



AFRL-RB-WP-TR-2011-3057

**Development of an Aero-Optics Software Library and
Integration into Structured Overset and Unstructured
Computational Fluid Dynamics (CFD) Flow Solvers**

William J. Coirier and James Stutts

KRATOS/Digital Fusion, Inc.

**April 2011
Final Report**

Approved for public release; distribution unlimited.

See additional restrictions described on inside pages

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
AIR VEHICLES DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the USAF 88th Air Base Wing (88 ABW) Public Affairs Office (PAO) and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RB-WP-TR-2011-3057 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

**//Signature//*

SCOTT E. SHERER, Program Manager
Computational Applications Branch
Aeronautical Sciences Division

//Signature//

CAPT SARAH SUMMERS, Acting Branch Chief
Computational Applications Branch
Aeronautical Sciences Division

//Signature//

MICHAEL STANEK, Technical Advisor
Aeronautical Sciences Division
Air Vehicles Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

*Disseminated copies will show “//Signature//” stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YY) April 2011		2. REPORT TYPE Final		3. DATES COVERED (From - To) 09 March 2010 – 15 January 2011	
4. TITLE AND SUBTITLE Development of an Aero-Optics Software Library and Integration into Structured Overset and Unstructured Computational Fluid Dynamics (CFD) Flow Solvers				5a. CONTRACT NUMBER In-house	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 62201F	
6. AUTHOR(S) William J. Coirier and James Stutts				5d. PROJECT NUMBER 4QCQ97091401	
				5e. TASK NUMBER BY09-008SP	
				5f. WORK UNIT NUMBER A08A0B	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) KRATOS/Digital Fusion, Inc. 5030 Bradford Drive, Building 1, Suite 1 Huntsville, AL 35805				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Air Vehicles Directorate Wright-Patterson Air Force Base, OH 45433-7542 Air Force Materiel Command United States Air Force				10. SPONSORING/MONITORING AGENCY ACRONYM(S) AFRL/RBAT	
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) AFRL-RB-WP-TR-2011-3057	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES PAO case number 88ABW-10-6237, cleared 29 Nov 2010. Report contains color.					
14. ABSTRACT An object-oriented software library has been developed that is used to integrate a Paraxial Beam solver within both an overset (OVERFLOW) and an unstructured (AVUS) CFD model. The Paraxial Beam equation is used to propagate optical waveforms through the turbulent, unsteady flow fields produced by the structured overset and unstructured solvers. To achieve this new capability, an object-oriented, software library approach has been undertaken and has been used to develop the following components; a standalone mesh preprocessor, an easily integrated Paraxial Beam solver, a standalone Paraxial Beam solver and an assortment of post-processing tools. The new capability has been integrated within OVERFLOW and AVUS, and demonstrated for evaluating the Aero-Optical propagation through unsteady, turbulent flows past a circular cylinder and compared to aerodynamic and aero-optical data for a hemispherical turret..					
15. SUBJECT TERMS aero-optics, computational fluid dynamics, paraxial beam equations, OVERFLOW, AVUS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT: SAR	18. NUMBER OF PAGES 32	19a. NAME OF RESPONSIBLE PERSON (Monitor) Scott E. Sherer
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include Area Code) N/A

Table of Contents

List of Figures	ii
Acknowledgements	iii
1 Background	1
2 Introduction	2
3 Spectral/Parabolic Solution of the Paraxial Beam Equation	3
4 Aero-Optics Class Library, Framework and Data Flow	4
4.1 Library Components	5
4.2 Data Processing Work Flow	6
5 Mesh Pre-Processor	7
6 Application Programming Interface to CFD Models	8
7 Standalone Paraxial Beam Model	9
8 Post-Processing Tools and Models	10
8.1 Jitter and Boresight Error	10
8.2 Wavefront Error Processing	11
9 Optical Path Difference-based Wavefront Error Processing	11
10 Demonstration: Unsteady Flow Past a Circular Cylinder	12
11 Validation: 1 Foot Conformal, Hemispherical Turret	14
12 Conclusions and Recommendations	17
13 References	20

List of Figures

1	Convergence rate for the Gaussian beam problem.	5
2	Aero-Optics Class Library Components and Framework	5
3	Illustration of beam grids, interpolator grids and spectral grids	7
4	Streamwise aligned cut through the CFD grid and interpolator grid (left), OVERFLOW solution and interpolated density (middle) and AVUS solution and interpolated density (right) at a particular time step of an unsteady DES calculation	9
5	Unsteady flow past a circular cylinder. Entropy measure contours (left), Mach contours in a spanwise plane (middle), locations and directions of beam propagator grids (right).	12
6	Comparison of Wave Front Errors for OVERFLOW (top) and AVUS (bottom) on beam grids 1, 2 and 3 (across). Shown are computed rms values of the wavefront error in microns	13
7	Comparison of jitter for the three beam grids between OVERFLOW and AVUS.	13
8	Comparison of Mach contours mid-span for OVERFLOW (left) and AVUS (right). Both contours made with same range and levels of Mach number (21 levels from 0 to 0.2).	14
9	Grid system showing the upstream fetch, wall clustering and nested wake grids.	15
10	Beam grids and hot wire probe traverse locations in the CFD model	15
11	Centerline surface pressure comparison (upper left) and averaged streamwise velocity and streamwise velocity fluctuations at the traverse locations	17
12	Comparison of measured to predicted wave front distortions. Shown are the mean values +/- the standard deviations	18

Acknowledgements

The authors gratefully acknowledge the support of this project through a contract with the Air Force Research Laboratory, Computational Sciences Branch, Technical Monitor, Dr. Scott Sherer. We also thank Ralph Noack of Penn State University/Army Research Laboratory for his contribution of the Alternating Digital Tree, which formed the basis of the nearest-neighbor searching algorithms used in the pre-processing phase. Thanks are also due to Jennie Barber at Kratos/DFI for her assistance in developing the overset mesh system for the conformal turret. This work was supported in part by a grant of computer time from the DOD High Performance Computing Modernization Program at the USASMDC Simulation Center and at the Air Force Research Laboratory.

1 Background

This final report details the progress made under the Aero-Optics Code Development Program managed by the Computational Sciences Branch, Air Vehicles Directorate, Air Force Research Laboratory. This program was tasked with developing an in-house capability to perform computational analysis of the aero-optical environment associated with configuration of interest to the Air Vehicles Directorate. The progress in the first two years focused on two main areas: 1) identify and obtain high quality experimental data for various aero-optic configurations to serve as possible validation data for future code development, and 2) evaluate the unstructured flow solver AVUS for use on canonical cases representing the types of problems expected to be encountered in the aerodynamic analysis of future aero-optics configurations and improve the predictive capabilities of the code if feasible. The results of the effort in these two areas may be found in Reference [1]. The third area addressed during this effort involved the development of specialized versions of the CFD solvers OVERFLOW and AVUS to study aerodynamically-induced aberrations to optical wave forms as they are propagated through an unsteady, compressible and turbulent flow field. The initial work in this area may be found in Reference [2], where an aero-optics capability was added to the OVERFLOW solver. The current work details the follow-on effort to improve the implementation of the aero-optics solver in OVERFLOW based on lessons learned, and to extend the approach to the unstructured solver AVUS.

2 Introduction

The propagation of electromagnetic beams through turbulent flows has been an important topic for many years, with applications ranging from missile defense, directed energy, to target designation and tracking. An important aspect of these applications is determining the effective beam-on-target characteristics after the beam has propagated through both the near-field turbulent flow field of the vehicle and the far-field turbulent atmosphere. Near-field propagation (i.e. aero-optics) maintains some similarities to the far-field (i.e. atmospheric) propagation, but due to the interactions between turbulence length scales, beam wavelengths, apertures and distances, the two often require different approaches [3, 4, 5, 6]. As shown in Sutton [4], the techniques needed to evaluate the propagation of a beam from the vehicle to the target, or to evaluate the image imposed on a sensor from the targets signature are similar, although the engineering challenges from the two are quite different (e.g.; adaptive optics versus remote sensing).

Many noteworthy efforts have been made applying Computational Fluid Dynamics models for aero-optics predictions. For example, Tsai [7] in 1990 performed Direct Numerical Simulations (DNS) of a two-dimensional, planar mixing layer, and evaluated phase differences via Optical Path Difference (OPD) integration. Large Eddy Simulations (LES) have been made by many, including Jones [8], Sinha [9] Mani [10, 11] and recently by Visbal [12], and others. In these calculations, OPD variations, or equivalently, phase differences, have been used to characterize the beam distortion caused by the unsteady turbulent flow field. A Partially-Averaged Navier-Stokes (PANS) approach has been applied by Cenicerros [13, 14] to a turret model, where significant correlation to experimental statistics was shown, by comparing the unsteady variations of OPD about a mean. An approach which solves the Reynolds-Averaged Navier-Stokes (RANS) equations coupled with a compressibility corrected, two-equation turbulence model and a transport equation for the density perturbation variances have been made in Pond [15], where, again, OPD is used to characterize the aero-optical distortion. The approach of solving a density perturbation variance transport equation was also used previous to this work by Smith [16] within a parabolized/thin-layer solver, valid for shear layers.

The assumptions often made to apply a purely geometric optics approach, such as those noted above, may not be met for all flow and beam propagation conditions. As noted in Truman [17], diffraction may be neglected when $\lambda L/l^2 \ll 1$ (i.e.: the Fresnel condition), where the beam wavelength is λ , the propagation length is L and the turbulence length scale is l . The geometric optics approach is based upon the reduction of Maxwells equations to a parabolic (paraxial) form, and then further reducing this form to the solution of a phase-shift operator [17]. This assumes that the phase shift has no impact upon the beam amplitude, although the interaction between the phase and the beam spreading may not be negligible for all beam forms, propagation distances and index of refraction profiles. Since many applications need to predict effects such as scintillation, the application of a purely geometric approach might not be broadly applicable, and care needs to be taken in determining when to apply it. For shorter wavelengths, the phase shifting is more effective (due to the increased indices of refraction), and the scale difference between turbulence and beams are closer, so their interaction with the beam diffraction is more pronounced. Assumptions regarding compressibility, turbulence equilibrium, scales and isotropy are also often made [3, 4, 5, 6] that may not be applicable to the flow in the vicinity of an aerospace vehicle. As noted in Siegenthaler [5] and in Gordeyev [18, 19, 20], Visbal [12] and others, the complex flow patterns produced by protuberances, such as turrets, induce anisotropic turbulence fields, possibly not in turbulent equilibrium, with a wide range of length scales. With an increase in vehicle speeds, the assumptions often made regarding incompressibility are also suspect.

In Reference [1, 21], a specialized version of the OVERFLOW 2.1x overset CFD solver was developed to study aerodynamically-induced aberrations to optical wave forms as they are propagated through an unsteady, compressible and turbulent flow field. A reduced form of Maxwells equations, the Paraxial Beam equation, was solved concurrently with the Navier-Stokes equations to propagate a given wave form through the CFD generated density field, produced by Large Eddy Simulation (LES) or Detached Eddy Simulation (DES). The integration with OVERFLOW was made through a Fortran90 module containing procedures invoked by the master OVERFLOW process. A collection of post-processing utilities produced optical quality metrics to characterize the effects of flow treatment devices upon aero-optical transmission quality. As noted in Reference [21], the integration of the spectral/parabolic paraxial solver within the CFD model was relatively efficient, and for most cases, reduced the processing speed of the model by only 1-3%. The solver compared very well with theory/analytical solutions where applicable, and demonstrated spectral

convergence on certain problems [21]. Due to the tight integration of the paraxial solver within the CFD (OVERFLOW) model, some limitations and restrictions were encountered. The most restrictive limitation was the requirement that the paraxial equations be solved solely on separately identified grid zones that are part of the overall overset grid system. This means that the beam propagation directions and spans have to be determined during the mesh generation phase, and cannot be changed afterwards. In addition, during the grid compositing process, the beam path grids cannot be interpolated from any other donor grids placing unnecessary restrictions on grid topology and compositing. The procedures-based, tightly integrated approach within the CFD model resulted in a standalone model that was difficult to embed in other models, such as adaptive-optics or propagation models.

In order to provide a cleaner computational interface to other CFD models, we have developed an Aero-Optics software class library, and have used it to embed the spectral/parabolic paraxial beam solver within both an overset, structured solver (OVERFLOW) and an unstructured, finite-volume solver (AVUS). The integration within the CFD models is made as straightforward as possible, by providing four Application Programming Interfaces (API) called by the CFD codes. A mesh pre-processing phase finds interpolation stencils and their weights for a given mesh system and beam grid layout, and stores this data in a commonly accessed file. To speed up the software development, this common file format has been constructed using a specialization of the NetCDF library, and provides direct file access of commonly shared data by all of the applications in the aero-optics library framework. To provide maximum flexibility, the paraxial beam solver may be operated in a standalone fashion, where it uses the CFD generated density field, stored only at the limited number of points needed to interpolate the data from the CFD mesh to the beam grids. Post processing tools are provided that perform wave front processing tasks, and are used to characterize the aero-optical fields in terms of metrics, such as beam jitter and bore sight error. Other tools are provided that compute wave front error using Zernike polynomials, and may also be used to characterize the aero-optical distortions.

In the sections that follow, we describe the spectral/parabolic operator splitting method used to solve the parabolized Maxwells equations. The top-level framework of the library is described next, showing the typical data paths and steps taken to produce beams and post-process them. A detailed description of the mesh pre-processor is then given, followed by a description of the CFD model API provided by the aero-optics library and how we used these to integrate OVERFLOW and AVUS with the library. Post processing tools, including the standalone beam solver and aero-optics metrics generation are shown next. Finally, we demonstrate the model for the DES simulations past a circular cylinder using both OVERFLOW and AVUS, and compare measured to computed beam metrics and aerodynamic quantities for the flow past a 1-foot diameter, conformal, hemispherical turret mounted in the Air Force Academy wind tunnel.

3 Spectral/Parabolic Solution of the Paraxial Beam Equation

The aero-optics model is based upon solving the parabolized Helmholtz equation, also referred to as the paraxial beam equation [17, 22]. This is a reduced form of Maxwells equations which govern electromagnetic propagation through general media. For the electromagnetic propagation considered here, the medium, air, is considered to be non-conducting with a constant magnetic permeability[3]. In addition, since the time scale for the electromagnetic propagation is orders of magnitude faster than the flow time scales, the temporal variation of the electromagnetic field is assumed to be negligible. Depolarization is also neglected [17, 22], since the propagating wavelength is assumed to be much smaller than the turbulence length scale [17]. Neglecting propagation normal to the beam direction, assuming planar wave propagation and applying the paraxial approximation results in the following, where the coordinates are scaled by the propagation beam wavelength ($\xi = x/\lambda, \eta = y/\lambda, \zeta = z/\lambda$):

$$\frac{\partial^2 A}{\partial \xi^2} + \frac{\partial^2 A}{\partial \eta^2} + 4\pi j \frac{\partial A}{\partial \zeta} + 4\pi^2 \left[\left(\frac{n}{n_r} \right)^2 - 1 \right] A = 0 \quad (1)$$

The Gladstone-Dale constant relates the index of refraction to the fluid density for a given wavelength (in meters) as

$$n = 1 + K_{gd}\rho \quad (2)$$

where

$$K_{gd} = 2.24 \times 10^{-4} \left(1 + \frac{7.52 \times 10^{-3}}{(\lambda/1 \times 10^{-6})^2} \right) \frac{m^3}{kg} \quad (3)$$

We solve the paraxial beam equation (1) by a spectral/parabolic method using operator splitting[21]. The first parabolic operator is termed a phase shift advance since it only alters the phase of the planar wave, which is followed by the Fourier-based wave advance, which represents the Laplacian operator in the beam tangential direction via a Fourier decomposition, and advances the spectral coefficients in the propagation direction. The spectral/parabolic method marches the optical beam through the density field by solving successively a phase shift (4) and a wave advance (5) operator.

$$j \frac{\partial A}{\partial \zeta} + \pi \left[\left(\frac{n}{n_r} \right)^2 - 1 \right] A = 0 \quad (4)$$

$$\frac{\partial^2 A}{\partial \xi^2} + \frac{\partial^2 A}{\partial \eta^2} + 4\pi j \frac{\partial A}{\partial \zeta} = 0 \quad (5)$$

The final form of the wave advance equation in spectral space is:

$$\hat{A}_{kl}(\zeta + \Delta\zeta) = \hat{A}_{kl}(\zeta) e^{-j \left[\left(\frac{2\pi k}{N_x \Delta\xi} \right)^2 + \left(\frac{2\pi l}{N_y \Delta\eta} \right)^2 \right] \frac{\Delta\zeta}{4\pi}} \quad (6)$$

The phase shift equation is advanced in physical space as:

$$A_{kl}(\zeta + \Delta\zeta) = A_{kl}(\zeta) e^{j\pi \left[\left(\frac{n}{n_r} \right)^2 - 1 \right] \Delta\zeta} \quad (7)$$

The paraxial solver has been validated previously by comparison to accepted propagation code results for a turbulent field, as well as to theory for propagation of a beam through a vacuum. A Gaussian beam may be expressed in non-dimensional form as:

$$A = \frac{A_0}{(1 + j\tilde{\alpha}\zeta)} e^{-\frac{\pi\tilde{\alpha}\tilde{r}^2}{(1 + j\tilde{\alpha}\zeta)}} \quad (8)$$

where

$$\tilde{\alpha} = \frac{1}{\pi\tilde{r}_0^2} + \frac{j}{\tilde{f}} \quad (9)$$

and $\tilde{r}_0 \equiv r_0/\lambda$ and $\tilde{f} = f/\lambda$. We compute the solution on spectral grids with densities of (N_x, N_y) ranging from 8 to 1024. The Gaussian beam parameters are $A_0 = 1$, $\tilde{f} = 20,000$, $\tilde{r}_0 = 10$, $\lambda = 0.1m$, $\tilde{L}_x = \tilde{L}_y = 400$ and $\tilde{L}_z = 1000$. Error norms are computed that compare the magnitudes of the computed and exact solution at the propagation distance noted above, and are shown in Figure 1 plotted against the mesh size $h = L_x/N_x$. As can be seen from the plot, the error norms exhibit spectral convergence. As defined in Boyd [23], spectral convergence (which is synonymous to exponential convergence) results in an error decreasing faster than $\epsilon \approx 1/N_x^k$ for any power k .

We note that although we have taken an operator splitting approach here to solve the paraxial equation, it is a simple matter to replace the operator splitting approach with an unsplit method, and use the spectral model to represent the beam variation in the plane of propagation, and use, say, a Runge-Kutta or other explicit method to march in the propagation direction. An adaptive Runge-Kutta method that uses a compact scheme has been successfully applied for solving the paraxial beam equation[24].

4 Aero-Optics Class Library, Framework and Data Flow

The Aero-Optics capability is built upon a series of public-domain libraries and specially written classes which form a software library. Using this library, a number of standalone applications have been written, including the API used by the CFD models to invoke the software library. The block diagram shown in Figure 3.1 outlines the different components.

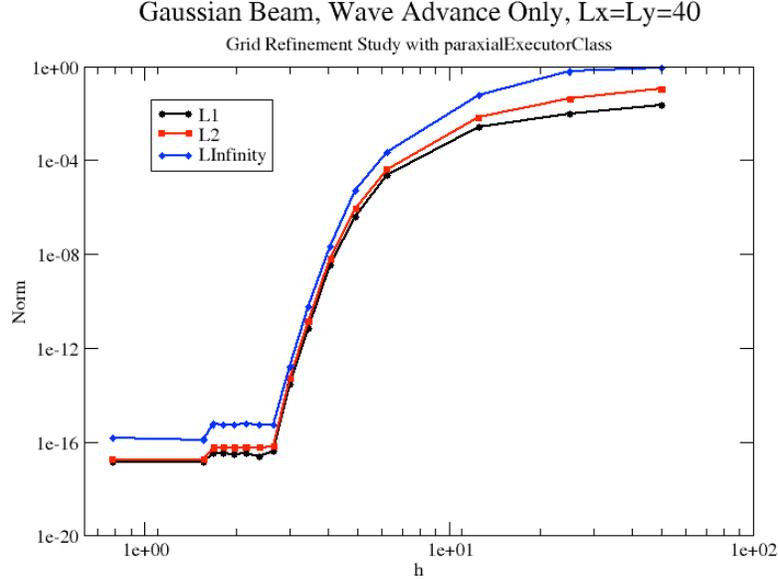


Figure 1: Convergence rate for the Gaussian beam problem.

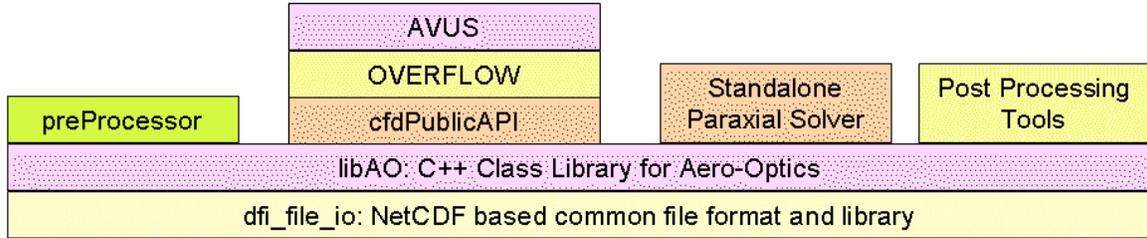


Figure 2: Aero-Optics Class Library Components and Framework

4.1 Library Components

Referring to Fig. 2, the components of the Aero-Optics capability are (from bottom to top):

- df_file_io: This is a C++ library and namespace which uses a specialization of the NetCDF [25] common file format and library to provide a common set of API that are used to read/write named data from/to a commonly accessed file. The data supported include vectors (singly subscripted arrays) of basic data types (unsigned ints, ints, floats, doubles) as well as complex data and strings. Access by Fortran applications is also provided. By allowing direct access to data in a commonly shared file, the software development process has been sped up considerably. Relevant classes in the libAO library have readFromCommonFile and writeToCommonFile functions.
- libAO: This is the C++ class library that contains all the classes and functions used by the Aero-Optics applications. The standalone applications (described below) are all constructed from classes supplied by this library. The paraxial solver invoked by the CFD models, and all the ancillary processing, is also handled by classes that make up this library. A sampling of some of the classes includes:
 - paraxialExecutorClass: This class controls and marshals the individual paraxial solvers, and is invoked by the standalone paraxial solver and the CFD model API.
 - paraxialSolverClass: This class encapsulates the paraxial solver itself.
 - boxGridClass::beamGridClass::paraxInterpolatorClass: This is a set of derived classes which are used to define beam grid extents and provide a fast interpolator to be used by the parabolic solver to find the density at the spectral grid points as the wave is propagated through the beam grid.

- preProcessor: This performs mesh pre-processing for given beam propagation paths and interpolator grid densities to find donor interpolants from the CFD mesh, and store them in the commonly accessed file. Section 5 describes the pre-processing steps and approach.
- cfdPublicAPI: Four simple functions are provided that are invoked by the CFD model. The functions are described fully in Section 6, and permit invocation of the paraxial beam propagator, and store donor density data at the limited set of unique CFD donor points used to define the density on the beam propagator grids.
- Standalone Paraxial Solver: The paraxial beam propagator can be run in a standalone manner, and uses the donor density data stored when the CFD model is run. The input to the standalone paraxial solver is identical to that when it is invoked from within the CFD model, and produces identical data files as well.
- Post Processing Tools: A variety of post processing tools are provided that process propagated beams and density fields and produce aero-optical metrics, waveforms and related information. The data generated by these tools includes:
 - Wavefront Errors: Zernike polynomials are used to remove piston, tip and tilt from the beams, and compute unsteady wave front error wave forms, that are further processed to compute statistical descriptions.
 - Aero-Optical Metrics: Quantities such as jitter, bore sight error and OPD are computed by processing the beams in time.

In addition to the libraries noted above, we make use of a number of public-domain libraries as well, including:

- fftw3: The FFTW3 library[26] is used to perform the Discrete Fourier Transformation (DFT) between physical and spectral space. It is publicly available and provides a very useful set of API for performing DFT related operations.
- boost: The boost C++ library (<http://www.boost.org/>) is a widely used set of classes and libraries and contains a variety of very useful classes that are an addition to the standard namespace C++ libraries.
- std_namespace: The std namespace distributed with the standard C++ language maintains some very useful container classes and related functionality.

4.2 Data Processing Work Flow

The data flow for the Aero-Optics capability is based upon a number of shared, commonly accessed files that are produced at various phases of the processing. The typical processing steps are as follows:

1. Mesh Pre-Processing: Given a CFD grid and definition of the beam interpolator grids (described in Section 5), the mesh pre-processor (preProcessor) finds the unique set of CFD donor data points and all the interpolants and weights to interpolate the CFD solution onto the set of beam interpolator grids. This data is stored, for the N-th beam grid, in a data file (beamGrid.interp.N.nc)
2. CFD Phase: In this phase, the CFD model is run. The CFD code input is modified by the addition of a small namelist (for the OVERFLOW and AVUS codes), which tells the solver when to invoke the beam propagator (if selected), and at what frequency to call it, as well as the name of the common file which defines the propagator input data. The propagator input data file defines the number of paraxial solvers to run, and associates the (pre processed) beam grid interpolator files with the paraxial solvers. This approach allows the re-use of beam grids for different solvers, so that, for example, different waveforms can be run through the same propagation path. The solver writes out, for each beam, the CFD donor density values, for each time step, to a data file (rhoDonor.N.nc). If the propagator is turned on, the propagated beam, for each beam grid, for each time step, is written to a data file (paraxSolver.N.beam). These two CFD model produced files (rhoDonor.N.nc and paraxSolver.N.beam) are used subsequently by the following standalone processors and post-processors.

3. (Optional) Standalone Paraxial Solver: A standalone paraxial solver is available that reads a propagator input file (that defines the beam grids and data needed by each paraxial solver), and for each beam grid, reads and processes the donor density data, propagates the given beam waveform for the prescribed time steps, and deposits the propagated beams into the beam file (paraxSolver.N.beam). This allows one to re-use the computationally expensive CFD data.
4. Post Processing Phase: The post processors (described in Section 8) are used to process given beam files (ie; propagated waveforms) and produce aero-optical quality metrics and descriptions. The post processors operate on given propagator input files (again, defining beam grids, interpolator data and rhoDonor data), and produce the metrics.

The following sections describe the different components in more detail.

5 Mesh Pre-Processor

In order to make the paraxial solver computationally efficient, it must be able to quickly compute the index of refraction on the integration points of the spectral grid planes as they are marched in the propagation direction. The index of refraction is directly related to the density via the Gladstone-Dale constant, so this means that a fast interpolation method is needed to interpolate the density from the CFD grid onto the spectral grid as it is marched in the propagation direction. This is accomplished by a pre-processing step that finds the CFD mesh donor interpolants (donor cells and weights) on a Cartesian grid that lies in the beam propagation path. This Cartesian grid is aligned in space with what is termed a beam grid. The Cartesian grid itself, upon which the CFD donor data and interpolants are precomputed is termed an interpolator grid. The paraxial solver marches in the beam propagation direction, and needs the index of refraction at the points that are termed the spectral grids. The following diagrams explain the relationship between beam grids, interpolator grids and spectral grids.



Figure 3: Illustration of beam grids, interpolator grids and spectral grids

The preprocessor processes a given set of interpolator grids for a particular CFD mesh and finds, for every point on the interpolator grid, the nearest CFD grid element (nodes for structured grids, cells for unstructured grids). For each of these elements a corresponding stencil is found dependent upon the particular flow solver type (ie; structured overset, or unstructured), and the weights needed to interpolate the CFD solution onto the interpolator grid point are computed. Once all the donor points (ie: CFD grid points that participate in the interpolation) are found, a set of unique donor points is created. This set of unique donor points is what needs to be gathered by the particular CFD solver, and is passed to the paraxial beam model during the time-varying CFD computations. Since this set of unique donor points is a significantly smaller subset of the entire CFD grid, we provide the ability to store the density at these points during the transient CFD solution, which form the rhoDonor.N.nc files. The pre-processor uses a modification of an ADT tree[27] that was supplied to us by Ralph Noack at PSU. This space partitioning tree permits very fast nearest neighbor searches that are needed to find the donor grid points for the interpolator.

For either structured overset or unstructured grids, the interpolation from the CFD grid to the interpolator grid is represented as a sum:

$$\rho_i = \sum_{n=1}^N \hat{\omega} \rho_n \quad (10)$$

Structured overset grid weights are found using a hexahedral, iso-parametric representation in local, natural coordinates. For a given interpolator point, the natural coordinates are found by solving, via Newton iteration:

$$\begin{aligned} x &= \sum_n \phi_n(\xi, \eta, \zeta) x_n \\ y &= \sum_n \phi_n(\xi, \eta, \zeta) y_n \\ z &= \sum_n \phi_n(\xi, \eta, \zeta) z_n \end{aligned} \quad (11)$$

Then, the interpolation weights are readily found from the iso-parametric interpolation basis functions.

Unstructured grid interpolator weights are found by first finding the grid nodes that make up the given nearest cell to the interpolator point (which is found by the ADT nearest neighbor search) using cell-to-node connectivity. This cloud of points is then used in a linearity preserving reconstruction scheme to find the grid node weights in the donor value reconstruction. For the interpolations used here, we find the weights using the linearity preserving Laplacian scheme [28] which finds the weights so that an interpolation operator,

$$L(f) = \sum_n \omega_n (f - f_n) = 0 \quad (12)$$

is exactly satisfied when f is a linear function of the coordinates. That is, find the Lagrange multipliers in is exactly satisfied when f is a linear function of the coordinates. That is, find the Lagrange multipliers λ_n in

$$\omega_n = 1 + \lambda_x (x - x_n) + \lambda_y (y - y_n) + \lambda_z (z - z_n) \quad (13)$$

so that

$$L(x) = L(y) = L(z) = 0 \quad (14)$$

The resulting 3×3 system of equations is inverted, which then gives the Lagrange multipliers, which then yield the interpolator weights.

Figure 4 shows an example CFD grid with an interpolator grid, and the interpolated solution overlaid with the CFD grid solution for an OVERFLOW and AVUS solution for the unsteady flow over a circular cylinder.

6 Application Programming Interface to CFD Models

The API supplied by the Aero-Optics library to be called by the CFD models has been made as minimal as possible. There are a total of four API called by CFD model, and are represented by the following pseudo-code:

```
initializeParaxialBeamSolvers(commonFormatFileName);
for each timestep in timesteps
  for each beamNum in beams
    setRhoDonor(beamNum, rhoDonor, timestep);
    if(paraxSolve) runParaxialBeamSolver(beamNum, timestep);
finalizeParaxialBeamSolvers();
```

- **InitializeBeamSolvers** instantiates the paraxial solver executor and instructs it to load all its data from the supplied common file named commonFormatFileName.

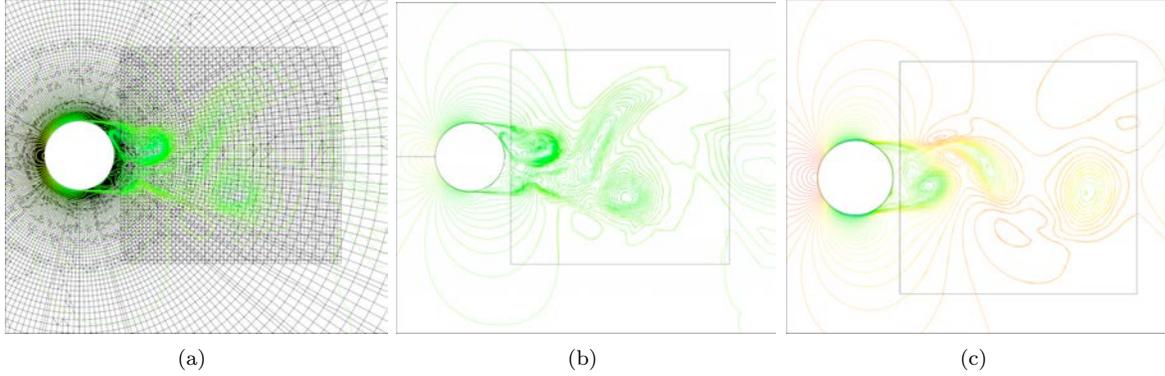


Figure 4: Streamwise aligned cut through the CFD grid and interpolator grid (left), OVERFLOW solution and interpolated density (middle) and AVUS solution and interpolated density (right) at a particular time step of an unsteady DES calculation

- *setRhoDonor transfers* the CFD supplied donor density values to the interpolator grid, which then loads up the interpolator with the density data using the precomputed interpolants. The donor density data is appended to a local NetCDF file
- *runParaxialBeamSolver* runs the paraxial beam solver at this time step for this beam using parameters specified in the common file.
- *finalizeParaxialBeamSolvers* performs all closeout/finalization for the solvers.

We have integrated the paraxial solver capability within both OVERFLOW and AVUS. Both solvers are modified to read in a small namelist that defines whether or not the paraxial model is to be invoked, whether or not to run the beam propagator, how often to invoke the model, and the name of the paraxial model common file. A Fortran90 module is created for each solver to store this data, as well as the donor mesh connectivity data that is read from the pre-processor produced files. Specific to each solver, a density gather operation has been written to gather and store only the donor density data, which is passed to the model through the setRhoDonor API. For the AVUS code, a specialization of the procedure that interpolates data from cell centers to nodes has been made to perform this density gather. Integration within both CFD models has been straightforward, and lends itself to integration in other models as needed.

7 Standalone Paraxial Beam Model

The paraxial library software framework has allowed the straightforward development of a standalone paraxial beam solver. The standalone paraxial beam solver is designed to use data generated from the CFD model in the form of the donor density files, and produces identical beam files that are the beams propagated through the unsteady flow field. By following the same data flow model, it is straightforward to reuse the computationally expensive, transient, CFD data for other applications as well. The potential applications include:

- Integration within Far-Field Models: We see this as being directly integrated within engineering level models which may be used to evaluate the effect of the near field aero-optics upon the full scale engineering system. Potential models include the WaveTrain model produced by MZA Associates.
- Multiple-Path Propagation: The standalone model allows one to propagate a beam out of the near field, perform a modification to this beam (through, for instance, an engineering model as noted above) and then propagate it back through the near-field, and measure its beam quality.
- Multiple Wave Forms and Wavelengths: The standalone model allows many different wave forms at different wavelengths to be modeled through the same CFD generated flow field.

- **Evaluation of Beam Propagator Methods:** It is a relatively simple matter to replace the factored paraxial beam propagator developed here with an unsplit method, such as an adaptive RK method, similar to that shown in Reference [24], and use the standalone approach to evaluate and compare the different solution strategies.

8 Post-Processing Tools and Models

A variety of post-processing tools have been developed using the software library, and are designed to post-process given waveforms and density fields to produce aero-optical quality metrics. For given waveforms, the beam jitter and boresight error are constructed for each timestep, and output to ASCII files for plotting and processing. In order to compare with experimental data, wavefront error is computed using Zernike polynomials and statistical representations of the propagated beams are also made available. As an alternative to using propagated beams to characterize the aero-optical field, the Optical Path Difference (OPD) computed using a purely geometric integral approach is also made available, and these phase errors may also be statistically processed by computing wavefront error. The sections below describe the approaches taken for each of the methods.

8.1 Jitter and Boresight Error

Using beams contained in the “paraxialSolver.N.beam files, beam jitter and boresight error are calculated using beam power-weighted integrals as follows. For each stored time step, the beam centroid is found by using a power-weighted area integral. For a general function defined in the imaging plane, we define a generic power-weighted area integral to be:

$$I(f) = \int_x \int_y AA^* f(x, y) dx dy \quad (15)$$

where AA^* is the beam power found by multiplying the beam magnitude by its complex conjugate. Then we find, at the n^{th} time slice, the centroids to be:

$$x_n = \frac{I(x)}{I(1)}, \quad y_n = \frac{I(y)}{I(1)} \quad (16)$$

The average beam centroid is found from the beam centroids and the N time intervals as:

$$\bar{x}_n = \frac{1}{N} \sum_{n=1}^N x_n, \quad \bar{y}_n = \frac{1}{N} \sum_{n=1}^N y_n \quad (17)$$

We define jitter as the root mean square of the time-varying beam centroid about the average beam centroid:

$$\sigma_x = \sqrt{\frac{1}{N} \sum_{n=1}^N (x_n - \bar{x}_n)^2}, \quad \sigma_y = \sqrt{\frac{1}{N} \sum_{n=1}^N (y_n - \bar{y}_n)^2} \quad (18)$$

The angular deviation over the propagation length is:

$$\theta_x = \sin^{-1} \left(\frac{\sigma_x}{L} \right), \quad \theta_y = \sin^{-1} \left(\frac{\sigma_y}{L} \right) \quad (19)$$

We define boresight error using the difference between the beam centroid and image center. The translational error is:

$$\epsilon_{x,n} = x_n - x_{0,image}, \quad \epsilon_{y,n} = y_n - y_{0,image} \quad (20)$$

where the boresight error is

$$\beta_n = \sqrt{\epsilon_{x,n}^2 + \epsilon_{y,n}^2} \quad (21)$$

8.2 Wavefront Error Processing

We provide the ability to compute the wavefront error for each beam, and then to statistically process these wavefront errors. The approach taken here to produce the wavefront error is to compute the phase of the beam, and then using Zernike polynomials compute and subtract from the phase the piston, tip and tilt. Zernike polynomials are a set of orthogonal polynomials defined on a unit disk that are often used in optics to characterize waveforms in a consistent manner, of which the piston, tip and tilt are the first three polynomials. We use ANSI standard Zernike defined for the n-th radial and m-th azimuthal order as:

$$Z_n^m(\rho, \theta) = \begin{cases} N_n^m R_n^{|m|}(\rho) \cos(m\theta); m \geq 0 \\ -N_n^m R_n^{|m|}(\rho) \sin(m\theta); m < 0 \end{cases} \quad (22)$$

where

$$R_n^{|m|} = \sum_{s=0}^{(n-|m|)/2} \frac{(-1)^s (n-s)!}{s! [(n+|m|)/2 - s!] [(n-|m|)/2 - s!]} \rho^{n-2s} \quad (23)$$

and

$$N_n^m = \sqrt{\frac{2n+2}{1+\delta_{m0}}} \quad (24)$$

For a given wave form, the wavefront error is computed by the following:

1. Compute a no flow beam phase (ϕ_∞) by propagating the beam through the specified propagation distance using a constant density field, set at the freestream density value.
2. Compute the difference between the n-th timesteps beam phase and the no flow phase: $\phi_1 = \phi - \phi_\infty$
3. Compute the piston (ϕ_p), tip (ϕ_{tip}) and tilt (ϕ_{tilt}) using the orthogonality properties of the Zernike polynomials, and subtract these from the phase difference: $\phi_2 = \phi_1 - (\phi_p + \phi_{tip} + \phi_{tilt})$
4. Find the mean phase error: $\bar{\phi}_2 = \langle \phi_2 \rangle$
5. Find the n-th timesteps wave front error by subtracting the mean from the phase error: $\tilde{\phi} = \phi_2 - \bar{\phi}_2$

This wave front error is then statistically processed by computing a mean and the root mean square deviation from the mean. The procedure described above requires the definition of an aperture mask, since the Zernike polynomials are defined on a unit radius disk.

9 Optical Path Difference-based Wavefront Error Processing

Here, instead of propagating a beam through the beam grid, we integrate the density along the propagation direction, generating the Optical Path Difference (OPD). The OPD is defined as:

$$OPD = K_{gd}\lambda \int_0^{\tilde{L}_z} \rho d\zeta \quad (25)$$

The OPD is related to the phase shift since

$$\Delta\tilde{\phi} \approx 2\pi K_{gd} \int_0^{\tilde{L}_z} \rho d\zeta = \frac{2\pi}{\lambda} OPD \quad (26)$$

The same wavefront error processing shown above is used to compute a statistical description of the OPD-based wavefront error.

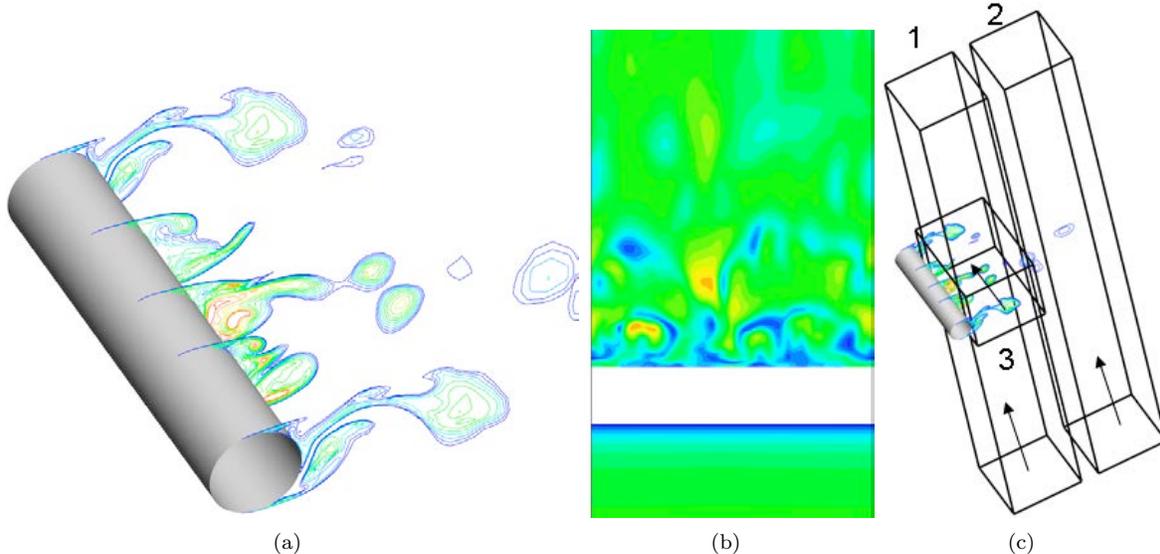


Figure 5: Unsteady flow past a circular cylinder. Entropy measure contours (left), Mach contours in a spanwise plane (middle), locations and directions of beam propagator grids (right).

10 Demonstration: Unsteady Flow Past a Circular Cylinder

The paraxial solver is first demonstrated for the unsteady flow past a circular cylinder using both the OVERFLOW and AVUS CFD models. The cylinder diameter is 2 inches and the free stream Mach number is 0.2 with static conditions corresponding to 1 atmosphere at 288 degrees K. The time step is chosen to be $\Delta t = 8.5 \times 10^{-6}$ seconds which resolves one shedding period with 200 time steps when using an estimate of the shedding frequency based upon a Strouhal number of 0.22. The cylinder span is taken to be 10 inches. Both models use the same mesh, which is a structured, single domain grid of $101 \times 151 \times 201$ grid points in the spanwise, radial and azimuthal directions. For the AVUS code, a mesh processor was written that converts the structured grid into AVUS input format. The mesh normal to the wall is clustered to give the first cell center a value in log-normal coordinates of $y^+ \approx 30$ which is adequate for both models using wall functions. Both codes were run with the Spalart-Allmaras Detached Eddy Simulation [29] turbulence model for 16,000 time steps, followed by 1,000 time steps with the paraxial model activated. OVERFLOW was run using a dual time stepping scheme, the HLLC approximate Riemann solver and third-order upwinding with the default limiter. AVUS was run with dual time stepping, a least-squares reconstruction scheme, the HLLC approximate Riemann solver and limiting turned off. The paraxial model recorded the donor density at 5 time step intervals, which was subsequently used by the post processing models to compute various aero-optical beams and beam metrics. The beam grids shown on the right of Fig. 5 shows that the first two grids are oriented so that the beam shines upwards, through the shear layer sheet, while the third beam grid shines along the spanwise direction, just aft of the cylinder. The interpolator grids associated with each beam grid were $61 \times 61 \times 61$, from which the pre-processing found 63,895, 15,440 and 277,419 donor points. The pre-processor took 71 and 99 seconds to process the overset and unstructured grid systems, respectively, to find the donor points and interpolator weights.

Using the donor density deposited every 5 time steps, top hat beam profiles of roughly 5 inch diameter were propagated through the beam grids for wavelength $\lambda = 1 \mu m$ on spectral grids of 256×256 using the standalone propagator. Figure 6 shows the wave front errors computed from the propagated beams using the procedure outlined in Section 8.2 for a mask with a diameter of 4.5 inches.

An example of post processing the beams for aero-optical propagation metrics is shown in Fig. 7, where we compare the computed jitter for the three beam grids for the OVERFLOW and AVUS computations. It is interesting to note that the OVERFLOW model gives results that appear to have a finer turbulent scale than AVUS. This is evident from the jitter shown in Fig. 7, where the levels of jitter from OVERFLOW are higher than those from AVUS. Examining the wave front error from beam grid 1 in Fig. 6 shows that

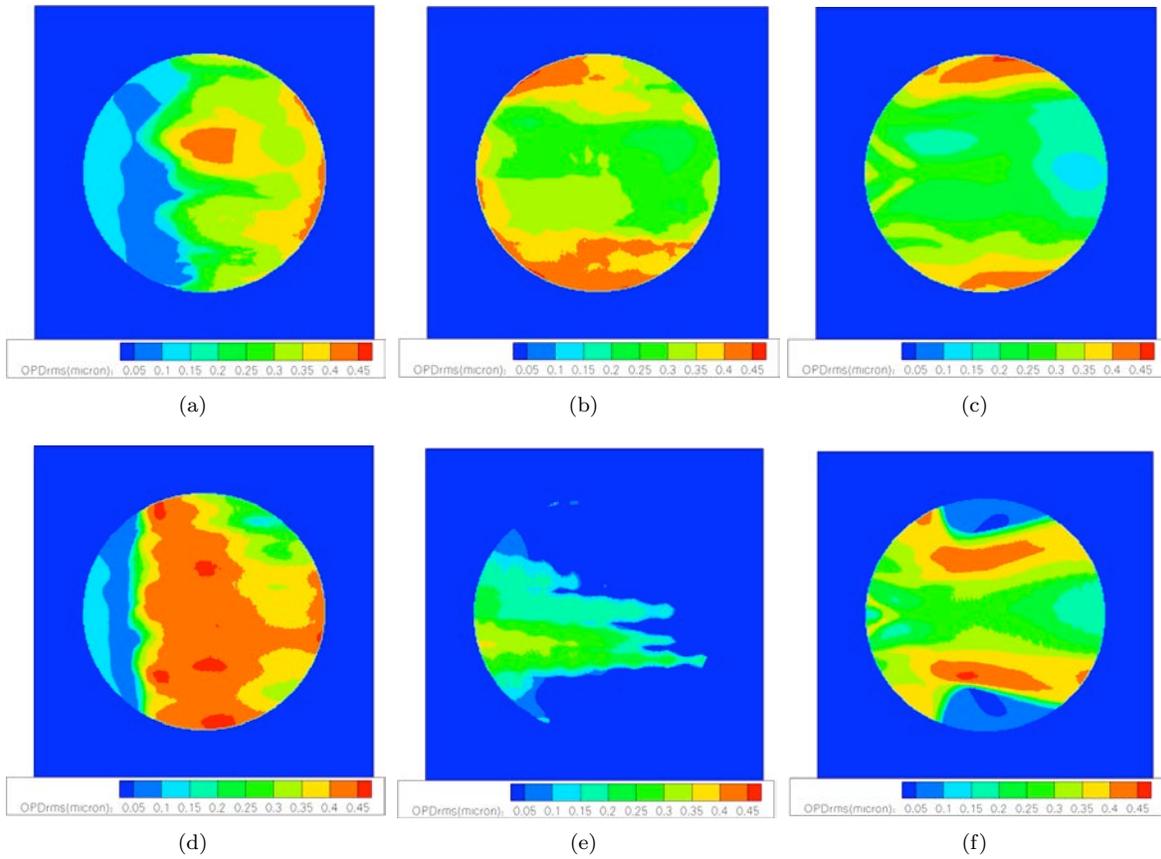


Figure 6: Comparison of Wave Front Errors for OVERFLOW (top) and AVUS (bottom) on beam grids 1, 2 and 3 (across). Shown are computed rms values of the wavefront error in microns

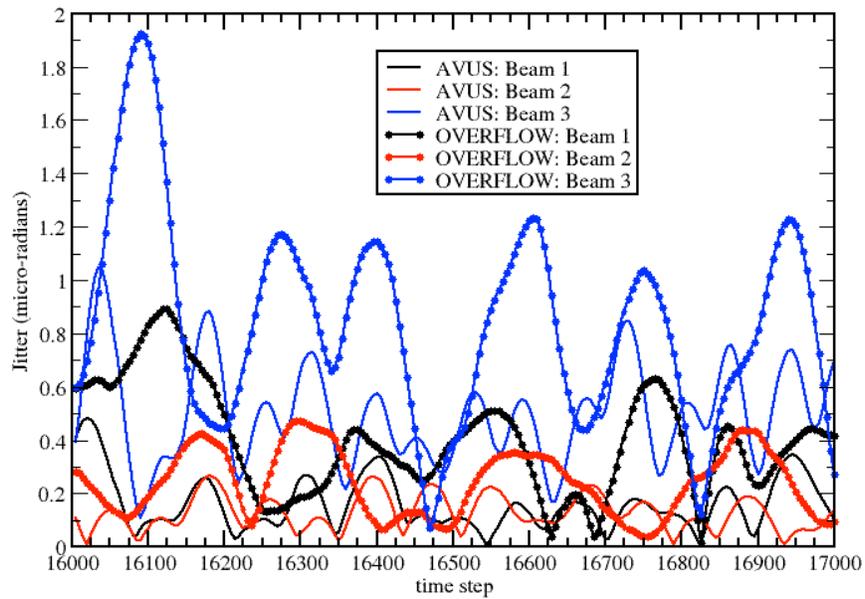


Figure 7: Comparison of jitter for the three beam grids between OVERFLOW and AVUS.

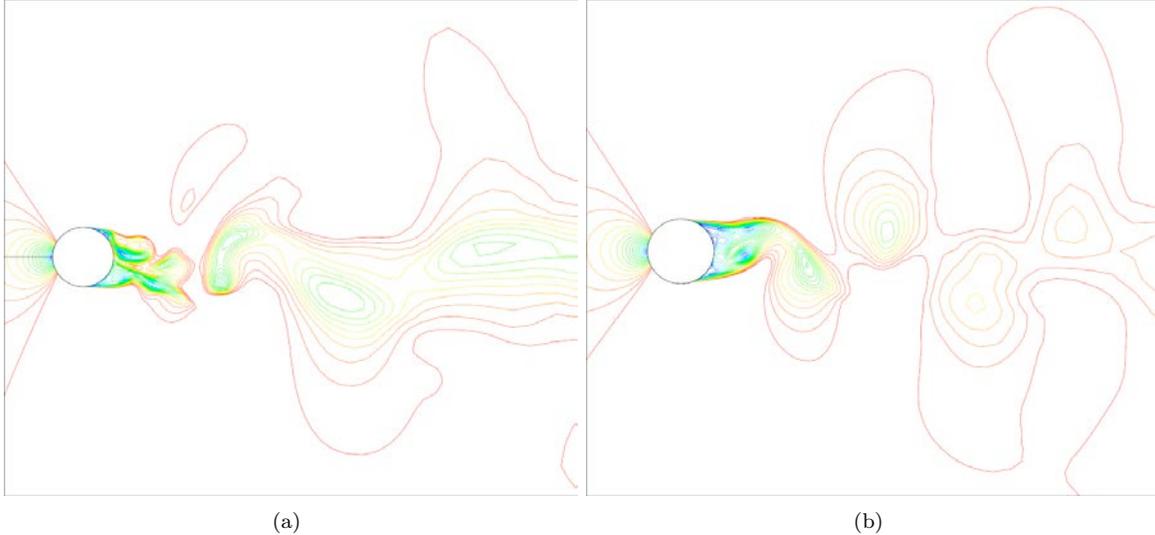


Figure 8: Comparison of Mach contours mid-span for OVERFLOW (left) and AVUS (right). Both contours made with same range and levels of Mach number (21 levels from 0 to 0.2).

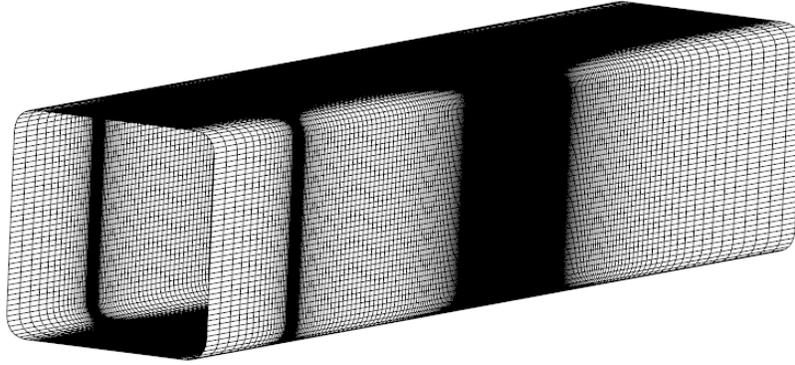
both codes predict spanwise turbulent structures, but those from the AVUS produce higher wave front error near the cylinder than the OVERFLOW model. These get dissipated more quickly downstream at beam grid 2 from AVUS than from OVERFLOW, while the third beam grid, which shines along a spanwise beam next to the cylinder, shows a much more compact/necked structure from AVUS. We speculate that this may be caused by differences in the implementation of the HLLC scheme, as well as differences in the numerics related to the turbulence models. Shown in Fig. 8 are Mach contours mid-span of the cylinder at the same time step from OVERFLOW and AVUS. The AVUS results show a more regular vortex street than the OVERFLOW results.

11 Validation: 1 Foot Conformal, Hemispherical Turret

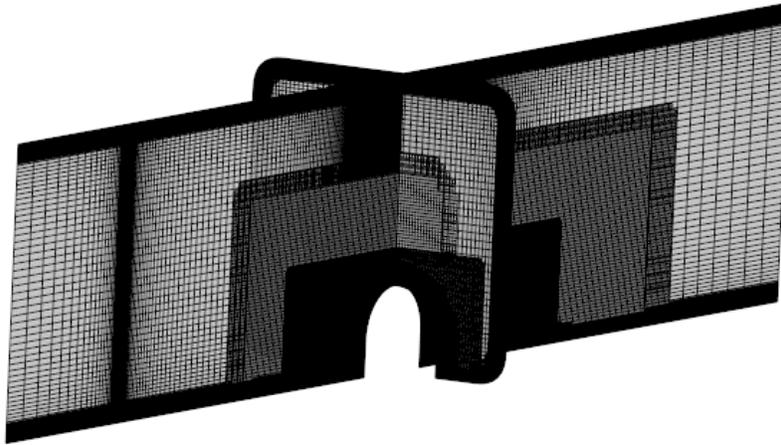
The combined aero-optics CFD modeling capability has been validated by comparison to both aerodynamic and aero-optic data taken in a wind tunnel test in the Air Force Academy 3ft. x 3ft. subsonic wind tunnel [13, 30]. A 1 foot diameter hemispherical turret was mounted in this tunnel on a 4.5 inch high cylinder, which itself is mounted to a wall of the wind tunnel. Details behind the experiment can be found in References [13, 19, 30], which has proven the basis for a number of CFD model validation efforts, including those shown in References [31, 14] and others. The data provided from the experiment includes:

- Aerodynamic Data:
 - Hot Wire Measurements: Time-averaged streamwise velocity and streamwise velocity fluctuations at five traverse locations: One upstream of the turret and 4 downstream, in the wake.
 - Surface Pressure Measurements: Pressure coefficients along the hemisphere surface on a curve aligned with the free stream.
- Aero-Optic:
 - Malley Probe
 - 2-D Shack-Hartmann Wavefront sensor
 - 8x8 High-Bandwidth Wavefront sensor

For the comparisons shown here, we focus upon the aerodynamic data and the data taken from the 2-D Shack-Hartmann Wavefront sensor for the $M = 0.4$ test.

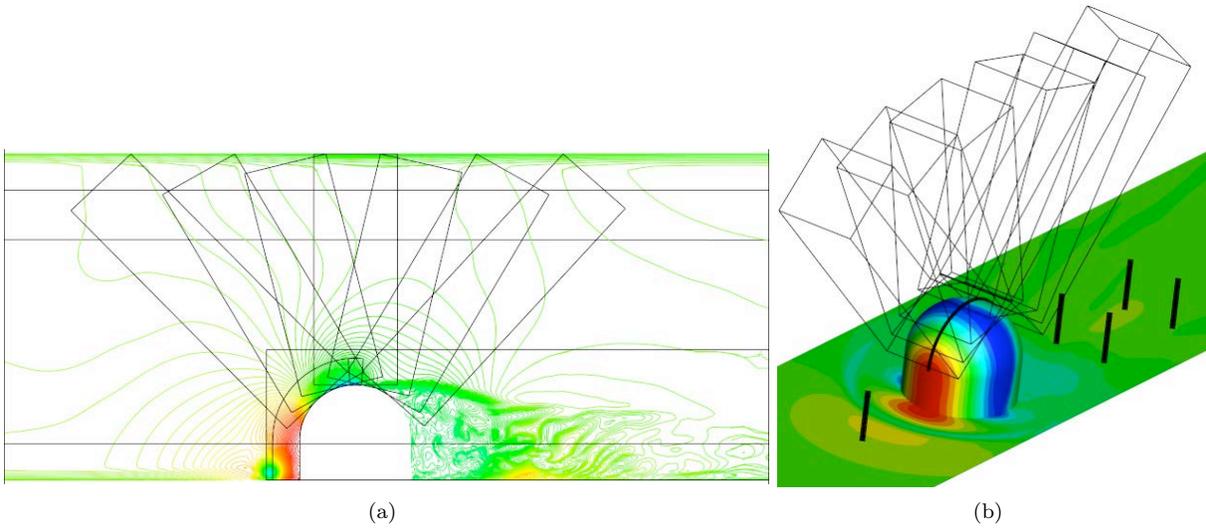


(a)



(b)

Figure 9: Grid system showing the upstream fetch, wall clustering and nested wake grids.



(a)

(b)

Figure 10: Beam grids and hot wire probe traverse locations in the CFD model

An overset mesh system was developed with six grids that models the wind tunnel test section, including the upstream fetch of the incoming boundary layer to the test section. The mesh along the tunnel walls and turret walls was clustered to have an initial spacing so that the first cell is located at a y^+ value of approximately 1, with the first four layers at this same spacing, followed by a growth out to $y = 0.2$ feet with a stretching ratio of 1.15. Two, nested, Cartesian grids are clustered over the turret, and are shown in Fig. 9. The total mesh has approximately 14×10^6 nodes.

To compare to the experimental aero-optic data, six beam grids are constructed to lie at elevation angles of 60, 76, 90, 103, 120 and 132 degrees, and are shown in Fig. 10, along with the locations of the hot wire traverses and surface pressure measurement locations. Each beam grid was used to define interpolator grids with dimensions $61 \times 61 \times 101$, which were preprocessed to produce the donor connectivity data and interpolators. Using a modified Strouhal number, the shedding frequency of the turret was estimated, and based upon a single period, the time step was chosen so that 200 steps are taken for each period. OVERFLOW was run with the SST-DES model, using dual-time stepping, with the third-order upwinding option with default limiter, without wall functions, for a total of 12,500 steps. The collection of time-averaged statistics began at 7,500 steps, and the paraxial model was invoked during the last 1,000 steps, depositing donor density every five time steps. The time-averaged data collected by OVERFLOW during the DES calculation is used to compute the averaged data shown in Fig. 11 where only the resolved turbulence is used to compare to the velocity fluctuations. The traverse locations are numbered so that location 1 is upstream, 3 and 5 are located directly downstream of the turret, and 2 and 4 are offset.

Inspection of the data in Fig. 11 shows that the surface pressure on the turret is adequately predicted, although the incoming turbulent boundary layer is too thin, and the turbulence levels are underpredicted. As shown in the data for probes 2 and 4, the turbulence levels off-centerline are overpredicted, with accompanying lower momentum regions near the wall when compared to the experimental data. Along the cylinder centerline, the opposite behavior is shown, with the CFD model under-predicting the turbulent fluctuations, and over-predicting the velocity profiles. This could be due to the low levels of turbulence being imposed on the inflow to the turret and warrants future investigation.

The collected donor density data was then used to propagate beam waveforms through the beam grids, on spectral grids of 256×256 dimensions. Top hat profiles of beams with wavelengths of $\lambda = 1\mu\text{m}$ were propagated for each of the 200 collected time steps, and wave front errors computed and statistically analyzed. In order to compare the statistical data, both the experimental waveforms and the paraxial beam propagator generated waveforms use the processing noted in Section 8.2, from which the mean rms and standard deviations of the rms of the waveforms are found (as noted in Reference [30]) as:

$$OPD_{rms} = \sqrt{\langle E(\phi^2) - E(\phi)^2 \rangle} \quad (27)$$

The computed wave front error ϕ is used, and the spatial operator, E , is defined as:

$$E(f(r, \theta, t)) = \frac{\int_r \int_\theta f r dr d\theta}{\int_r \int_\theta r dr d\theta} \quad (28)$$

Here, r varies over the aperture mask, taken to be 4.5 inches in diameter. This procedure is used on the experimental data contained in the Wave Front Sensor data files accompanying the experimental data set, as well as the beam produced phase differences found from the beam propagator model, and produces data similar shown in Reference [30]. Figure 12 compares the measured and predicted statistics by plotting mean wave front error root mean square values with the standard deviation added and subtracted to the mean.

What is noted from the comparison is:

- The mean wave front errors appear to be well predicted, except for the last beam at 132 degrees.
- The mean wave front error deviation is underpredicted at the 60 and 76 degree elevations (upstream), but well predicted at the other elevations.

This behavior can be attributed to the underprediction of the turbulence levels upstream. Other contributing factors could be sub-iterative convergence, choice of upwind schemes, limiters and turbulence model related issues. These could be investigated in subsequent studies.

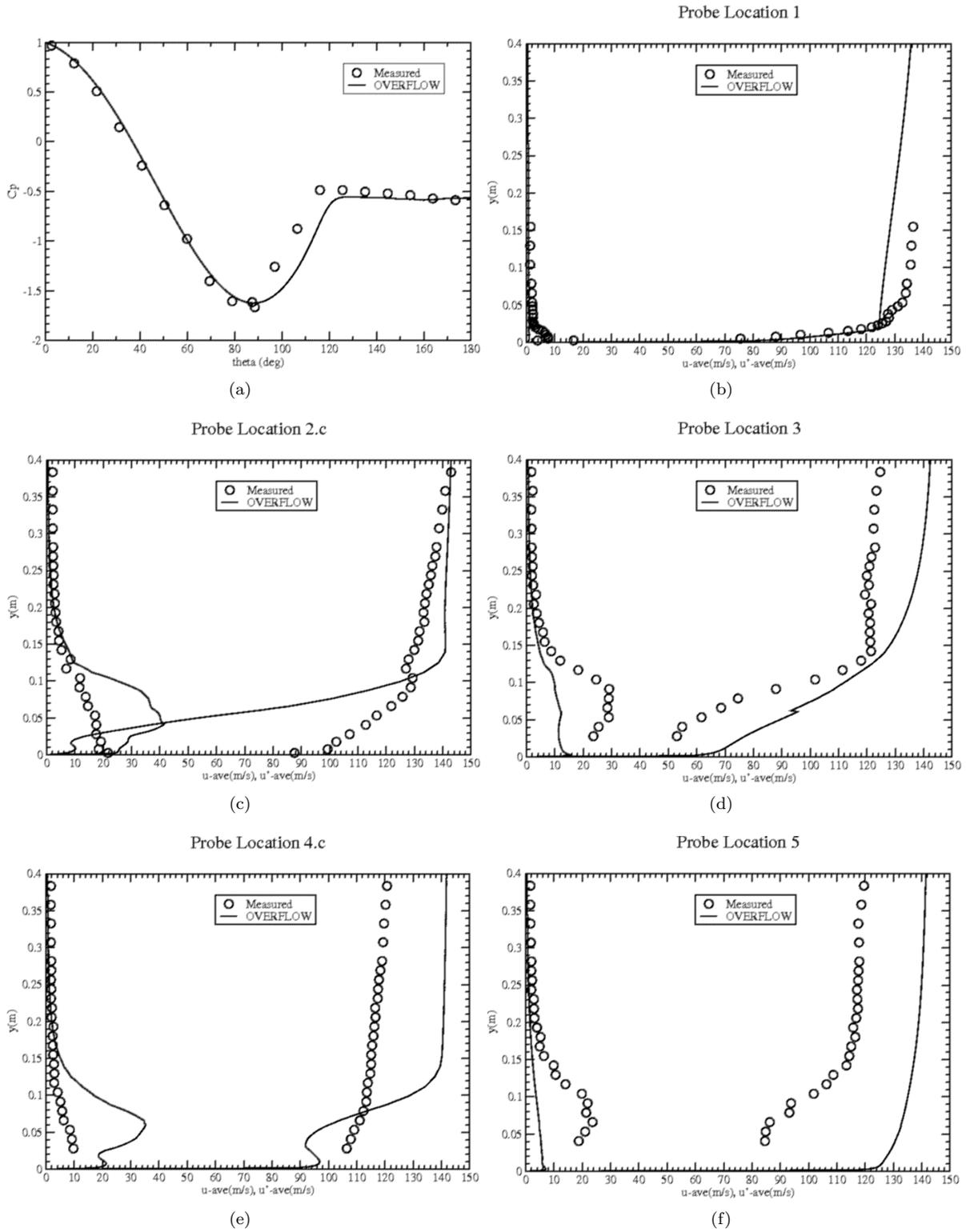


Figure 11: Centerline surface pressure comparison (upper left) and averaged streamwise velocity and streamwise velocity fluctuations at the traverse locations

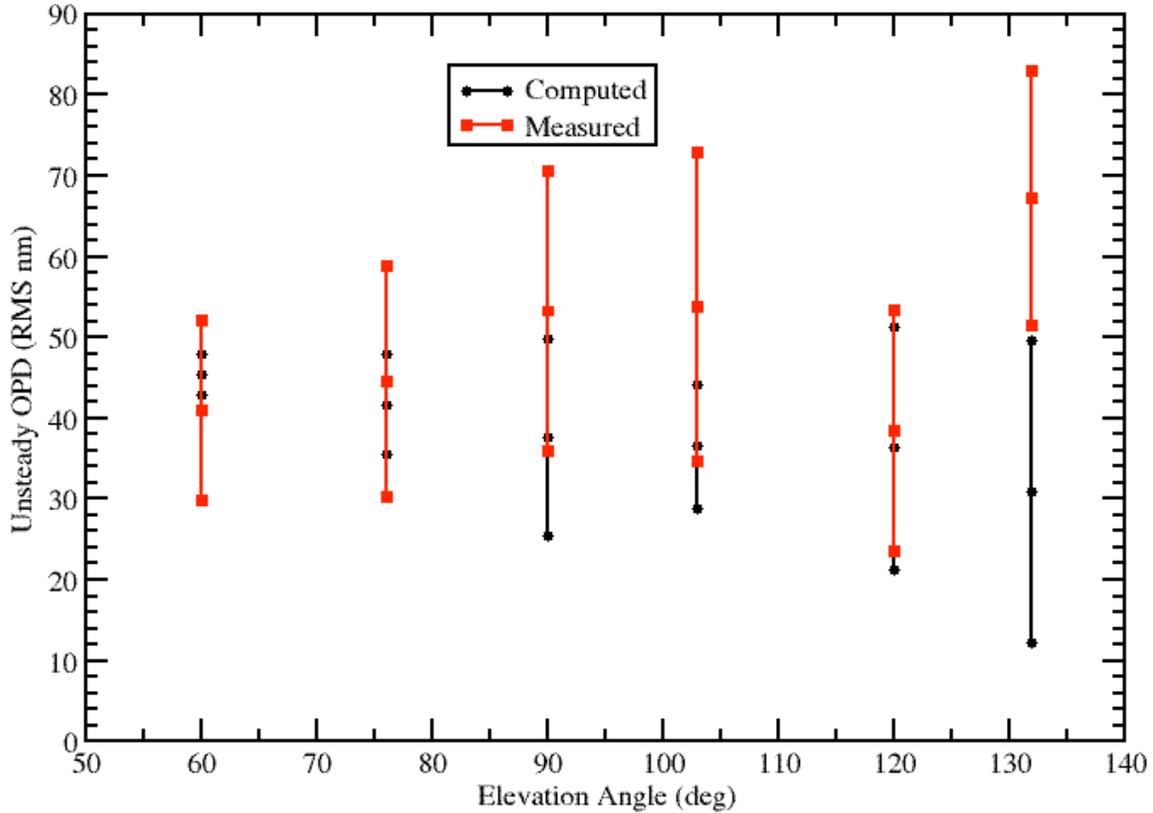


Figure 12: Comparison of measured to predicted wave front distortions. Shown are the mean values +/- the standard deviations

12 Conclusions and Recommendations

An object-oriented software library has been developed that is used to integrate a Paraxial Beam solver within both an overset (OVERFLOW) and an unstructured (AVUS) CFD model. The Paraxial Beam equation is a parabolized form of Maxwells equations, and is used to propagate optical waveforms through the turbulent, unsteady flow fields produced by the structured overset and unstructured solvers. Previous work, where a tightly coupled, procedures based integration was made with OVERFLOW, identified a number of deficiencies, which have been addressed with this work. To achieve this new capability, an object-oriented, software library approach has been undertaken and has been used to develop the following components; a standalone mesh preprocessor, an easily integrated Paraxial Beam solver, a standalone Paraxial Beam solver and an assortment of post-processing tools. The new capability has been integrated within the OVERFLOW and AVUS CFD solvers, and demonstrated for evaluating the Aero-Optical propagation through unsteady, turbulent flows past a circular cylinder and validated by comparison to aerodynamic and aero-optical data for a conformal, hemispherical turret.

Based upon our work, we recommend the following areas to be investigated:

- Evaluate CFD Model Accuracy and Resolution Requirements. For the wavelengths under consideration, it is critical to capture the small scale turbulent structures in order to provide accurate near-field propagation characteristics. A compact-scheme based CFD code, such as FDL3DI, could be needed to model this behavior properly, but until a systematic study is conducted, this is purely speculative. We recommend performing such a study by first integrating the paraxial model into FDL3DI, and comparing canonical flows with various turbulence scales produced with varying levels of discretization accuracy in order to ascertain these effects. The hemispherical turret cases (both flat windowed¹⁷ and conformal¹⁶) can provide experimental data on which to base the comparisons as well.

- Evaluate Paraxial Model Accuracy and Resolution Requirements: The effect of paraxial model resolution (marching step size and spectral grid dimensions) must be understood to characterize the effect of the paraxial beam solution accuracy upon the predicted near-field aero-optical distortions. Implementation of a non-factorized scheme should be investigated, and where possible, techniques such as the Method of Manufactured Solutions, as shown in reference 27, should be used to formally characterize the accuracy of the beam propagation through non-vacuum fields.
- Integration with Engineering Scale Propagation Models: The true effect of near-field optical distortions can only be measured in terms of their impact upon the total system performance. This is most readily achievable by integrating the results from near field CFD produced aero-optical distortions into a large-scale engineering propagation model which models the entire end-to-end process. The approach we have taken here, by storing donor density data in commonly accessed files, and providing for standalone paraxial beam propagators, forms the basis for such an integration. We recommend pursuing this integration with an appropriate engineering scale propagation modeling system such as WaveTrain by MZA Associates.
- Further Application to Large-Scale Problems: We anticipate applying this new modeling capability to aircraft and systems of interest to the Air Force and other agencies.

13. References

- [1] Sherer, S. E., “Aero-Optics Code Development: Experimental Database and AVUS Code Improvements,” Technical Report AFRL-RB-WP-TR-2009-3076, Air Vehicles Directorate, Air Force Research Laboratory, March 2009.
- [2] Weeks, T. and Coirier, W., “Aeronautical Sciences and Advanced Studies Program: Aero-Optics Analysis for Relevant Air Force Configurations,” Technical Report AFRL-RB-WP-TR-2009-3182, Air Vehicles Directorate, Air Force Research Laboratory, July 2009.
- [3] Sutton, G., “On Optical Imaging Through Aircraft Turbulent Boundary Layers,” AIAA Paper 80-1413.
- [4] Sutton, G., “Aero-optical Foundations and Applications,” *AIAA J.*, Vol. 23, No. 10, pp. 1525–1537.
- [5] Siegenthaler, J., Jumper, E. J., and Gordeyev, S., “Atmospheric Propagation vs. Aero-Optics,” AIAA Paper 2008-1076.
- [6] Clark, R., Banish, M., and Hammer, J., “Fundamentals of Aero-Optics Phenomena,” AIAA Paper 94-2545.
- [7] Tsai, Y. and Christiansen, W., “Two-Dimensional Numerical Simulation of Shear-Layer Optics,” *AIAA J.*, Vol. 28, No. 12, pp. 2092–2097.
- [8] Jones, M. and Bender, E., “CFD-Based Computer Simulation of Optical Turbulence Through Aircraft Flowfields and Wakes,” AIAA Paper 2001-2798.
- [9] Sinha, N., Arunajatesan, S., Seiner, J., and Ukeiley, L., “Large Eddy Simulation of Aero-Optics Flowfields and Control Application,” AIAA Paper 2004-2448.
- [10] Mani, A., Wang, M., and Moin, P., “Computational Study of Aero-Optical Distortion by Turbulent Wake,” AIAA Paper 2005-4655.
- [11] Mani, A., Wang, M., and Moin, P., “Resolution requirements for aero-optical simulations,” *Journal Comp. Phys.*, Vol. 227, 2008, pp. 9008–9020.
- [12] Visbal, M. and Rizzetta, D., “Effect of Flow Excitation on Aero-Optical Aberration,” AIAA Paper 2008-1074.
- [13] Cenicerros, J., Nahrstedt, D., and Hsia, Y.-C., “Wind Tunnel Validation of a CFD-Based Aero-Optics Model,” AIAA Paper 2007-4011.
- [14] Ladd, J., Mani, M., and Bower, W., “Validation of Aerodynamic and Optical Computations for the Flow about a Cylindrical/Hemispherical Turret,” AIAA Paper 2009-4118.
- [15] Pond, J. and Sutton, G., “Aero-Optics Performance of an Aircraft Forward-Facing Optical Turret,” *AIAA J. of Aircraft*, Vol. 43, No. 3, 2006, pp. 600–607.
- [16] Smith, R., Truman, C., and Masson, B., “Prediction of Optical Phase Degradation Using a Turbulence Transport Equation for the Variance of Index-of-Refractive Fluctuations,” AIAA Paper 90-0250.
- [17] Truman, C. and Lee, M., “Effects of organized turbulence structures on the phase distortion in a coherent optical beam propagating through a turbulent shear flow,” *Phys. Fluids A*, Vol. 2, No. 5, 1990.
- [18] Gordeyev, S., Jumper, E., Ng, T., and Cain, A., “Aero-Optical Characteristics of Compressible, Subsonic Turbulent Boundary Layers,” AIAA Paper 2003-3606.
- [19] Gordeyev, S., Post, M., and Mclaughlin, T. e. a., “Aero-Optical Environment Around a Conformal-Window Turret,” *AIAA J.*, Vol. 45, No. 7, 2007, pp. 1514–1524.
- [20] Gordeyev, S., Hayden, T., and Jumper, E., “Aero-Optical and Flow Measurements Over a Flat-Windowed Turret,” *AIAA J.*, Vol. 45, No. 2, 2007, pp. 347–357.

- [21] Coirier, W. and Tramel, R., “Development of an Integrated Aero-Optics Modeling Capability: OVERFLOW-aeroOptics,” AIAA Paper 2010-0557.
- [22] White, M., “High-Order Paraxial Beam Approximation for Aero-Optics,” AIAA Paper 2008-1075.
- [23] Boyd, J., *Chebyshev and Fourier Spectral Methods*, Springer-Verlag, Heidelberg, 1989.
- [24] White, M., “High-order parabolic beam approximation for aero-optics,” *J. Comp. Phys.*, Vol. 229, 2010, pp. 5465–5485.
- [25] Rew, R. K. and Davis, G., “NetCDF: An Interface for Scientific Data Access,” *IEEE Computer Graphics and Applications*, Vol. 10, No. 4, 1990, pp. 76–82.
- [26] Frigo, M. and Johnson, S., “The Design and Implementation of FFTW3,” *Proceedings of the IEEE 93*, Special Issue on Program Generation, Optimization, and Platform Adaptation, 2005, pp. 216–231.
- [27] Feng, Y. and Owen, D., “An augmented spatial digital tree algorithm for contact detection in computational mechanics,” *Int. J. Numer. Meth. Eng.*, Vol. 55, 2002, pp. 159–176.
- [28] Holmes, D. and Connell, S., “Solution of the 2D Navier-Stokes Equations on Unstructured Adaptive Grids,” AIAA Paper 89-1932-CP.
- [29] Nichols, R., Tramel, R., and Buning, P., “Solver and Turbulence Model Upgrades to OVERFLOW 2 for Unsteady and High-Speed Applications,” AIAA Paper 2006-2824.
- [30] Nahrstedt, D., Hsia, Y.-C., Jumper, E., Gordeyev, S., Cenicerros, J., Weaver, L., DeSandre, L., and McLaughlin, T., “Wind tunnel validation of computational fluid dynamics-based aero-optics model3,” *Proc. IMechE Vol. 223 Part G*, : J. Aerospace Engineering, pp. 393–406.
- [31] Morgan, P. and Visbal, M., “Numerical Simulations Investigating Control of Flow Over a Turret,” AIAA Paper 2009-0574.