the construction of a slender body and interference shell is unnecessary. And although HABP concerns itself with hypersonic flow, and its attendant simplifications, a judicious choice of solution procedures can lead to rather accurate solutions even in the supersonic (Mach 1.5 - 4) range.

It is strongly recommended that a circumspect user of HABP be familiar with hypersonic flow theory; namely the hypersonic flow approximations, that form the heart of HABP. Only by understanding the equations and knowing their assumptions and limitations can a user feel confident that the solution procedures he or she chooses are representative of actual flow structure. An excellent reference for these approximations is chapter 2 of John Anderson's *Hypersonic and High Temperature Gas Dynamics*.

Begin HABP by typing:

***habp***

```
input the complete name for permanent geometry data direct access file
<enter>
** open file 15 for permanent geometry data direct access file old **
input the complete name for plotting & run data direct access file
<enter>
** open file 16 for plotting & run data direct access file old **
```

If, during HABP run definition, a laminar-to-turbulent transition (option 3) was selected, HABP will prompt for the location of the transition point on both axisymmetric and 2D geometries. Experience has shown that the default values offered by HABP are adequately accurate for this level of analysis.

```
momentum thickness b.l. transition method.
user must select values for retheta/mach
at the onset of transition.
recommended values: axis. ---> 150
                   2-d ---> 350

enter value chosen for axisymmetric transition:
150

enter value chosen for 2-d transition:
350
```

Similarly, if skin friction analysis was chosen during HABP run definition, HABP will prompt for the wall conditions. Reference enthalpy at the radiation equilibrium temperature (option 3) is the best approximation to the adiabatic wall condition chosen in GASP, so it is selected. Conversely options 2 and 4 give the option for a constant wall temperature, which would be suitable for an actively cooled structure.

```
skin friction analysis methods:

1 - ref. temperature (rad. equ. temp.)
2 - ref. temperature (constant t-wall)
3 - ref. enthalpy (rad. equ. temp.)
4 - ref. enthalpy (constant t-wall)
note: ref. enthalpy uses equil. air properties

choose a method:
3
```

After these inputs, and a few moments, HABP will display a list of aerodynamic coefficients for the current run and prompt for the boundary layer transition settings for the next run. It is recommended that the “recommended values” of 150 and 350 be chosen. For skin friction analysis, again the “recommended value” is 3, reference enthalpy for radiation equilibrium temperature. Unfortunately for a user with many runs, these values have to be entered for each run, which can become rather tedious.

Post-processing of APAS data in the MDO environment usually consists of writing a “deck” of aerodynamic coefficients suitable for use in a trajectory optimization program, such as POST. A simple program called APASDAT greatly facilitates this process. Enter the program by typing:

***apasdat***

```
input the complete name for plotting & run data direct access file
<enter>
if file status of *( old )* is correct hit return - if not enter no

*( open file 4 for plotting & run data direct access file old **
run numbers
    1          2          3          4          5
    6          7          8          9         10
   11         12         13         14         15
   16

0 - exit
1 - print all runs
2 - print 1 run
3 - change coefficient
4 - write post aero deck
5 - plot
6 - Olds special
```

Run numbers are listed above the menu; if they have dashes beside them (such as -12) it indicates that that run has not been analyzed yet. Runs can be examined using options 1 and 2, and the POST aero deck is written by choosing option 4 and following the instructions. Exit the program by typing 0.

This concludes the APAS tutorial for Stargazer. Hopefully, these explanations will enable a user to rapidly recreate the analyses reported in this paper, and avoid many of the pitfalls that ensnared the author as he attempted it.

## Appendix C

### Geometric Equations for Wing Definition

---

A simple wing is essentially a trapezoid, and can be designed using four parameters. Wing design typically specifies eight parameters, which can be expressed as functions of each other.

Theoretical Wing Reference Area	$S_{ref}$
Wingspan	$b$
Theoretical Aspect Ratio	AR
Theoretical Taper Ratio	TR
Wing Leading Edge Sweep	$\theta_1$
Wing Trailing Edge Sweep	$\theta_2$
Theoretical Root Chord	$c_r$
Tip Chord	$c_t$
Average Chord	$\bar{c}$

A distinction is made between theoretical values of  $S_{ref}$ , AR, and TR because wings typically do not extend to the centerline of the vehicle fuselage. In such cases, the actual values of these quantities will differ from the theoretical quantities, and adjustment must be made.

### Useful Geometric Equations for Wing Definition

---

$$S_{ref} = \left(\frac{b^2}{4}\right)(\tan \theta_1 - \tan \theta_2) \left(\frac{1-TR}{1+TR}\right)$$

$$S_{ref} = bc_t + \left(\frac{b^2}{4}\right)(\tan \theta_1 - \tan \theta_2)$$

$$S_{ref} = b(c_r + c_t)$$

$$b = \sqrt{\frac{4S_{ref}}{(\tan \theta_1 - \tan \theta_2) \left(\frac{1-TR}{1+TR}\right)}}$$

$$\theta_2 = \arctan \left( \tan \theta_1 - \frac{4}{AR} \left(\frac{1-TR}{1+TR}\right) \right)$$

$$AR = \frac{b^2}{S_{ref}}$$

$$c_r = \left(\frac{1}{b}\right) \left( S_{ref} + \left(\frac{b}{2}\right)^2 (\tan \theta_1 - \tan \theta_2) \right)$$

$$c_t = \left(\frac{1}{b}\right) \left( S_{ref} - \left(\frac{b}{2}\right)^2 (\tan \theta_1 - \tan \theta_2) \right)$$

$$\bar{c} = \frac{c_r + c_t}{2} = \frac{S_{ref}}{b} = \frac{b}{AR}$$

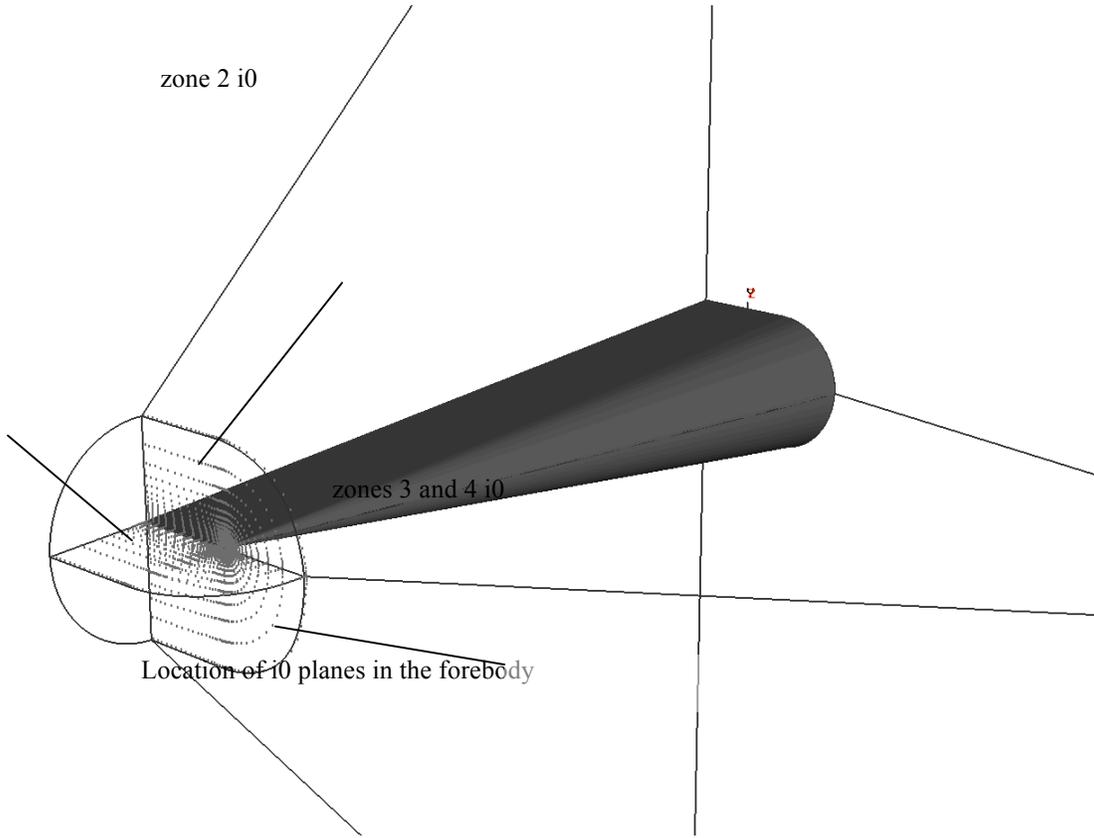
$$TR = \frac{c_t}{c_r}$$

## Appendix D

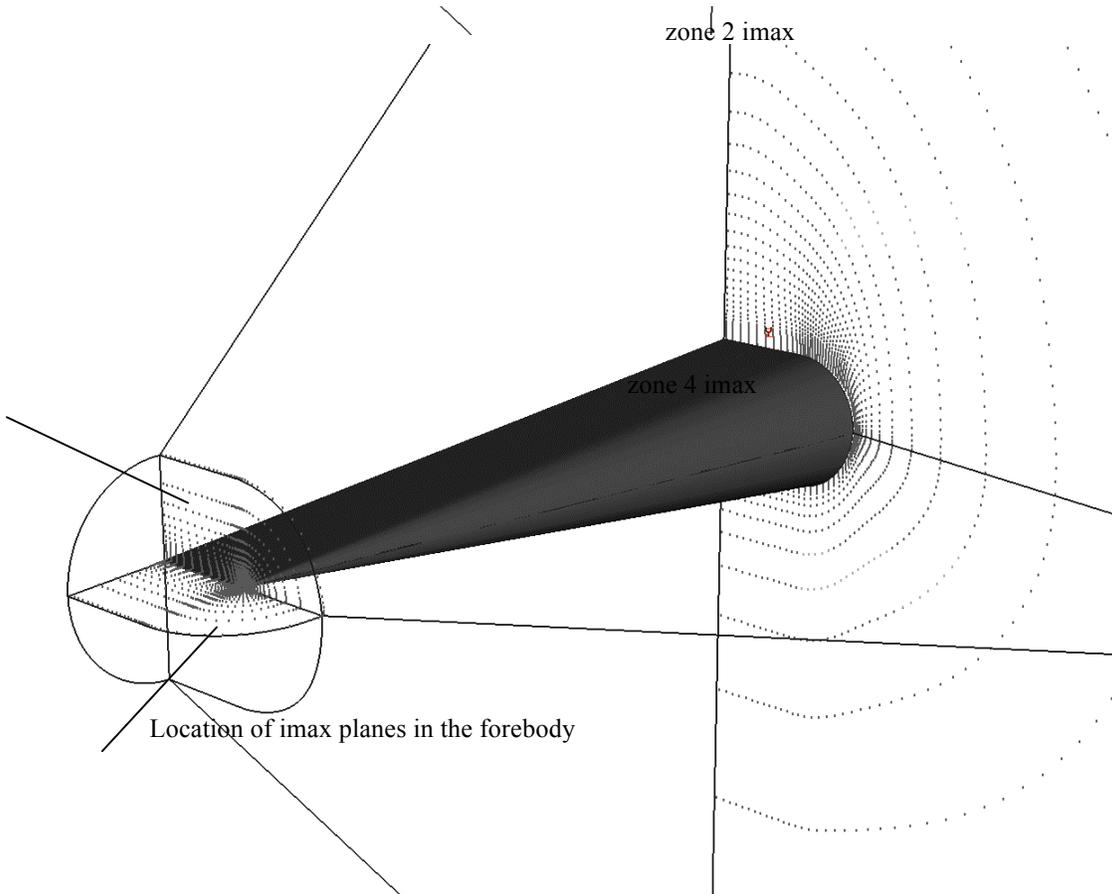
### **Zonal Boundaries and Visualization**

---

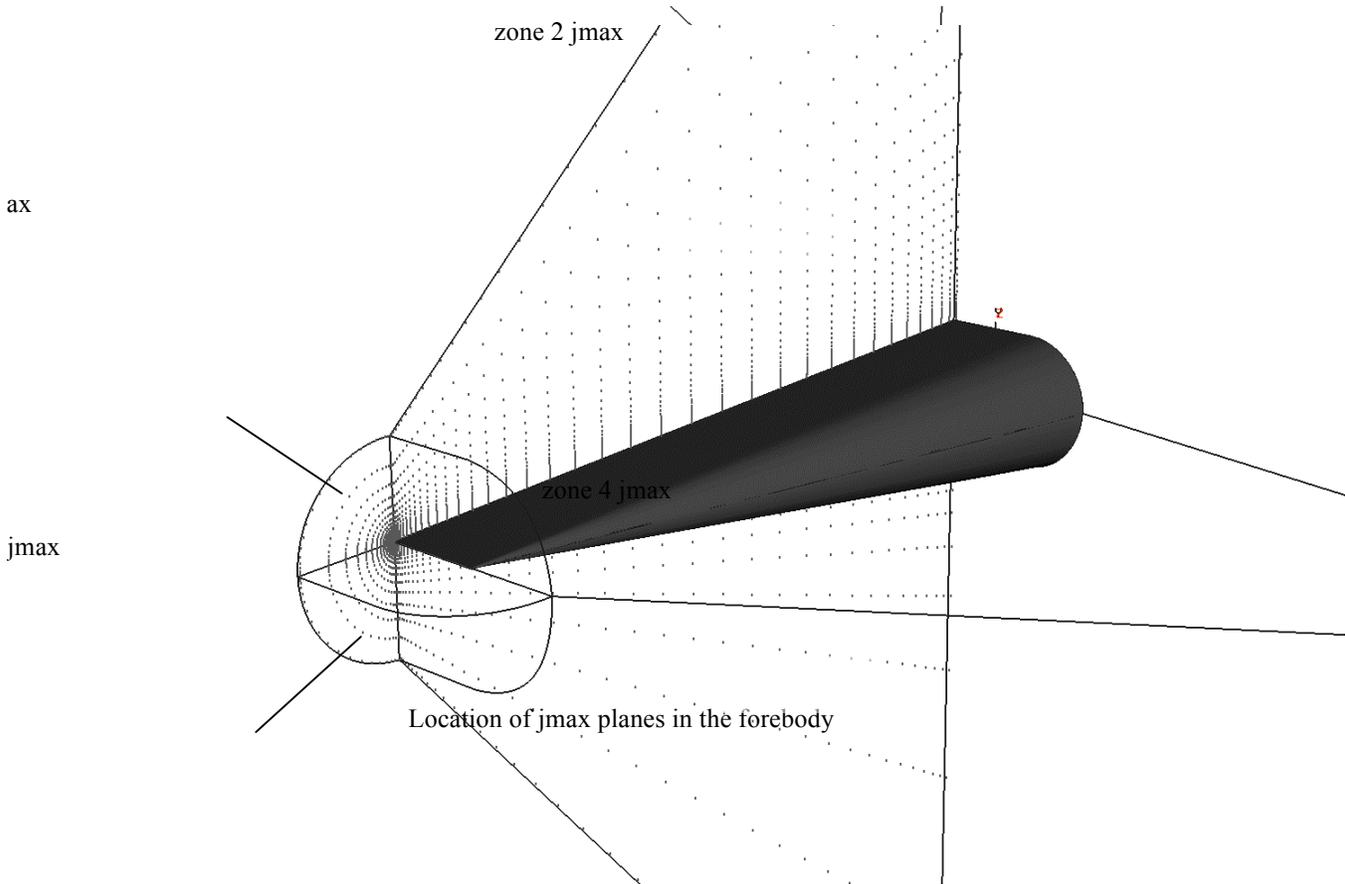
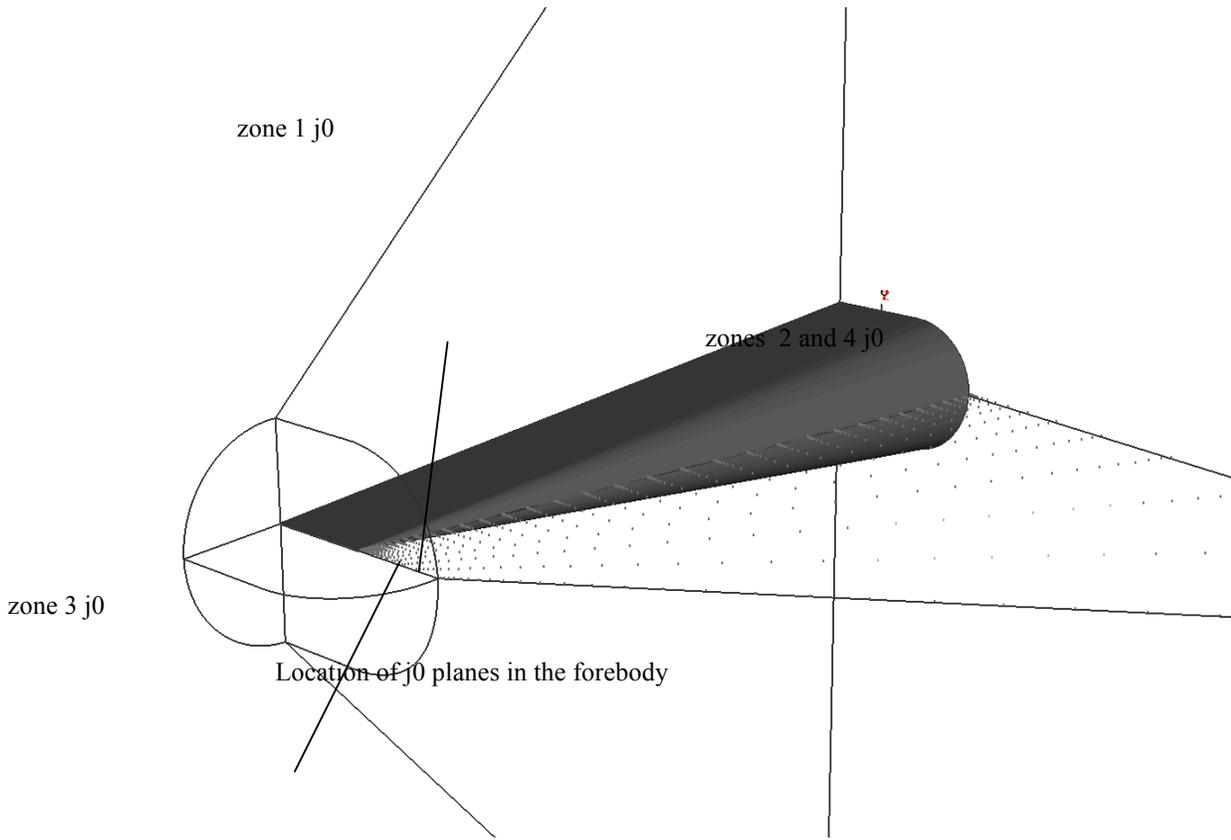
$i_0$



x



zone 3 imax



1 k0

1ax

1ax

