Examination of FlightStream for Proximity Flight with Stores

by

Eric Matthew Todd

A thesis submitted to the Graduate Faculty of
Auburn University
In partial fulfillment of the
Requirements for the Degree of
Master of Science

Auburn, Alabama
August 5, 2017

Keywords: proximity flight, FlightStream, store separation, panel codes

Approved by

Roy Hartfield, Chair, Walt and Virginia Waltosz Professor of Aerospace Engineering
Brian Thurow, W. Allen/Martha Reed Associate Professor of Aerospace Engineering
Dudley Nichols, Assistant Professor of Aerospace Engineering

Abstract

The goal of the following work was to validate and demonstrate the capability of the FlightStream solver to model proximity flight. To this end, wind tunnel proximity data for a MK-82SE bomb released from an F-16 was used as a point of comparison. Models of the aircraft and store were generated using NASA Open Vehicle Sketch Pad and exported as .STL files to FlightStream. FlightStream was then used to validate the models, to generate solutions for the scenario described in the wind tunnel tests, and to measure the aerodynamic coefficients in a plane near the aircraft. Comparisons have been provided both for the validation of the models and the replication of wind tunnel results. This thesis shows the utility of FlightStream for modeling proximity flight data with good fidelity in most scenarios.

Acknowledgements

The author would like to thank Dr. Roy Hartfield for providing the opportunity to conduct this research and for support, guidance, and encouragement throughout the process. Thanks are also due to Dr. Vivek Ahuja both for developing the FlightStream solver and for his continual support and assistance throughout the duration of the project. Finally, the author would like to thank his parents for their years of unwavering support and encouragement, enabling both this research and all other academic endeavors.

Table of Contents

List of Figures

Nomenclature

α	Angle of attack
β	Relative incidence
Γ	Circulation around marked spatial loop
γ	Local Vorticity
ε	Cone half-apex angle
μ	Freestream viscosity
ρ	Freestream density
4T	Arnold Air Force Base Aerodynamic Wind Tunnel
C_A	Axial force coefficient
C_F	Skin friction coefficient
CFD	Computational Fluid Dynamics
C_m	Rolling moment coefficient
C_N	Normal force coefficient
C_n	Pitching moment coefficient
CTS	Captive Trajectory System
FLTSM	FlightStream
GUI	Graphical User Interface
L	Lift force
NEAR	Nielsen Engineering & Research

Re_x	Reynolds number at a given location
STL	Stereolithography
V	Freestream Velocity
VSP	NASA Open Vehicle Sketch Pad
x	Distance from local leading edge

Chapter 1:

Introduction

1.1: Store Separation

When in close proximity, bodies produce disturbances in the flowfield that result in different aerodynamic loads on the bodies than they would experience singly. This is of particular relevance to military fighter, attack, and bomber aircraft, as their missions most frequently require the carriage and release of stores, either in the form of weapons loads, external fuel tanks, electronic countermeasures pods, or similar. The analysis of store separation is concerned with ensuring that the release of these stores can be conducted in a safe and consistent manner. Under certain conditions, released stores have been known to collide with their parent aircraft or other unreleased stores, a scenario which can be extremely costly and hazardous. Even prior to release, the presence of stores can have a significant impact on the performance of an aircraft, which can be determined and adjusted for as part of store separation testing. [1]

The longest-standing method of conducting store separation testing is to conduct a series of flight drop tests using the aircraft and store combination of interest. While this offers the advantage of ensuring that both the store and the aircraft are represented with ultimate fidelity, and remains a critical step in certifying store-aircraft combinations, full scale flight testing poses certain problems. Firstly, the necessary series of flights required are extremely expensive, time consuming, and labor intensive. To illustrate, the costs associated with a single F/A-18 flight, to include aircraft operating cost, range operating cost, test preparation, the store to be dropped, and all necessary measurements and telemetry, are estimated to be between $80,000 and $90,000. Testing all required configurations a store will be expected to safely release from can require a

great many flights, easily bringing the costs of a full test program into the range of multiple millions of dollars. [2] In addition to the significant costs, flight testing can be potentially hazardous to both pilot and aircraft in the event of a mid-air collision between store and parent aircraft. [3] This was significantly less of a problem in the earliest days of aviation, when airspeeds were subsonic and payloads were relatively small and spaced relatively far apart, meaning that the effects of interaction between store and parent aircraft were fairly minimal. Drop testing stores was therefore generally safe, and primarily focused on determining trajectories for the purposes of sighting. With the advent of faster aircraft capable of carrying greater payloads during and after the Second World War, however, the dynamics of store separation became more complex and the risks involved became greater. [4] Therefore, it became desirable to have alternative methods by which to test store separation scenarios in order to reduce dependency on, and ensure the safety of, full-scale flight tests. [3]

Starting in the 1960s, wind tunnel instrumentation became sufficiently capable to conduct separation tests. A representative of this sort of wind tunnel system is the Captive Trajectory System (CTS) installed in the Arnold Air Force Base 4T wind tunnel. Aerodynamic loads on a store are measured though the use of an internal strain gauge, which are then fed to a computer that calculates the trajectory of the store. The store is then moved to a position corresponding to the next timestep. This process is then repeated until the store reaches a position corresponding to the edge of the system. Positions of released stores can be calculated to within a tolerance of 0.05 inches. [5] While wind tunnel testing eliminates the safety concerns inherent to flight testing, it is not without its own difficulties. Constructing models of the aircraft and store is a time consuming, expensive, and specialized task in and of itself, as is conducting the actual tests in the wind tunnel. Furthermore, the scaled-down models used in a wind tunnel will still not

necessarily match the behavior of a full-scale aircraft and store. [3] This imperfect modeling in a wind tunnel can produce results which do not capture potentially hazardous scenarios, to include collisions between parent aircraft and store. [6] Even discounting scale effects, a wind tunnel setup by its nature introduces discrepancies in the form of the supports to which the models are mounted; under certain conditions the presence of a sting has been seen to have a significant detrimental impact on the accuracy of wind tunnel results [7]. Wind tunnel tests have become much more accurate as time has progressed, though they still represent a significant investment in both time and effort. [3] For example, a 1989 test program for clearing the Joint Stand off Weapon to be launched from the F/A-18 required in excess of 400 hours of wind tunnel testing, costing approximately $1,500,000. [8]

1.2: CFD Application

Beginning in the 1970s, Computational Fluid Dynamics (CFD) software offered the ability to generate solutions for store separation problems purely through the use of computer programs. The most successful of the early CFD programs intended for the analysis of store separation was a subsonic six-degree-of-freedom program developed by Nielsen Engineering & Research (NEAR) for the US Air Force between 1968 and 1974. [11] The NEAR program was relatively limited by modern standards, using a series of sources and sinks to model the fuselage and a vortex lattice to model aerodynamic surfaces. These simplifications were seen to introduce some weaknesses to the program, particularly with regard to the vertical position and pitch rate of the separating store. [10] In general, however, it proved to be a useful tool, with most results being in reasonable agreement with experimental results it was compared against. [11, 12] The continuing development and improvement of CFD offered the potential to reduce or eliminate the necessity for wind tunnel testing. However, the computers of the day initially handicapped

CFD solutions, with solutions taking as much or more time than a comparable wind tunnel test. Even into the 1990s, running a single solution on a supercomputer could require runtimes on the order of several days. [13] In addition, solutions generated did not always fully capture the interactions between the two bodies. [3]

Continuing development of both hardware and software systems allowed CFD store separation testing to mature by the 2000s. However, generating an accurate separation solution using a CFD program requires an extremely high number of runs in order to model the position of the separating store in an acceptably fine resolution. Because of this, running a volume-based CFD test still requires a significant investment of time and energy, potentially to the point of being unfeasible for certain circumstances. This point was dramatized during Operation Iraqi Freedom; the US Navy required the certification of the F/A-18C carrying the GBU-12 "Paveway II" guided bomb at a station adjacent next to a 330 gallon external fuel tank. Generating and testing a CFD model was judged to be excessively time consuming, meaning the Navy elected to rely primarily on practical, full-scale, flight testing, with Air Force CFD simulations on a previously generated model acting in a supplementary role. [3]

1.3: Experimental Data

In order for the FlightStream solver to be acceptable for use on store separation and proximity flight problems, validation against experimental data is necessary. For this purpose, data from the United States Air Force report, "Aerodynamic Loads and Separation Characteristics of the BLU-27B, MK-82SE, and GBU-8 Weapons in the F-16 Aircraft Flow Field at Mach Numbers from 0.4 to 1.2" has been used. Relatively few openly available references were found with comparison data for store separation, but this report provided high-quality data of the type necessary to validate FlightStream. Specifically, in addition to modeling

the separation of a store as it left the aircraft, the report gave aerodynamic coefficients measured at discrete, specified positions in the flowfield below the aircraft. Data of this type allowed for the most effective testing of FlightStream, as it reduced the problem to its most basic form without requiring the modeling of bodies in motion relative to each other, a capability not currently native to FlightStream.

1.4: Overview

This thesis will seek to validate FlightStream for application toward the problem of store separation. An overview of the experimental data against which FlightStream has been compared will be presented in the second chapter, explaining the procedures, equipment, and facilities used in the experiment. The physical principles behind the operation of FlightStream will be broadly explained in the third chapter, along with practical implementation. The development of the outer mold line models used in FlightStream testing will be covered in the same chapter. Chapters four and five will present the results of testing, with the sixth chapter offering suggestions for future work and concluding the thesis.

Chapter 2:

Experimental Reference Data

2.1: Background, Experimental Reference Data

The data against which FlightStream results have been compared has been extracted from Reference 15. The experimental data used a 1:15 scale F-16 and stores placed in the Arnold Air Force Base Aerodynamic Wind Tunnel (4T). A photograph of aircraft and store models used in a similar test is shown in Figure 2.1:

Figure 2.1: F-16 and Store Models in Arnold AFB 4T [14]

The 4T is a closed-loop, variable-density, continuous-flow wind tunnel with the capability of operating at a range of Mach numbers from 0.1 to 1.2. Its test section measures 4 by 4 by 12.5 feet, with perforated walls that may be adjusted to have between 0.5 and 10 percent open area as necessary. As seen in the above figure, the 4T is equipped with two different mounting systems for the store and the aircraft. The parent aircraft was mounted, inverted, to an offset sting extending from the floor of the tunnel, while the store was mounted to the CTS sting extending from a support above the model. The F-16 sting could be adjusted in pitch only, while the CTS allowed for six-degree-of-freedom movement of the store. The motion of the store

could be controlled either through direct input of the user or automatically in response to aerodynamic forces on the store. In the series of runs that have been examined in this work, the position was manually selected. A side view diagram of the wind tunnel setup is shown in Figure 2.2. [15]

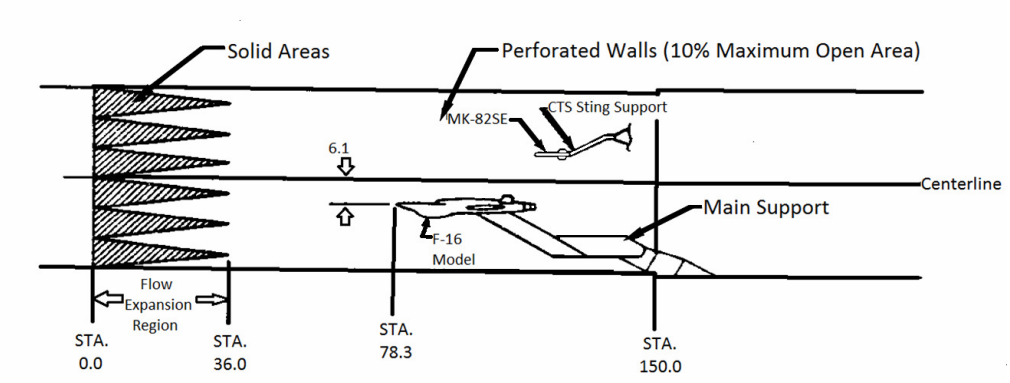

Figure 2.2: 4T Test Section Diagram (dimensions in inches) [15]

All tests were conducted at a simulated altitude of 5,000 feet above mean sea level, at a series of Mach numbers ranging from 0.4 to 1.2. Coefficients were calculated based on load data from the six-component internal strain gauge connected to the store sting support, and all results were believed to be ninety-five percent accurate. The experiments conducted in Reference 15 included testing a variety of stores in proximity to an F-16, with run series both at discrete, pre-determined locations and with trajectories simulated via a six-degree of freedom solver. [15] The work that has been conducted in FlightStream is narrower in scope than the original Air Force report; only data obtained from a single store, the MK-82SE "Snakeye" high-drag unguided bomb, at predetermined positions was used for comparison, and all comparison data came from Mach 0.6 test runs. Given that the goal of this thesis is to validate the ability of FlightStream to model proximity flight, this data was seen as being the best suited to the task. Comparing against runs in which the position of the store was manually set simplified the problem compared to runs in which the store followed a realistic trajectory, decreasing the

number of variables had to be accounted for, and reducing the problem to purely a test of FlightStream. It was decided to examine a single store instead of the three used in Reference 15 for a similar reason; if FlightStream was capable of predicting the loads on any single store, then it should reasonably be expected to do so for any store. Selection of the MK-82SE over the other stores tested in Reference 15 was fairly arbitrary, however, with it being chosen based on the fact that other tests had used it in the past. The runs at Mach 0.6 were selected with the goal of staying within the range of consistent operation of FlightStream; higher Mach numbers would have entered the transonic regime, which FlightStream is not currently intended to analyze.

2.2: Model Geometry

The parent aircraft in the wind tunnel tests was a 1/15-scale model of an F-16. A diagram of this model is shown in Figure 2.3:

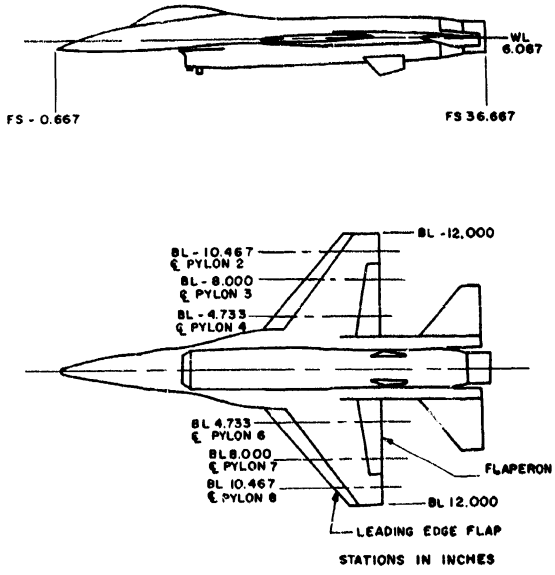

Figure 2.3: Sketch of F-16 Model [15]

Note that the vertical stabilizer is not included; as seen in Figures 2.1 and 2.2, this omission is due to the fact that the portion of the fuselage to which the stabilizer is normally attached was

needed as a mount point for the sting. Also note that, while horizontal stabilizers are shown in Figure 2.3, they were not present during testing. Upon being placed in the wind tunnel, it was found that the horizontal stabilizers interfered with the CTS sting to which the store was mounted, and therefore were removed. [15]

A diagram of the MK-82SE model is shown in Figure 2.4:

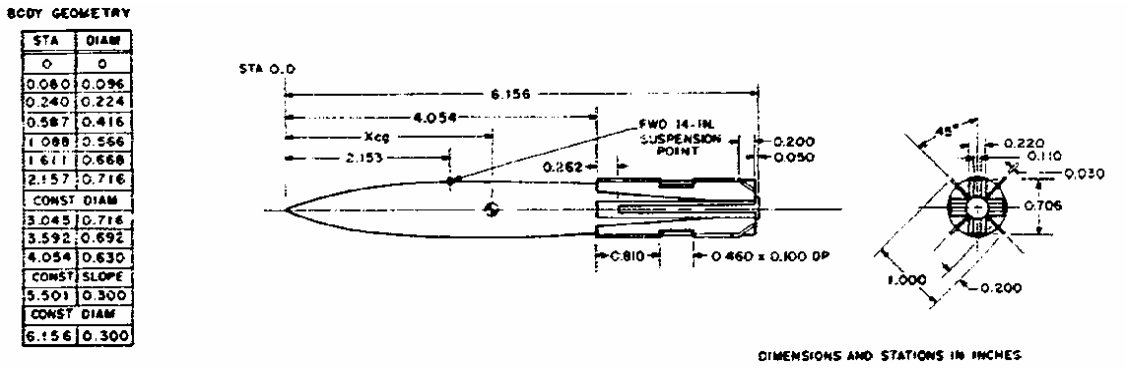

Figure 2.4: Sketch of MK-82SE Model [15]

For the sake of clarity, it should be noted that the wind tunnel model behaved slightly differently than a normal, full-scale version of the MK-82SE. In normal operation, the aft section of the "Snakeye" opens very soon after release in order to act as an airbrake, as shown in Figure 2.5. This was not the case with the wind tunnel model, which was modeled as a single, solid object with no moving parts.

Figure 2.5: Deployed MK-82SE, Illustrating Normal Operation [16]

Chapter 3:

FlightStream and Setup

3.1: FlightStream Background and Principles of Solver Operation

The FlightStream solver is a vorticity-based potential flow solver developed by Vivek Ahuja as a robust, efficient alternative to typical CFD solvers. It is capable of modeling incompressible or compressible flow at mach numbers less than 0.8. [17] FlightStream calculates induced lift and induced drag using the Kutta-Joukowski Theorem, and skin friction drag from local vorticity distribution. The computed lift and drag distributions on the body are then used to determine moments about a user-specified point. Unlike volume-based CFD solvers, FlightStream is capable of producing solutions using only an unstructured surface mesh. This capability greatly reduces the amount of time and effort needed to create a mesh that will be suitable for FlightStream use. [17, 18] FlightStream so far has been shown to consistently yield good results for individual aircraft. Of passing relevance to store separation, FlightStream has been seen to provide good results for optimizing the position of a turbofan engine pod mounted under the wing of a transport, which bears some similarity to a store carried below a fighter aircraft. [17, 19]

3.1.1 Background, Panel Codes

Panel codes have been in common use for aerodynamic analysis beginning in the early 1970s. [20] They offer a very natural method for modeling an aircraft; the outer mold line of an aircraft can be modeled as it exists in reality, whereas older codes require the use of approximations to yield the same effect as the body. [21] At the same time, they offer significant runtime reductions compared to volume-based CFD solvers, both through the elimination of the

volume mesh, which in itself yields a drastic runtime reduction, and through their generally efficient nature. [17, 22]

The tradeoff for this improved runtime is that panel codes do not generally resolve boundary layers or other viscous regions. Some success has been seen in this regard, with NASA codes implementing pressure-based models to predict the effects of flow separation and viscous effects [17]. Unfortunately, these adaptations have been an imperfect solution, sacrificing much of the efficiency of a panel code without fully matching the fidelity of a viscous Navier-Stokes code. [17] This is a natural limitation of a pressure based panel code, which, depending on geometry, may require a very fine mesh to ensure pressure gradients are modeled correctly and a good solution may be generated. [17, 23] Further, small, pronounced irregularities in the shape of a mesh, such as dents, bumps, or kinks, will have a large adverse effect on the quality of a solution. [17, 24] The importance of mesh quality has a twofold impact on the efficiency of a panel solver; firstly, additional time and effort must be taken to ensure the mesh will run effectively, and secondly, a finer mesh will naturally require a longer runtime.

Alternatively, vorticity-based panel codes have been seen to be significantly more robust compared to pressure-based solvers with regards to mesh fidelity, allowing for the use of coarser, lower-quality, or less consistent meshes. [17, 25] This makes them the more reliable option for analysis which requires rapidly generated meshes, such as optimization problems and evaluation of aircraft in the early stages of the design process, in which the geometry of a model may be subject to rapid and dramatic changes. [17] The objective driving the development of FlightStream was to combine the advantages of the two types of potential solvers, giving fidelity comparable to a successful pressure-based solver, while retaining the efficiency and robustness of a vorticity solver. [17]

3.1.2: FlightStream Induced Velocity Calculations

The foundational principles underlying the operation of the FlightStream solver are explained in full detail in References 17 and 18. The following is intended as a brief overview of the workings of FlightStream.

The Kutta-Joukowski theorem is central to the operation of FlightStream; this theorem states that the lift per unit span of an airfoil will be described by equation 3.1:

$$L = \rho_\infty V_\infty \Gamma_\infty \tag{3.1}$$

In which Γ represents the circulation around the airfoil. The Kutta condition states that the circulation value is such that the airflow leaves the trailing edge of an airfoil in a smooth manner, so long as the circulation is taken over a curve which encompasses the entire surface. [26] The FlightStream solver enforces this condition at specified trailing edges, which allow for a starting point for calculating the lift over the entirety of the model.

[17, 18]

FlightStream operates through the application of a vortex ring to each of the triangular facets of a model. The direction of the vortex rings is aligned to match the windings of the facets, and because each the edges of the facets forms a discreet, closed loop, the vortex ring is modeled in accordance with Helmholtz's theorems. [17, 20] The application of the vortex ring is illustrated in Figure 3.1:

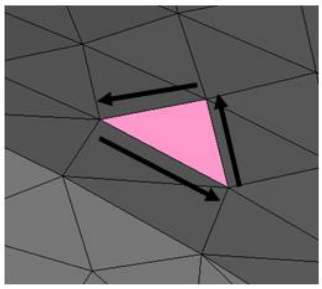

Figure 3.1: Vortex Ring for a Single Facet [17]

For a brief mathematical illustration, Figure 3.2 diagrams a vortex ring inducing a velocity “V” at a point “P”.

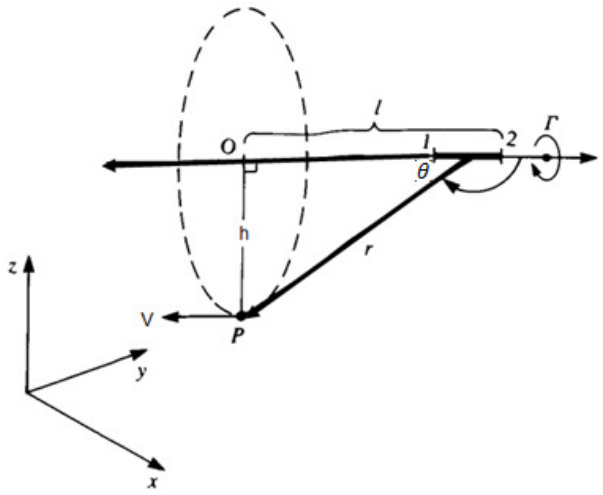

Figure 3.2: Vortex Ring [17]

Velocity induced by a vortex ring on a single facet side can be determined from equation 3.2:

$$V_i = \frac{\Gamma}{4\pi h}(\cos(\theta_2) - \cos(\theta_1)) \tag{3.2}$$

Net velocity induction for each facet can be calculated by summing the induced velocity from each facet edge:

$$V_j = \Gamma_j \sum_{i-1}^{N_{edges}} \left[\frac{1}{4\pi h_i}(\cos(\theta_{2i}) - \cos(\theta_{1i})) \right] = \Gamma_j A_{j,P} \tag{3.3}$$

Where $A_{j,P}$ is referred to as a geometric influence coefficient. A full derivation of these equations can be found in Reference 17. [17]

As with most other surface-based potential flow solvers, the vortex strengths are determined to satisfy the Neumann boundary condition for a solid surface; the dot product of the velocity vector and normal vector of a face must be equal to zero, which is to say that the velocity must be directed parallel to the surface of a facet. [17]

3.1.3: FlightStream Vorticity Shedding and Wake Model

Prior to running the solver, segments to be considered as trailing edges are specified either manually by the user or automatically by FlightStream. It is at these edges that the Kutta Condition is enforced; if left unmarked, the solver will treat the simulation as purely a potential flow problem, and compute results for pressure distribution but no aerodynamic loads. Once the solver begins operation, wake strands are generated at nodes located at the endpoints of marked trailing edge segments. The vorticity shed into each wake strand is determined by taking the summation of the vorticities of each of the facets in contact with the point of origin of the wake strand, as shown in Figure 3.3.

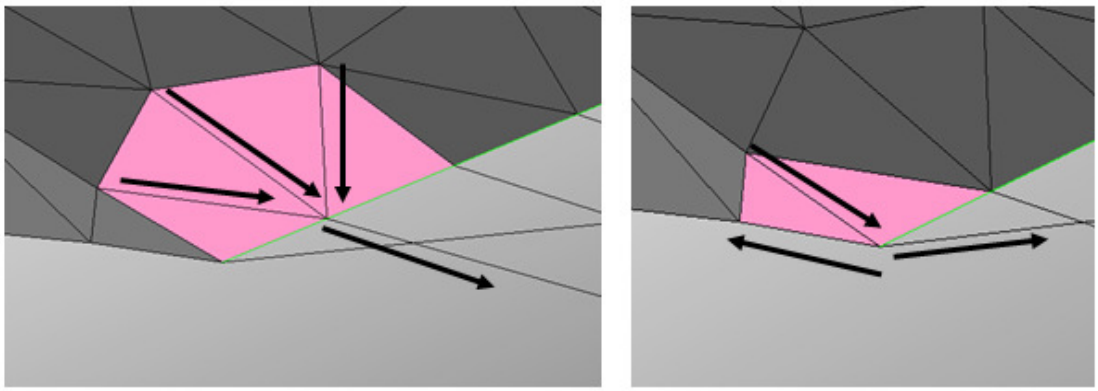

Figure 3.3: Trailing Edge Wake Model [17]

Each of the arrows in the illustration indicates an edge contributing to the vorticity of the wake strand, with the arrows themselves being representative of the difference between the vorticites of the facets in contact at that edge. Note that in the realistic case of a trailing edge which joins two sides of a body, as would be seen for a wing with nonzero thickness, vorticity contributions will be made from both surfaces. The action shown in Figure 3.3 is represented mathematically by Equation 3.4:

$$\gamma_{k,wake} = \sum_{i=1}^{N_{SOURCES}} \gamma_i + \sum_{j=1}^{N_{SINKS}} \gamma_j \tag{3.4}$$

Marked trailing edges are assigned no vorticity; therefore, they do not contribute to the vorticity of the wake strand, and only the vorticity necessary to satisfy the Kutta condition at the trailing edge is shed into the wake strand. [17, 18]

3.1.4: FlightStream Calculations

In Equation 3.3, the only unknown variable is the strength of the vortex ring. Based on boundary physics, velocity is a known value at certain control points; therefore, it is possible to solve for the vorticity in each of the loops in contact with known velocity points. To include the entirety of the mesh, this concept can be expanded into the matrix equation:

$$\begin{bmatrix} A_{1,1} & \cdots & A_{1,N} \\ \vdots & \ddots & \vdots \\ A_{N,1} & \cdots & A_{N,N} \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \cdots \\ \Gamma_N \end{bmatrix} = \begin{bmatrix} B_i \\ \cdots \\ B_N \end{bmatrix} \tag{3.5}$$

Matrix "A" is the N-by-N matrix of geometric influence coefficients, matrix "B" contains the velocity requirements on control points known from external physics requirements, and matrix "Γ" is comprised of the unknown vorticities at each of the facets. This is effectively a system of N equations and N unknowns, and it therefore can be solved exactly. Unfortunately, extremely large matrices must be used to fully describe an aircraft model, which results in an unacceptably high degree of machine error. In order to limit the influence of machine errors, FlightStream partitions the mesh into smaller sections and solves them one at a time. [17, 18]

The process of dividing the mesh means that the process for evaluating the mesh is naturally iterative. The solver ceases iterating once it has converged to a prescribed convergence criterion, determined through the use of the area-averaged vorticity strengths on the mesh. This threshold is selectable by the user but by default is set to 1.0×10^{-4}, which has been seen to be an appropriate value in most cases. Once an accurate solution has been arrived at, lift, induced drag, skin friction drag, and moment coefficients are computed in post processing. [17, 18]

3.1.5: FlightStream Skin Friction Drag Calculations

The skin friction drag coefficient for each facet is calculated using the following semi-empirical equation, developed by Prandtl-Schlichting in 1932 [17]:

$$C_F = \frac{0.455}{(\log_{10} \mathrm{Re}_x)^{2.58}} - \frac{1700}{\mathrm{Re}_x} \tag{3.6}$$

Because FlightStream does not compute velocity at each of the facets, Reynolds number must be arrived at through alternative means. Vorticity at each face is computed by the solver, so the Reynolds number of a given panel is determined using Equation 3.7 [17]:

$$\mathrm{Re} = \frac{\rho_\infty V_\infty \left(x + \frac{\gamma}{V_\infty} \right)}{\mu} \tag{3.7}$$

In which x represents distance from the local leading edge, μ represents dynamic viscosity, and γ represents the vorticity of the panel being examined. The result for this equation can then be inserted into the previous equation, giving the local friction coefficient. The total skin friction on the aircraft is then determined by summing the friction coefficients of all of the individual facets. [17, 18]

3.1.6: Fuselage Vortex Shedding

In the course of validating the models, it became necessary to shed vortices in a manner atypical to the normal operation of FlightStream. In the case of a slender bodied aircraft, such as a bomb or missile, set at a non-trivial angle of attack, a significant portion of its lift is generated on the forward portion of the fuselage. As it was primarily intended for the analysis of aircraft that are not slender bodies, FlightStream does not automatically account for vorticies shed from the forward fuselage of a body. In order to accurately model the behavior of a slender body,

trailing edges must be manually marked on the front portion of the fuselage, as shown in Figure 3.4. [27]

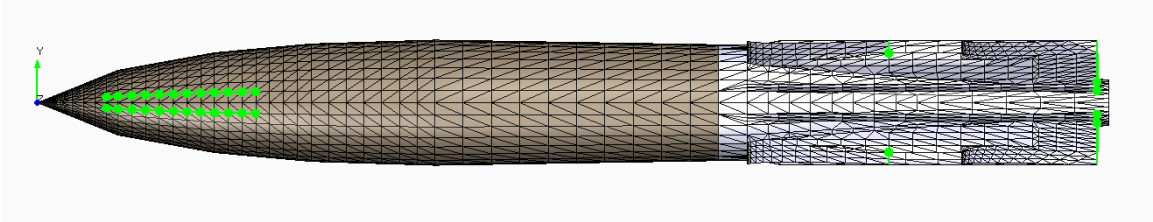

Figure 3.4: MK-82SE Model set up for Fuselage Vortex Shedding

Note that all vertices on the marked lines must be designated as wake termination nodes to ensure that the solver runs correctly; if wake strands are propagated from the forward section of the model, they will not behave realistically and will cause the trailing strands to generate inaccurate load results. The front fuselage must also be split into a separate surface in order to ensure that accurate results can be ascertained. This is necessary because FlightStream currently gives lift results which are the negative of the correct loads on the forward fuselage. Therefore, the model must be divided and the results for both sections added together by the user after the solver finishes running. This division is seen in Figure 3.4 with the front and rear sections being differently colored. Continuing development of FlightStream is expected to eliminate this requirement at some point in the future.

For this work, the position at which the vortices are shed, and therefore which lines should be marked, is determined from the experimental curves shown in Figure 3.5: [27]

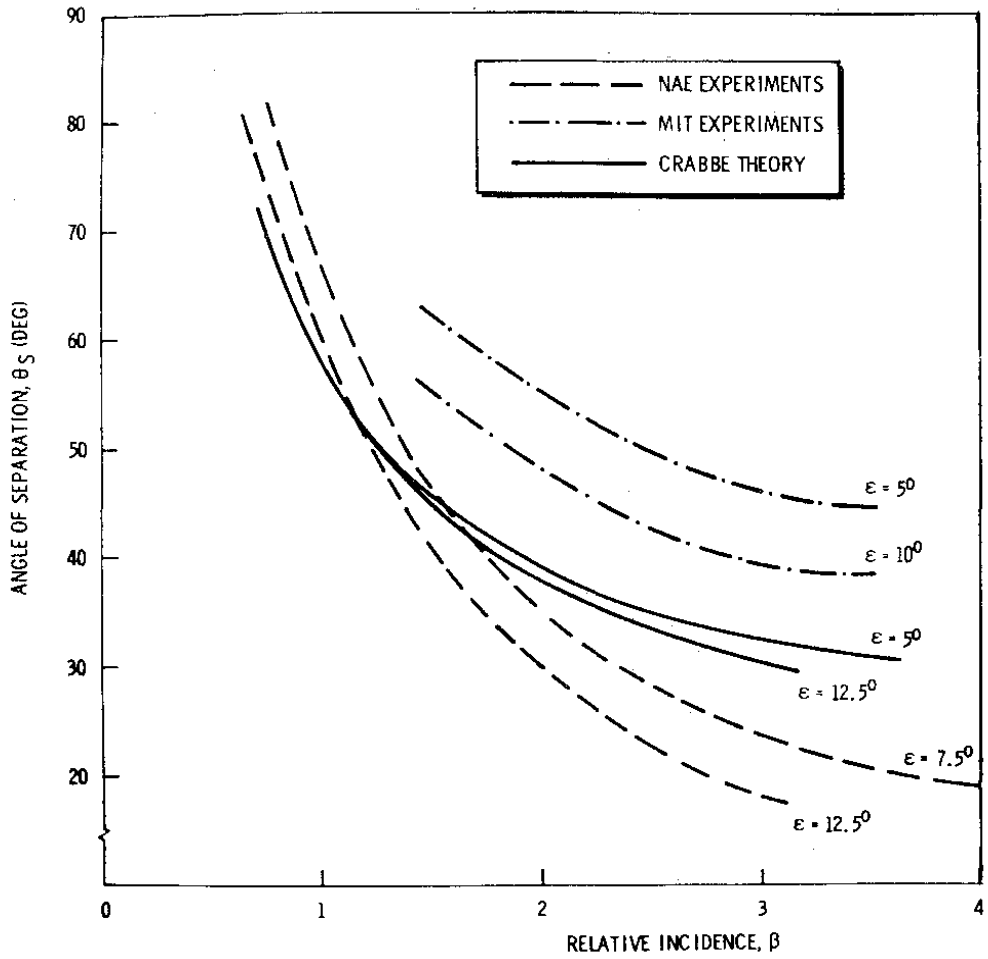

Figure 3.5: Angle of Separation for Circular Cones [27]

Relative incidence, represented here as β, is defined in Equation 3.8:

$$\beta = \frac{\tan(\alpha)}{\tan(\varepsilon)} \tag{3.8}$$

In which α represents the angle of attack of the cone and ε represents the half-apex angle of the cone. Once β has been computed, the angle of separation can be determined by finding the location of β on the x-axis of Figure 3.5 and following a vertical line upwards until it intersects the curve which corresponds to the cone being examined. For example, a 10-degree cone set at a 20 degree angle of attack produces a relative incidence value of approximately 2.06. After tracing upwards from 2.06 to the intersection of the "ε=10°" curve, it can be seen that the angle

of separation is approximately 47 degrees from the horizontal. For the application to this thesis, the MK-82SE is most closely approximated by a 12.5 degree cone, and the angle of attack varies between zero and four degrees. Computing β and tracing upwards gives an angle of separation in the region of 80-90 degrees, or in the case of an angle of attack less than one degree, no apparent separation angle at all. [27] Future versions of FlightStream are expected to improve upon this empirical approach by calculating the vortex shedding line and strength from the induced velocity distribution.

3.2: Background, NASA Open Vehicle Sketch Pad and Construction of Models

All models used were constructed through the use of NASA Open Vehicle Sketch Pad (VSP). Developed from the earlier NASA Rapid Airplane Modeler, VSP is an open-source tool designed for the purposes of generating three-dimensional aircraft models in the early conceptual phase of aircraft design. As such, it has been designed with the ability to generate models very quickly and efficiently. It includes the capability to export models in various file formats compatible with aerodynamic analysis programs, including the 3D Systems Stereolithography (STL) files used by FlightStream. [28] These STL files can be generated either in a high-quality unstructured mesh, or a simple lower-quality mesh grid. The simple mesh grid has been seen to function well and yield good results, so it is typical practice to use that option when generating STL files for FlightStream. [29] Because it is specifically designed for modeling aircraft through manipulating a series of parameters via a relatively simple graphical user interface, VSP offers significant advantages over traditional computer aided design (CAD) programs, such as Solid Edge. Pre-loaded airfoil profiles allow aerodynamic surfaces to be rapidly created in VSP, all other key geometry design features can easily be accessed and adjusted from the GUI options

windows, and the fidelity of the model mesh may be easily adjusted. Each of these functions tends to be extremely time consuming or impractical in CAD programs; airfoils must be manually defined, geometry must be methodically constructed piece by piece, and meshes are generally less than ideal. While they offer extreme precision with regard to the shape of the object being generated, they typically generate an excessively fine mesh for the purposes of FlightStream analysis. In the best case scenario, this leads to undesirably long runtimes, requiring several hours to import and run. In the worst case scenario, overly fine CAD meshes have been seen to completely crash FlightStream after several hours of attempting to import. The latest versions of VSP also offer the ability to export CAD files; this functionality provides maximum flexibility for the latest and future versions of FlightStream. [28]

The F-16 mesh used is shown in Figure 3.6. This mesh was generated based on the model sketch shown in Figure 2.3. However, as there are no size restrictions in FlightStream, the model was constructed at full scale. This ensured that reference dimensions could be matched to their real-world values without need for adjustment. Cursory testing indicated that scaling the model up to full size had no apparent effect on the accuracy of results generated.

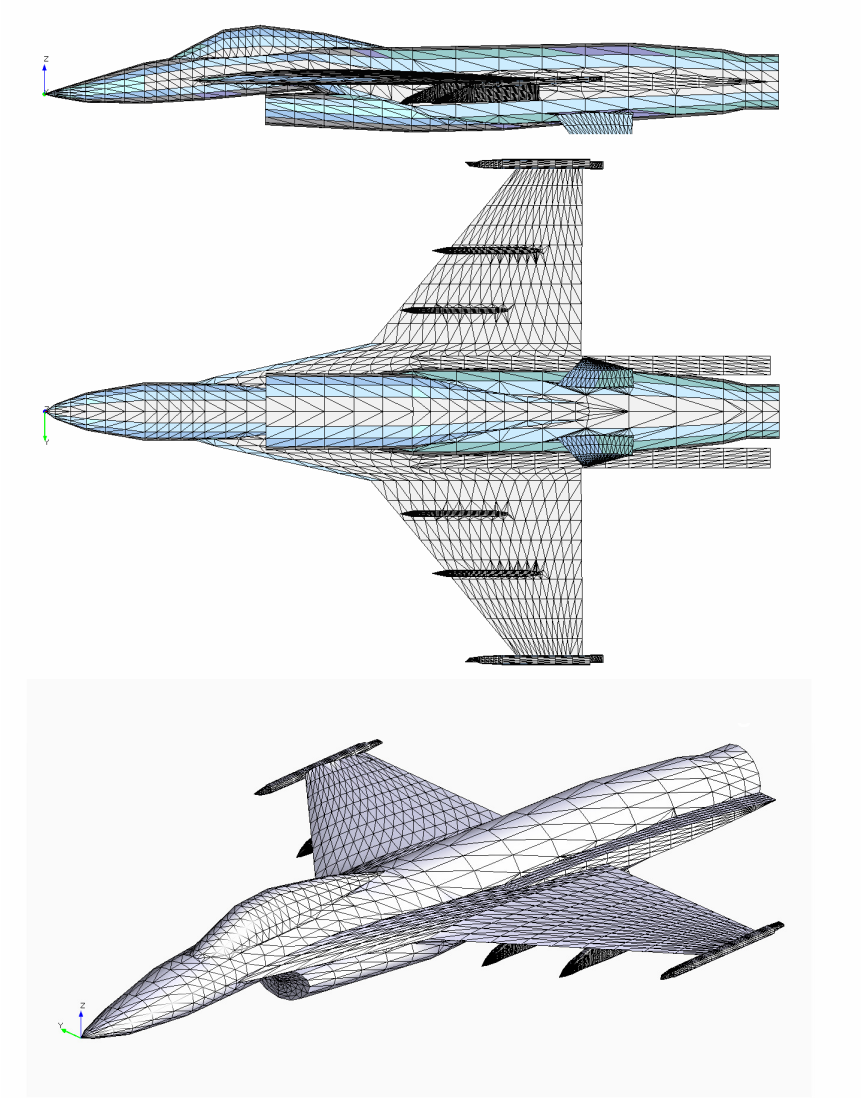

Figure 3.6: F-16 Computer Model

As mentioned in Chapter 2, the lack of horizontal and vertical stabilizers is correct to the experimental data; the vertical stabilizer was never included in the model, and the horizontal stabilizers were removed after they were found to interfere with the sting mount on which the store was placed. Underwing pylons and wingtip rails were also included in the computer model. Though they were not shown in the sketch, these features were accurate to the wind tunnel model, and necessary to ensure that all surfaces present in the original experiment were modeled in FlightStream. Conversely, the movable surfaces shown in the sketch were not included in the

VSP model. All runs of interest held the surfaces in their neutral position; therefore, the inclusion of separate control surfaces would have added a significant degree of complexity to the model and increased computation time for no benefit.

Two deviations from the physical model were present; In order to ensure the solver would run properly, the lower portion of the ventral fins and the aft-most row of facets on the lower surface of the wings were omitted from the solver, effectively converting the surfaces to single-layer thin wings. Surfaces or portions of geometries which are extremely thin, such as these, have essentially the same boundary conditions on both the upper and lower surfaces, a situation which poses problems for the solver and will typically lead to errors. While a specific criterion for predicting which geometries will exhibit this behavior has not been determined, omission of similar sets of facets is common practice when using FlightStream, often being an early step in ensuring that a mesh will run accurately and consistently. These omissions have not been seen to noticeably impact the quality of results generated. [29]

Figure 3.7 shows the three models of the MK-82SE that were used, each with a progressively finer mesh structure.

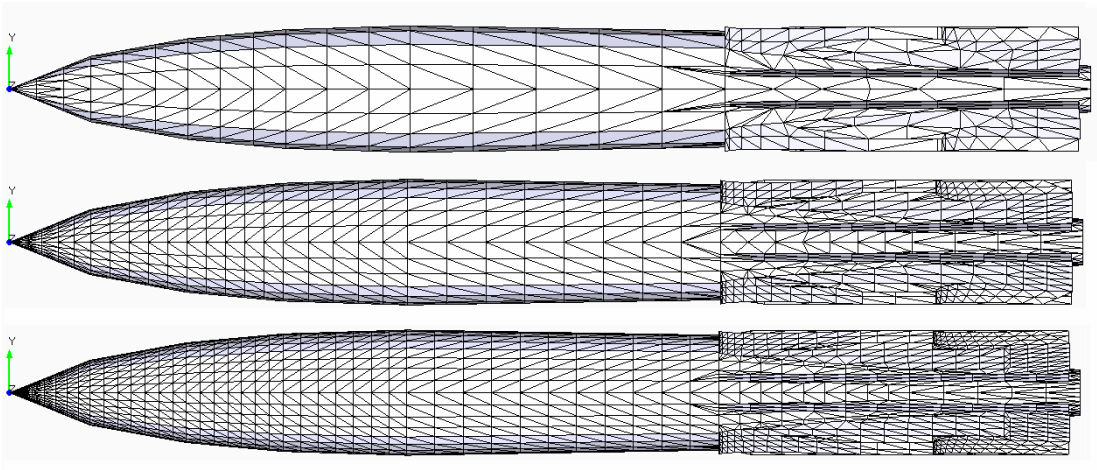

Figure 3.7: MK-82SE Meshes

All three models were based on the drawing in Figure 2.4, but as with the F-16, scaled up to full scale. A mesh refinement study was conducted to ensure numerical fidelity of the solutions generated. The most coarse mesh, comprised of 1,620 facets, was selected as a starting point, as it visually recreates the MK-82SE well and similar densities have been seen to work well on other models in unrelated work. The finest mesh was generated by approximately doubling the number of longitudinal and latitudinal sections in VSP, resulting in a facet count of 6,324. When paired with the F-16 model, the finest mesh also approximately coincided with the maximum mesh size that the hardware used was consistently capable of processing without errors. The medium fineness level mesh was generated by setting the latitude and longitude sections to approximately 1.5 times that of the coarsest mesh, and was comprised of a total of 3800 facets.

As the meshes output by VSP are not specifically designed to be compatible with the FlightStream solver, some inspection and minor alterations are typically necessary in order to ensure effective and consistent operation of the solver. In many cases, undersized faces, "inside out" faces with reversed surface normals, overlapping faces, or extremely high aspect ratio "needle" faces are present. All of these instances can keep FlightStream from successfully running a simulation, and if present, they must be dealt with manually by the user. In the case of the MK-82SE, three quarters of the initial mesh was deleted, with the remaining single quarter of the mesh being edited manually to assure its quality. The edited quarter was then mirrored across the XY and XZ planes in order to re-create the missing sections of the mesh. In addition to allowing for a less time consuming process to edit the full mesh, this approach ensured that the mesh would be completely symmetrical across both the XY and XZ planes.

The fins of the models are an approximation of those on the actual MK-82SE. The fins of the real store are effectively thin, flat planes; VSP is unable to effectively model them as such,

so a NACA 0005 symmetric airfoil was used instead. While the limited thickness of this airfoil section offered a good approximation of the flat fins, an aerodynamic surface in which the upper and lower surface are in extremely close proximity to each other poses some problems in FlightStream. This is a known limitation of the solver, as previously mentioned in regards to the ventral fins and trailing edges of the F-16 model. In the case of geometry in which the entirety of an aerodynamic surface is very thin, the accepted solution is to delete half the aerodynamic surface, as well as the small faces at the wingtips, again creating effectively a single layer mesh to approximate the wing. To ensure that the model would behave as similarly as possible to the actual, symmetric MK-82SE, the lower surfaces of the upper fins and upper surfaces of the lower fins were deleted. Simulations revealed that this geometry was capable of accurately modeling the behavior of the MK-82SE.

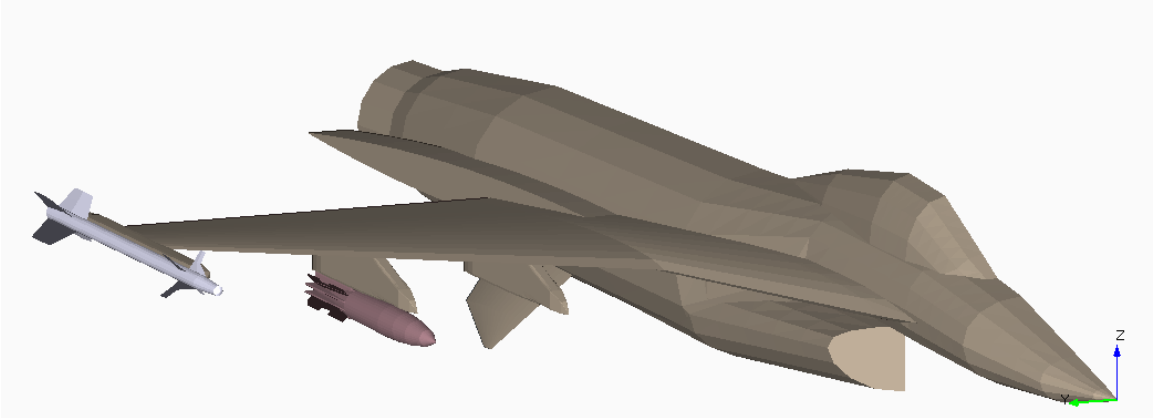

Figure 3.8: F-16 Mesh with MK-82SE

Figure 3.8 shows the configuration in which the aircraft and store were analyzed. The MK-82SE was positioned on the starboard side outboard pylon for all tests. This carriage position sets the store at a three degree angle downward relative to the centerline of the aircraft; the aircraft itself was at a positive two degree angle of attack, giving the store a negative one degree angle of attack to the freestream. As in the experimental data to be replicated, the inboard

pylon was empty, and an AIM-9 “Sidewinder” missile was modeled on the starboard wingtip rail. In order to reduce runtimes as much as possible, only the starboard half of the aircraft was modeled, as the effect of the port side of the aircraft on a store released from the starboard-most pylon is negligible.

Chapter 4:

FlightStream Analysis

4.1: Validation of Models

The STL models of the MK-82SE and the F-16 were compared against experimental results from reference 15 and 17. The results of these FlightStream runs are shown in Figure 4.1 to 4.3.

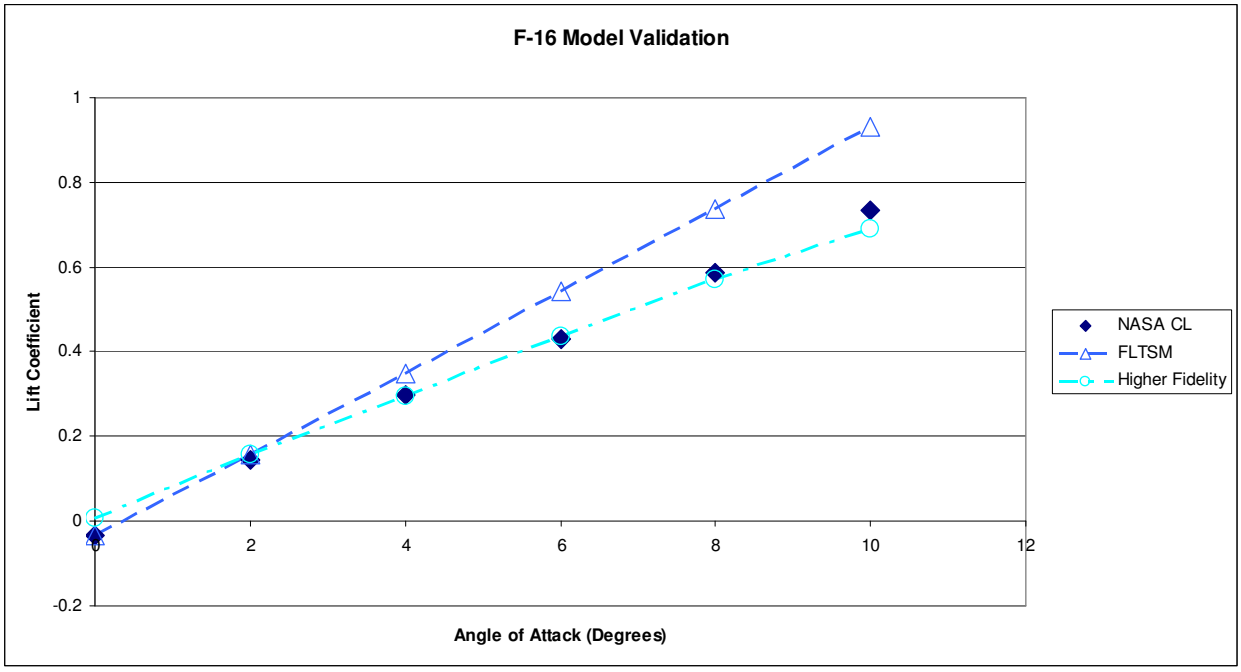

Figure 4.1: Results for F-16, Lift Coefficient vs. Angle of Attack [17]

So as to ensure an accurate representation of the model used to generate this experimental data, the underwing pylons were removed from the F-16 mesh and horizontal stabilizers were added. As seen in Figure 4.1, the model proved to be in generally good agreement with the experimental data, though with decreasing accuracy as angle of attack is increased. This has been attributed to the fidelity of the mesh used here; for comparison, a series of results generated with a higher fidelity FlightStream model used in Reference 17 has been included. That older

model, which was unavailable for use in this work, is seen to be markedly improved at higher angles of attack over the one used here. However, given that the results of the current model were seen to be reasonably accurate between zero and four degrees, and that the model was to be run at a fixed two degree angle of attack, it was judged to have a sufficient degree of fidelity for the task at hand. As such, it was not considered necessary to generate a finer, higher-fidelity mesh for the F-16; while the potential for FlightStream to better re-create the behavior of the aircraft clearly exists, further refining the mesh would have potentially added significant runtime for little to no benefit in the area of interest. [17]

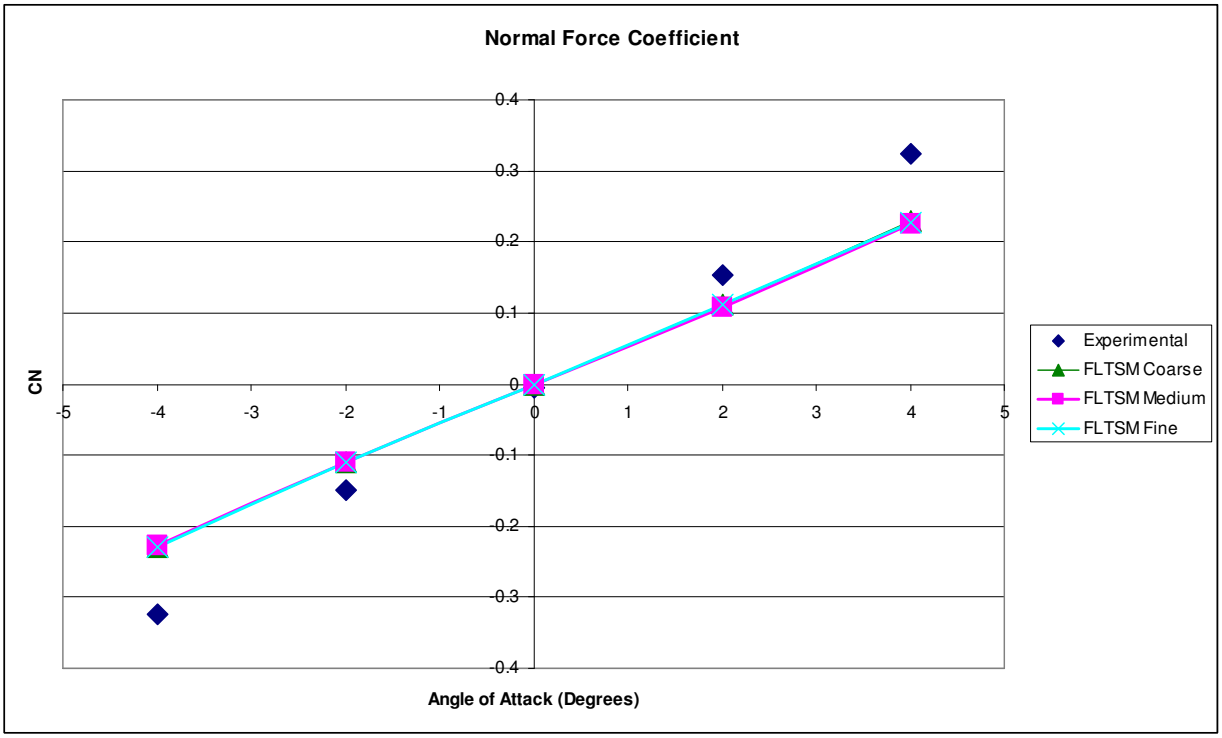

Figure 4.2: MK-82SE, Normal Force Coefficient vs. Angle of Attack [15]

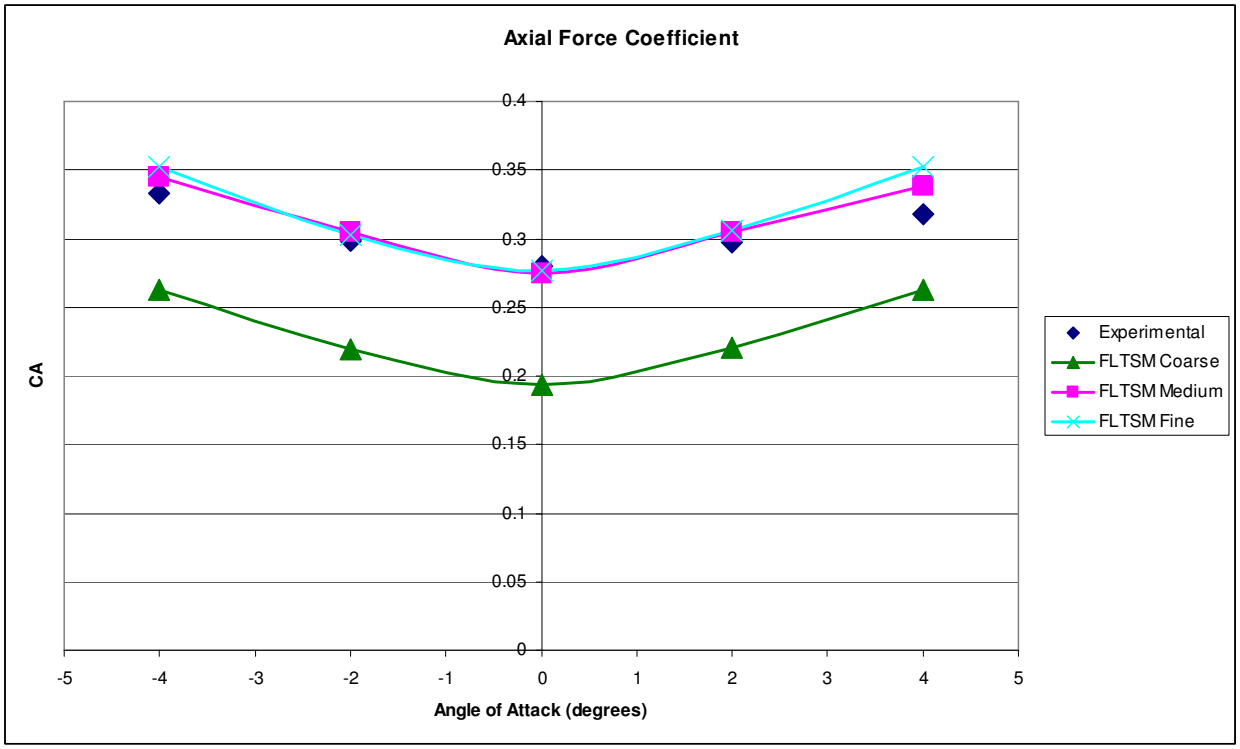

Figure 4.3: MK-82SE, Axial Force Coefficient vs. Angle of Attack [15]

In order to validate the MK-82SE model, the three meshes were run through a series of simulations at a Mach number of 0.6 and an altitude of 5000 feet, with the angle of attack being varied between negative and positive four degrees in increments of two degrees. The results for the three meshes are shown in figures 4.2 and 4.3, with the labels "coarse," "medium," and "fine" applying to the meshes in ascending order of fineness. The data from Reference 15 gave values for normal force coefficient and axial force coefficient; in order to match, the normal and axial force values were calculated from FlightStream lift and drag output using equations 4.1 and 4.2, respectively:

$$C_N = C_L \cos(\alpha) + C_D \sin(\alpha) \tag{4.1}$$

$$C_A = -C_L \sin(\alpha) + C_D \cos(\alpha) \tag{4.2}$$

In regards to axial force, the two finer meshes proved to be both accurate to experimental data and consistent to each other. For normal force, all three meshes were consistent with each other, though the model proved to be somewhat less accurate than ideal. As seen in Figure 4.2, the normal force for the FlightStream simulation becomes less accurate as angle of attack moves away from zero. In general, FlightStream is very accurate in calculating lift on an aircraft; the discrepancy in this case is likely due to two factors: the shape of the MK-82SE and the effects of separation. FlightStream operates most effectively when analyzing models in which aerodynamic surfaces with clearly defined trailing edges generate most of the lift acting on the model, but in the case of the MK-82SE, a significant portion of the lift produced is generated by the fuselage. Because the fuselage cannot be marked with a well-defined trailing edge, FlightStream results for lift, and therefore normal force, are less accurate because the lift on the forward body is not captured correctly. In addition, FlightStream is not currently equipped to model separated flows, which becomes a more significant contributor to normal force as angle of

attack increases. Future versions of FlightStream are expected to more accurately model the effects of both of these factors, but in the current version of FlightStream, it was necessary to find an alternative method to improve the fidelity of normal force results.

Initial attempts to improve normal force results were focused on making minor alterations to the geometry. The rationale for attempting these changes was that these slight changes could allow FlightStream to measure fuselage lift without affecting the overall accuracy of the mesh geometry in any significant manner. Two of the slightly altered meshes are shown in Figure 4.4, with comparison against the unaltered mesh and experimental results shown in Figures 4.5 and 4.6:

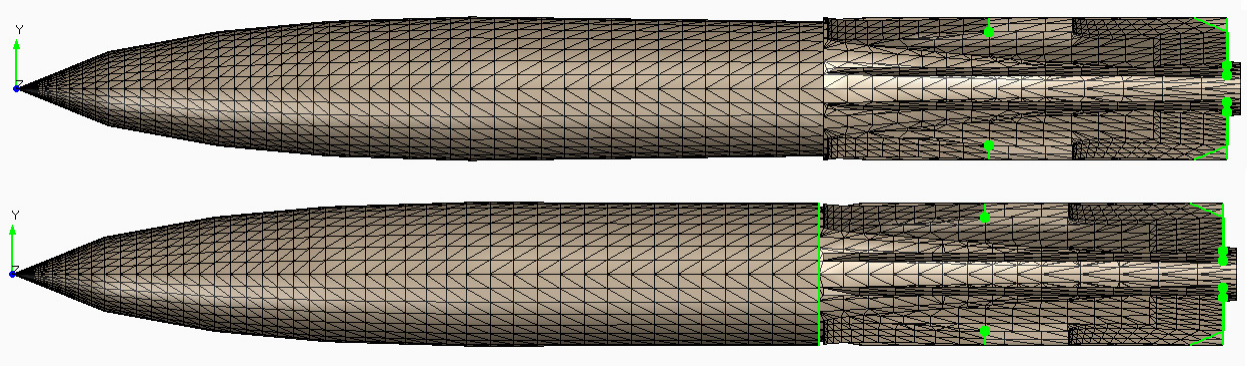

Figure 4.4: Altered MK-82SE Meshes

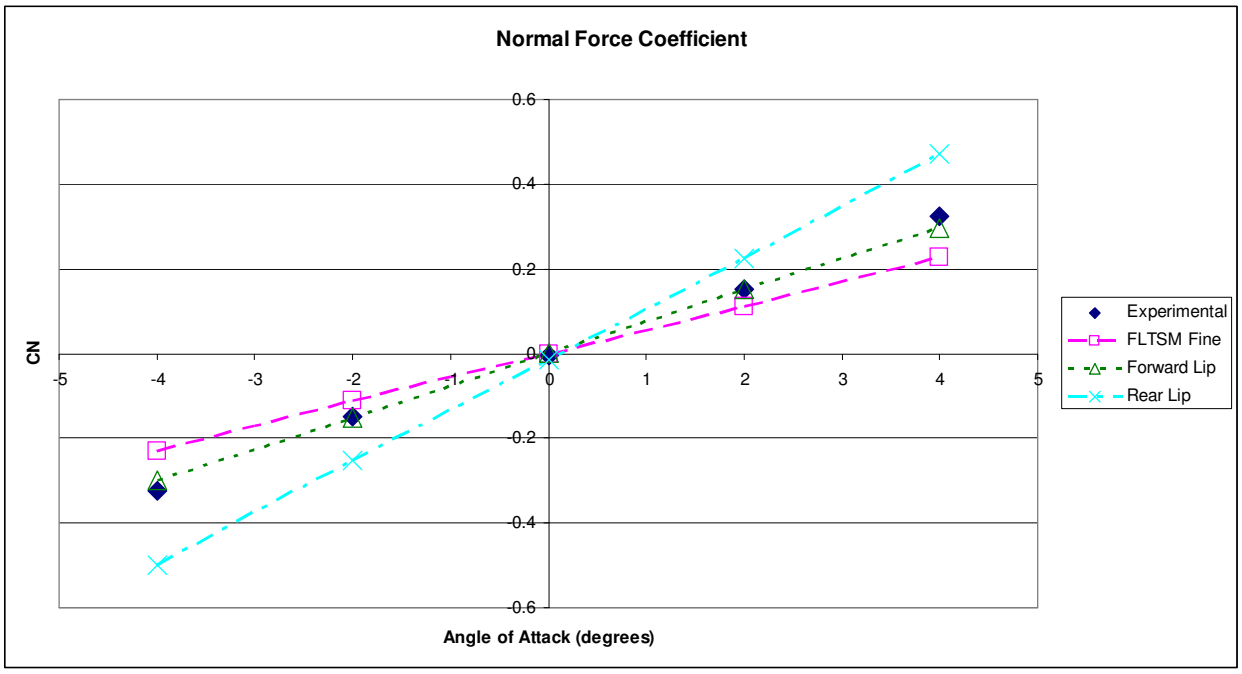

Figure 4.5: Normal Force Coefficient Comparison [15]

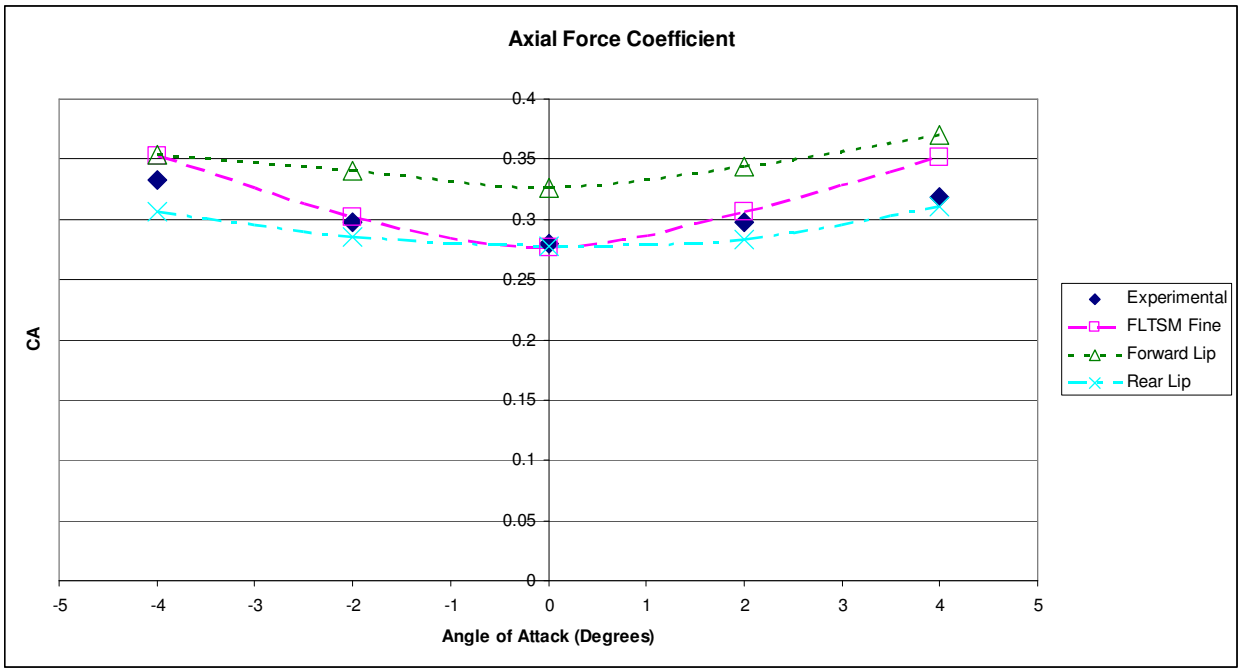

Figure 4.6: Axial force Coefficient Comparison [15]

Each of the meshes in Figure 4.4 differs from the normal fine mesh in that a small lip is present between the fuselage and fin sections of the mesh. In the first mesh, the lip faces forwards, which is to say that the diameter of the mesh increases between the forward fuselage and fin sections. This alteration was made based in some extent on the shape of the real-world MK-82SE. It was thought that the airbrake section, shown deployed in Figure 2.5, may create a small lip when retracted, and that the absence of this lip may have accounted for the erroneous normal force results. While there was some improvement seen in terms of normal force from this geometry, axial force results were seen to be significantly less accurate. In addition, closer examination of both drawings of the wind tunnel model and a full-scale MK-82SE indicated that a forward-facing lip was unrealistic.

The second mesh has a rearward-facing lip, with trailing edges marked around the circumference of the store. Note that in order to ensure consistent operation of the solver, the facets connecting the edge of the lip to the fin section of the store were omitted. FlightStream does not function well if trailing edges are marked on a surface which does not come to a point, so this type of omission is common practice. While the addition of the forward-facing lip was an attempt to improve the realism of the mesh geometry, the addition of the rearward-facing lip was an attempt to allow FlightStream to measure lift on the forward fuselage. By adding the marked trailing edges along the lip, it was hoped that FlightStream would be able to treat the fuselage in the same way that it does a more typical aerodynamic surface, and therefore produce accurate normal force results. Instead, the opposite effect was seen, with normal force results being far more inaccurate than they had been with the unaltered model.

Following the rejection of the two meshes shown in Figure 4.4, a series of various small adjustments were made to the meshes, to include modifying the shape of the lips, removing

sections of the lips in small increments, and cutting out a section of the mesh and marking trailing edges on the forward portion of the resulting gap. Unfortunately, these tests cannot be expounded upon in any great detail here, as data corruption has rendered their meshes and simulation files inaccessible at the time of writing. However, their results can collectively be summarized as having produced no results which were an acceptable improvement on the unaltered mesh. This led to the conclusion that modifying the mesh geometry would not produce acceptably accurate results, and therefore a different approach was needed.

After the failure of adjusting the MK-82SE geometry, the application of fuselage vortex shedding was determined to be a viable solution to this limitation. The same series of runs was again conducted on the finest MK-82SE mesh, with adjustments applied to allow for the modeling of axisymmetric shedding. Specifically, a series of edges on the forward fuselage were marked as trailing edges, as shown in Figure 3.3. For a given angle of attack, the edges marked were on the leeward half of the model; for a positive angle of attack, edges were marked on the upper half, while for a negative angle of attack, edges were marked on the lower half. No edges were marked in the case of a zero degree angle of attack, as no vortices would be shed from the forward fuselage in that instance. The axisymmetric shedding method markedly improved results with regard to normal force, while maintaining a similar degree of accuracy for axial force, as shown in Figures 4.7 and 4.8. This improvement for normal force results can be directly attributed to the fact that allowing for the shedding of vorticity from the forward portion of the fuselage allows the Kutta-Joukowski theorem to capture fuselage loads.

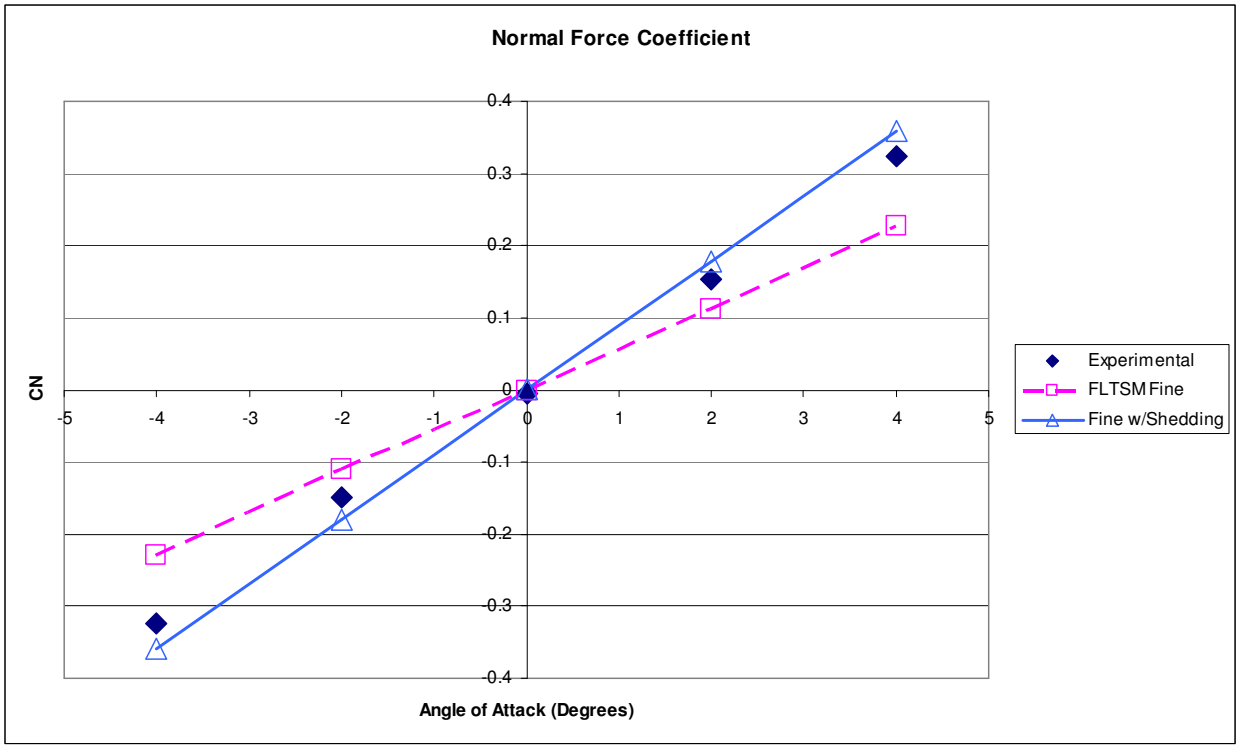

Figure 4.7: Normal Force Coefficient for MK-82SE with Fuselage Shedding [15]

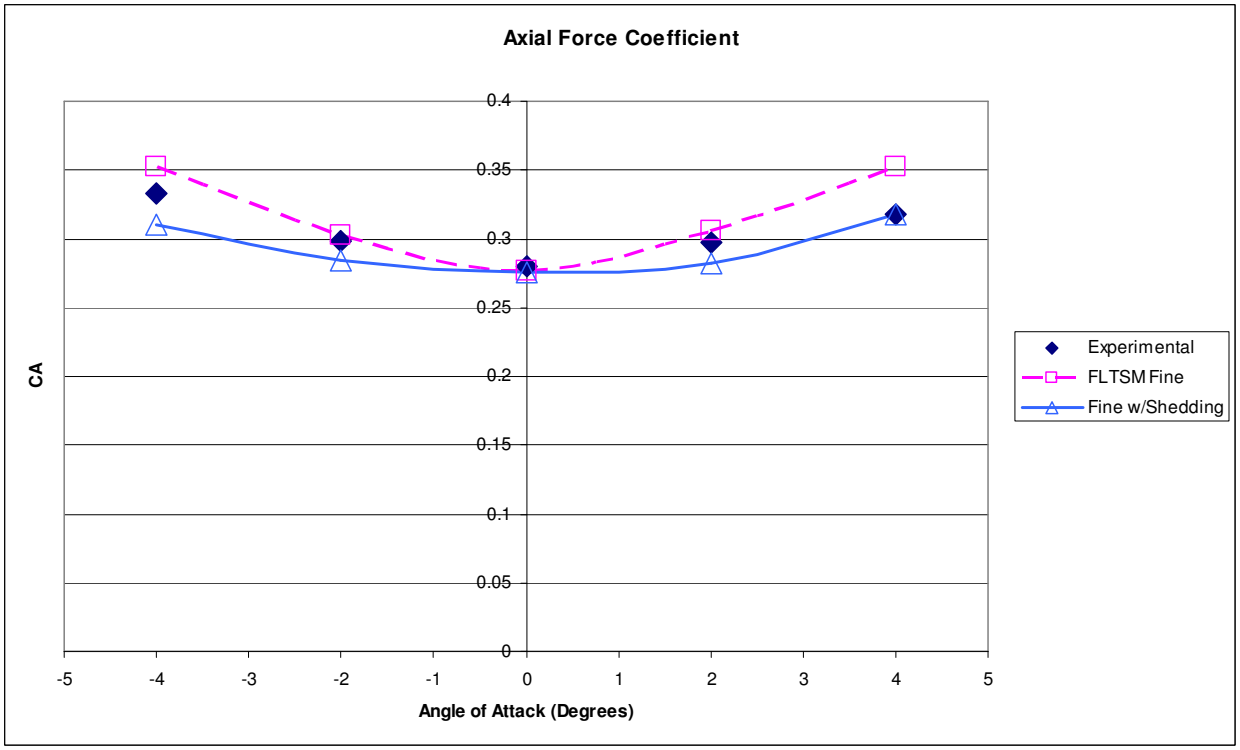

Figure 4.8: Axial Force Coefficient for MK-82SE with Fuselage Shedding [15]

Following these results, the model was judged have sufficient fidelity for the purposes of proximity testing. Note that all subsequent plots will display only the results for the finest mesh, as the two less fine meshes were seen to be less accurate and therefore their inclusion would only serve to make plots more difficult to interpret.

4.2: Proximity Analysis

The results of FlightStream analysis are shown in Figures 4.9 to 4.12. All runs were conducted at a freestream Mach number of 0.6, with the F-16 at a two degree angle of attack. This resulted in the MK-82SE being held at a negative one degree angle of attack relative to the incoming freestream. It was determined that because of this low angle, the vortices shed from the fuselage of the bomb were effectively nonexistent, and therefore no edges were marked on the front portion of the bomb. The computer used was a Dell Latitude D830 laptop equipped with a 2.5 gigahertz dual-core processor and 4 gigabytes of RAM. Both cores were used in parallel, and runtimes for a single simulation ranged from 45 to 60 minutes. For clarity, it should be emphasized that these runtimes are consequence of using an older, lower-power, model computer. The same simulation files have been seen to run much faster on more recent and powerful hardware.

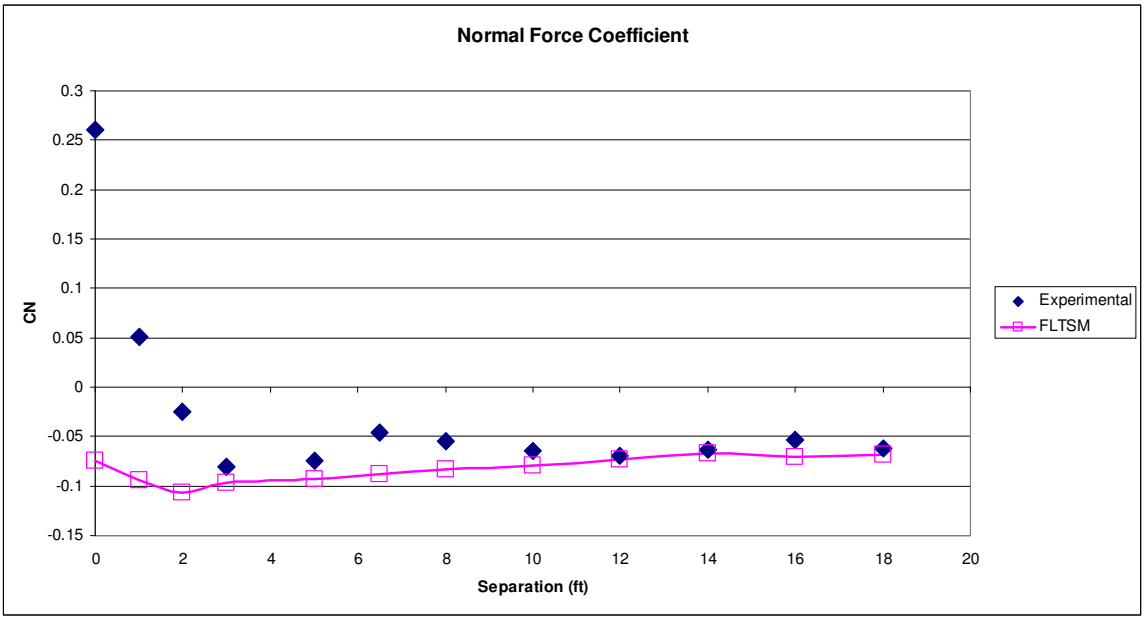

Figure 4.9: Normal Force Coefficient vs. Separation, Mach 0.6, Altitude 5000 ft [15]

Normal force coefficient, shown in Figure 4.9, has been seen to be generally in very good agreement with the experimental results. At positions closer than three feet from the carriage position, however, the FlightStream results noticeably diverge from the experimental results, though they do trend in the correct manner. This is possibly a result of the very close spacing between the MK-82SE and F-16 creating a Venturi effect. Given that the freestream is moving at a Mach number of 0.6, it is believed that this Venturi effect produces a region of locally transonic flow, a regime which FlightStream is not currently equipped to analyze. Further discussion of the possible causes and effects of this discrepancy will be addressed in section 4.3.

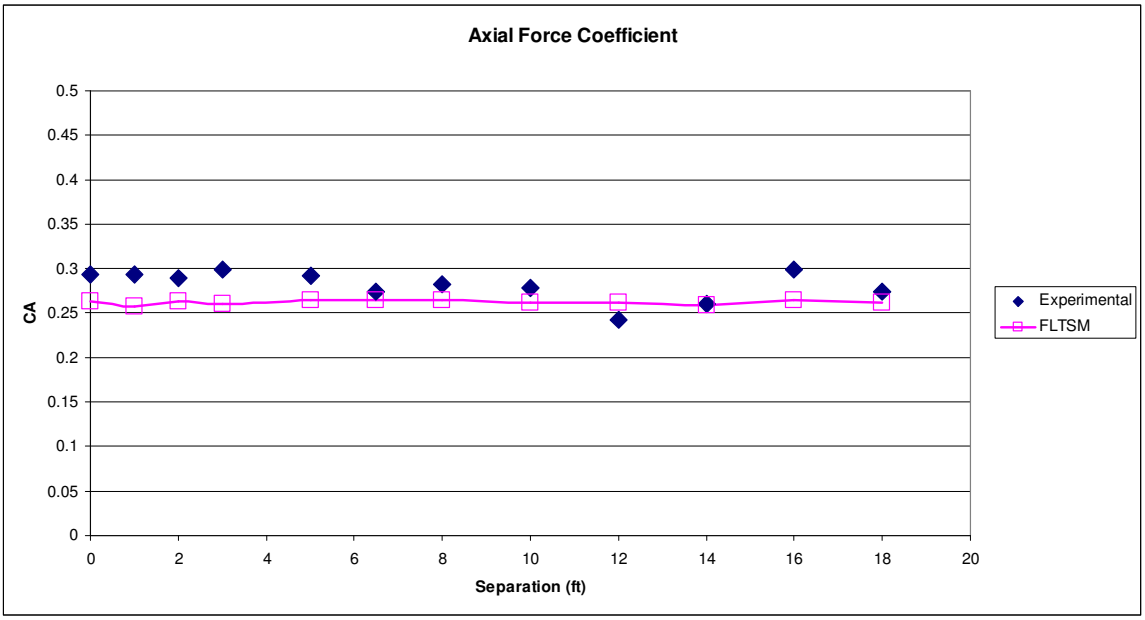

Figure 4.10: Axial Force Coefficient vs. Separation, Mach 0.6, Altitude 5000 ft [15]

The results for axial force coefficient proved to be very accurate to the experimental data, as shown in Figure 4.10. The noticeable departure from experimental results seen on the normal force plot was not present for axial force. This is likely due to the fact that for both FlightStream and the experimental data the axial force coefficient was fairly insensitive to the position of the store, meaning that the interaction which produced the upward trend in normal force was not a factor for axial force. Therefore, FlightStream was still able to produce accurate results in this case. The degree of accuracy for these results was encouraging, as determining axial force has, in the past, been particularly troublesome for the FlightStream solver, in addition to being one of the more difficult aerodynamic loads to accurately measure experimentally.

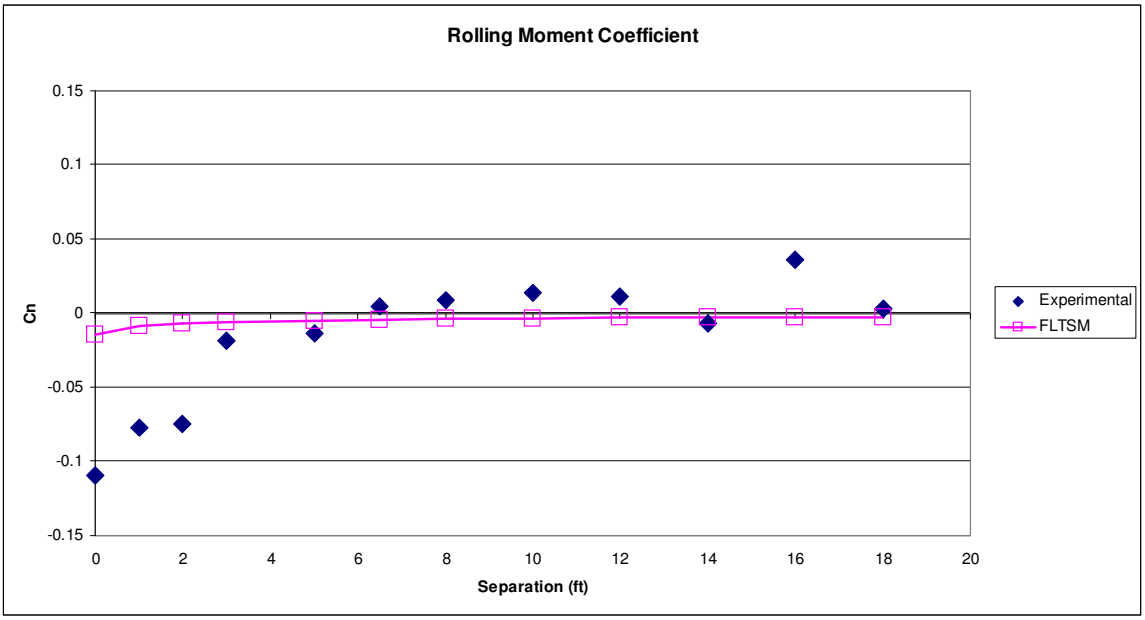

Figure 4.11: Rolling Moment Coefficient vs. Separation, Mach 0.6, Altitude 5000 ft [15]

Rolling moment results, shown in Figure 4.11, were in good agreement with both the experimental data and expected results. As would be expected for a symmetrical body, most of the FlightStream results were effectively zero, with slight numerical errors taken into consideration. At closer separation distances, a more significant negative rolling moment was observed, both in FlightStream and in the experimental data. The FlightStream data differs from the experimental data in the magnitude of the moment coefficient, although the negative trend is still apparent. It is probable that this is the result of the same effect that has been seen to disrupt other results. At the time of writing, FlightStream cannot accurately model this effect, though future versions are expected to have the capacity to do so. Section 4.3 will discuss in more detail possible causes of and solutions for this inaccuracy.

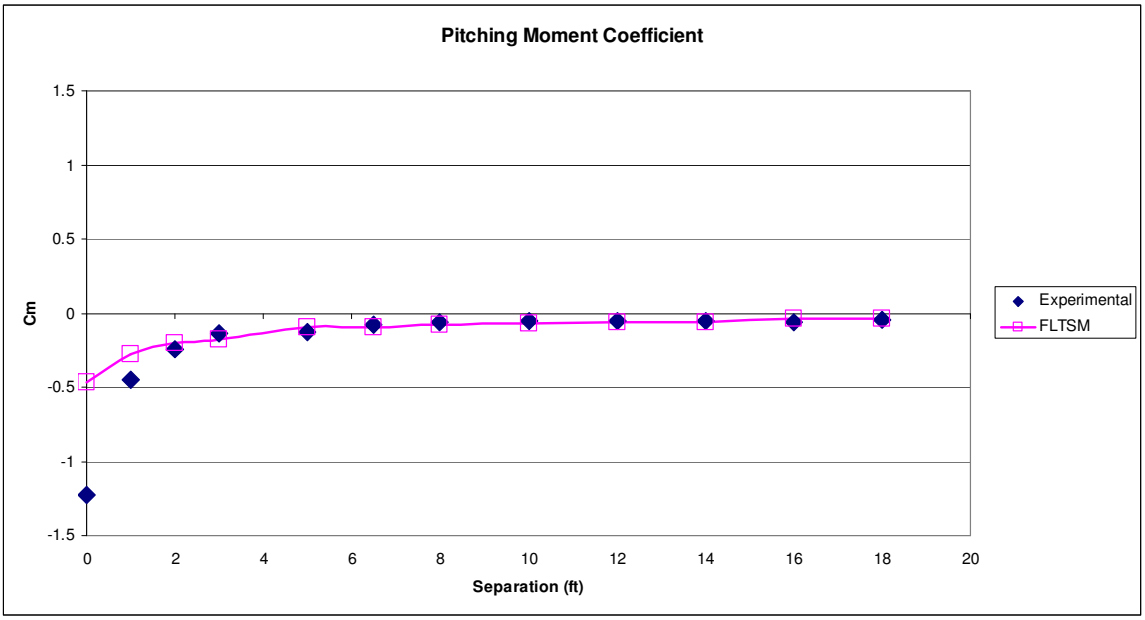

Figure 4.12: Pitching Moment Coefficient vs. Separation, Mach 0.6, Altitude 5000 ft [15]

Pitching moment results were very accurate for all positions except one, to the point that most of the data points shown in Figure 4.12 overlap. In the carriage position, the pitching moment seen in the wind tunnel was significantly more negative than the moment calculated by FlightStream. As with the discrepancies seen when testing the normal force coefficient, it has generally been believed that this may be a result of a Venturi effect producing a region of transonic flow at very close separation distances, though further possibilities will be explained in section 4.3.

Section 4.3: Possible Explanations and Solutions for Observed Discrepancies

The discrepancies seen near the aircraft have consistently been present throughout the entirety of the thesis work. A definitive explanation has not been determined as of yet, but a number of possible explanations have been proposed. Firstly, it is possible that the positioning of the aircraft and store produces a region of flow that FlightStream is not capable of analyzing. As has been briefly discussed previously, the possibility exists that the close spacing of the store

and aircraft produces a Venturi effect, driving airflow in the space between the two bodies into the transonic regime. As FlightStream is not currently intended to analyze transonic flow, this would be a reasonable cause for generating unreliable results. Future versions are expected to function more effectively in transonic flow, which should improve the quality of results seen here if the errors have in fact been caused by a locally transonic region.

Throughout most of the process of conducting work for this thesis, the generally accepted explanation for the incorrect results has been that a region of transonic flow is present. However, further possibilities exist which may more accurately describe the situation. In the original wind tunnel tests, the positioning of the store and aircraft may be producing a region in which unsteady flow patterns are present. FlightStream is only capable of modeling steady state flow conditions, meaning that any loads produced because of unsteady effects will not be accurately captured. While it is possible that further FlightStream development may eventually allow for modeling unsteady flow, at the time of writing this it remains a long-term goal. As such, it should be regarded as being beyond the capabilities of FlightStream for the foreseeable future.

An additional possibility unrelated to the behavior of the airflow is that the errors are a product of the nature of FlightStream itself as a vorticity-based solver. In order to successfully apply the Kutta-Joukowski theorem to calculate lift, the contour around which circulation is taken must be sufficiently large that the circulation on the body can be approximated by a point vortex. Multiple bodies being present in close enough proximity to each other can prevent this from being possible, leading to incorrect results.[30] In this case, erroneous results were consistently seen at separation distances closer than three feet, which approximately corresponds to half the length of the store. Because of this correspondence, it appears most likely that the errors seen can be attributed to this limitation. A possible solution for this method would be

modifying how FlightStream operates, forcing the solver to measure the loads on the store and aircraft using two distinct contours adjacent to each other, but which do not intersect each other. This operation would likely add runtime, but would be unnecessary after a certain separation distance. After that point, FlightStream could revert to its normal, less time-consuming, mode of operation.

An alternative method that could be applied to account for this condition would be switching the operation of FlightStream from that of a vorticity-based solver to that of a pressure-based solver. This would require that the mesh for both aircraft and store be refined to a higher degree than what is generally acceptable for FlightStream, and would therefore lead to longer runtimes than FlightStream would normally require. However, because a pressure-based solver is not subject to the contour requirement that hampers a vorticity-based solver, this should be an effective method to generate results at very close separation distances. As in the case of the contour adjustment method, running FlightStream as a pressure solver would become unnecessary after a point, allowing it to resume normal operation. Neither of the previously discussed alterations to FlightStream are planned to be implemented at the time of writing, though based on their relative simplicity and potential for significant improvements in the capabilities of the solver, they may be implemented at some point.

Chapter 5:

Analysis of Coefficients in a Plane near the Aircraft

Following the completion of the validation runs, an additional series of one hundred runs were conducted by placing the MK-82SE in a 10 foot by 10 foot grid pattern below the wing of the F-16. The results for these runs have been presented in figures 5.1 to 5.4. As before, all runs were conducted at a Mach number of 0.6, an altitude of 5000 feet above mean sea level, and an angle of attack of two degrees for the aircraft, which placed the store at a negative one degree angle of attack relative to the freestream. Note that in the figures, (0, 0) indicates the store carriage position, and the inboard pylon is approximately located at (4, 0). Coordinates are negative in the down and inboard directions, consistent with the aircraft-centered coordinate system used in FlightStream.

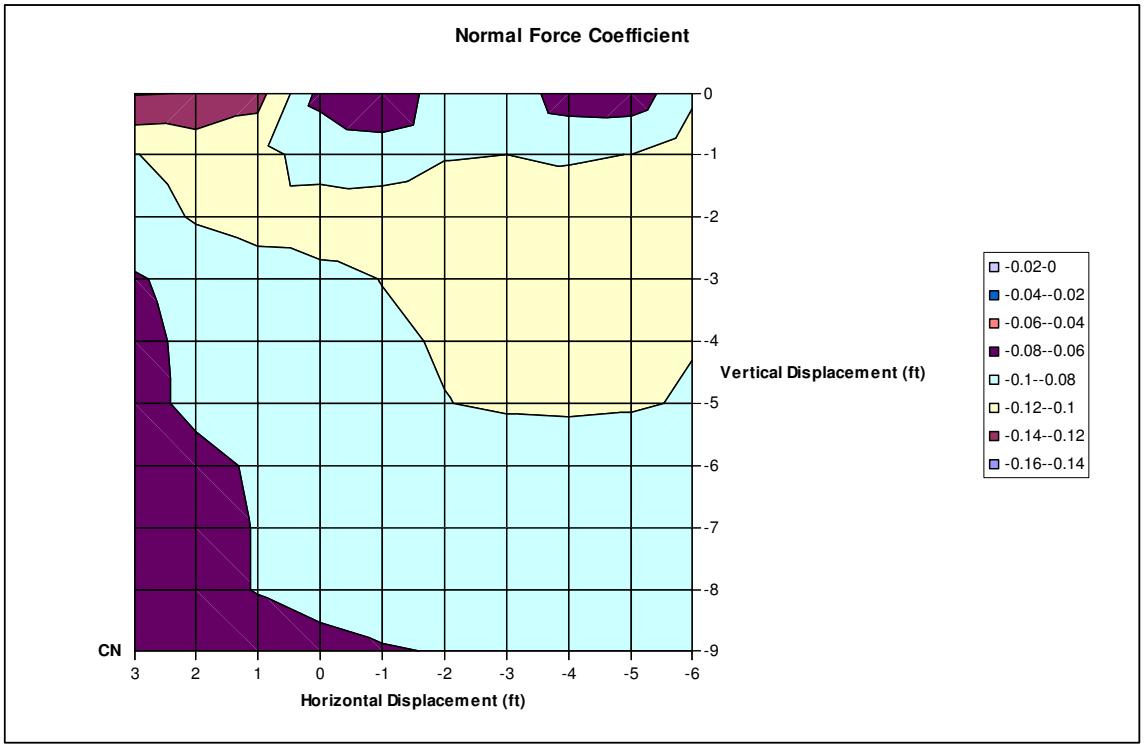

Figure 5.1: Normal Force Coefficient Map, Mach 0.6, Altitude 5000ft

Normal force coefficient is shown in Figure 5.1. The results are generally what would be expected, with the normal force being noticeably influenced by proximity to the underwing pylons, the wingtip-mounted Sidewinder and the fuselage, with influence decreasing as the store is moved farther from the aircraft.

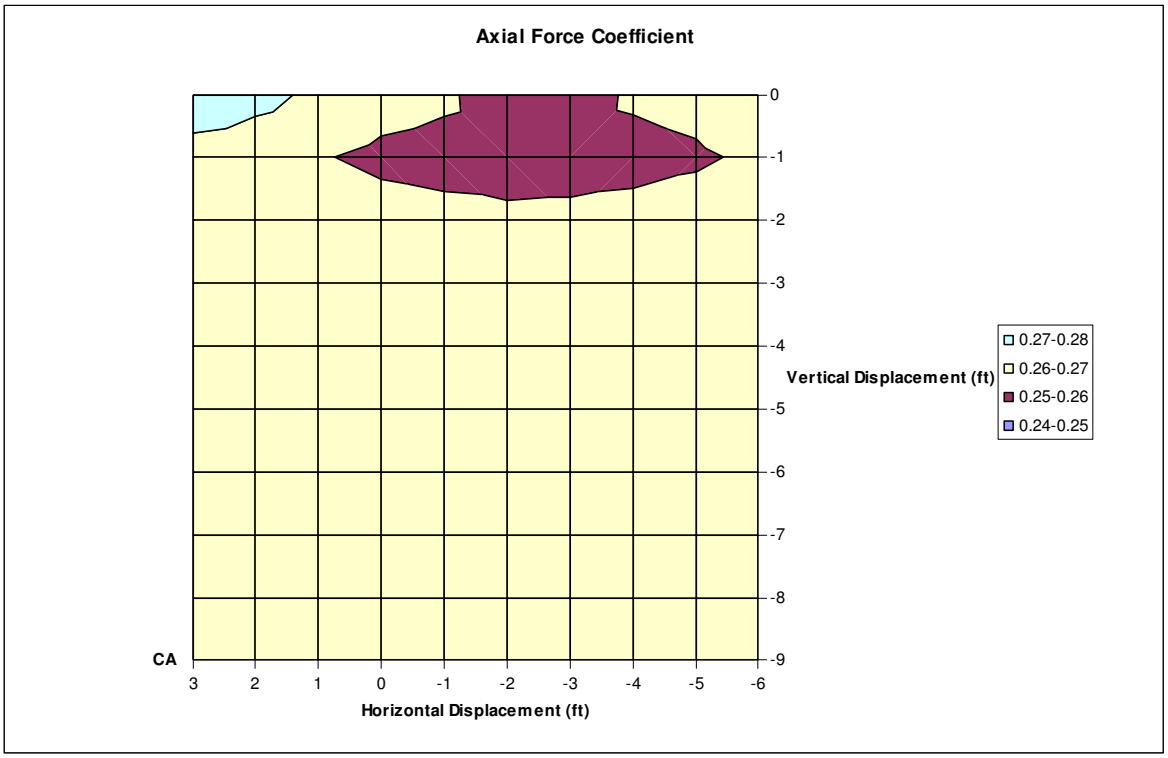

Figure 5.2: Axial Force Coefficient Map, Mach 0.6, Altitude 5000ft

Axial force results are shown in Figure 5.2. As was the case in the comparison against the experimental data, axial force is seen here to be significantly less sensitive to position than normal force. However, proximity to the underwing pylons and wingtip missile can be seen to notably influence axial force. As would be expected, the affect of the aircraft on the store decreases as the store is moved farther away from the aircraft, with axial force becoming generally uniform at vertical separations from the carriage position greater than three feet.

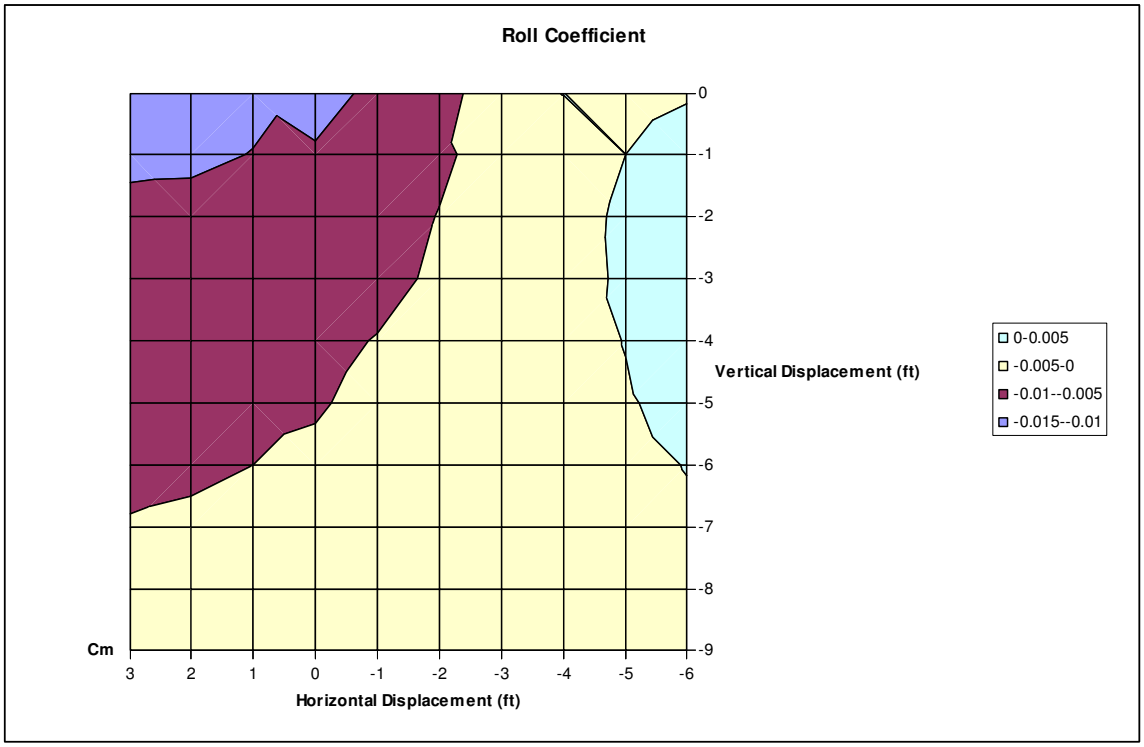

Figure 5.3: Rolling Moment Coefficient Map, Mach 0.6, Altitude 5000ft

As would be expected for a symmetric body, rolling coefficient was small and fairly consistent, ranging between -1.3×10^{-2} and 4.5×10^{-3} across the entire grid. Proximity to the fuselage can be seen to influence the roll results, but the wing and pylons were less of an influence than seen for axial and normal force. The small irregularity at (-4, 0) appears to be an effect of proximity to the inboard pylon. A similar spike is seen at the location of the outboard pylon, (0, 0), but the value at (-4, 0) appears more extreme on the graph because it is a positive value, 1×10^{-4}, surrounded by negative values. A value of zero located at (-5, 1) results in the narrow feature connecting (-4, 0) to the region of positive values at the right-hand side of the plot.

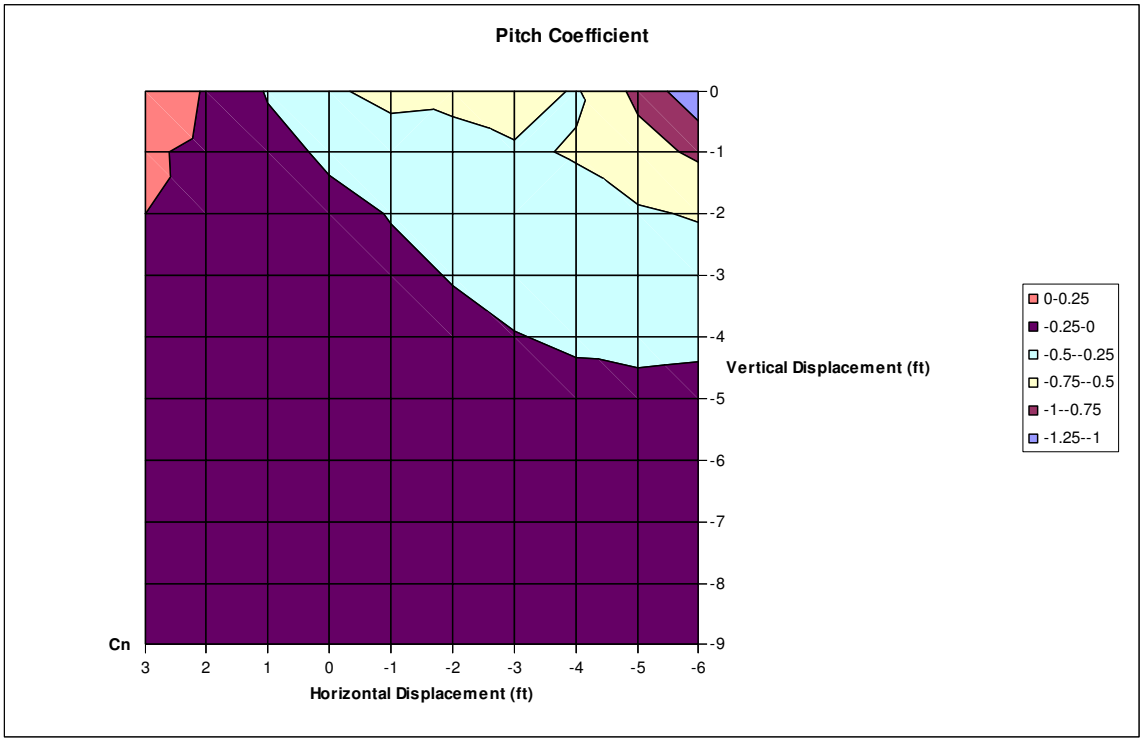

Figure 5.4: Pitching Moment Coefficient Map, Mach 0.6, Altitude 5000ft

Similar to the force coefficient results, pitching moment coefficient results were seen to be strongly influenced by proximity to the fuselage, pylons, and Sidewinder. As would be expected, pitching moment is seen to decrease as the store is moved farther away from the aircraft, eventually becoming fairly uniform.

Based on the above results, FlightStream has been generally seen to generate good results for regions around an aircraft. In the future, an expanded version of this operation may be used with surrogate modeling to form a comprehensive loads model for six-degree-of-freedom simulations.

Chapter 6:

Conclusions and Future Work

FlightStream has proven capable of analyzing proximity flight to a reasonable degree of accuracy, and a method for using FlightStream to efficiently produce a loads map for store separation has been demonstrated. After the implementation of fuselage vortex shedding, FlightStream has also been shown to be capable of calculating the loads on a body for which the fuselage generates a significant portion of lift. Though FlightStream currently produces inaccurate results for analysis of very close separation distances, it is possible that future developments will positively identify and eliminate the cause of this problem. At present, it is thought that these inaccuracies are caused either by the nature of a vorticity-based solver or a characteristic of the airflow between the aircraft and store. Possible solutions have been proposed in either case, and the continuing development of the solver will likely incorporate one or more of them. In the meantime, the speed and ease of use offered by FlightStream still make it a valuable tool in both the area of proximity flight and general aerodynamic analysis.

One of the preliminary areas for future work is the automated calculation for fuselage vortex shedding, to include the calculation of separation line positions. This is the subject of work currently being funded by NASA for the purposes of calculating high lift loads. This development is also applicable to store separation problems, such as the one examined here, as well as other scenarios in which a non-trivial portion of the lift on a body is generated by a fuselage. In this thesis, this problem was addressed manually, with separation lines being determined from experimental data and lift from the fuselage being manually summed to yield the total lift on the MK-82SE. Automation will significantly improve both the speed and the accuracy of this process.

Future work in the area of proximity flight and separation analysis may also include improvements specifically intended to address the inaccuracies seen in this work. Altering the operation of the solver to ensure bodies in very close proximity are analyzed using distinct circulation loops is expected to offer a significant improvement through small alterations to the solver. Similar improvements may also be seen from using a pressure-based solver to determine loads at close separation distances. No timeframe is currently set for the development of either of these features, but their relative simplicity should allow them to be implemented with relatively little difficulty. By contrast, the ability to simulate unsteady conditions remains a very long-term goal. While this may improve on the results seen here and would broaden the applicability of the solver as a whole, the complexity of modeling such conditions mean that adapting FlightStream to unsteady conditions is unfeasible for the time being.

Additional future work may include coupling FlightStream with a six degree of freedom solver for on-demand aerodynamic load calculations, enabling FlightStream to actively model the motion of a separating store throughout its trajectory. Further, linking FlightStream to both a six degree of freedom solver and an optimizer will enable FlightStream to find optimal store placements for a given combination of parent aircraft and stores.

References:

[1] Schindel, L.H., *"Store Separation"*, North Atlantic Treaty Organization Advisory Group for Aerospace Research and Development AGARDograph No. 202, June 1975.

[2]Cenko, A. et al. *"IHAAA Applications to Reducing Store Separation Flight Testing"*, AIAA Paper 2007-1653, February 2007

[3] Cenko, A. "*Lessons Learned in 30 years of Store Separation Testing"*, 47th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition. Orlando, Florida.

[4] Arnold, R.J. and Epstien, C.S. *"AGARD Flight Test Techniques Series Volume 5 on Store Separation Flight Testing"*, North Atlantic Treaty Organization Advisory Group for Aerospace Research and Development AGARDograph No. 300 Vol. 5, April 1986.

[5] Carman, J.B., Jr., Hill, D.W., Jr., and Christopher, J.P. *"Store Separation Testing Techniques at the Arnold Engineering Development Center. Volume II: Description of Captive Trajectory Store Separation Testing in the Aerodynamic Wind Tunnel (4T)"*, Arnold Engineering Development Center, Arnold Air Force Station, Tennessee. June 1980

[6]Maddox, A.R. *"Store Separation Trajectory Analysis",* Naval Weapons Center, China Lake, California. January 1980.
[7] Cenko, A., *"Captive Trajectory System Sting Effects on Store Loads"*, AIAA Paper 1994-0195, January 1994.

[8] Cenko, A. *"ACFD Applications to Store Separation – Status Report",* The Aeronautical Journal, October 2000.

[9] Powell, A.C., *"Computer Prediction of Store Aerodynamic Loading During Separation"*, Thesis, Air Force Institute of Technology, December 1982.

[10] Hansen, D.M., *"Store Separation Methodology Analysis"*, Thesis, Naval Postgraduate School, September 1991.

[11] Dillenius, M.F.E., et al. *"Extension of the Method for Predicting Six-Degree-of-Freedom Store Separation Trajectories Up to the Critical Speed to Include a Fuselage with Noncircular Cross Section. Volume I-Theoretical Methods and Comparisons With Experiment."* Nielsen Engineering and Research, Incorporated. November 1974.

[12] Maddox, A.R., *"Store Separation – A Review"*, Naval Air Systems Command, May 1981.

[13] Ryckebusch, C. et al. *"Evaluation of the Capablilties of CFD to Predict Store Trajectories from Attack Aircraft"*, AIAA Paper 2002-0279, January 2002.

[14] "Installing test article is detailed work", http://www.afmc.af.mil/News/Article-Display/Article/155750/installing-test-article-is-detailed-work/, November 2005. Retrieved 6/29/17.

[15] *"Aerodynamic Loads and Separation Characteristics of the BLU-27B, MK-82SE, and GBU-8 Weapons in the F-16 Aircraft Flow Field at Mach Numbers from 0.4 to 1.2."* Propulsion Wind Tunnel Facility, Arnold Engineering Development Center, Air Force Systems Command, Arnold Air Force Station, Tennessee 37389. 1976.

[16] https://commons.wikimedia.org/wiki/File:Mk._81,_Mk._82_Bombs_freigestellt.jpg September 2013. Retrieved 6/29/17.

[17] Ahuja, V., *"Aerodynamic Loads Over Arbitrary Bodies by Method of Integrated Circulation"*, Auburn University, 2013

[18] Ahuja, V., and Hartfield, R.J., *"Aerodynamic Loads Over Arbitrary Bodies by Method of Integrated Circulation"*, Journal of Aircraft, Vol. 53, No. 6 (2016), pp. 1719-1730.

[19] Ahuja, V., Burkhalter, J., and Hartfield, R.J., *"Optimizing Engine Placement on an Aircraft Wing using Bio-mimetic Optimization and FlightStream ™"*, AIAA 2017-0235 (Engine Placement), January 2017

[20] Anderson, J.D. Jr., *"Fundamentals of Aerodynamics"*, Fifth Edition, McGraw-Hill, Inc., ISBN 978-0-07-339810-5, 2010

[21]Derbyshire, T. and Sidwell, K. W., *"PAN AIR Summary Document* (*Version 1.0)*", NASA CR-3250, March 1990.

[22] Ahuja V., and Hartfield R. J., *"Optimization of UAV Geometries for Aerodynamic Performance Using Advanced Paneling Aero-Propulsive Paneling Models and Genetic Algorithms"*, 6th AIAA Multidisciplinary Design Optimization Specialist Conference, Orlando, Florida, April 2010

[23]Epton, M.A, and Magnus, A.E., *"PAN AIR- A Computer Program for Predicting Subsonic or Supersonic Linear Potential Flows About Arbitrary Configurations Using a Higher Order Panel Method"*, Version-I, Theory Document, NASA CR-3251, March 1990.

[24]Maskew, B., *"Program VSAERO, A Computer Program for Calculating the Nonlinear Aerodynamic Characteristics of Arbitrary Configurations, User's Manual"*, NASA CR-166476, November 1982

[25]Katz, J. and Plotkin, A., *"Low-Speed Aerodynamics: From Wing Theory to Panel Methods"*, McGraw-Hill Inc., ISB-0-07-050446-6, 1991

[26] Anderson, J.D. Jr., "*Introduction to Flight"*, Seventh Edition, McGraw-Hill, Inc. ISBN 978-0-07-338024-7, 2012

[27]Silvio B. Angelucci. *"A Multivortex Method for Axisymmetric Bodies at Angle of Attack"*, Journal of Aircraft, Vol. 8, No. 12 (1971), pp. 959-966.

[28] Hahn, A.S., *"Vehicle Sketch Pad: A Parametric Geometry Modeler for Conceptual Aircraft Design"*, AIAA Paper 2010-657, January 2010.

[29] Olson, E. D. and Alberton, C.W. *"Aircraft High-lift Aerodynamic Analysis Using a Surface-Vorticity Solver"*, 55th AIAA Aerospace Sciences Meeting. San Diego, California

[30] Chiu, P. K. and Wirz, R. E. *"Effects of airfoil thickness in potential flow over biplanes"*, International Journal of Aerospace Engineering, 2015