SAND99-1259C

AIAA 99-1726

An Overview of the Development of a Vortex Based Inflation Code for Parachute Simulation (VIPAR)

Vance L. Behr, Walter P. Wolfe, Carl W. Peterson
Sandia National Laboratories
Albuquerque, New Mexico

and

Christine E. Hailey
Utah State University
Logan Utah

15th CEAS/AIAA Aerodynamic Decelerator Systems Technology Conference
8-11 June 1999
Toulouse, France

For permission to copy or republish, contact the American Institute of Aeronautics and Astronautics
1801 Alexander Bell Drive, Suite 500, Reston, VA 20191

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

AIAA-99-1726

AN OVERVIEW OF THE DEVELOPMENT OF A VORTEX BASED INFLATION CODE FOR PARACHUTE SIMULATION (VIPAR)

Vance L. Behr[*], **Walter P. Wolfe**[*], **Carl W. Peterson**[†]
Sandia National Laboratories[‡]
Albuquerque, New Mexico

Christine E. Hailey[§]
Utah State University
Logan, Utah

Abstract

Sandia National Laboratories has undertaken an ambitious, multiyear effort to greatly improve our parachute system modeling and analysis capabilities. The impetus for this effort is twofold. First, extending the stockpile lifetime raises serious questions regarding the ability of the parachutes to meet their requirements in the future due to material aging. These aging questions cannot currently be answered using available tools and techniques which are based upon the experience of expert staff and full-scale flight tests and are, therefore, not predictive. Second, the atrophy of our parachute technology base and the loss of our experienced staff has eroded our ability to respond to any future problems with stockpiled parachutes or to rapidly design a new parachute system on an experience base alone. To assure a future in-house capability for technical oversight of stockpile nuclear weapon parachutes, Sandia must move from our present empirically based approach to a computationally based, predictive methodology. This paper discusses the current status of the code development and experimental validation activities. Significant milestones that have been achieved and those that are coming up in the next year are discussed.

Introduction

The goal of the Vortex Based Inflation Code for Parachute Simulation (VIPAR) development project is to produce a computational tool that can accurately simulate the entire parachute deployment sequence from initial deployment to ground impact for all weapon deployment conditions. The code must be capable of computing the fluid mechanics of the flow about a parachute at speeds ranging from low subsonic through transonic, the structural response of the parachute material due to aerodynamic and inertial loads, and the complex fluid/structure interactions. Parachute deployment is a complex, dynamic event. A parachute decelerates a body by transferring momentum from the body to the surrounding air. Although lightweight, a weapon parachute transfers a lot of momentum very quickly. While this is ideal for decelerating weapons, it complicates modeling. The challenges in this project can be categorized into those dealing with the fluid mechanics, structural dynamics, fluid/structure coupling, parallelization of the computations, experimental validation, and user interfaces. Each of these areas could be the subject of a complete paper in its own right, thus discussion here will be limited to a summary nature only. Discussion will concentrate on the progress made since the initial report on this effort[1] and goals for the immediate future.

Fluid Mechanics

The fluid mechanics nature of the problem has been reviewed in depth in past articles.[2-4] The fluid mechanics portion of the VIPAR code is targeted to solve the unsteady Navier-Stokes equations using gridless vortex methods. Vortex methods were chosen for several reasons.[5] The geometry of a parachute undergoes radical change during inflation. Since these large geometry changes would require significant computational effort be spent to regrid the flow field at

[*] - Principal Member of Technical Staff, Associate Fellow of AIAA
[†] - Department Manager, Fellow of AIAA
[§] - Associate Professor, Senior Member of AIAA

[‡] - Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-Aco4-94AL85000

every time step, grid-based CFD methods would encounter great difficulties for parachute inflation calculations. Because vortex methods only need to resolve the wake region generated by the parachute, they have a reduced computational domain compared with grid-based methods and, therefore, may turn out to be more computationally efficient. They are also more stable and have less numerical diffusion than other CFD formulations. Traditionally, vortex methods have been used to model incompressible flow. However, our parachute application also requires calculations in the transonic and low supersonic regimes. One of the future milestones for the program and biggest unknowns to date is determining the feasibility of extending the vortex formulation into the compressible flow regime.

To understand some of the pressing issues of the VIPAR code development effort, it is instructive to look at the governing equations. The vorticity transport equation (valid for constant viscosity, incompressible fluids) can be obtained by taking the curl of the momentum equation.[6]

$$\frac{D\omega}{Dt} = \omega \cdot \nabla \mathbf{u} + \nu \nabla^2 \omega \qquad (1)$$

This formulation is particularly attractive when analyzing two-dimensional, incompressible flows since the first term on the right hand side of the equation vanishes identically. However, for three-dimensional flows it remains and is generally referred to as the "vortex stretching" term since the gradient of velocity is involved. It is noteworthy that the pressure does not appear in the vorticity transport equation, meaning that the vorticity, and hence the velocity fields, can be determined without solving for the pressure field.

Another equation can be obtained in a similar way that describes the pressure. By taking the divergence of the incompressible momentum equation for a fluid with constant viscosity, one can obtain the following equation.[7]

$$\nabla^2\left(\frac{p}{\rho}\right) = \omega \cdot \omega + \mathbf{u} \cdot \left(\nabla^2 \mathbf{u}\right) - \tfrac{1}{2}\nabla^2\left(\mathbf{u} \cdot \mathbf{u}\right) \quad (2)$$

Notice that the equation for pressure does not involve the viscosity of the fluid. Any viscous effects will be introduced into the pressure only secondarily through the vorticity transport equation.

The velocity can be obtained from the vorticity field by integrating the influence of the entire vorticity field on any point in space. This yields the following equation (for incompressible flow):

$$\mathbf{u} = \frac{1}{4\pi}\int_V \frac{\omega(\boldsymbol{r}') \times (\boldsymbol{r} - \boldsymbol{r}')}{\left|\boldsymbol{r} - \boldsymbol{r}'\right|^3} dV \qquad (3)$$

which, when discretized for solution by numerical means, becomes a summation of the following form.[16]

$$\mathbf{u}(\boldsymbol{r}) = \boldsymbol{g}(\boldsymbol{r}) \sum_{i=1}^{N_V} \boldsymbol{h}(\boldsymbol{r}'_i)\delta V(\boldsymbol{r}'_i) \frac{1}{\left|\boldsymbol{r} - \boldsymbol{r}'\right|^2} \qquad (4)$$

This summation can then be broken down into schemes which allow for faster computation as will be discussed in the parallelization section.

In the more common grid based schemes, the fine grid near any surface with a no-slip boundary condition allows for velocity gradients and hence shear stress to be modeled. While vortex methods can be used to resolve boundary layer velocity gradients, one of the advantages to the method is that it is not necessary to do so. One of the critical areas for achieving success in the vortex method is how vorticity is added to the flow due to the shear layer at the surface. Indeed, introducing the boundary conditions in vortex methods requires careful attention.

We know that both the normal and tangential relative velocity must be zero at any impermeable surface. For two-dimensional flow, the zero normal velocity boundary condition at a surface can be met by introducing a sheet of vortices along the surface as shown (end on view) in Figure 1. Commonly the surface is broken in to panels and a vortex associated with each panel.

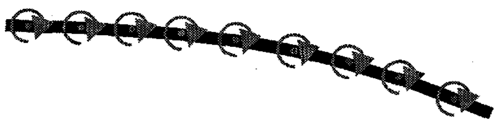


Figure 1

By splitting the vortex sheet about the surface as shown in Figure 2, the zero tangential boundary condition can also be satisfied.

Once the velocity field is calculated from the vorticity field, the vorticity created by the shear layer can be convected and diffused away from the surface and the process at the surface repeated.

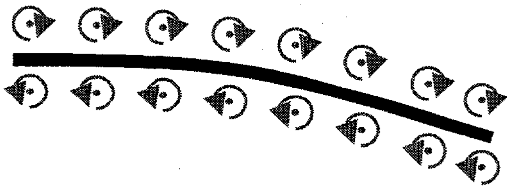


Figure 2

Conceptually then after one time step the vortex field may appear something like that shown in Figure 3 (before splitting the vortex sheet to handle the zero tangential velocity boundary condition.)

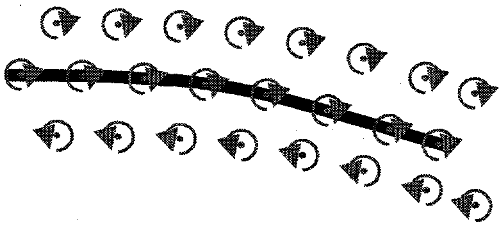


Figure 3

Following this procedure leads to two vortons being created per surface panel per time step (one on each side of the surface panel). This is a characteristic that will be discussed later in the parallelization section.

While this treatment of the boundary conditions seems fairly straightforward in these two-dimensional examples, things become more complicated when treating a full three-dimensional problem. Considerable effort has gone into the proper handling of boundary conditions for the 3-D case. In order to describe the problem, consider a triangular surface element for which the boundary conditions must be satisfied. It can be shown that the boundary conditions can be satisfied in an integral sense, with a constant value of circulation co-located with the element edges as shown in Figure 4. This can also be shown to be equivalent to a constant jump in the flow potential (*Δφ*) from one side of the surface to the other at any given point as depicted in Figure 4.

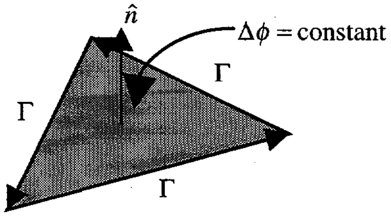


Figure 4

While this treatment of the surface panel is simple, it leads to noisy vorticity fields near the surface due to the concentration of circulation at the edges. This in turn leads to noisy surface differential pressures, which frustrates the coupling of the structural and fluid aspects of the problem. Thus, a different more regular description of the circulation was sought. The current approach is to allow the jump in flow potential to be spatially linearly dependent as shown in Figure 5. The circulation per unit area, *γ*, is then defined as a vector quantity in the plane of the element and related to the potential jump as shown. This treatment allows for much smoother vorticity fields and pressures and seems to lend itself better to our problem.

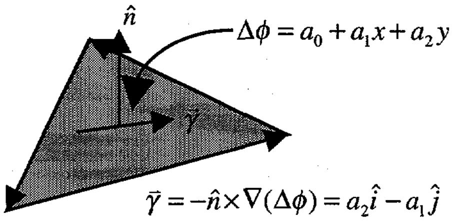

$\hat{n}$
$\Delta\phi = a_0 + a_1 x + a_2 y$
$\vec{\gamma}$
$\vec{\gamma} = -\hat{n}\times\nabla(\Delta\phi) = a_2\hat{i} - a_1\hat{j}$

Figure 5

The last schedule reported to this community[1] called for the investigation and initial development of a gridless compressible flow solver by this point in time. While the review of such methods has taken place[8] nothing was found in the literature that fit our problem and the development of this aspect of the solution has been delayed due to limited resources. The diffusion and stretching terms of the 3-D vorticity transport equation are just beginning to be addressed. The first implementation of the 3-D flow code will likely ignore these terms in favor of demonstrating a highly parallelized flow solver coupled to a structural code. These terms will then be addressed as additional modules to the code.

Parallelization

The coupled fluid/structure problem of inflating parachutes is a very difficult one. The code would likely never be able to produce meaningful results if run on a single processor computer in a scalar fashion. Instead, these computations require taking advantage of recent advances in massively parallel computing architectures. A significant portion of the VIPAR effort to date has been spent on investigating which solution techniques lend themselves to parallelization and which do not. This has at times meant rewriting code to take advantage of solution schemes that were found to be far superior in the parallel environment vs. a more traditional scalar-computing environment.

The most common approach taken in vortex methods is to represent the vorticity field by discrete vortex elements, ω_i, at various locations, $\boldsymbol{r}_i$. A summation of the influence of each vortex elements over all of the elements is then used to approximate the integral for the velocity. Unfortunately, as the number of vortex elements, N_V, grows, the computation time grows as N_V^2. However, due to the $1/\boldsymbol{r}^2$ behavior of the velocity equation, vortex elements far removed from any given region of the flow field can be considered to have an aggregate affect on the velocity in that given region. This means that vortex elements can be grouped and the influence calculated on an aggregate basis rather than on an individual basis. Criteria have been developed that allow the determination of how close vortex elements must be before they must be considered individually. Indeed, the error due to this approximation is directly related to the proximity criterion used to determine whether individual or aggregate behavior is utilized. Applying this computational scheme reduces the rate of growth in computational requirements from N^2 to $N{\cdot}lnN$, thereby significantly reducing the computation time. The process of grouping vortons and calculating their influence on other vortons and the velocity field uses Fast Multipole Methods (FMM)[16] to approximate the integral in Equation 3 for vorticity and velocity.

Even with the grouping of vortices, however, the computational requirements can quickly become excessive for a single processor machine. Taking full advantage of a parallel computer requires that the numerical algorithms be of a nature distribute the overall workload rather evenly between the processors. One of the significant milestones in the past two years has been the development of a FMM that demonstrates good parallelization characteristics. Our current procedure is based upon the Clarke-Tutty[9] scheme which recursively divides the computational region into binary number of "boxes" having the same number of vortices (± 1). The calculation is then performed individually between vortices within individual boxes or in closely located boxes, and on an aggregate basis between remotely located boxes as discussed earlier. It turns out that this procedure parallelizes very nicely.

Ideally the "wall clock" time, t_{calc}, for a given calculation on a single processor computer would be reduced to t_{calc}/P for a multi-processor computer, where P is the number of processors. Figure 6 shows the run time for solving a problem with the Clarke-Tutty FMM as a function of number of vortons and processors. These runs were made on the Sandia teraflop class computer using as many as 2048 processors. As you can see in Figure 6, run time is reduced as more processors are utilized and increases as the total number of vortices increases.

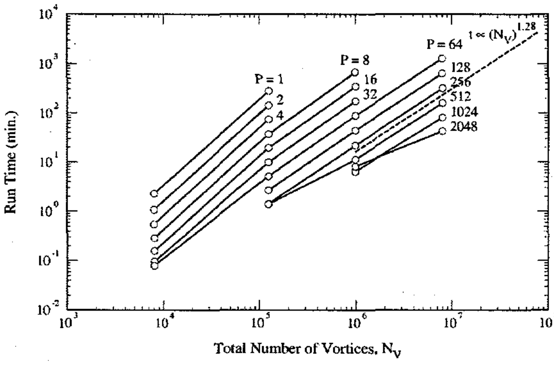


Figure 6

Figure 7 shows how the aggregate run time (i.e., the average run time per processor multiplied by the number of processors) increases with the number of vortices. We see that indeed for large numbers of vortices the aggregate time grows at a rate very close to the $N_V \, ln \, N_V$ limit.

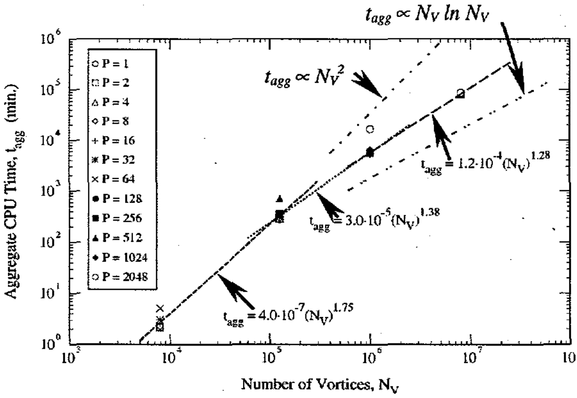


Figure 7

If we define a "Speedup" parameter as the time for a calculation on one processor divided by the time for a calculation on multiple processors, we can plot the speedup factor versus the number of processors and vortices as shown in Figure 8.

In summary, the FMM solver has been exercised on a parallel machine with 8,000 to 8,000,000 vortons and utilizing 1 to 2048 processors. In these computations we find, not surprisingly, that the larger the number of vortons, the more accurate the solution. This is roughly

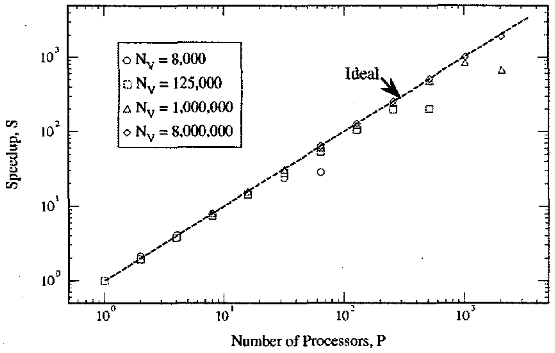


Figure 8

equivalent to smaller elements giving more accurate solutions in grid-based methods.

The calculations show that for a fixed number of processors, the "wall clock" run time increases as $(N_V)^{1.28}$, where N_V is the number of vortices. The parallel efficiency was found to be in excess of 80% for $N_V/P > 1{,}000$ and in excess of 90% when $N_V/P > 10{,}000$, where P is the number of processors. It was also determined that the Clark-Tutty FMM scheme leads to very good load balancing between processors for large problems with only a 1% variation between the minimum and maximum run times per processor.

The current scheme is limited by available memory because all the multi-pole coefficients are stored on every processor. (Each processor has only 128MB of memory.) A scheme has been devised to distribute this storage across all processors and is currently being implemented. If this scheme is successful, each processor should be able to handle even more vortices and allow the total number of vortices in the entire computational domain to grow beyond the current limit of 8,000,000 (for reasonable accuracy in the multi-pole expansion). This may well be necessary if we are to solve problems with realistic surface representations.

Structural Mechanics

The structural portion of VIPAR uses PRONTO3D. PRONTO3D is a three-dimensional, transient, solid dynamics code for analyzing large deformations of highly nonlinear materials subjected to extremely high strain rates. Since PRONTO3D has been under development at Sandia for many years for other applications, it does not need extensive code development. This code has already been parallelized for purely structural calculations. A detailed description of the code will not be offered here, but can be found elsewhere.[10] The challenging portions of this aspect of the program include: 1) adequately characterizing the woven textile materials used in parachutes,
2) structurally characterizing the behavior of sewn joints, 3) developing ways to easily specify the parachute structure as input to PRONTO3D, and 4) to develop a type of element that adequately describes the textile material that can be characterized physically.

To facilitate the initial attempts at solving a sample fluid-structure interaction problem, a structural mesh was developed for a small ribbon parachute. This mesh was generated using CUBIT,[11] a Sandia developed code for the creation of three-dimensional geometries with simple commands. However, the mesh was not generated in a general sense from a user specified geometry or by any preprocessor, but rather by "hand" using intuition. This mesh, representing the parachute canopy in a folded state, can be seen in Figure 9. The mesh generation process is confounded by the fact that a conical, continuous ribbon parachute is not well defined in a fashion like a machined metal part that has a well defined geometry in its zero stress state. The continuous ribbons are actually small cylindrical segments connected by the radials and mini-radials (roughly equivalent to verticals). The vent lines are also typically foreshortened to some length less than the vent band's equivalent diameter. These two items in particular create real difficulties for writing a generalized meshing algorithm for a ribbon parachute. Nevertheless, following the methodology used to generate the mesh in Figure 9, a generalized preprocessor can be developed.

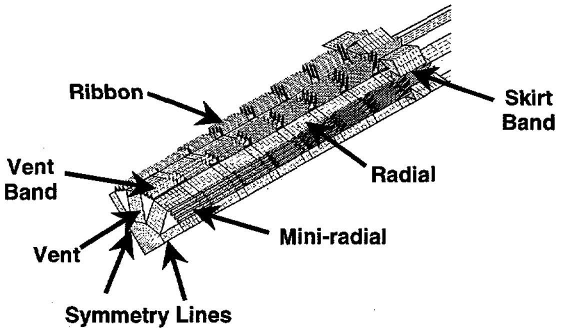


Figure 9

The features of PRONTO3D that are especially pertinent to the parachute problem are, shell and membrane elements, the handling of orthotropic materials and nonlinear deformation, and the element contact algorithm.

An initial calculation of this test case showed several problems. Severe hourglassing of membrane elements occurred. Tangled elements lead to very small time steps. This indicates that the contact algorithm may need additional intelligence to avoid tangling of large numbers of membrane or shell elements. Outside reviewers have also suggested that an implicit (or hybrid explicit-implicit) solution scheme might be of use to avoid the small time steps necessary in the explicit solution scheme of PRONTO3D. This will be an area of future examination for the structural side of the VIPAR code. The orthotropic materials option is currently under development for other applications and we will take advantage of that effort.

Fluid/Structural Coupling

As stated previously, the mass of the parachute is small compared with the mass of the air that is accelerated as the parachute inflates. In fact, the mass of the air is the dominant mass. The consequence of this is that a coupled fluid/structure-calculation is difficult. The highest degree of fluid/structure coupling would be to solve both the fluid and structure equations as one set of equations. However, this is feasible computationally because of the VERY large set of equations that would need to be solved. This approach would also result in inefficient use of resources since the structural solution, while cheap per time-step, takes time-steps on the order of microseconds, whereas the fluid calculation is very expensive but takes time-steps on the order of milliseconds.

The next logical scheme to consider would be a tightly coupled calculation in which an iteration is performed between a separate fluid and structure calculation until the same accelerations, velocities, and positions are calculated at the end of the time step for both codes. Unfortunately, this scheme is also impractical due to the cost of the fluid calculation

However, a loosely coupled calculation is possible if the added mass matrix is used to account for the surrounding fluid mass. The added mass matrix is readily available from the vortex formulation of the fluid mechanics problem.[12] In this formulation, the vortex solution passes pressures and the added mass matrix to the structural code. The structure code calculates (over many time steps) the new structural element positions and velocities which are then passed back to the vortex code. The added mass terms tend to control the movement of the structure to more physical levels and add stability to the calculation. This means of approach allows for both the structural and fluid computation to be made at their natural timesteps.

Since a three-dimensional flow solver was not yet ready for coupling to PRONTO3D, this coupling scheme was evaluated by coupling VPARA[17] (a 2D axisymetric, vortex based, flow solver) to PRONTO2D (using the two-dimensional cartesian option to model a single radial, vent line and suspension line.) PRONTO2D was modified to account for the (locally) perpendicular acceleration of the added mass at each structural node. (The full added mass matrix computed in VPARA was diagonalized[12] so there was a single added mass term for each structural node.) Since the purpose of the 2D coupling was to evaluate the added mass coupling scheme rather than a detailed analysis, some simplifying assumptions were made. The major assumption was ignoring the gore structure between two radials. The force from the pressure on the gore was applied directly to the radial, based on the circumferentially swept area between two radials. Any effect of the gore structural strength on determining the position of the radial (e.g. hoop stress) was ignored. The positions and velocities passed to VPARA were those of the radial member. However, this 2D coupling did allow the early investigation of issues and the education of the staff regarding the added mass, loose coupling approach.

Early attempts at the coupling proved to be unstable. Investigation showed that this was due to noisy pressures in the flow solution (causes mentioned in earlier fluid mechanics section). Thus, an artificial method of spatially smoothing the surface pressures and added mass was devised as shown below. This smoothing scheme assumes a regularly spaced structural grid. In fact, for this initial coupling attempt, the structural grid and surface grid for the flow field were taken to be the same with equally spaced nodes.

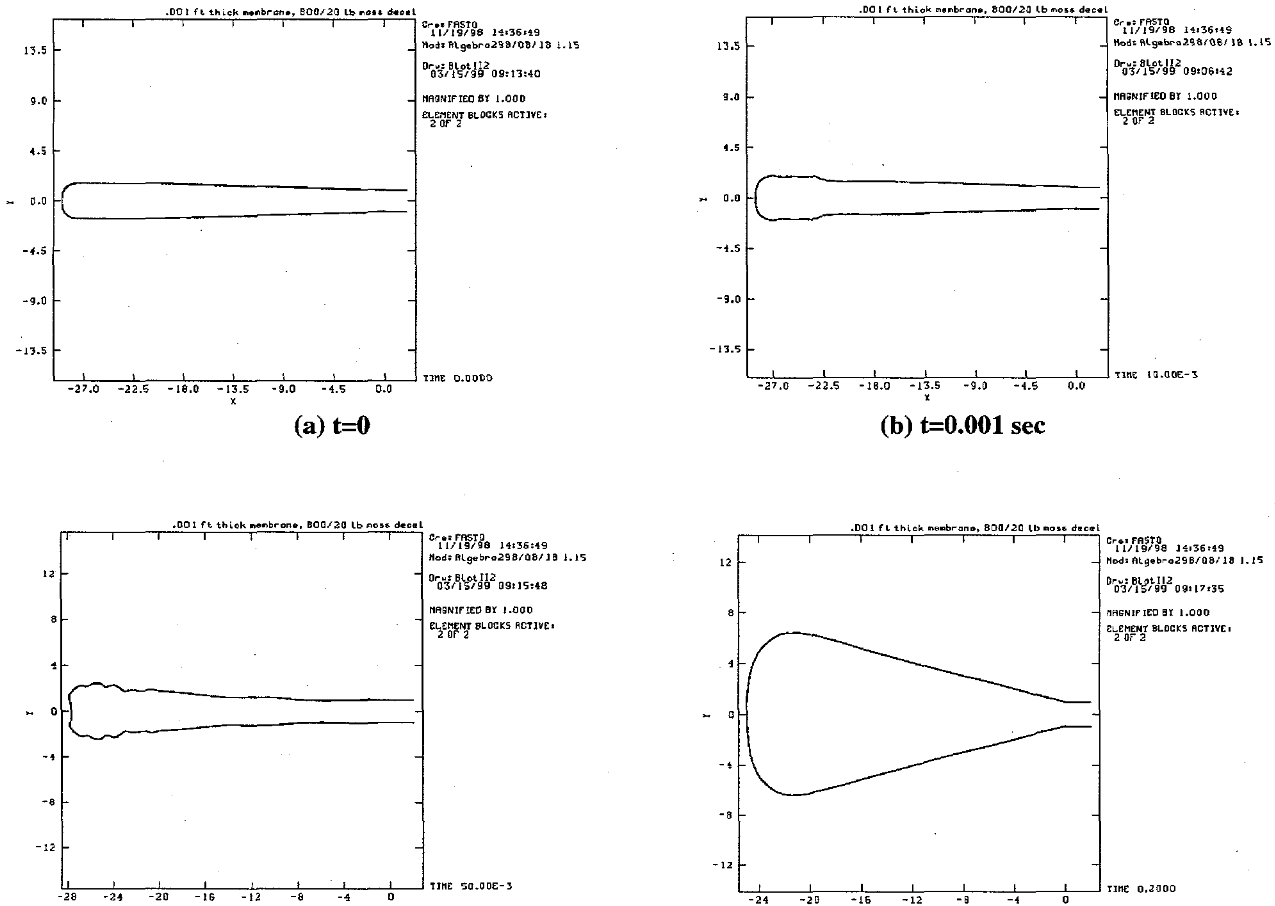
$$\Delta p_i = \frac{1}{16}\Delta p_{i-2} + \frac{1}{4}\Delta p_{i-1} + \frac{3}{8}\Delta p_i + \frac{1}{4}\Delta p_{i+1} + \frac{1}{16}\Delta p_{i+2} \qquad (5)$$

$$m_{i,added,sent} = \frac{1}{4}m_{i-1,added} + \frac{1}{2}m_{i,added} + \frac{1}{4}m_{i+1,added} \qquad (6)$$

A sample problem was considered as follows:

400 f/s initial velocity (q=190 psf)
800 lb payload
Gravity acting in x-direction
Aerodynamic forces on canopy only

Results were obtained for this simplified-model calculation that look remarkably realistic. Figure 10 shows the initial (assumed) canopy shape and the shape at 0.01, 0.05, and 0.2 seconds.



(a) t=0

(b) t=0.001 sec

(c) t=0.05 sec

(d) t=0.20 sec

Figure 10

It is important to note that while two years ago[1] calculated canopy shapes were presented, they were computed based upon an empirically based fluid pressure history acting on the canopy. The results shown here, while from a simplified model, are a result of a coupled fluid/structure calculation and are thus vastly closer to actually simulating the complex fluid-structure interaction.

Obviously much work remains to be done in this area. However, we must wait until some initial version of the 3-D flow solver is completed before a fully 3-D fluids/structure coupled calculation can be attempted. It also seems prudent to wait on improvements being made in the 3-D flow solver that will lead to smoothersurface pressures that drive the structural code. To not take advantage of these improvements would mean that much effort could be spent in developing smoothing techniques that would ultimately not be needed and hence the effort would have been wasted. Nevertheless, much has already been learned in this 2-D attempt at coupling two codes very similar to those to ultimately be used and will prove valuable in the very near future when the 3-D vortex solver is more mature.

Experiments for Validation

Each module of the VIPAR code will be subjected to verification checks as it is developed. These checks will ensure that a module accurately performs its stated task, the model equations are solved correctly, and the interaction between modules is correct. The results of these checks will be contained in the documentation for that module.

The code will also be subject to a validation process that ensures the code accurately models physical reality. High quality experimental data are required for validation. We have scheduled a series of experiments during each year of the project to obtain validation data. We will also examine existing data to determine what is available that can be used for code validation. During the early years of the project our validation experiments focus on single physics phenomenon and simple geometry. In later years, the focus of the experiments will shift more toward multiple physics phenomenon and more complicated geometry. The results of these validation comparisons will be used to quantify the accuracy of the code.

The first validation experiment devised consisted of testing a rigid hemispherical model with four "rings" and correspondingly three "slots" in a water tow tank. The model can be seen in Figure 11. The "rings" when measured along the surface of the model were 1-3/4 inches wide and the "slots" 7/16 inches wide. There was a 2-1/4 inch diameter vent, yielding an overall geometric porosity of 17.8%.

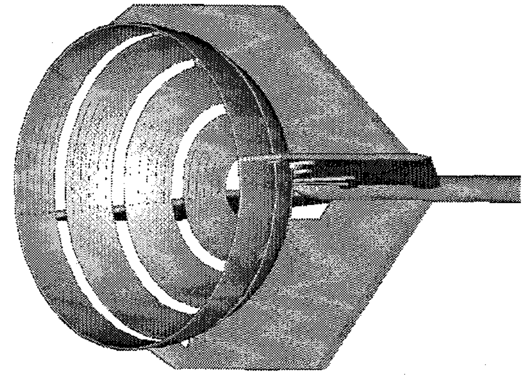


Figure 11

The model was mounted on a sting that allowed the model to be driven through the water at zero or some finite angle of attack. The horizontal velocity profile could be specified as a function of time. The model was equipped with numerous pressure ports to allow differential pressures between the front and back sides of the ribbons to be measured. The results of this experiment are discussed in detail elsewhere.[13] However, a sample of the differential pressure measurements can be seen in Figure 12.

This experiment required developing some rather innovative means of reducing the data in order to remove the acoustical delay time from the raw data. This technique has been described in elsewhere.[14]

A second experiment was just completed in March of this year. It consisted of stretching three steel ribbons, representative of typical nylon ribbons in width and thickness, crosswise to the flow in a wind tunnel. The ribbons were two inches wide, 3/32 of an inch thick, and were spaced 1/2 inch apart in a frame set in flow splitter plates as shown in Figure 13. Time accurate pressures were taken via pressure ports in the "ribbons". Particle Image Velocimetry (PIV) was also used to characterize what was intended to be a two dimensional flow field. Video cameras were also used to capture gross movement of the ribbons.

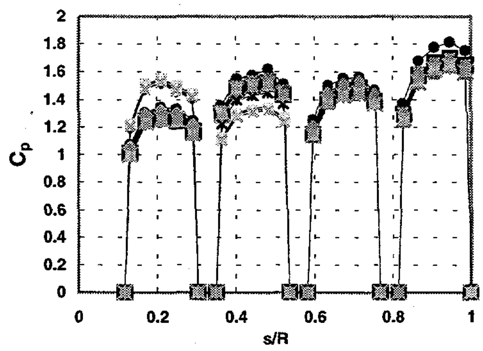


Figure 12.

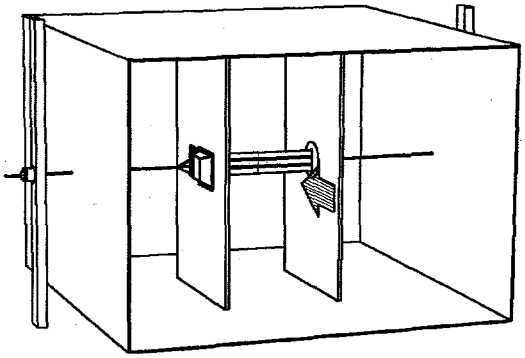


Figure 13

Some runs were also made with 1000-lb rated Nylon ribbons substituted for the instrumented steel ribbons as a comparison of the more flexible textile material with the steel ribbons. While we can say subjectively that a lot of very interesting data was acquired in this series of tests, the data is still in a state of reduction and no quantitative results can be presented at this time.

Experiments will continue to be devised that allow us to step progressively closer to the real world problem of a deploying/inflating parachute. Ultimately, we hope at the end of the program to do a full scale flight test with adequate instrumentation to validate our code. Certainly advances being made in the area of in-situ solid state data acquisition devices make this more of a possibility.

User Interface

Complex codes such as VIPAR can frequently be daunting for the designer to use. User interface routines will be written that will ultimately allow a typical parachute engineer to utilize this code for system design and evaluation. This effort will include developing suitable processing modules to generate the appropriate input for the VIPAR code as well as user-friendly post processing routines to analyze the voluminous data expected to be generated by such a code. These modules are considered absolute necessities in order for the project to be considered a complete success. Unfortunately, with limited resources, this aspect of the code development effort has been delayed and there is no significant progress to report.

Program Direction

A key programmatic goal was reached recently when an external review panel was assembled. Each of the areas discussed in this paper were reviewed, including the approaches taken, those that didn't work, those that did, and future direction. The panel members included experts in vortex fluid methods, fluid/structure interactions, structures, experimental fluid techniques, and code verification and validation. While the panel was composed of primarily members from the academic community, members from DoD laboratories and industry were invited as well. This review served as a "sanity check" on our approaches. The reviewers had many comments on what was being done right and what could be improved upon. The review panel assimilated all of their observations in a report[15] that we plan to use in part to help establish the future direction and priorities for the program.

Our immediate goals for the program include demonstrating a 3-D Vortex flow solver by the end of the fiscal year. We should also have the entire results of the wind tunnel experiments reduced and work should have commenced on the user interface for input to the VIPAR code. Emphasis will be placed upon determining the correct way to deal with the hourglassing and entanglement of elements in the structure code.

The current program plan states that next fiscal year we will begin looking at compressible flow solution techniques, continue to look at ways to model the vortex diffusion and stretching terms, and begin looking at means for vortex coalescence. The wind tunnel test data will be documented in a report and the next validation experiment(s) devised. The ultimate goal is to have an incompressible version of the code running at the end of fiscal year 2000 or the beginning of 2001 and a compressible version of the code completed to a "beta" version in fiscal year 2002.

Summary and Conclusions

In summary, significant progress has been made towards having a three-dimensional coupled fluid/structure code for modeling the flow about an inflating parachute. While significant amounts of work lie ahead, progress in the area of vortex methods, surface element treatment, added mass effects, and structural grid generation encourage us to keep striving for the final goal. Resources continue to be limited to below what would be ideally critical levels requiring a constant re-prioritization of tasks to ensure the best of possibilities for success.

References

[1] Peterson, C. W., et al, "Development of a Massively Parallel Parachute Performance Prediction Code," AIAA-97-1504, presented at the 14th AIAA Aerodynamic Decelerator Systems Technology Conference, June 3-5, 1997, San Francisco, CA.

[2] Strickland, J. H., and H. Higuchi, "Parachute Aerodynamics: An Assessment of Prediction Capability," J. Aircraft, Vol. 33, No. 2, March-April 1996, pp 241-252.

[3] Peterson, C. W., J. H. Strickland, and H. Higuchi, "The Fluid Dynamics of Parachute Inflation," Annu. Rev. Fluid Mech., Vol. 28, 1996, pp. 361-87.

[4] Wolfe, W. P. and J. H. Strickland, "A Hybrid Vortex Method for Two-Dimensional Flow Over Tube Bundles," presented at 3rd Int. Workshop on Vortex Flows and Related Numerical Methods, Toulouse, FR, August, 1998, to be published in ESAIM Proceedings.

[5] Gharakhani, A., "A Survey of Grid-Free Methods for the Simulation of 3-D Incompressible Flows in Bounded Domains," SAND 97-2256, Sandia National Laboratories, NM, September 1997.

[6] Batchelor, G. K., An Introduction to Fluid Dynamics, Sec. 5.2, Cambridge University Press, Cambridge, 1967.

[7] Currie, I. G., Fundemental Mechanics of Fluids, Section 3.4, McGraw Hill, Inc. 1974.

[8] Homicz, G. F., "FY98 ESRF Effort on Gridless Methods for Compressible Flows," internal memorandum, Sandia National Laboratories, Albuquerque, New Mexico, October, 14,1998.

[9] Clarke, N. R. and O. R. Tutty, "Construction and Validation of a Discrete Vortex Method for the Two-Dimensional Incompressible Navier-Stokes Equations," Computers and Fluids, Vol. 23, No. 6, pp. 751-783, 1994.

[10] Attaway, S. W., et.al., "PRONTO3D User's Instructions: A Transient Dynamic Code for Nonlinear Structural Analysis," SAND98-1361, Sandia National Laboratories, Albuquerque, NM, June 1998.

[11] Blacker, T. D., et. al., "CUBIT Mesh Generation Environment, Volume 1: Users Manual," SAND94-1100, Sandia National Laboratories, Albuquerque, NM, May 1994.

[12] "PRONTO2D/VPARA Coupling," Internal Memorandum from W. D. Sundberg to W. P. Wolfe, Sandia National Laboratories, Albuquerque, NM, 11 Jan., 1999.

[13] McBride, D. D., J. F. Henfling, and E. L. Clark, "Pressure Data for the Experimental Validation of a Vortex Based Parachute Inflation Code," AIAA-99-1747, presented at the 15th CEAS/AIAA Aerodynamic Decelerator Systems Technology Conference, Toulouse, FR, 9-11 June, 1999.

[14] McBride, D. D., J. F. Henfling, and E. L. Clark, "Determining the Transfer Function for Unsteady Pressure Measurements Using a Method of Characteristics of Solution," presented at 18th Int. Congress on Instrumentation in Aerospace Simulation Facilities, Toulouse, FR, 14-17 June, 1999.

[15] "External Program Review of VIPAR: comments from the March 29, 1999 Meeting," submitted to the VIPAR Program, Sandia National Laboratories, April 30, 1999.

[16] Strickland, J. H. and R. S. Baty, "A Pragmatic Overview of Fast Multipole Methods," Lectures in App. Math., Vol. 32, 1996.

[17] Strickland, J. H., "Axisymmetric Bluff-Body Flow: A Vortex Solver For Thin Shells," SAND91-2760, Sandia National Laboratories, Albuquerque, NM, May, 1992.