ADVANCES IN HIGH-SPEED STORE SEPARATION FOR UPWARD-EJECTED STORES AND DYNAMIC CAVITY DOORS

By

Timothy P. Wolfe

Kidambi Sreenivas
Professor, Mechanical Engineering
(Chair)

Charles H. Margraves
Associate Professor, Mechanical Engineering
(Committee Member)

James C. Newman III
Department Head, Mechanical Engineering
(Committee Member)

Distribution is Unlimited/AFRL-2023-5139; Cleared 10/26/2023

ADVANCES IN HIGH-SPEED STORE SEPARATION FOR UPWARD-EJECTED STORES AND DYNAMIC CAVITY DOORS

By

Timothy P. Wolfe

A Thesis Submitted to the Faculty of the University of
Tennessee at Chattanooga in Partial
Fulfillment of the Requirements of the Degree
of Master of Science of Engineering

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

May 2024

ABSTRACT

As technology continues to evolve for aircraft-deployed weapons, computational approaches to store separation analysis face new challenges. Anticipating vehicle designs that require upward store ejection, this study uses computational fluid dynamics and 6DoF motion analysis to predict such store trajectories in high-speed flow. Several store designs are analyzed at various ejection velocities in Mach 4 and Mach 6 conditions. The trajectory results show that streamlined store geometries may not induce sufficient drag to clear the aircraft after ejection. However, store designs with drag-enhancing features show potential for safe separation trajectories. To account for the unsteady effects of a cavity door which opens just prior to store ejection, dynamic cavity door simulations are presented, comparing their results to a quasi-static approach. The results show that such cavity opening effects impact pressure loads on the store, which may be critical for accurate store trajectory predictions.

DEDICATION

I dedicate this work to my wife, Rebekah. Throughout this academic pursuit, you have been a constant source of encouragement, reminding me of what we can achieve together. You have made countless sacrifices on my behalf – without them, this endeavor would not have been possible, and because of them, we prepare a brighter future for our family. I thank you, and I love you.

ACKNOWLEDGEMENTS

I would like to acknowledge and thank those who have contributed to the success of this work. Your efforts, creative ideas, and continual support have made this possible.

Thank you, Dr. Kidambi Sreenivas, for the many months of mentorship you have provided. All that I have learned from your expertise makes me a stronger professional, and I hope that our field of study strengthens as a result.

I am grateful for Dr. Charles Margraves and Dr. James Newman for their role on my thesis committee alongside Dr. Sreenivas. My appreciation also extends to the rest of the UTC Department of Mechanical Engineering – the professors and staff whose work has equipped me for the career opportunities ahead. Thanks are also due to the UTC Research Institute at the University of Tennessee Chattanooga for providing computational resources.

For their contributions, I would like to thank Dr. Jon Poggie (Purdue University) and Dr. Brett Hauber (University of Dayton Research Institute). Dr. Poggie's suggestion for the backward cone configuration accelerated the success of the study. Dr. Hauber has provided designs and mass properties for various store concepts. Thank you, also, to Ralph Noack (Celeritas Simulation Technology, LLC) for your support to apply Suggar++ to this application.

This work would not have been possible without the funding provided through the Design of Resilient Hypersonic Vehicle Structures (DRHVS) program by the Air Force Research Laboratory, Aerospace System Directorate, High Speed Systems Division (AFRL/RQH), Prime Contract Number FA8650-21-C-2408 with Dr. Jerrod Hofferth as the technical monitor.

TABLE OF CONTENTS

ABSTRACT iii

DEDICATION iv

ACKNOWLEDGEMENTS v

LIST OF TABLES viii

LIST OF FIGURES ix

LIST OF ABBREVIATIONS xiv

LIST OF SYMBOLS xv

CHAPTER

1. INTRODUCTION 1

1.1 Background 1
1.2 Objectives 2

2. LITERATURE REVIEW 4

2.1 Store Separation Historical Overview 4
2.2 Cart3D Review for 6DoF Simulations 6
2.3 The Effects of Doors in Weapons Bay Cavities 8
Experimental Case Study 9
Computational Case Study 1 13
Computational Case Study 2 16
Computational Case Study 3 19

3. TRAJECTORY PREDICTIONS OF UPWARD-EJECTED STORES 24

3.1 Problem Setup 25
Test Case Descriptions 25
Freestream Conditions, Mass Properties, and Non-Dimensionalization 28
3.2 Grid Generation 29
3.3 Results and Discussion 31
Simple Cone 31

Mach 4 Conditions – Simple Cone 32
Mach 6 Conditions – Simple Cone 35
Reversed Cone 41
Mach 4 Conditions – Reversed Cone 41
Mach 6 Conditions – Reversed Cone 44
Modified Store – Solid Disk 48
Modified Store – Ring Disk 52
6DoF Time Step Studies 57
3.4 Concluding Remarks and Future Work 66

4. DYNAMIC CAVITY SIMULATIONS 68

4.1 Problem Definition and Setup 69
Geometry Definition 69
Freestream Conditions 70
Computational Methods 71
4.2 Review of Overset Grid Assembly 72
4.3 Domain Discretization 75
Vehicle Grid 75
Door Grid 77
Store Grid 78
Background Grid 79
4.4 Lessons Learned with an Empty Cavity 80
Steady-State Quiescent Cavity Solution 80
Dynamic Door Opening Time Study 86
4.5 Challenges to the Overset Methodology with a Store in Carriage 90
Steady-State Solutions Using Yoga 91
Additional Overset Challenges with a Dynamically Opening Door 95
Dynamic Solution of an Ejecting Store 98
4.6 Cavity and Door Simulations Using Suggar++ 102
Steady State Solution Using Suggar++ 104
Discussion of Results for a Dynamically Opening Door 108
Comparison of Overset and Single Mesh Static Solutions 109
Comparison of Dynamic and Quasi-Static Solutions 112

5. CONCLUSIONS 123

5.1 Summary of Results 123
5.2 Future Work 124

REFERENCES 126

VITA 130

LIST OF TABLES

Table 3.1 Test Cases 27

Table 3.2 Store Mass Properties 29

Table 3.3 Grid Statistics 30

Table 4.1 Dynamic Simulation Time Step Data 87

Table 4.2 Pressure and Temperature Data Probe Positions 116

LIST OF FIGURES

Figure 2.1 6DoF trajectory analysis workflow for each time level, *n* [17] 8

Figure 2.2 Experimental setup of a rectangular cavity with a downstream sliding door [19] 9

Figure 2.3 Pressure data on the front cavity wall, (a) signal frequencies and (b) strengths [19] 11

Figure 2.4 Coefficient of pressure (average and range) for quasi-static and dynamic cases [19] 12

Figure 2.5 Rectangular cavity and door configurations for wedge and sliding doors [20] 14

Figure 2.6 Streamlines for each cavity/door configuration [20] 15

Figure 2.7 Streamwise velocity contour plots with streamlines at the mid-span cross-section [20]....... 15

Figure 2.8 Streamwise velocity contour plots with streamlines at the half-depth cross-section [20]..... 16

Figure 2.9 Geometry of dynamic opening doors [21] 17

Figure 2.10 RMS and mean pressure over the depth of a rectangular cavity [21].................................. 18

Figure 2.11 Streamwise velocity contour plots with (right) and without (left) cavity doors [21] 18

Figure 2.12 Mean pressure on the cavity floor with (right) and without (left) cavity doors [21] 18

Figure 2.13 RMS surface pressure on the cavity floor as the doors open in a dynamic simulation [21] ... 19

Figure 2.14 Mach number contours with and without a store, with the doors 5° and 20° open [22]....... 21

Figure 2.15 Vertical force (left) and pitching moment (right) as the door opens [22] 22

Figure 3.1 Geometry of high-speed carrier vehicle and store.. 25

Figure 3.2 Geometry of four store designs, with close-up of “Ring Disk” (right) 26

Figure 3.3 Refined grids for the (a) simple cone, (b) reversed cone, (c) solid disk, and (d) ring disk 31

Figure 3.4 Mach number contours at time, t = 0, in Mach 4 freestream conditions 32

Figure 3.5 Store trajectories – Case A1: Mach 4, 3m/s (left) and Case A2: Mach 4, 5m/s (right) 33

Figure 3.6 Store trajectory – Case A3: Mach 4, 8m/s 33

Figure 3.7 Store trajectory – Case A4: Mach 4, 10m/s ... 34

Figure 3.8 Center of mass trajectories for Cases A1 through A4 ... 35

Figure 3.9 Mach number contours at time, t = 0, in Mach 6 freestream conditions ... 36

Figure 3.10 Store trajectories – Case A5: Mach 6, 3m/s (left) and Case A6: Mach 6, 5m/s (right) ... 36

Figure 3.11 Store trajectory – Case A7: Mach 6, 8m/s ... 37

Figure 3.12 Store trajectory – Case A8: Mach 6, 10m/s ... 37

Figure 3.13 Center of mass trajectories for Cases A1 through A8 ... 38

Figure 3.14 Streamwise “x” (left) and vertical “y” (right) trajectories for Cases A1 through A8 ... 39

Figure 3.15 Time t = 0.503 s in Mach 4 and Mach 6 freestream conditions, 8 m/s ejection velocity ... 40

Figure 3.16 Store pitch for Cases A1 through A8, where negative values are nose-down pitch ... 40

Figure 3.17 Mach number contours at time, t = 0, in Mach 4 freestream conditions ... 42

Figure 3.18 Store trajectory – Case B1: Mach 4, 3m/s ... 42

Figure 3.19 Store trajectory – Case B2: Mach 4, 5m/s ... 43

Figure 3.20 Store trajectory – Case B3: Mach 4, 8m/s ... 43

Figure 3.21 Center of mass trajectories for Cases B1 through B3 ... 43

Figure 3.22 Mach number contours at time, t = 0, in Mach 6 freestream conditions ... 44

Figure 3.23 Store trajectory – Case B4: Mach 6, 3m/s ... 45

Figure 3.24 Store trajectory – Case B5: Mach 6, 5m/s ... 45

Figure 3.25 Store trajectory – Case B6: Mach 6, 8m/s ... 45

Figure 3.26 Center of mass trajectories for Cases B1 through B6 ... 46

Figure 3.27 Streamwise “x” (left) and vertical “y” (right) trajectories for Cases B1 through A6 ... 48

Figure 3.28 Mach number contours at time, t = 0, in Mach 4 freestream conditions ... 49

Figure 3.29 Store trajectory – Case C1: Mach 4, 3m/s ... 50

Figure 3.30 Store trajectory – Case C2: Mach 6, 3m/s ... 50

Figure 3.31 Center of mass trajectories for Cases C1 and C2 ... 50

Figure 3.32 Store pitch and yaw versus streamwise “x” position for Cases C1 (left) and C2 (right) 51

Figure 3.33 Store pitch time history for Cases C1 and C2 52

Figure 3.34 Mach number contours at time, t = 0, in Mach 4 freestream conditions 53

Figure 3.35 Store trajectory – Case D1: Mach 4, 3m/s.. 53

Figure 3.36 Store trajectory – Case D2: Mach 6, 3m/s.. 54

Figure 3.37 Center of mass trajectories for Cases D1 and D2 54

Figure 3.38 Center of mass trajectories for Cases C1 and D1 (left) and Cases C2 and D2 (right) 55

Figure 3.39 Store pitch and yaw versus streamwise “x” position for Cases D1 (left) and D2 (right) 55

Figure 3.40 Store pitch time history for Cases D1 and D2.. 56

Figure 3.41 Store pitch time history for Cases C1 and D1 (left) and Cases C2 and D2 (right) 56

Figure 3.42 Simple cone trajectory comparisons using time steps $\Delta t*$ = 1 and $\Delta t*$ = 10 58

Figure 3.43 Simple cone trajectory comparisons using time steps $\Delta t*$ = 0.1 and $\Delta t*$ = 1 59

Figure 3.44 Simple cone “x” and “y” position time histories using time steps $\Delta t*$ = 1 and $\Delta t*$ = 10 60

Figure 3.45 Simple cone “x” and “y” position time histories using time steps $\Delta t*$ = 0.1 and $\Delta t*$ = 1 61

Figure 3.46 Reversed cone trajectory comparisons using time steps $\Delta t*$ = 1 and $\Delta t*$ = 10...................... 62

Figure 3.47 Reversed cone “x” and “y” position time histories using time steps $\Delta t*$ = 1 and $\Delta t*$ = 10..... 63

Figure 3.48 Solid disk store trajectory comparisons using time steps $\Delta t*$ = 1 and $\Delta t*$ = 10..................... 64

Figure 3.49 Solid disk “x” and “y” position time histories using time steps $\Delta t*$ = 1 and $\Delta t*$ = 10............ 65

Figure 3.50 Ring disk store trajectory comparisons using time steps $\Delta t*$ = 1 and $\Delta t*$ = 10...................... 65

Figure 3.51 Ring disk store “x” and “y” position time histories using time steps $\Delta t*$ = 1 and $\Delta t*$ = 10..... 66

Figure 4.1 Vehicle and cavity (blue) with a downstream sliding door (green) and store (purple)........... 70

Figure 4.2 Mid-span section view of the vehicle component grid 76

Figure 4.3 A closer view of the refined cavity volume in the vehicle component grid 77

Figure 4.4 Mid-span section view of the door component grid.. 78

Figure 4.5 Mid-span cut view of the store component grid.. 79

Figure 4.6 Mid-span cut view of the background grid 80

Figure 4.7 Mach number contours for solution initialization with zero velocity around the cavity 81

Figure 4.8 Mid-span Mach number contours for the steady-state solution 83

Figure 4.9 Overset node statuses where vehicle and door component grids are in contact 84

Figure 4.10 Orphan nodes (red) occur where viscous vehicle and door grids intersect 85

Figure 4.11 Mach number contours for the steady-state solution with a closed cavity door 88

Figure 4.12 Plot of residuals for the steady-state solution with a closed cavity door 88

Figure 4.13 Coefficient of pressure on the rear cavity wall for the "1K", "5K", and "50K" resolutions 89

Figure 4.14 Store component grid which has been added to the composite assembly 92

Figure 4.15 A closer view of the refinement zone in the store component grid 92

Figure 4.16 Mach number contours of a steady-state solution with a closed cavity door 93

Figure 4.17 Modified store grid such that grid boundaries are confined to the cavity space 94

Figure 4.18 Steady-state solution before and after modifying the store component grid 95

Figure 4.19 Composite grid composition at time, t = 0.56s, colored by component grid ID 96

Figure 4.20 False velocity spikes in the dynamic simulation 97

Figure 4.21 Modified door and store components grids 98

Figure 4.22 Mach contours in a dynamic simulation with the door 25, 50, 75, and 100% open 98

Figure 4.23 Modified vehicle component grid with extended cavity refinement zone 100

Figure 4.24 Mach contours as the store is ejected out of the cavity in a dynamic simulation 100

Figure 4.25 Location of low-density solution failure behind the store as it is ejected 101

Figure 4.26 Mach contours for the steady-state solution shown on the door component grid only 103

Figure 4.27 Grid composition using Yoga with a closed door, colored by component grid ID 104

Figure 4.28 Grid composition using Suggar++ with a closed door 104

Figure 4.29 Steady-state solution using Suggar++ and fully overlapping component grids 105

Figure 4.30 Mach number contours for the steady-state solution using Suggar++ 107

Figure 4.31 Grid composition using Suggar++ at t = 0.56s 108

Figure 4.32 Mach contours for steady-state solutions using each of the three grid configurations 110

Figure 4.33 Store force coefficients from steady-state solutions using different grid configurations 111

Figure 4.34 Mach number contours for quasi-static and dynamic door opening simulations 114

Figure 4.35 Mach number contours for quasi-static and dynamic simulations with 100% open door ... 114

Figure 4.36 Numbered locations for pressure (black) and temperature (red) data probes 115

Figure 4.37 Dynamic pressure history and quasi-static data points at probe locations 116

Figure 4.38 Dynamic store force history and quasi-static data points in the x-, y-, and z-directions 118

Figure 4.39 Pressure distribution on the store when the door is 100% open 119

Figure 4.40 Temperature contours for quasi-static and dynamic door opening simulations 120

Figure 4.41 Dynamic temperature history and quasi-static data points at probe locations 121

LIST OF ABBREVIATIONS

6DoF	6 Degrees of Freedom
BAM6QT	Boeing/AFOSR Mach-6 Quiet Tunnel
BDF2	Backward Difference Formula (2nd Order)
CFD	Computational Fluid Dynamics
CFL	Courant–Friedrichs–Lewy Number
CSD	Computational Structural Dynamics
CTS	Captive Trajectory System
DCI	Domain Connectivity Information
DDES	Delayed Detached Eddy Simulation
GMP	Geometry Manipulation Protocol
IHAAA	Institute for HPC Applications to Air Armament
JDAM	Joint Direct Attack Munition
KEC	Kinetic Energy Consistent
LES	Large Eddy Simulation
NASA	National Aeronautics and Space Administration
RANS	Reynolds-Averaged Navier Stokes
RMS	Root Mean Square
SAS	Scale Adaptive Simulation
TALon	Tennessee Aerothermodynamics Laboratory
UTSI	University of Tennessee Space Institute

LIST OF SYMBOLS

$(\ )_{\infty}$	Freestream value
$(\ )_{x,y,z}$	Directional component value
a	Speed of sound
C	Sutherland's constant
C_P	Pressure coefficient
$C_{x,y,z}$	Force coefficient (x, y, z direction)
D	Depth
e_t	Total internal energy
f_{ac}	Acoustic frequency
f_R	Rossiter frequency
$F_{x,y,z}$	Force (x, y, z direction)
g	Gravitational acceleration
I	Moment of inertia
L	Length
m	Mass
M	Mach number
M'	Modified Mach number
$M_{x,y,z}$	Moment (x, y, z direction)
n	Frequency mode number
P	Pressure

Pr	Prandtl number
r	Temperature recovery factor
R	Gas constant
Re_x	Reynolds number (per unit length)
Str	Strouhal number
T	Temperature
T_0	Freestream stagnation temperature
T_∞	Freestream static temperature
T_c	Cavity temperature
T_{ref}	Reference temperature
T_{wall}	Wall temperature
t	Time
Δt	Time step
U_∞	Freestream velocity
u	X-direction velocity
v	Velocity
W	Width
x, y, z	Coordinate direction/position
α	Empirically tuned constant
γ	Ratio of specific heats
κ	Empirically tuned constant
μ	Dynamic viscosity
ρ	Density
ϕ	Angular door position

CHAPTER 1

INTRODUCTION

1.1 Background

As weapons stores are certified for deployment from military aircraft, it is important that the trajectory of the store away from the carrier vehicle is carefully studied to ensure safe separation. Within an approved flight envelope, there must be a high degree of certainty that the store will not collide with the vehicle after its release. Over the years, there have been many developments in the techniques used to study store separation. Each development in the field seeks to improve the accuracy of trajectory predictions and reduce the time and resources required for the aircraft/store certification process. Techniques used to study store separation include flight testing, wind tunnel experimentation, and the use of computational fluid dynamics (CFD). Each of these methods is continuously improved to meet the changing needs of evolving aircraft technology.

Considering the inherent cost and safety risks involved in flight testing, as well as the time and cost associated with wind tunnel experimentation, an approach to store separation analysis using CFD provides a safe and efficient alternative. However, CFD methods must be applied carefully to ensure quality store trajectory predictions without excessive computational cost. The application of CFD is challenged by the evolution of aircraft technology as it pertains to weapons carriage and ejection. For decades, store separation analysis predicted trajectories for externally mounted stores dropped from aircraft wings. The next generation of aircraft features internal weapons bay cavities which are opened during flight and from which a weapons store is ejected. The inherent unsteadiness and viscous effects of cavity flows have presented new challenges for CFD to address.

This work, however, looks forward to the potential of a new store release configuration – one in which the store is ejected upward from the vehicle out of a sky-facing cavity. The motivation for such a design applies to high-speed scramjet designs where spatial constraints may leave no room on the underside of the vehicle to include a traditional weapons bay cavity. Instead, the weapons bay must be designed into the top of the vehicle, and the store ejected upward. This presents similar challenges to the internal cavities studied over the last couple of decades, but it also presents new challenges to ensure safe and reliable store clearance while gravity tends to bring the store back toward the vehicle.

1.2 Objectives

This work provides exploratory computational studies that investigate the main effects of the upward ejection of weapons stores in high-speed flow. Two primary objectives are pursued.

The first objective is to simulate trajectories of a store being ejected upward from an aircraft at various ejection velocities into Mach 4 and Mach 6 flow. The investigation to meet this objective is presented in Chapter 3. The store and vehicle models are simplified conical geometries. The weapons bay cavity is not modeled for these computations, and it is assumed at the start of the simulation that the store has already ejected to a position slightly above the vehicle and has a specified velocity. The freestream Mach numbers are chosen to match conditions in UTSI's Mach 4 TALon wind tunnel facility [1] and Purdue University's Mach 6 BAM6QT tunnel [2].

Early in the investigation, it becomes clear that safe store separation is not attainable for the given configuration at reasonable ejection velocities. The store simply falls back onto the vehicle. Efforts were then focused on simulating trajectories for alternate store designs which were modified for enhanced drag to help push the store downstream of the vehicle before it falls. These design modifications resulted in angular oscillations in the store's attitude, which were investigated further.

The second objective of this work is to investigate the dynamic effects of opening a cavity in high-speed flow with a store in carriage. The computational investigation pursuant of this objective is the subject of Chapter 4. The trajectory predictions of Chapter 3 provide useful preliminary insights for stores ejected upward, but they also assume that the store has safely left the cavity, which is not modeled. Cavity flow is characterized by highly unsteady and resonant behavior which may influence a store's trajectory. These cavity effects should be considered for upward store ejection.

One could note, though, that studies which consider cavity flow effects in store separation often only consider the quasi-steady resonant state of cavity flow when the cavity has been open for an extended time. This is not the case in real store ejection where a cavity door opens, the store is ejected, and the cavity is closed in a matter of seconds. A dynamic analysis is warranted that computationally simulates an opening door in real time with a store in carriage prior to ejection. Such a simulation is useful for two main purposes. It provides the proper initial conditions for store ejection and trajectory analysis, and it provides a dynamic history of pressure and temperature loads in the cavity which must be considered in mechanical design.

The contents of Chapter 4 pursue this objective by considering a simple rectangular cavity with a downstream sliding door in Mach 4 flow. First, steady solutions are obtained with a closed door which seals the cavity and ensures a quiescent solution within the cavity at the start of the dynamic simulations. Moving door simulations are then conducted using overset grid assembly techniques to open the door. This configuration of multiple bodies (vehicle, door, and store) in close proximity proves to be a challenge to overset methodologies. The challenges faced and lessons learned are presented using multiple overset assembly codes – NASA's Yoga assembler and the commercial Suggar++ code. The results of dynamic door opening simulations are compared to solutions with the door in various static positions, including effects on store loads. The methodological development to conduct these computational experiments lays the groundwork for future analysis of stores ejected into high-speed flow.

CHAPTER 2

LITERATURE REVIEW

The literature that is available concerning store separation analysis in various applications is abundant. This review considers highlighted publications that meet three objectives. First, they provide a historical overview of store separation analysis techniques. Second, they review the computational tools used for trajectory prediction in this study – specifically, NASA's Cart3D flow solver. Third, they describe the state-of-the-art for simulating the effects of doors in weapons bay cavities, pertinent to the investigations of Chapter 4.

2.1 Store Separation Historical Overview

Cenko provides a comprehensive history of the technical development in store separation trajectory analysis [3, 4]. The following summarization of his work highlights the use of CFD methods as they are of interest for this work.

In the 1960s, captive trajectory system (CTS) wind tunnel experimentation was developed to address the inherent safety risks in flight testing procedures at the time [5]. Flight testing utilized a "hit-or-miss" methodology, gradually pushing the limits of the flight envelope during store release until it was considered too dangerous to proceed. Within the safe confines of the wind tunnel, CTS measures the aerodynamic forces on a model-scale store at various positions near the vehicle model to predict its trajectory. This has provided reasonable results compared to flight testing data, but sometimes there are discrepancies, especially considering the similarity laws that must be utilized to accurately scale wind tunnel results to real flight conditions. With hopes to one day reduce the time and expense of physical testing, computational methodologies have been developed using CFD.

CFD techniques to predict store separation trajectories have continued to develop over the last few decades. At their conception, computational methods were difficult to implement due to the computational resources and long run times required for a solution. Since then, the field has matured through various stages of development and become an important tool in modern store separation analysis. By the late 1970s and early 1980s, CFD development had reached the point where it could provide solutions for complex geometries and predict flow conditions around an aircraft wing. Soon after, potential flow models were used in codes, such as PAN AIR [6], to predict the aerodynamic forces on a released weapons store. The forces could then be utilized by a 6-degree-of-freedom (6DoF) algorithm to predict the store's trajectory. By the 1990s, both the Air Force and Navy were pushing programs to further develop the role of CFD in store separation analysis, and the resulting studies [7-9] validated the use of full potential and Euler codes for accurate trajectory predictions.

Until the 2000s, most studies were limited to vehicle designs with externally mounted stores like the wing-pylon-store configuration [10]. It was also standard practice to use inviscid, "quasi-static" computational methods, simplifying the analysis to a series of steady-state solutions through the store's trajectory rather than a time-dependent, dynamic simulation. Because aircraft at the time were generally designed to have non-separating, steady flow around the wing within the flight envelope of store release, the inviscid, quasi-static approach produced sufficiently accurate trajectories. However, the next generation of military aircraft has reconfigured weapons deployment to use internal weapons bays to improve aerodynamic performance and reduce vehicle profile. The internal weapons bay presents additional challenges to store separation prediction. Once the weapons bay cavity opens, a highly resonant, unsteady flow results from flow over the cavity. In these configurations, CFD methods are challenged to account for the unsteady viscous shear layer that the store must pass through, potentially impacting its trajectory.

Addressing the challenges of the time, the Institute for HPC Applications to Air Armament (IHAAA) was created as a joint Army, Navy, and Air Force effort to unify and further develop the computational tools used for store separation analysis. IHAAA programs have resulted in studies [11-14] that further investigate store separation from internal weapons bay cavities. Such studies [15] have found that if the weapons bay is built with a "well-designed" leading edge spoiler, the resonant, unsteady characteristics of the cavity flow may be sufficiently dampened such that quasi-static simulations produce results similar to transient simulations. Other studies [12, 16] have investigated the bifurcation of store trajectories from the unsteady bay cavity in the absence of flow control features like leading edge spoilers. These show that, due to the unsteadiness of the cavity flow, different store release times may result in different trajectories.

This historical overview sets the stage for the store separation analyses presented here. The inviscid, quasi-static techniques developed will prove useful for the trajectory predictions of Chapter 3 where the presence of the cavity itself is not considered. The dynamic cavity simulations of Chapter 4, however, place an emphasis on the viscous, unsteady cavity effects which must be taken into account.

2.2 Cart3D Review for 6DoF Simulations

Because the trajectory predictions in Chapter 3 do not model the weapons bay cavity, an inviscid, quasi-static approach is considered sufficient. Given this approach, NASA's Cart3D suite of CFD tools is well-suited for the trajectory predictions of upward ejected stores. Cart3D not only provides a computational solution to the Euler equations, but it is also distributed with the capability to generate unstructured Cartesian (hanging node) grids, provide adaptive grid refinement using adjoint solutions, and pass aerodynamic force values into a 6-degree-of-freedom (6DoF) algorithm to update a dynamic body's position in time. This functionality meets the needs for store separation trajectory analysis. At each quasi-static time step, the grid is regenerated, a steady solution is computed, forces on the store are calculated, and the store's position is updated. This capability in Cart3D has been demonstrated in several publications [17, 18].

To facilitate trajectory analysis in Cart3D, the Geometry Manipulation Protocol (GMP) was introduced by Murman et al. [18] GMP was developed as a low-level convention for describing geometries and manipulating them as needed in a given application. The protocol unifies the process from mesh generation, through CFD solution, to 6DoF motion. GMP is implemented in an XML format, which, due to its hierarchical structure, is well-suited for describing geometries that lend themselves to parent-child relationships (i.e., rotors as children of a parent rotorcraft, or aerodynamic fin surfaces as children of a weapons store). Murman et al. demonstrated GMP's ability to handle dynamic rigid body simulations. Multi-body dynamics are demonstrated using both prescribed (by analytic functions) motion as well as aero-6DoF motion. In aero-6DoF simulations, bodies are positioned over time according to aerodynamic forces and moments when trajectories are unknown a priori. In Murman's work, this functionality is showcased by simulating the trajectories of the solid rocket boosters' ejection from the Space Shuttle. This capability is well-suited for the store separation problem.

In another publication [17], Murman et al. apply this methodology to store separation, simulating the trajectory of the GBU-31 Joint Direct Attack Munition (JDAM) store from the F/A-18C aircraft. Cart3D is used to compute inviscid CFD solutions coupled with a 6DoF routine to update store position. The trajectory analysis process is shown in Figure 2.1 below, which makes use of Cart3D's robust and efficient Cartesian mesh generator. At each time step, the volume mesh is regenerated, a flow solution is obtained, integrated forces and moments are processed, and the store's position is updated, all within the unified GMP framework. The simulations are carried out using both quasi-static and time-dependent solution methods, and the results are compared. Both solution methods provide results that agree with flight test data. These results help validate the quasi-static approach for similar applications. This approach is used with Cart3D's solver and utilities for the work published in this paper under the assumption that the store's inertia is great enough and near-body effects are small enough for sufficient analysis accuracy.

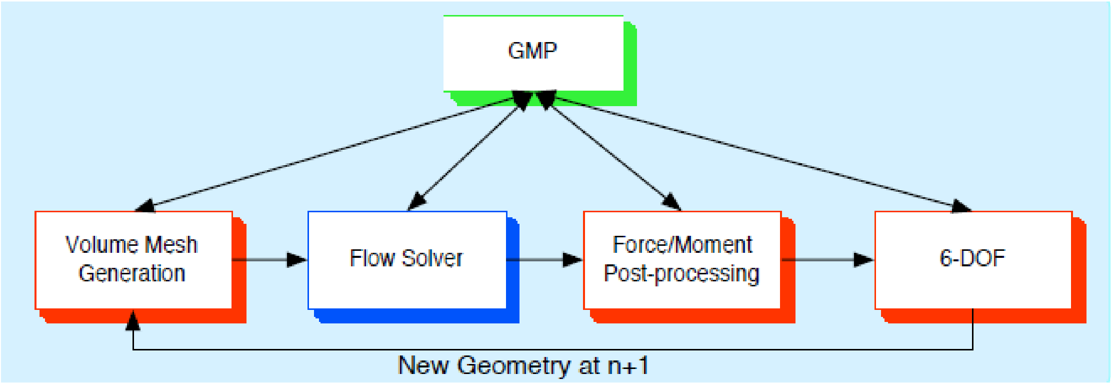


Figure 2.1 6DoF trajectory analysis workflow for each time level, *n* [17]

2.3 The Effects of Doors in Weapons Bay Cavities

Chapter 4 simulates a store in carriage in a weapons bay cavity while the cavity door opens dynamically. Anticipating this analysis, it is important to gain an understanding of the current state of published literature concerning the effects of doors over weapons bay cavities. The available publications in this field are limited, but four studies [19-22] are highlighted here which are considered most relevant. These studies include both experimental [19] and computational [20-22] approaches, as well as quasi-static [19-22] and dynamic [19, 21, 22] methodologies. These studies consider both sliding door [19, 20] and rotating double door [20-22] configurations. It should be noted that the highest freestream Mach number presented in these studies is Mach 1.6 [20], which is considerably lower than the Mach 4 conditions investigated in Chapter 4.

Experimental Case Study

Turpin et al.'s work [19] is reviewed first. It presents a strictly experimental approach to cavity flow investigation, highlighting the effects of a downstream sliding door. Turpin et al. recognize a gap in published cavity flow studies relating to the store separation problem. The bulk of existing studies have only considered the quasi-steady resonant state of an open cavity, allowing as much time as needed to achieve the fully resonant state before analyzing data. A natural question for store separation is whether this quasi-steady state is actually reached by the time the store is ejected. There is a flow development process from the time that the cavity starts to open until final quasi-steady resonant flow that Turpin et al. call "cavity startup". The primary goal of their study is to investigate the nature of cavity startup and identify any implications that it may have for store separation.

This experimental study investigates a simple rectangular cavity with the downstream sliding door shown in Figure 2.2. Simulating flight conditions in an intermittent blowdown wind tunnel, the freestream Mach number is 1.5 and the freestream temperature is 203K. The Reynolds number is 32.6x10^6. Their experimental methods include oil flow visualization for qualitative flow description, discrete pressure transducer measurements and pressure-sensitive paint for quantitative analysis, and door photogrammetry to synchronize the measured data with door position. Pressure data within the cavity is analyzed while the door is held in several quasi-static positions as well as for the dynamic opening case.

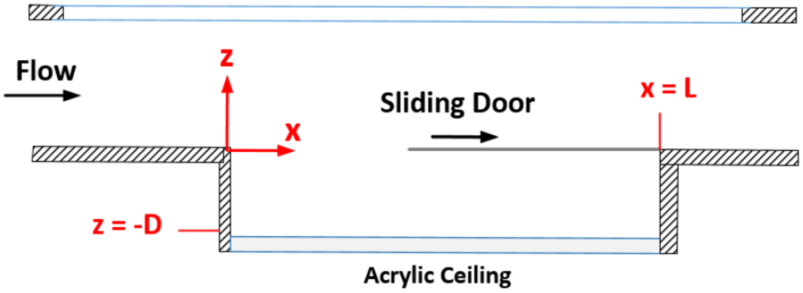


Figure 2.2 Experimental setup of a rectangular cavity with a downstream sliding door [19]

The resonant modes of the pressure data are compared to two analytical models. The first is the semi-empirical Rossiter model [23], modified by Heller et al. [24] for high-speed flow,

$$Str = f_R \frac{L}{U_\infty} = \frac{n - \alpha}{(\kappa^{-1} + M')}$$

$$M' = \frac{M_\infty}{\left[1 + \left(\frac{r}{2}\right)(\gamma - 1)M_\infty^2\right]^{0.5}}$$

Here, the Rossiter resonant modes are predicted based on a mode number, n, freestream Mach number, and two empirically tuned constants, α and κ, which are generally accepted as 0.25 and 0.57, respectively. Heller et al. [24] modify Rossiter's original formulation by introducing the modified Mach number, M', which considers the effects of high-speed flows. γ is the ratio of specific heats, and r is a temperature recovery factor,

$$r = \frac{T_c - T_\infty}{T_0 - T_\infty}$$

where T_c, T_∞, and T_0 are the cavity, freestream static, and freestream stagnation temperatures, respectively. The Rossiter tones are considered the benchmark for resonant open cavity simulations, which Turpin et al. expect to find after cavity startup.

The pressure frequency data is also compared to the classical acoustic resonant modes, described analytically by the equation,

$$f_{ac} = \frac{na_0}{2L}$$

Based on the work Unalmis et al. [25] and Bartel and McAvoy [26], it is expected that these frequencies would be present when the door is only partially open and mostly closed. Accordingly, Turpin et al. expect to see a transition from classical acoustic frequency content to the Rossiter tonal model. Their primary question is how this transition relates to the door opening motion and what that may infer about the nature of cavity startup.

The door opening process is first simulated in the wind tunnel by several static runs with the door in various discrete positions. This quasi-static approach yields the expected results, showing resonant content aligned with classical acoustic modes while the door is mostly closed, and shifting to the lower Rossiter tones by the time the door opens. Also, full resonance amplitudes are reached when the door is about 60% open. These results are best summarized by the data presented in Figure 2.3 for a pressure transducer on the front wall of the cavity. Figure 2.3(a) highlights the shift between the classical acoustic and Rossiter tones, and Figure 2.3(b) shows the full amplitude being reached prior to the cavity being fully opened. This suggests that full cavity startup takes place prior to the door fully opening. However, this quasi-static approach may neglect the unsteady effects of a dynamically opening door.

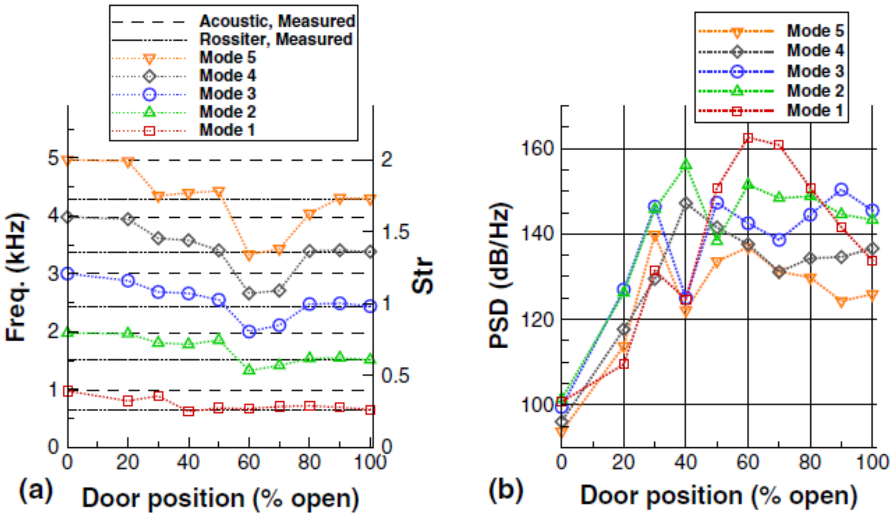


Figure 2.3 Pressure data on the front cavity wall, (a) signal frequencies and (b) strengths [19]

Turpin et al. next investigate the sliding door under a more dynamic approach. The sliding door is opened by a spring mechanism, and cavity pressure data is sampled as the door opens in a single wind tunnel run, repeated for statistical confidence. Results for the dynamic door opening are compared to the quasi-static results in Figure 2.4 for the front wall pressure transducer. It is noted that, except for an experimental discrepancy at the initial time, the mean pressure data for the quasi-static and dynamic cases are in general agreement. However, the pressure amplitudes exhibited around 60% open are much larger for the quasi-static analysis than for the dynamic study. By the time the door fully opens, though, the mean pressure value and amplitude are similar.

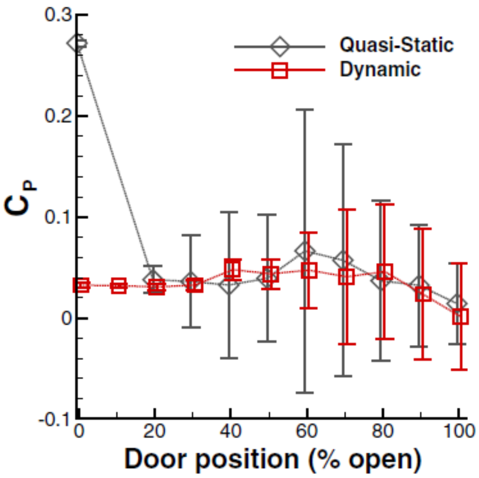


Figure 2.4 Coefficient of pressure (average and range) for quasi-static and dynamic cases [19]

For the dynamic door cases, Turpin et al. define the cavity startup time as the time when a running standard deviation of pressure data crosses the threshold of the pressure standard deviation when the door is left open for a long period of time. On average, cavity startup is reached when the door is 63% open. This suggests one of the primary conclusions of the study, which is of most interest here; the effects of cavity startup delay are limited to the time that the door is opening. For the store separation engineer, one would be hopeful that cavity startup is further delayed, avoiding the complexities of unsteady cavity flow as the store is ejected, but for Turpin et al.'s study, this is not the case.

In Chapter 4, sliding door cavity simulations are conducted and the results of time-accurate dynamic simulations are compared to a quasi-static approach. There are several key differences between what is presented in Turpin et al.'s work and what is presented in Chapter 4, besides the experimental versus computational approach. First, Chapter 4 considers much higher speed flow, Mach 4 rather than Mach 1.5. Also, the presence of a store in carriage is included in Chapter 4, which is not considered in Turpin's study. The effects of the store are unknown, and the impact of the flow on store separation can be more directly evaluated by measuring store forces. Finally, although store trajectory is only impacted by the state of the flow once the door is open, the intermediate flow states may contain critical insights to store and cavity designs which are subjected to the resulting pressure and thermal loads.

Computational Case Study 1

The remainder of this literature review is dedicated to computational studies on the effects of opening doors in weapons bay cavities. The first of these publications is the work of Baugher et al. [20] Their primary focus is a description of the three-dimensional flow effects which are present in a cavity bay and modified by the presence of doors. A rectangular cavity in Mach 1.6 flow is simulated using both Reynolds-Averaged Navier Stokes (RANS) and Delayed Detached Eddy Simulation (DDES) computational methods. The cavity model is 225.5mm long with a length to depth ratio, $\frac{L}{D}$ = 4.5. The freestream stagnation temperature, T_0, is 292K, and Reynolds number, Re_x, is 30x10^6/m. Double doors that rotate open ("wedge" doors) as well as a downstream sliding door are analyzed. These geometric configurations are depicted in Figure 2.5. Note that only the starboard wedge door is shown, and the port door is designed to open symmetrically.

While the open cavity flow is simulated without doors, the more computationally expensive DDES solution is used as a baseline to validate the RANS solution. The RANS solution is determined to capture the key characteristics of the flow sufficiently that it may be used to analyze configurations with cavity doors. It should be noted that these simulations are conducted with the doors in static positions. The wedge doors are simulated open, and the sliding door is simulated 50% open. Dynamic, moving door simulations are not conducted here.

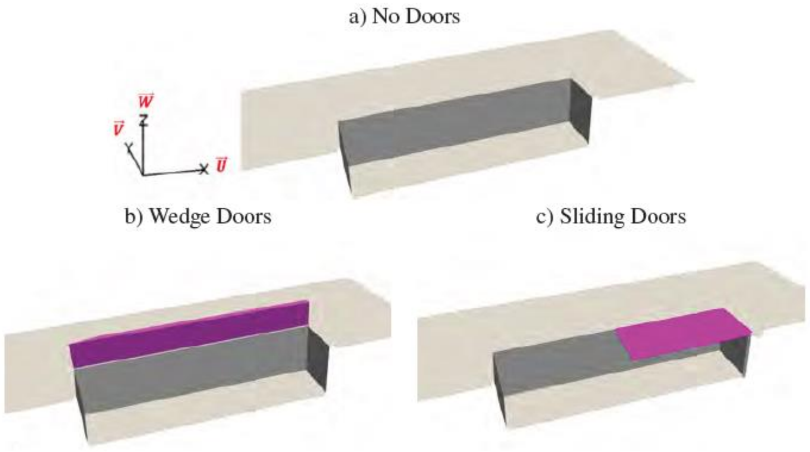
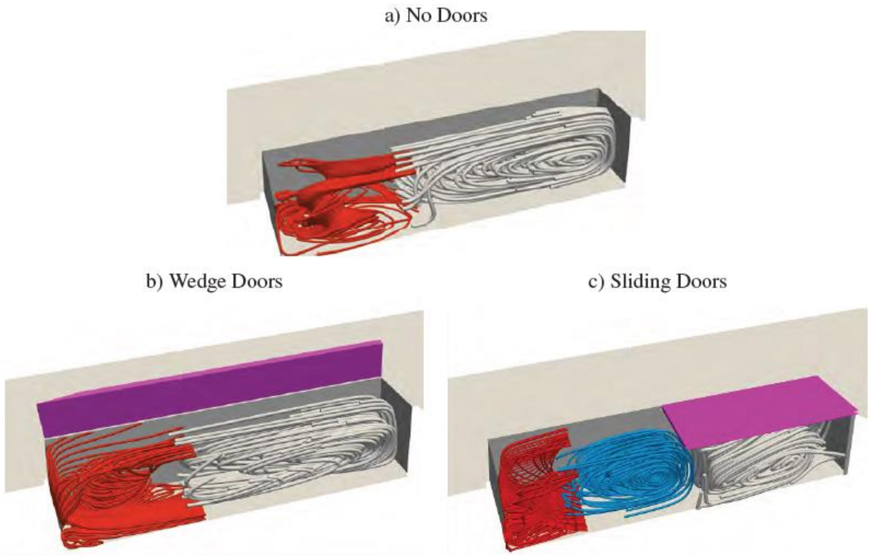


Figure 2.5 Rectangular cavity and door configurations for wedge and sliding doors [20]

Baugher et al.'s study describes the effects on large-scale flow structures when wedge or sliding doors are introduced. Three-dimensional streamlines are shown in Figure 2.6 for each of the three configurations – no doors, wedge doors, and a sliding door. The no doors solution features a large downstream recirculation region and an upstream tornado vortex. As the wedge doors are introduced, the tornado vortex moves slightly upstream and is centered on the cavity floor. The simulation with the sliding door shows a recirculation region beneath the door and a secondary vortex region just upstream of the door's leading edge. In addition, the tornado vortex is split into two twin counter-rotating vortices for the sliding door case. To help illustrate, streamlines and contours of the streamwise velocity component are plotted for a mid-span section (Figure 2.7) and for a section at the half-depth of the cavity (Figure 2.8).

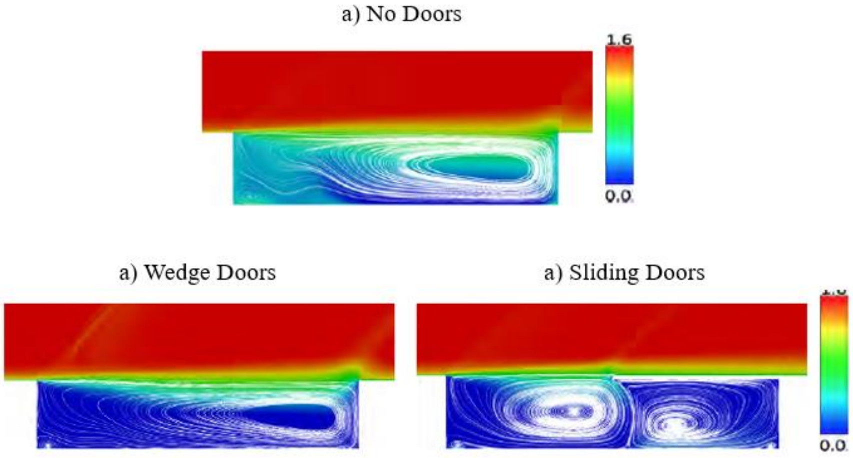


a) No Doors

b) Wedge Doors

c) Sliding Doors

Figure 2.6 Streamlines for each cavity/door configuration [20]



a) No Doors

a) Wedge Doors

a) Sliding Doors

Figure 2.7 Streamwise velocity contour plots with streamlines at the mid-span cross-section [20]

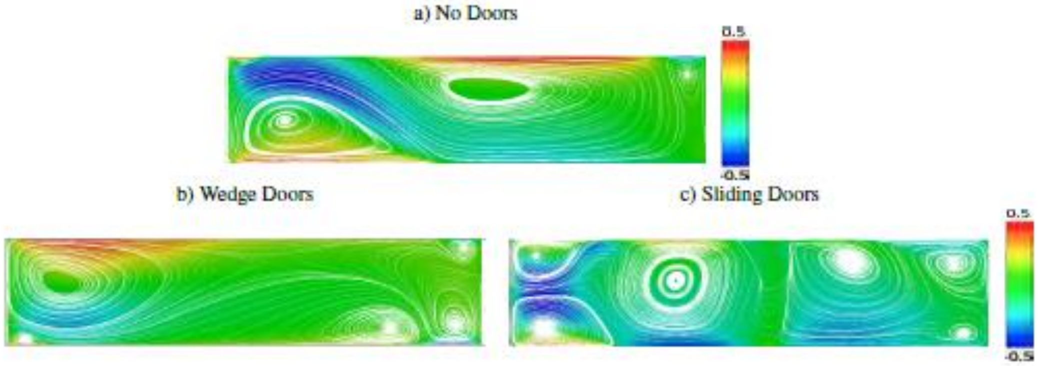


Figure 2.8 Streamwise velocity contour plots with streamlines at the half-depth cross-section [20]

This publication provides an excellent description of the inherently three-dimensional nature of complex cavity flows and the impact of the presence of doors. There are some key differences, though, between Baugher's work and the study presented in Chapter 4. These results are in much slower freestream conditions, Mach 1.6 rather than Mach 4. Also, the door simulations are static in contrast to the dynamic door-opening solutions presented later. Nonetheless, the work provides a descriptive baseline for understanding the three-dimensional nature of cavity flow.

Computational Case Study 2

Another study investigating the effects of doors in weapons cavity bays is presented by Sheta et al. [21]. This work takes both an experimental and computational approach. However, for the interests of this review, an emphasis is placed on the CFD study. The geometry of the doors is shown in Figure 2.9 below, featuring two doors which rotate open. For the computational portion of the study, freestream conditions are simulated where Mach number is 1.44, dynamic pressure is 74.4 kPa, and stagnation pressure is 172 kPa. The effects of static doors on cavity flow are investigated as well as a dynamic door-opening study.

Sheta et al. utilize a hybrid RANS/LES solution method, modeling turbulent effects closest to the wall and resolving them further away. They also highlight the use of a low dissipation scheme, specifically a Kinetic Energy Consistent (KEC) model as presented by Subbareddy and Candler [27]. It is pointed out that the high acoustic frequencies associated with cavity flow are likely to be lost in numerical dissipation using typical flux schemes. Using this low dissipation scheme and a small time step for solution advancement, Δt = 5x10^{-6} seconds, Sheta et al. seek to provide a high-accuracy computational solution for the tonal content of the flow solution.

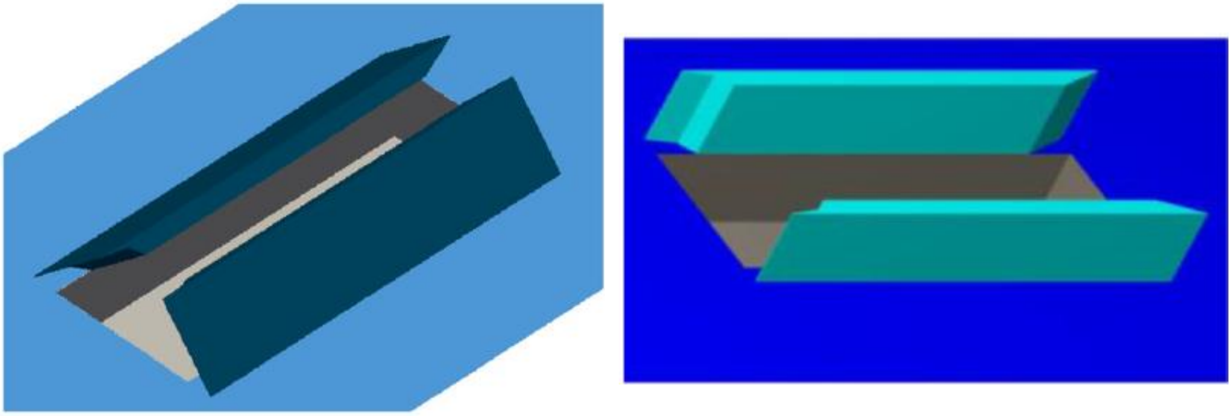


Figure 2.9 Geometry of dynamic opening doors [21]

From the static computations, one of the primary results of this study was the increase in mean pressure and pressure fluctuations within the cavity in the presence of cavity doors. These results are shown in the RMS pressure and mean pressure plots in Figure 2.10. Especially considering mean values, the inclusion of the doors dramatically increases the pressure at a given cavity depth. The mechanism for this pressure rise is explained by the shocks from the leading edges of the doors which cause a jump in pressure. This results in the shear layer going deeper and effectively pushing more flow into the cavity. This effect is depicted well by the streamwise velocity contours shown for the spanwise-symmetric cut plane in Figure 2.11 and by the cavity floor surface pressures shown in Figure 2.12.

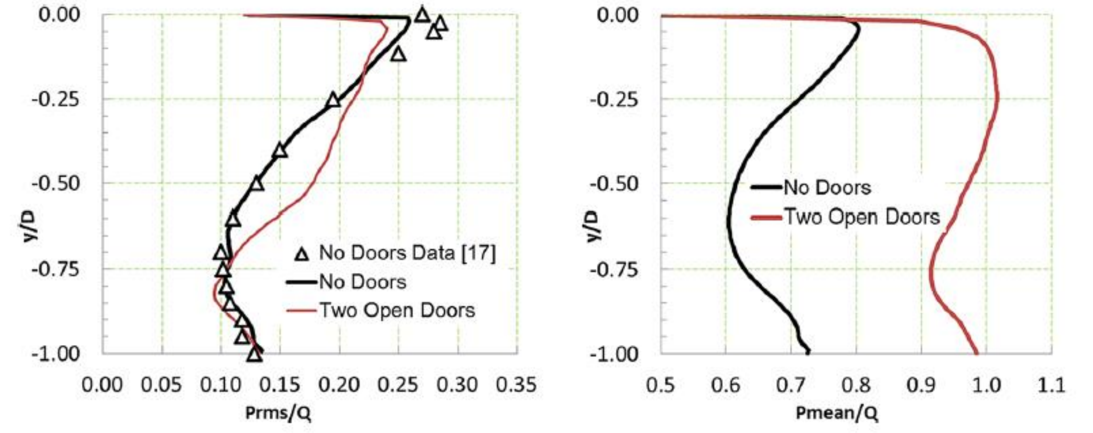


Figure 2.10 RMS and mean pressure over the depth of a rectangular cavity [21]

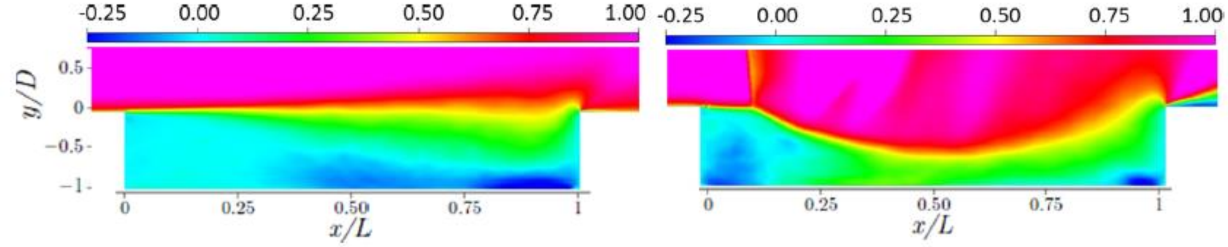


Figure 2.11 Streamwise velocity contour plots with (right) and without (left) cavity doors [21]

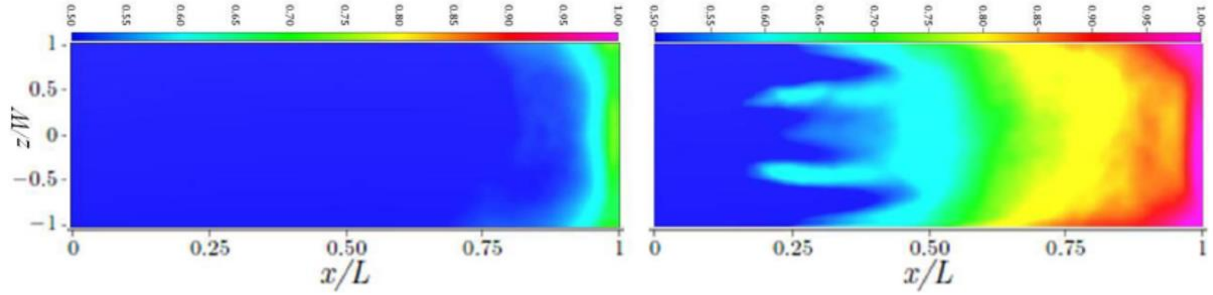


Figure 2.12 Mean pressure on the cavity floor with (right) and without (left) cavity doors [21]

Sheta et al. also provide results for the dynamic simulation with doors rotating open. Their computation starts with the doors in the 5° open position. A steady solution is obtained and used as the initial condition for the dynamic simulation which continues until the doors are 35° open. The initial condition with the doors partially open is significant for this review. One of the goals of Chapter 4 is to accurately capture the quiescent cavity condition at solution startup with a closed and sealed cavity. In Sheta et al.'s study, the solution starts with the doors partially open, and the initial conditions are not quiescent.

The most significant finding in this publication for the dynamic door simulation is that the RMS pressure values on the cavity floor and doors peak when the doors are 25°-30° open, rather than constantly increasing. These results are shown for the cavity floor in Figure 2.13. This is a fundamentally important result as it illustrates key loading effects inside the cavity that are only captured in transient analysis. Static cavity simulations would be insufficient to capture such intermediate results. The study presented in Chapter 4 works toward a similar simulation under different conditions (Mach 4 freestream with a sliding door), with a curiosity if similar main effects will be found by the nature of a dynamic, transient solution.

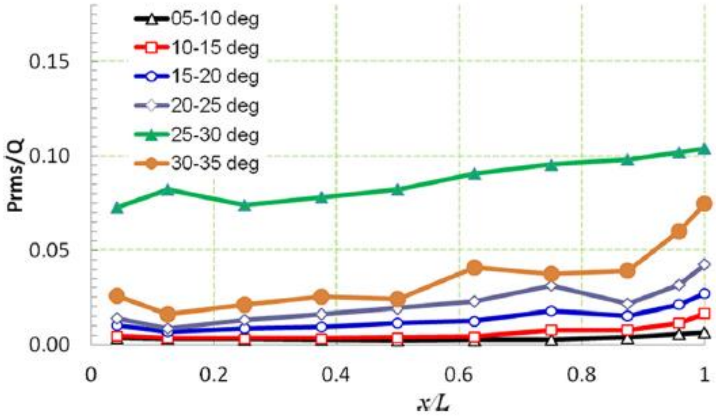


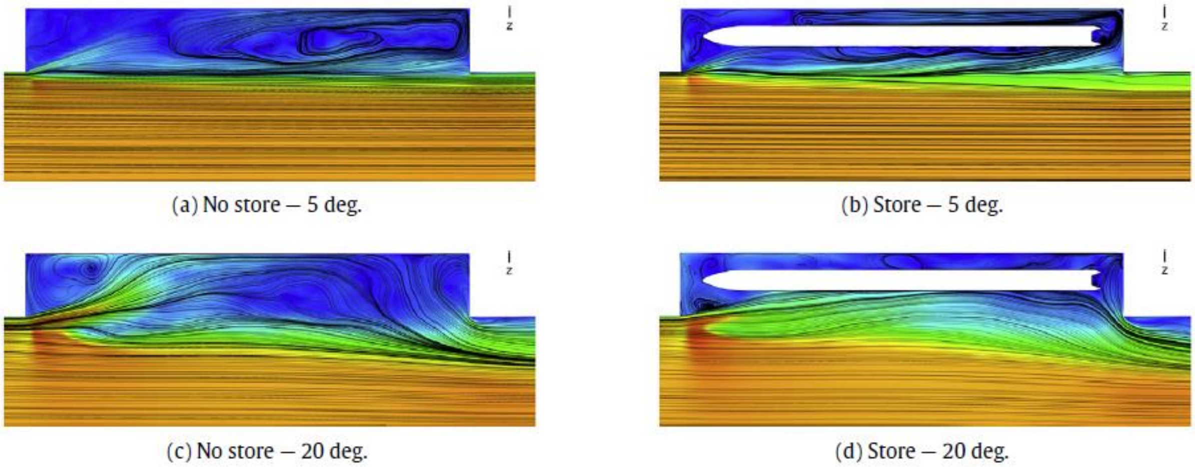
Figure 2.13 RMS surface pressure on the cavity floor as the doors open in a dynamic simulation [21]

Computational Case Study 3

The final publication reviewed here is the work of Loupy et al. [22]. This study uses CFD methods to simulate a rectangular weapons bay with a store in carriage. It also includes the effects of cavity doors which rotate open, similar to the work of Baugher et al. [20] However, here, the doors are simulated dynamically, capturing time-accurate flow data as the doors open. With the doors opening and with a store in carriage, perhaps this work is most similar to what is presented in Chapter 4, as far as the mechanics of the simulation itself are concerned. However, there are significant differences in Loupy et al.'s work, primarily concerning the door configuration and the simulated flow conditions.

This study simulates subsonic flight conditions at 3,000 feet altitude. Mach number is 0.85, freestream temperature is 281.5K, and Reynolds number is $6.5\text{x}10^6$ based on cavity length. The cavity is 3.59m long. Acknowledging the computational expense of DES methods commonly used for cavity flow simulations, Loupy et al. instead use a Scale-Adaptive Simulation (SAS) method as presented by Menter and Egorov [28] and shown suitable for cavity flow by Babu [29]. The SAS method is used, expecting accurate results at a reduced computational cost. This study also models the aeroelastic behavior of the store, interpolating the structural modes from the computational structural dynamics (CSD) model onto the CFD grid.

The aeroelastic behavior of the store from carriage through ejection and separation trajectory is of primary importance for Loupy et al. However, in this review, emphasis is placed on the impact that a dynamic door opening simulation has on a store in carriage. In the dynamic door simulation, this study points out that a jet forms at the leading edge of the cavity as the doors start to open. Without the store, the effect of this jet reaches the ceiling of the cavity. However, with the store in carriage, the cavity walls are shielded from the effects of the jet, and the loads on the store are slightly impacted by the jet impingement. This is illustrated in the Mach number contour plots in Figure 2.14.



(a) No store – 5 deg.

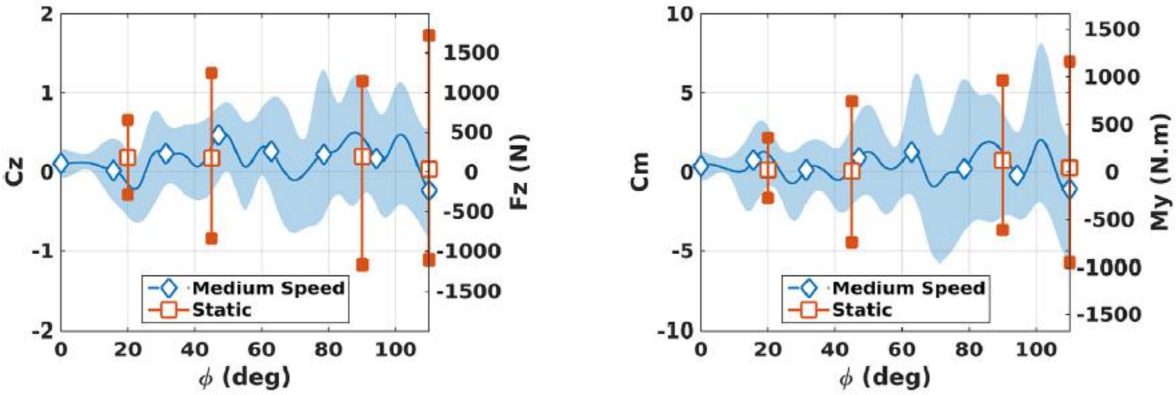
(b) Store – 5 deg.

(c) No store – 20 deg.

(d) Store – 20 deg.

Figure 2.14 Mach number contours with and without a store, with the doors 5° and 20° open [22]

Figure 2.15 shows the vertical force and pitching moment results on the store for both the dynamic door simulation as well as with the door in several static partially open positions. Plotted values for the dynamic case represent a running time average with the associated range of values shaded in blue. The static simulation results are averaged over the entire computation time, and Figure 2.15 plots their mean values and ranges in red. There are some slight differences in the store load results between the dynamic and static cases with the doors about 20 degrees open when the jet is impinging on the store nose. Ultimately, though, there is little difference in the mean store loads between the static and dynamic cases. One can see, though, that there are some differences in the load ranges throughout the door opening process. Especially around 40 degrees open, one can see that the static simulation overpredicts the load fluctuations relative to the dynamic solution. This is reminiscent of the results that Turpin found experimentally for a sliding door [19] where quasi-static door opening overpredicted pressure fluctuations as the door opened.



(a) C_z.

(b) C_m.

Figure 2.15 Vertical force (left) and pitching moment (right) as the door opens [22]

This review is interested in how the doors are represented in the composite grid scheme. Because of the low-speed nature of this case (Mach 0.85), Loupy et al. are afforded the opportunity to use undersized doors that are not flush with the cavity walls in the closed position. Each of the two doors are only 46% of the cavity width and 98% of the cavity length, leaving a small gap around the doors in the closed position. For the studies presented in Chapter 4, this strategy was not considered as it would contaminate results in high-speed flow. Including such gaps around the doors in high-speed flow would induce shocks around the door edges, causing excessive heat and pressure loads. Also, this would cause the initial state of the cavity to not be quiescent, which is desired so that initial conditions are most realistic for the dynamic simulation. As shown in Chapter 4, a setup which properly seals the cavity and provides a quiescent initial condition is a new challenge.

Of all the cavity studies reviewed, Turpin et al. [19] provide the most detail for flow over a rectangular cavity with a downstream sliding door, the door configuration used in the present work. Baugher et al. [20] describe the three-dimensional nature of cavity flows in the presence of doors while Sheta et al. [21] provide initial results for dynamic door simulations. Finally, Loupy et al. [22] show how the dynamic simulation of an opening cavity impacts store loads. These studies highlight the current state of the literature as it pertains to the dynamic weapons bay cavity simulation presented here. Unique to the current study, though, is its consideration of higher-speed flow than what is studied in these publications, with the added complexity of a cavity that is sealed from the freestream environment before door motion begins.

CHAPTER 3

TRAJECTORY PREDICTIONS OF UPWARD-EJECTED STORES

An exploratory investigation is presented which analyzes store separation trajectories from a high-speed vehicle using computational fluid dynamics (CFD). These analyses support wind tunnel experiments which will be performed at the University of Tennessee Space Institute (UTSI) and Purdue University. UTSI is to perform tests at Mach 4 freestream conditions at the HORIZON TALon facility [1]. Experimental studies at Purdue will be at Mach 6 conditions in the BAM6QT tunnel [2]. The CFD simulations presented here emulate these conditions accordingly. The novelty of this work is in the nature of the release of the store. Rather than dropping the store downward or ejecting the store rearward, the construction of the vehicle requires that the store be ejected upward. The geometry used to simulate upward store ejection is shown in Figure 3.1.

Upward store ejection presents obvious challenges that this study seeks to address. One of the primary goals of store separation analysis is to gain confidence in safe separation from the vehicle, ensuring the store does not collide with the carrier vehicle within its flight envelope. Upward store ejection begs the question, “Will the store simply fall back onto the carrier vehicle after ejection?” Validation of such store ejection designs requires the certainty that there is enough drag force on the store to drive it downstream of the vehicle by the time it falls. Initial simulations shown here reveal that without careful store design, there may not be enough drag to ensure safe separation. However, by enhancing the store geometry with drag features, there is potential for safe store designs.

Each trajectory analysis starts with the store just above the vehicle as if it has already been ejected from a cavity. The cavity itself is not modeled. For each computation, the store is prescribed an initial upward velocity. It is assumed that at the start of the trajectory simulation, the store is sufficiently separated from the vehicle such that viscous effects are negligible. The accuracy of this assumption is the subject of future study, but it is expected that it is sufficient for the preliminary exploratory investigation presented here. As such, an inviscid quasi-static approach is taken. The Euler equations are solved using NASA's Cart3D solver, presented in Section 2.2. The trajectory is predicted by a series of inviscid, steady-state solutions, updating the store's position and orientation each time step according to calculated forces and moments.

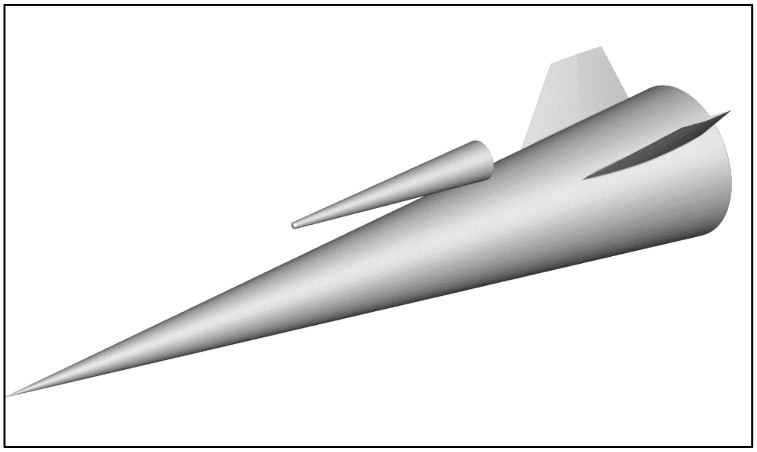


Figure 3.1   Geometry of high-speed carrier vehicle and store

3.1 Problem Setup

Test Case Descriptions

This investigation analyzes four store geometries over a single carrier vehicle design. Like canonical hypersonic test cases, these designs are simulated with simplified representative cone-like geometries. The geometry of the vehicle is shown in Figure 3.1, a 5° half-angle cone, 4.92m long. The four store geometries are shown in Figure 3.2. The first is a simple 4° half-angle cone with a 9-inch (0.229m) diameter base. The nose tip radius is 0.850 inches (21.59mm). For the second store design, the cone is simply rotated

180° such that its base faces directly upstream. Although impractical for actual use, it provides initial insight into store trajectory behavior with enhanced drag. The third store design adds a 0.336m diameter disk on the base of the store to increase drag during the store separation event. Its design intends for the disk to be jettisoned from the store after safe separation. The fourth store introduces 0.5-inch (12.7mm) slots in the drag disk around the base, generating the “ring disk” design that is shown. This design is explored in hopes of alleviating some of the oscillatory flight behavior of the store as shown later. Future work will include deployable drag fin designs that have the spatial advantage of a minimized profile in carriage.

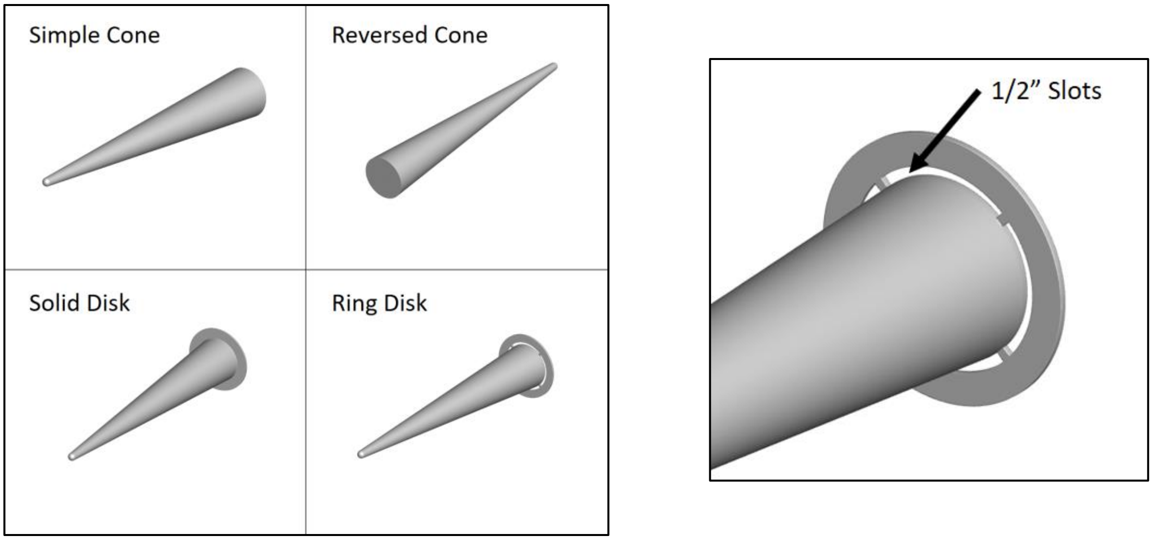


Figure 3.2 Geometry of four store designs, with close-up of “Ring Disk” (right)

The simulated test cases are summarized in Table 3.1 below. Cases A1 through A8 explore ejection velocities (initial vertical velocity of the store) from 3 m/s to 10 m/s. This is done to show how quickly the store must be released to safely clear the vehicle without any additional drag features. Realistically, such large ejection velocities are not feasible due to the magnitude of the reaction forces imposed on the carrier vehicle which are unsuitable for flight operation. The design target for store ejection velocity is 3 m/s.

Cases B1 through B6 simulate the reversed cone geometry to provide initial insights into store separation with increased store drag. Finally, the C and D cases investigate store separation using the solid disk and ring disk designs in Mach 4 and Mach 6 flight conditions, limited to a 3 m/s ejection velocity. Table 3.1 also records the initial position of the store for each case, relative to the nose of the vehicle. The solid and ring disk configurations require a slightly higher starting position to make room for the additional disk geometry.

Table 3.1 Test Cases

Case	**Store Geometry**	**M_{∞}**	**Ejection Velocity (m/s)**	**Initial Store X-Position* (m)**	**Initial Store Y-Position* (m)**
A1	Simple Cone	4	3	2.090	0.430
A2			5		
A3			8		
A4			10		
A5		6	3		
A6			5		
A7			8		
A8			10		
B1	Reversed Cone	4	3	2.090	0.430
B2			5		
B3			8		
B4		6	3		
B5			5		
B6			8		
C1	Solid Disk	4	3	2.090	0.480
C2		6			
D1	Ring Disk	4	3	2.090	0.480
D2		6			

*X-Position is in the streamwise direction. Y-Position is vertical. This is the position of the most forward point on the centerline of the store (nose tip for forward-facing stores, base center for the reversed cone) relative to the nose of the vehicle. Spanwise, the store and vehicle centerlines are aligned.

To simulate store separation, this investigation utilizes NASA's Cart3D inviscid flow solver and 6DoF (six degrees of freedom) motion capabilities. This is accomplished using the quasi-static approach, meaning a series of steady-state solutions are generated, and the resulting aerodynamic forces and moments are used to update the position and orientation of the store at each time step. This process is repeated until the necessary number of time steps are completed. The use of Cart3D with 6DoF motion is demonstrated in Reference [17] using both quasi-static and true unsteady methods. However, the unsteady implementation is not available for public release or use in this study. Imparting an initial velocity on a 6DoF motion object (in our case, the store) is demonstrated in Reference [18]. An obvious omission from the investigation, by nature of an inviscid solution, is the effect of skin friction on the forces and moments. However, it is assumed, for an exploratory investigation, that pressure drag effects are sufficient to learn more about key design considerations. (One may note that, through Chapter 3, "time step" generally refers to the actual time step from one steady-state solution to another, not the pseudo-time step used to generate each static solution.)

Freestream Conditions, Mass Properties, and Non-Dimensionalization

Freestream conditions are considered for flight at an altitude of 80,000 feet (24,384 meters). To simulate these conditions, a freestream temperature, T_∞, of 221K, and a freestream density, ρ_∞, of 0.0454kg/m^3 are used. This implies a freestream speed of sound, a_∞, of 298 m/s. Mass properties for the four store designs are summarized in Table 3.2 below. Dimensionless variables utilize the characteristic length, $L = 1\,m$. Mass properties are non-dimensionalized by freestream density and characteristic length. For mass,

$$m^* = \frac{m}{\rho_\infty L^3}$$

For principal moments of inertia about the store's center of mass,

$$I_x^* = \frac{I_x}{\rho_\infty L^5} \ , \qquad I_y^* = \frac{I_y}{\rho_\infty L^5} \ , \qquad I_z^* = \frac{I_z}{\rho_\infty L^5}$$

Time (applicable to time step inputs), velocity (applicable to ejection velocity inputs), and gravitational acceleration are non-dimensionalized by the freestream speed of sound and characteristic length,

$$t^* = t\frac{a_\infty}{L}$$

$$v^* = \frac{v}{a_\infty}$$

$$g^* = g\frac{L}{a_\infty^2} = -0.00011048$$

Table 3.2 Store Mass Properties

Store Design	***m*** **(kg)**	***I$_x$*** **(kg·m^2)**	***I$_y$*** **(kg·m^2)**	***I$_z$*** **(kg·m^2)**	***m****	***I$_x$****	***I$_y$****	***I$_z$****	**Center of Mass** (m)**
Simple Cone	92.229	0.455	10.808	10.808	2031.5	10.022	238.06	238.06	0.9061
Reversed Cone	92.229	0.455	10.808	10.808	2031.5	10.022	238.06	238.06	0.9061
Solid Disk	97.564	0.530	11.927	11.927	2149.0	11.674	262.71	262.71	0.9314
Ring Disk	97.021	0.522	11.819	11.819	2137.0	11.500	260.32	260.32	0.9290

**The center of mass is measured from the nose of the cone along its centerline.

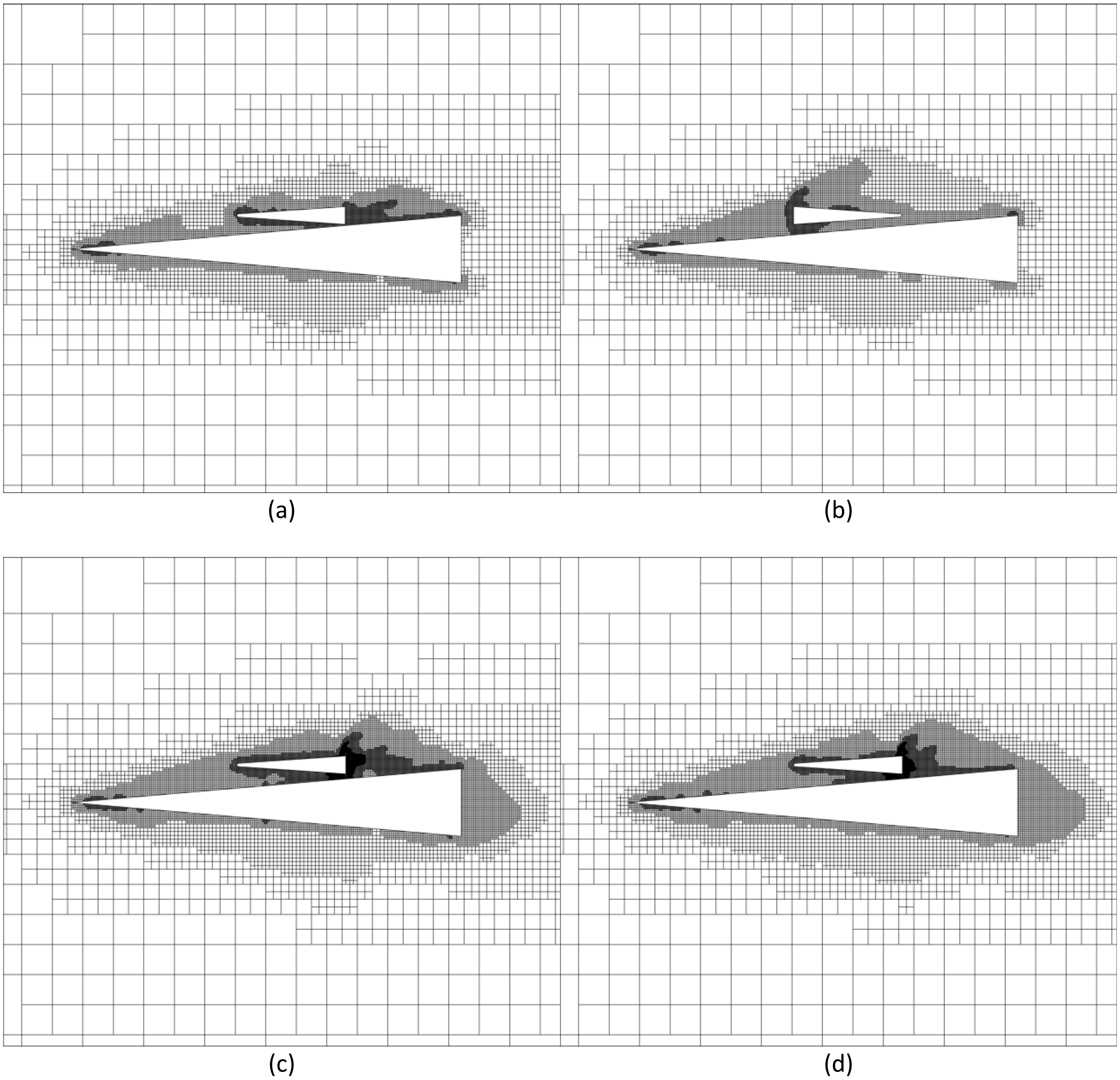
3.2 Grid Generation

One complexity in the CFD analysis of store separation is grid generation – specifically, updating the grid from one 6DoF time step to the next. Although not utilized here, one of the most common ways of doing this is overset grid assembly – superimposing a moving grid that surrounds the store onto a fixed background grid which includes the carrier vehicle. However, Cart3D utilizes simple octree, Cartesian, hexahedral grids, for which overset methods are not supported. So, for this study, the computational grid is regenerated for each 6DoF time step. Fortunately, this geometry is simple enough that this does not come at too great of a computational expense.

Because the grid is rebuilt at each time step, there is a need to automatically generate these grids with appropriate local refinement as the store moves. To this end, an adaptive meshing scheme is used. For each quasi-static solution (at each time step), a total of four solutions are generated, each computing the adjoint solution to the previous run to adaptively refine the grid in local regions that have the highest impact on store and vehicle aerodynamic forces. The fourth solution is computed on the most refined grid, and the resulting aerodynamic forces are used for 6DoF calculations. Refined grids with the store in its initial position are shown in Figure 3.3. Grid statistics are summarized in Table 3.3. The extents of each domain are a cube with a side length of about 150 meters. One may note that the smallest cells of the solid and ring disk configurations are much smaller than the simple and reversed cones. This is due to the additional refinement that is required in the neighborhood of the disk. Refinement levels around the body of the store and the carrier vehicle are comparable for all cases.

Table 3.3 Grid Statistics

Store Geometry	**Smallest Element Size (mm)**	**Total Number of Cells**
Simple Cone	12.0	409,220
Reversed Cone	12.0	413,076
Solid Disk	0.75	1,090,643
Ring Disk	0.75	1,098,749



(a) (b)

(c) (d)

Figure 3.3 Refined grids for the (a) simple cone, (b) reversed cone, (c) solid disk, and (d) ring disk

3.3 Results and Discussion

Simple Cone

The first store geometry analyzed is the simple cone, used in cases A1 through A8. The analysis starts with the target initial ejection velocity of 3 m/s. The ejection velocity was then increased as necessary to 5, 8, and 10 m/s for the store to safely clear the carrier vehicle. These simulations are first conducted with Mach 4 freestream conditions, and subsequently, Mach 6 conditions.

Mach 4 Conditions – Simple Cone

A steady-state solution is shown in Figure 3.4 below at the initial time in Mach 4 freestream conditions corresponding to the conditions in the UTSI TALon facility. It is characterized, as expected, by a conical oblique shock around the carrier vehicle, a bow shock at the nose of the store, and expansion behind each body. From this position, the store is ejected upward at the given ejection velocity.

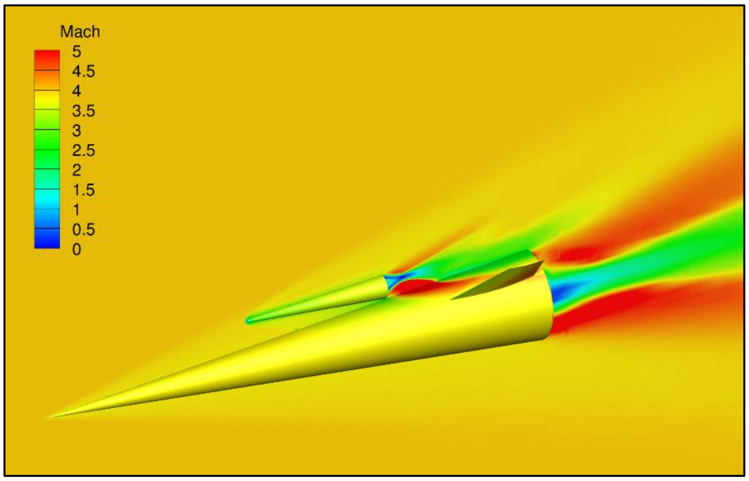


Figure 3.4 Mach number contours at time, t = 0, in Mach 4 freestream conditions

The trajectory of the simple cone at each ejection velocity is shown in Figure 3.5 through Figure 3.7. Clearly, the store does not clear the vehicle safely at 3 m/s or 5 m/s as the store simply falls back onto the carrier vehicle. The mass of the store is large enough that there is an insufficient drag to slow the store relative to the vehicle before it collides with the carrier vehicle. At 8 m/s, the store clears the vehicle, but only by a small margin, likely within the bounds of uncertainty at this point in the investigation. Finally, with an ejection velocity of 10 m/s, the store clears the vehicle with significant margin. However, the ejection force required to achieve a 10 m/s store velocity would result in upsetting the carrier vehicle.

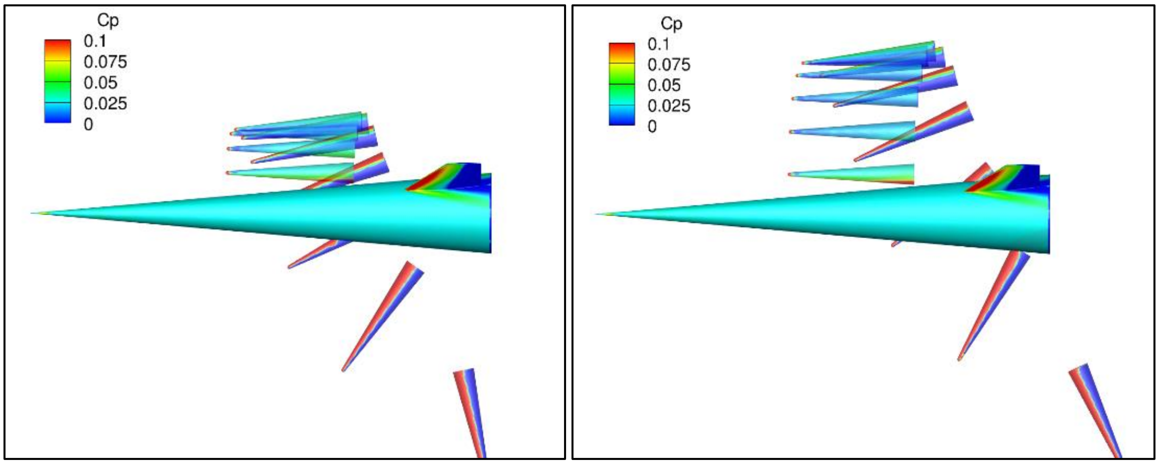


Figure 3.5 Store trajectories – Case A1: Mach 4, 3m/s (left) and Case A2: Mach 4, 5m/s (right)

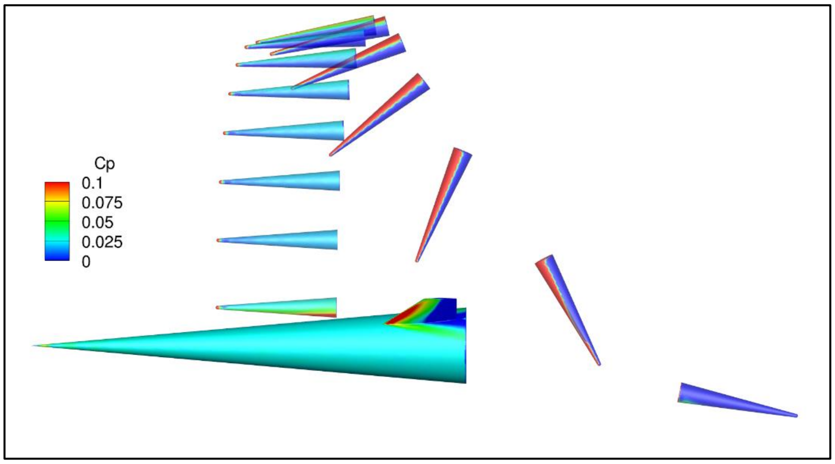


Figure 3.6 Store trajectory – Case A3: Mach 4, 8m/s

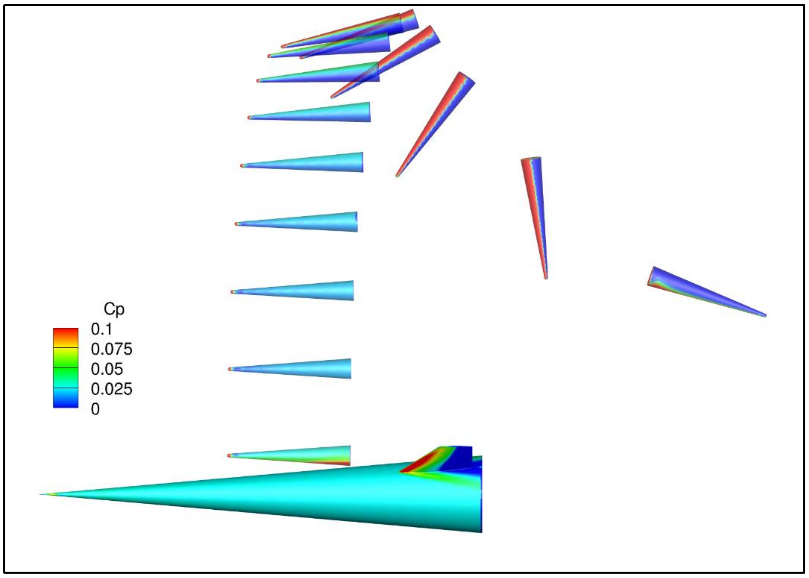


Figure 3.7 Store trajectory – Case A4: Mach 4, 10m/s

Figure 3.8 presents a trajectory comparison for the simple cone store at each ejection velocity. Specifically, these are the trajectories of the centers of mass for each case. It should be noted that the 8 m/s trajectory appears to clear the vehicle by a reasonable margin, but the extension of the store geometry from the center of mass to the nose eliminates most of the clearance. It is evident that in Mach 4 freestream conditions, the simple cone geometry does not provide a drag force sufficient for safe separation when applying reasonable ejection velocities. Before moving to other store geometries, it is worth investigating the effects of higher Mach number freestream conditions.

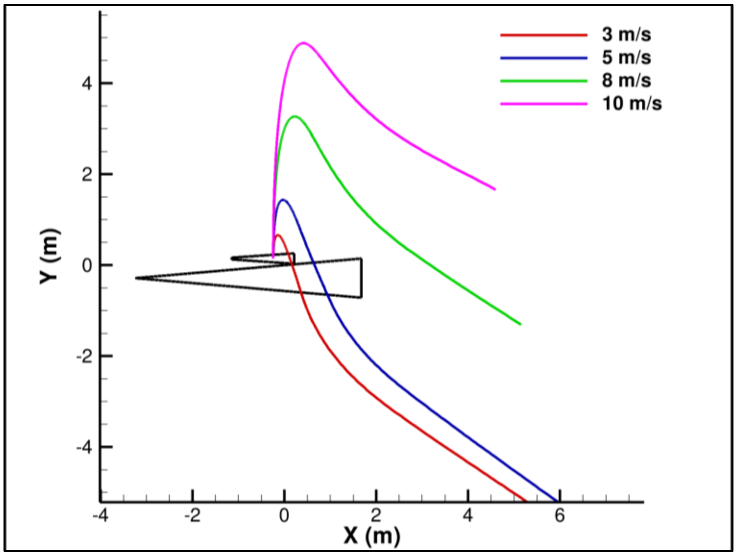


Figure 3.8 Center of mass trajectories for Cases A1 through A4

Mach 6 Conditions – Simple Cone

Store separation trajectories are also calculated using Mach 6 freestream conditions, simulating conditions in Purdue University's BAM6QT tunnel. These results provide insight into the effects of a higher-speed freestream condition for upward-ejected stores. One should note that freestream temperature is maintained at 221K, consistent with conditions at an altitude of 80,000 feet (24,384 meters). The initial, *t = 0*, solution is shown in Figure 3.9 below. The Mach 6 solution appears like the Mach 4 solution, featuring shallower oblique shock angles, consistent with compressible flow theory.

Store trajectories are shown in Figure 3.10 through Figure 3.12 for ejection velocities of 3, 5, 8, and 10 m/s at Mach 6 freestream conditions. As was the case in Mach 4 conditions, ejection velocities of 3 and 5 m/s are insufficient for the store to clear the vehicle. An ejection velocity of 8 m/s, again, clears the vehicle marginally, and 10 m/s clears sufficiently. Such close similarity between Mach 4 and Mach 6 calculations may be surprising as one would expect higher freestream velocity to result in much higher drag forces, leading to higher margins in separation clearance.

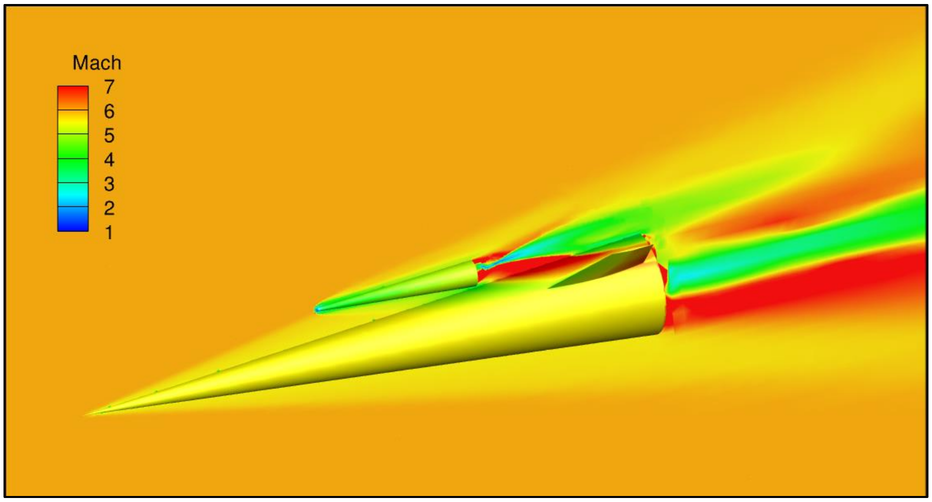


Figure 3.9 Mach number contours at time, t = 0, in Mach 6 freestream conditions

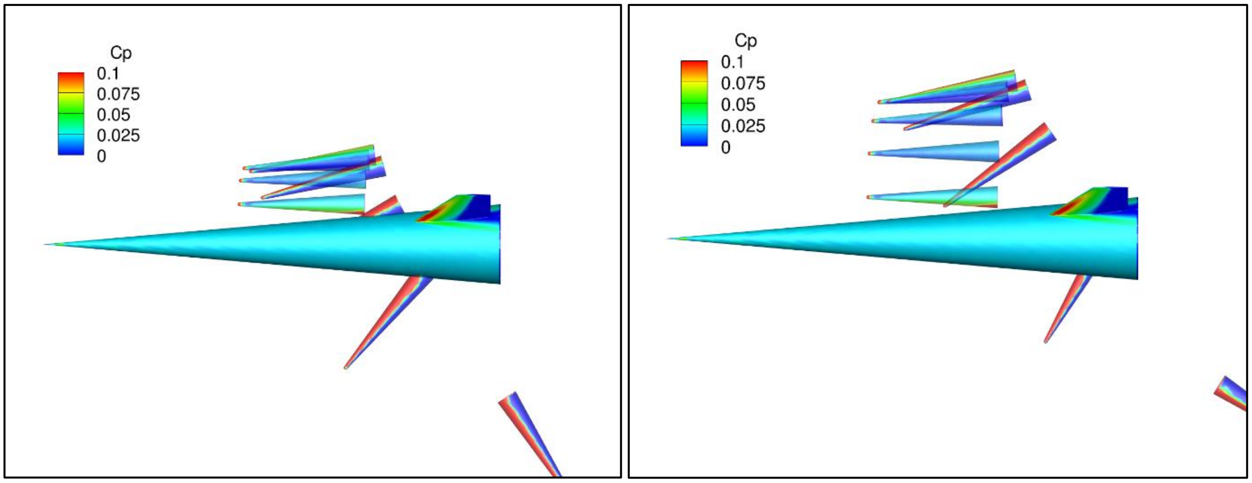


Figure 3.10 Store trajectories – Case A5: Mach 6, 3m/s (left) and Case A6: Mach 6, 5m/s (right)

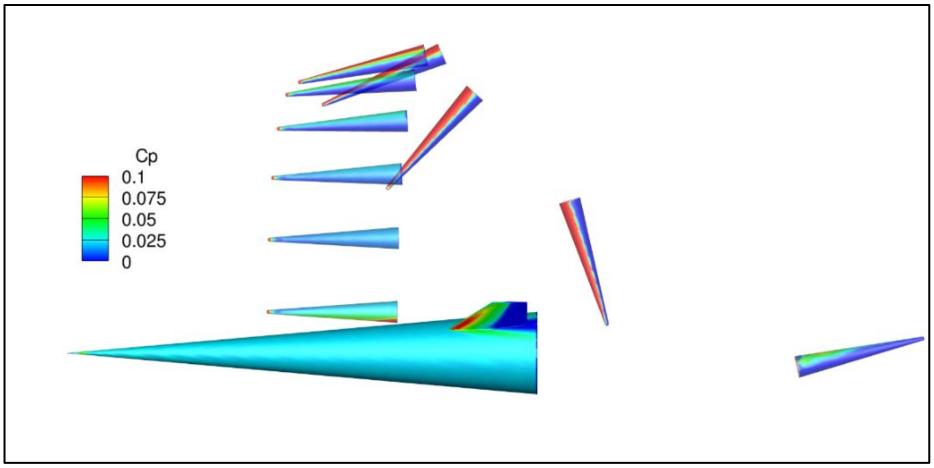


Figure 3.11 Store trajectory – Case A7: Mach 6, 8m/s

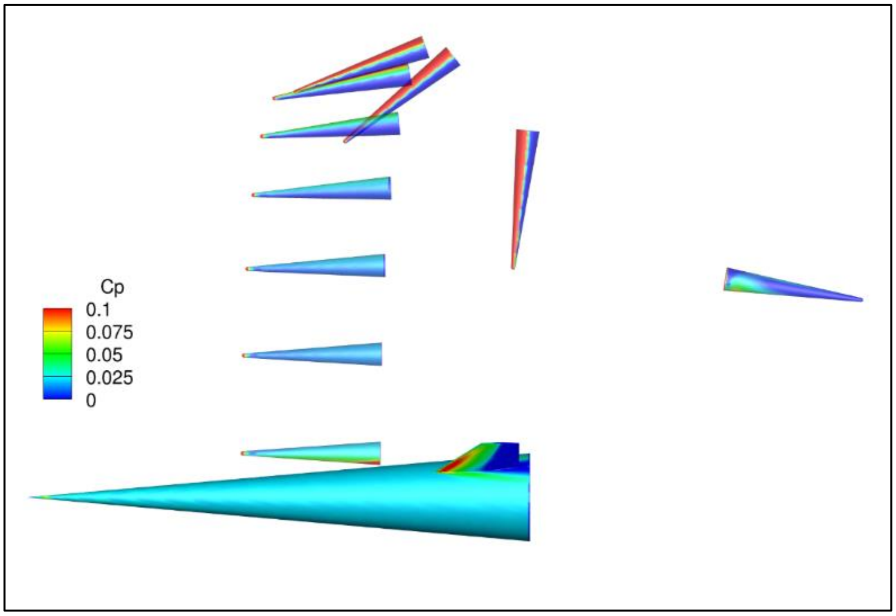


Figure 3.12 Store trajectory – Case A8: Mach 6, 10m/s

The separation trajectories show similar behavior between the Mach 4 and Mach 6 simulations, but closer comparison between the results reveals some important consequences of higher speed flow. Figure 3.13 compares the trajectories of the store in both Mach 4 and Mach 6 freestream conditions for all ejection velocities. One can observe that, especially at higher ejection velocities, the store does not reach as high of a trajectory peak in Mach 6 conditions as it does in Mach 4 flow. For 3m/s and 5m/s ejection velocities, this does not change the result that the store collides with the carrier vehicle. With an 8 m/s ejection velocity, the trajectory paths (for Mach 4 and Mach 6 conditions) coincide, or "cross", very close to the back of the vehicle, indicating very similar clearance characteristics. At 10 m/s ejection, the store still clears sufficiently, at a slightly smaller margin.

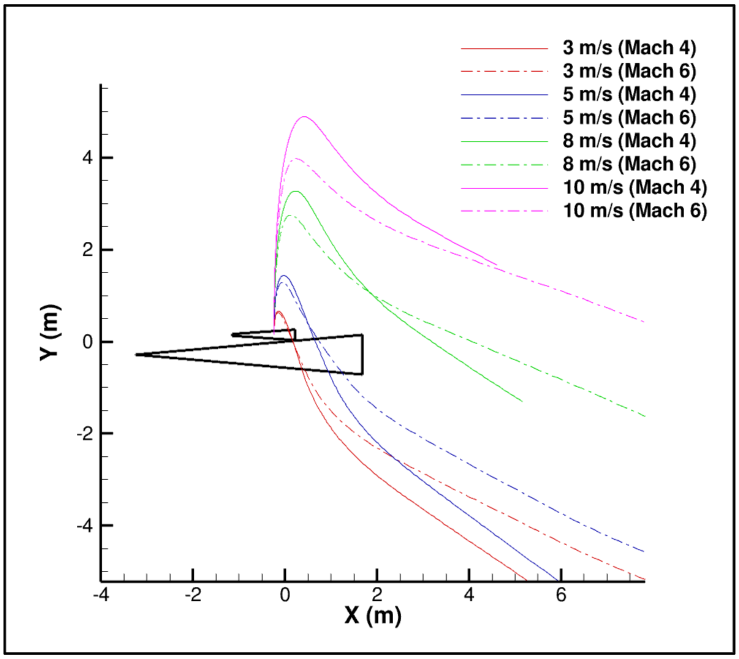


Figure 3.13 Center of mass trajectories for Cases A1 through A8

The comparison between Mach 4 and Mach 6 trajectories becomes clearer when time is considered. The streamwise "x" position and vertical "y" position time histories are plotted in Figure 3.14. From the *x-t* plots, one can see the expected trend that Mach 6 freestream conditions move the store more quickly in the downstream direction due to the increased drag force. However, the *y-t* plots show that the peak vertical positions in Mach 6 conditions are lower than those in Mach 4 flow and that those peaks are reached in less time. Although the drag force is greater in Mach 6 conditions, there is less time for the store to clear the vehicle before it drops.

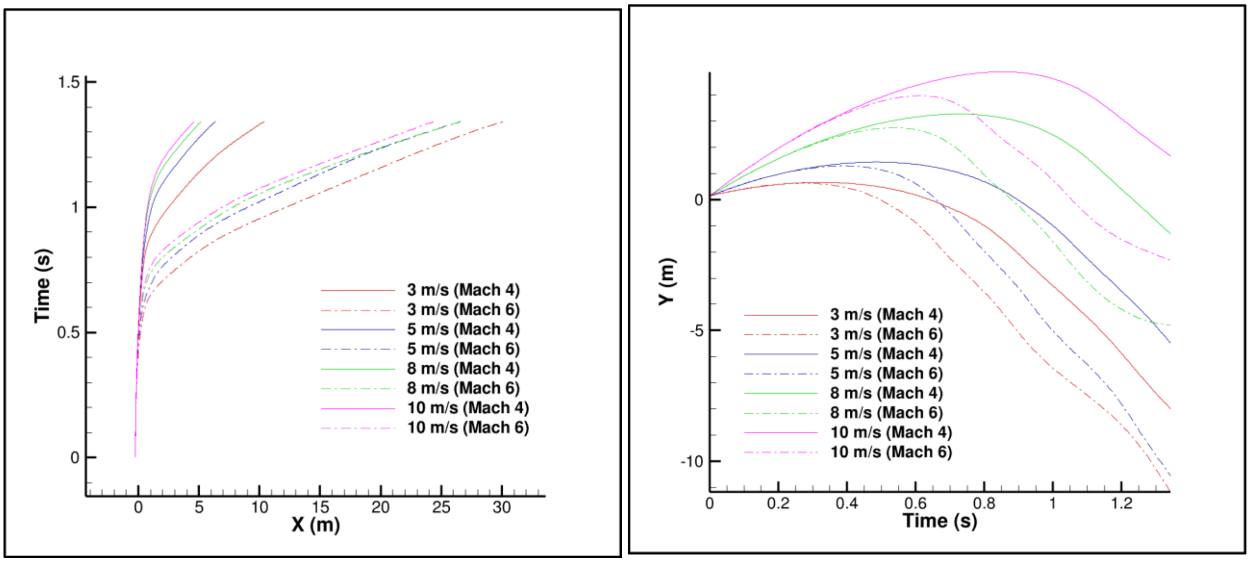


Figure 3.14 Streamwise “x” (left) and vertical “y” (right) trajectories for Cases A1 through A8

The reason for this behavior is related to the store’s rotation. In each of these simulations, the store tends to have a nose-down pitching rotation because of a high-pressure region between the base of the store and the vehicle in the store’s initial position. In Mach 6 freestream conditions, this high-pressure region is even more dramatic, and the rotation is even greater. As the store rotates forward more quickly in higher speed flow, the lift force decreases more quickly (that is, the net downward force increases). As an example, Figure 3.15 shows a snapshot of Cases A3 and A7 (8 m/s ejection, Mach 4 and Mach 6), both at *t = 0.503 s*, showing that the store is rotated further nose-down in the Mach 6 case. The pitching moment time history is shown in Figure 3.16 for all cases, A1 through A8, revealing the same effect. All Mach 6 freestream simulations show faster nose-down pitching than in Mach 4 conditions.

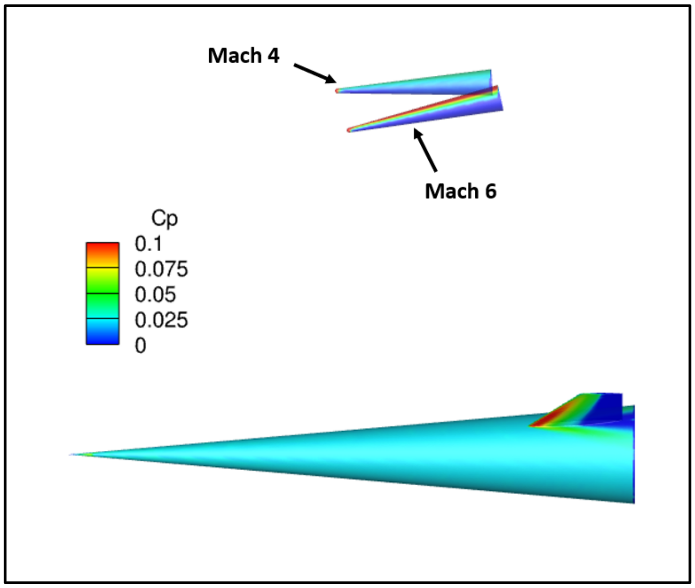


Figure 3.15 Time t = 0.503 s in Mach 4 and Mach 6 freestream conditions, 8 m/s ejection velocity

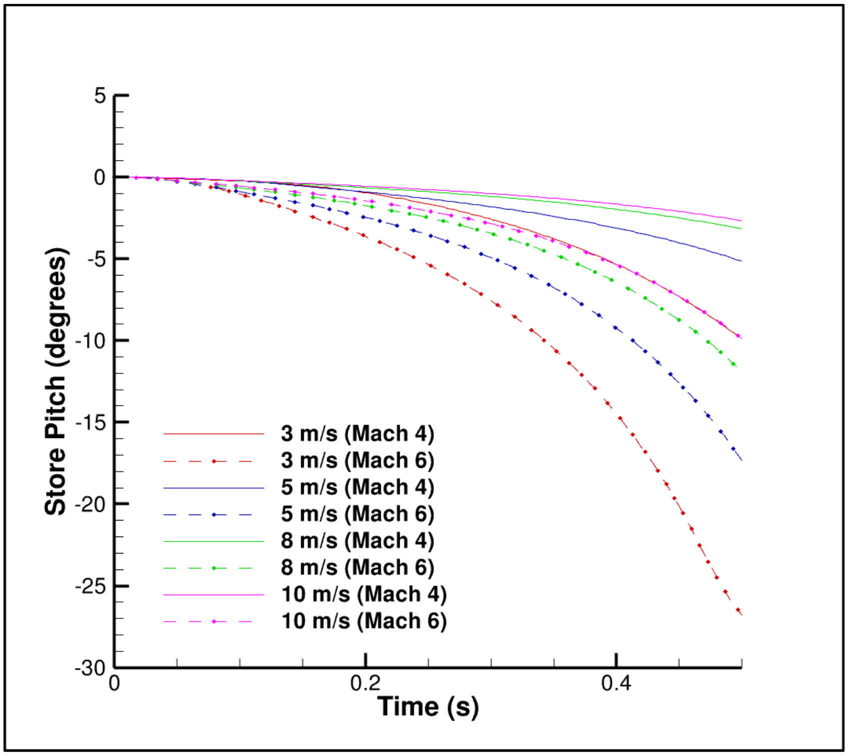


Figure 3.16 Store pitch for Cases A1 through A8, where negative values are nose-down pitch

It is clear from the simple cone store simulations that the cone does not provide sufficient drag on its own to clear the vehicle, especially when restricted to the lower ejection velocity required (about 3 m/s). In addition, it has been shown that increasing freestream velocity does not necessarily improve separation safety performance for this design. With the data shown thus far, safe separation in one set of freestream conditions may not guarantee safe separation at higher speeds, despite increased drag. It remains critical to study separation characteristics at higher speeds, even if lower-speed separations are deemed safe. With these things in mind, the investigation continues by modifying the store geometry to increase its drag. Perhaps such geometric changes could yield safe separation results at the required lower ejection velocities.

Reversed Cone

Before adding any drag features to the store, a proof-of-concept scenario was tested by simply reversing the store's orientation so that it is facing backward, exposing the flat base to the incoming flow. Mach 4 freestream conditions are simulated in Cases B1 through B3, and Mach 6 conditions in Cases B4 through B6.

Mach 4 Conditions – Reversed Cone

The reversed cone orientation results in a much stronger bow shock that sets up in front of the store, dramatically increasing the aerodynamic drag forces, aiding the store in safe separation from the vehicle. These flow features can be seen in the initial solution which is depicted in Figure 3.17 for Mach 4 freestream conditions. Like the forward cone case, the analysis starts with a 3 m/s ejection velocity and is then progressively increased.

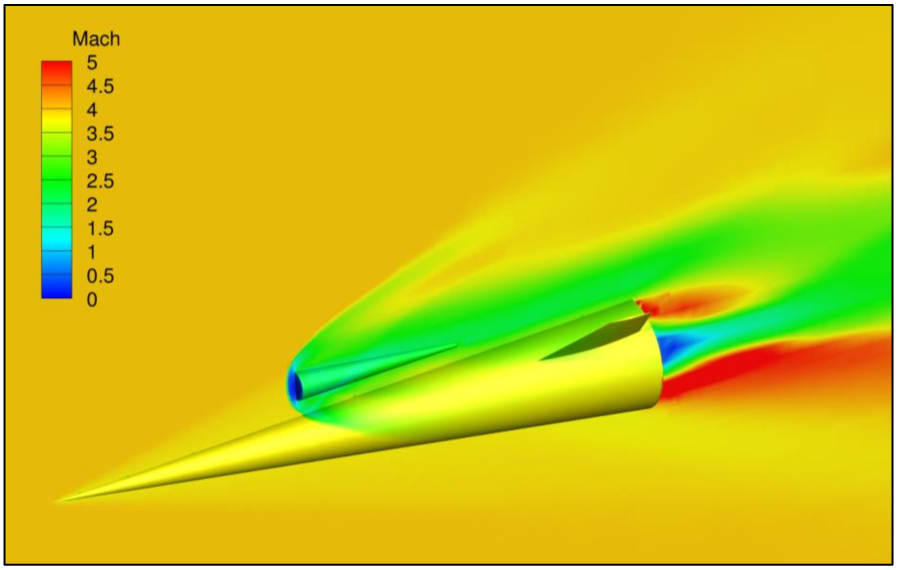


Figure 3.17 Mach number contours at time, t = 0, in Mach 4 freestream conditions

Individual trajectories are shown in Figure 3.18 through Figure 3.20 for Cases B1, B2, and B3 by a series of snapshots. Most notably, all cases clear the vehicle – the 3 m/s case by a small margin, the 5 m/s case by a much safer margin, and the 8 m/s case even more so. The trajectories of the centers of mass for each case are shown together in Figure 3.21. This analysis of the reversed cone shows the potential that additional drag features have for safe store separation in an upward ejection configuration. Intuitively, one may think that increasing the freestream Mach number increases the store drag, leading to safer separation. However, given the "less-than-obvious" results shown in the case of the forward-facing cone, it is important here, as well, to simulate higher-speed flows and compare the results.

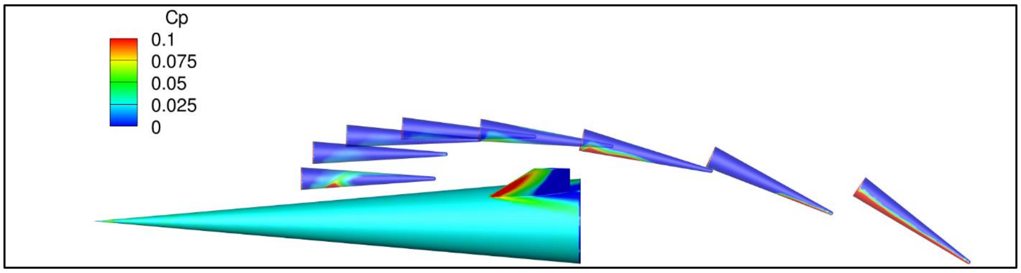


Figure 3.18 Store trajectory – Case B1: Mach 4, 3m/s

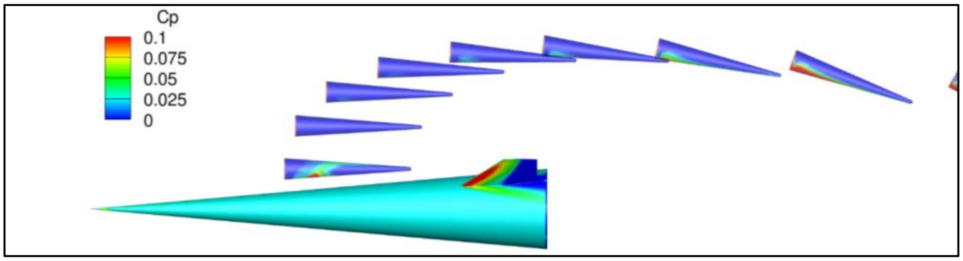


Figure 3.19 Store trajectory – Case B2: Mach 4, 5m/s

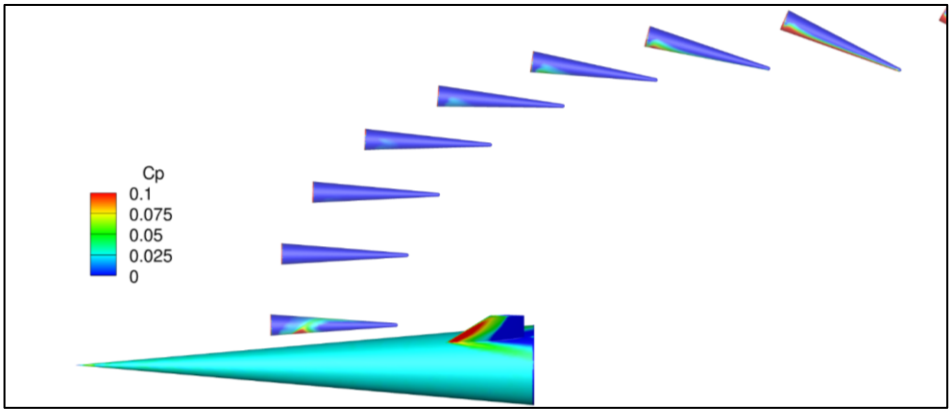


Figure 3.20 Store trajectory – Case B3: Mach 4, 8m/s

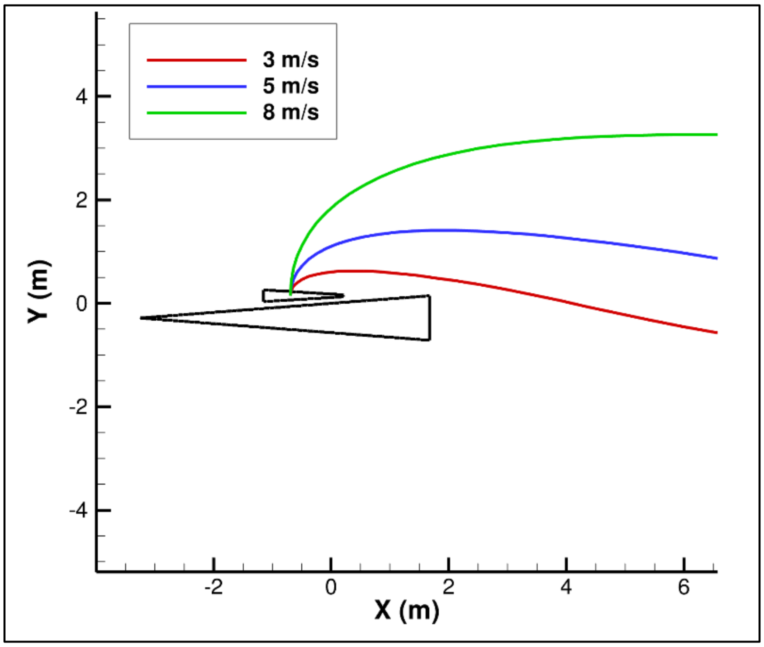


Figure 3.21 Center of mass trajectories for Cases B1 through B3

Mach 6 Conditions – Reversed Cone

Mach 6 freestream conditions are simulated for the reversed cone store in Cases B4, B5, and B6, which feature 3 m/s, 5 m/s, and 8 m/s ejection velocities. The initial, *t = 0*, solution is shown in Figure 3.22 below. The conical oblique shock around the carrier vehicle and the strong bow shock ahead of the store resemble those seen in the Mach 4 solution, with decreased shock angles as expected for higher speed flow.

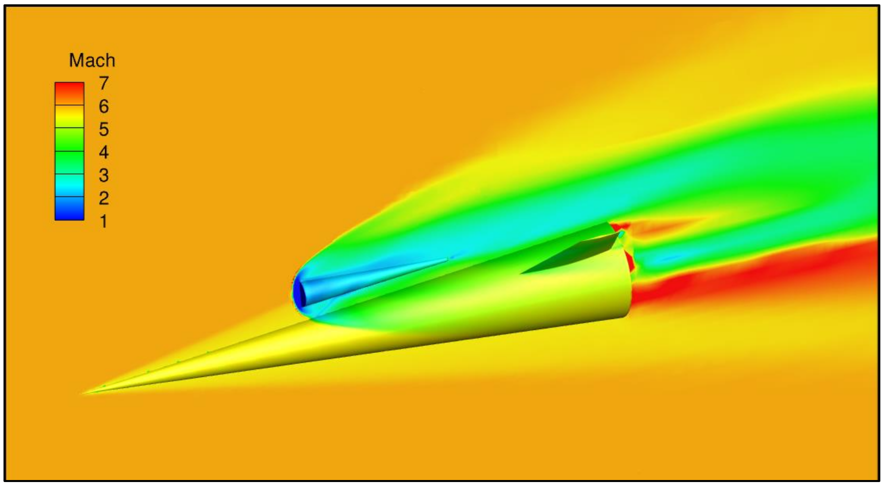


Figure 3.22 Mach number contours at time, t = 0, in Mach 6 freestream conditions

Store separation trajectories are visualized in Figure 3.23 through Figure 3.25. As in Mach 4 conditions, the 3 m/s ejection simulation clears the vehicle by a small margin at Mach 6. The 5 m/s case safely separates with a greater margin and the 8 m/s case with an even greater margin still. These trajectory paths are compared directly to the Mach 4 separation trajectories in Figure 3.26.

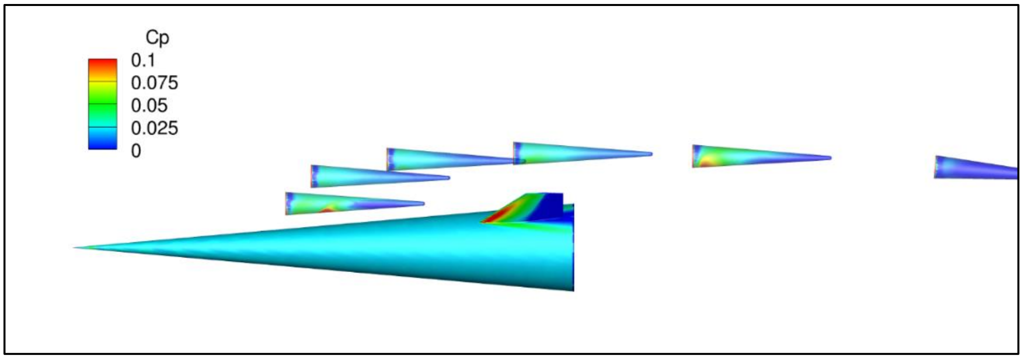


Figure 3.23 Store trajectory – Case B4: Mach 6, 3m/s

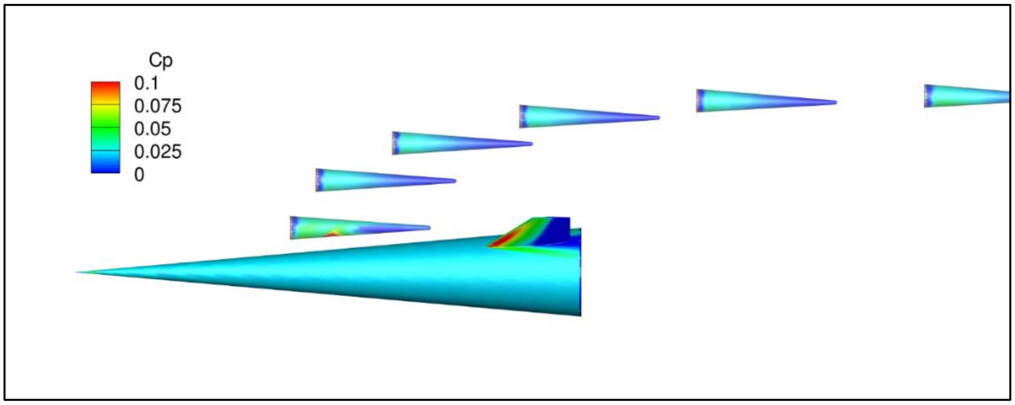


Figure 3.24 Store trajectory – Case B5: Mach 6, 5m/s

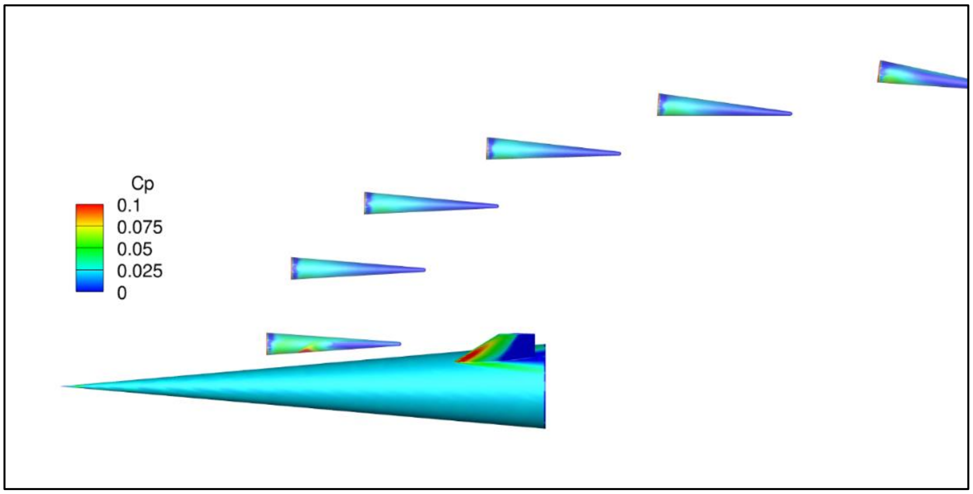


Figure 3.25 Store trajectory – Case B6: Mach 6, 8m/s

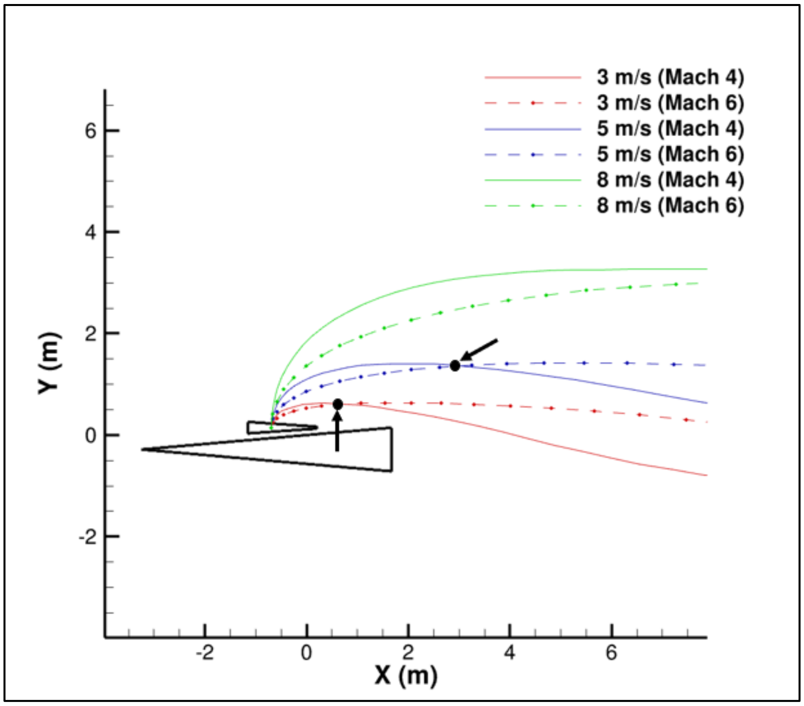


Figure 3.26 Center of mass trajectories for Cases B1 through B6

As shown in Figure 3.26, for a given ejection speed, the trajectory of the Mach 6 case starts vertically lower than the corresponding Mach 4 trajectory. After some point where the trajectories “cross”, the remainder of the Mach 6 trajectory is higher than its Mach 4 counterpart. The “crossing point” is significant because, if it is upstream of the aft end of the vehicle, the Mach 6 case has greater separation clearance than the Mach 4 case at the same ejection velocity. Otherwise, the Mach 4 case has greater clearance for safe separation. This crossing point moves further downstream at greater ejection velocities, and it is upstream of the end of the vehicle only for the 3 m/s ejection case. As other drag-enhancing designs are evaluated, this relationship should be investigated to understand the impact that freestream Mach number has on store separation clearance.

The streamwise “x” position and vertical “y” position time histories are plotted for all reversed cone cases in Figure 3.27. There are some key differences between the relationships shown here and those shown for the forward-facing cone (Figure 3.14). As expected, the streamwise position plot for the reversed cone shows that the store travels faster in the streamwise direction for Mach 6 cases than Mach 4. Interestingly, though, the streamwise component of the trajectory is independent of ejection velocity for all reversed cone cases, for each set of freestream conditions. That is, for all reversed cone Mach 4 simulations, the x-position versus time relationships are nearly identical, regardless of ejection velocity. The same is true for the reversed cone Mach 6 cases. This was not the case for the forward-facing cone, as shown in Figure 3.14. This indicates that the addition of drag-enhancing features to the store may be able to greatly reduce its initial sensitivities which impact drag forces. This may help trajectory analysis become more predictable and reliable.

The vertical component (y-position) trajectories in Figure 3.27 show a very consistent, but opposite, relationship. The vertical trajectory for all reversed cone cases appears to be independent of the freestream Mach number, dependent only on ejection velocity. The Mach 4 and Mach 6 vertical trajectories for a given ejection velocity do not deviate significantly until the store is well past the downstream end of the vehicle. This was not the case for the forward-facing store in Figure 3.14, where the Mach 4 and Mach 6 vertical trajectories deviate more significantly for a given ejection velocity. Again, this indicates that there may be reduced sensitivity in trajectories for stores with drag-enhancing features. This could be a significant advantage for design engineers looking to develop store drag features for safe, reliable separation.

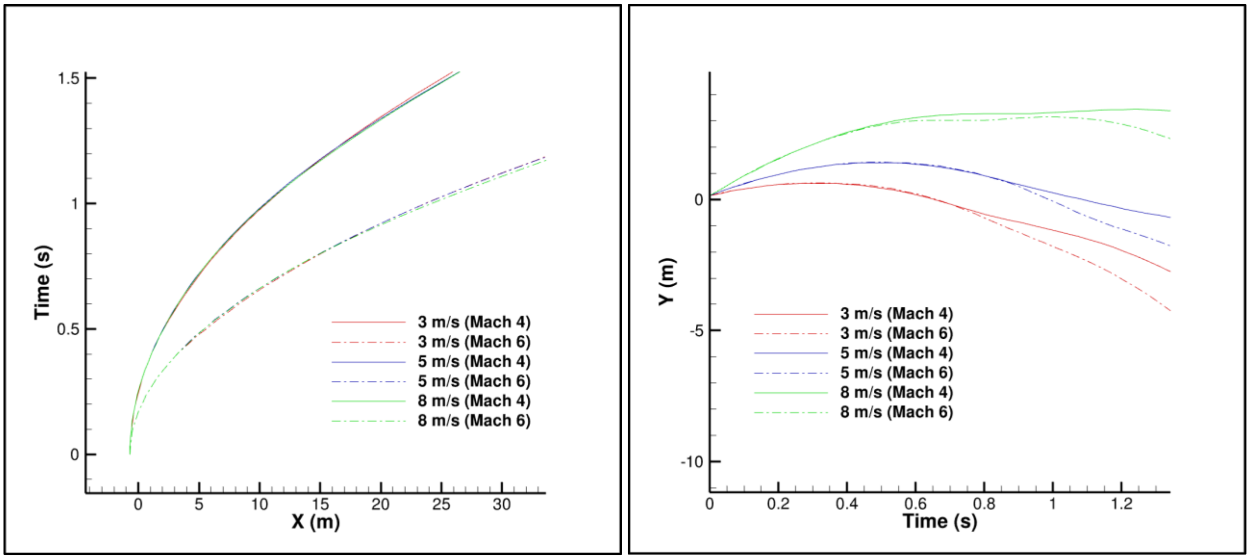


Figure 3.27 Streamwise “x” (left) and vertical “y” (right) trajectories for Cases B1 through A6

Despite the success shown in the reversed cone simulations, this design is not feasible in practice. A launch with the store oriented backward would necessitate some kind of maneuver to turn the store around. This could lead to a significant loss in the store’s momentum and is not desirable. For this reason, we look to alternate store designs that increase drag after store ejection while maintaining the forward-facing orientation.

Modified Store – Solid Disk

To increase the aerodynamic forces on the store after ejection, a 0.336m diameter drag disk was added to the base of the store as shown in Figure 3.2. The intent is that this disk would remain attached to the store during separation and then be jettisoned from the store after it has cleared the carrier vehicle. In addition to the bow shock at the store’s nose, this design creates an additional, much stronger shock that sets up in front of the disk on the base of the cone. This is clearly shown in the initial time solution as depicted in Figure 3.28.

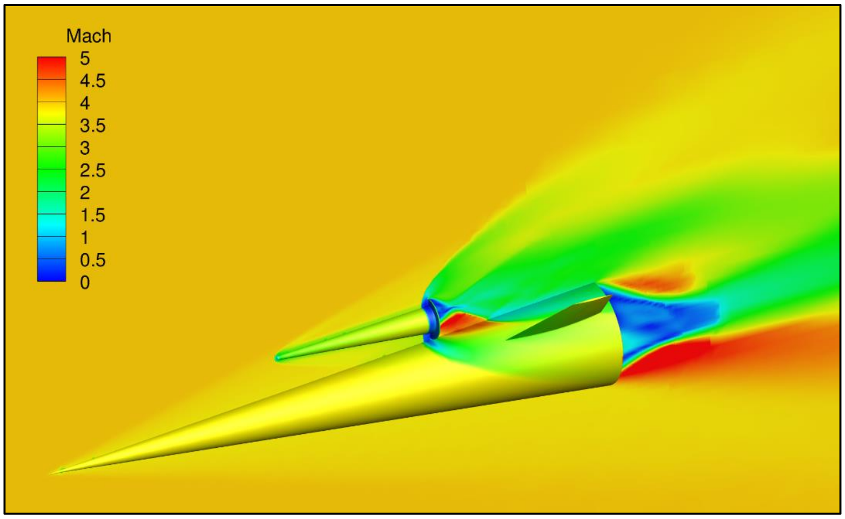


Figure 3.28 Mach number contours at time, t = 0, in Mach 4 freestream conditions

With sufficient aerodynamic forces to ensure safe separation from the vehicle, the scope of the analysis is now limited to 3 m/s ejection velocities, per the design target. Both Mach 4 and Mach 6 freestream conditions are simulated. The store trajectories for cases C1 and C2 are shown in Figure 3.29 and Figure 3.30, respectively. As desired, this design clears the carrier vehicle with a 3 m/s ejection velocity in both Mach 4 and Mach 6 freestream conditions. The Mach 4 and Mach 6 trajectories are compared in Figure 3.31. As was discussed for the reversed cone store, the Mach 4 and Mach 6 trajectories for the solid disk “cross” slightly behind the downstream end of the vehicle. This indicates that the Mach 4 trajectory has slightly better clearance characteristics than the Mach 6 trajectory, but both are satisfactory.

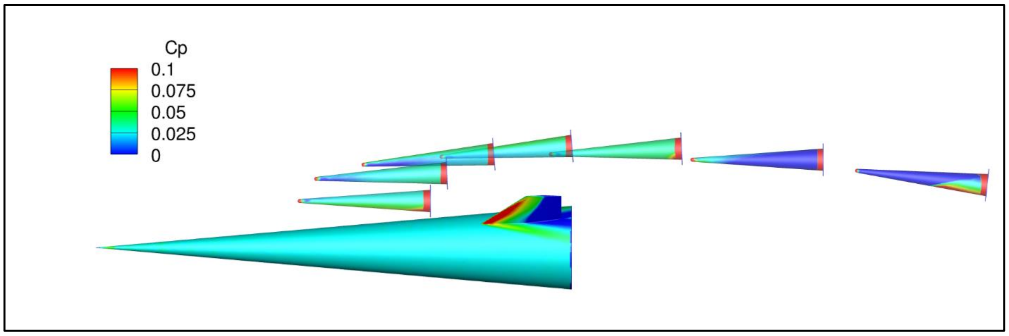


Figure 3.29 Store trajectory – Case C1: Mach 4, 3m/s

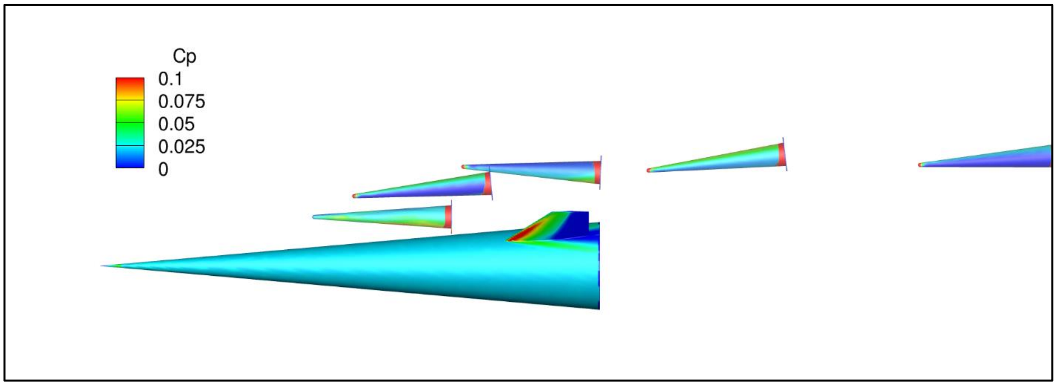


Figure 3.30 Store trajectory – Case C2: Mach 6, 3m/s

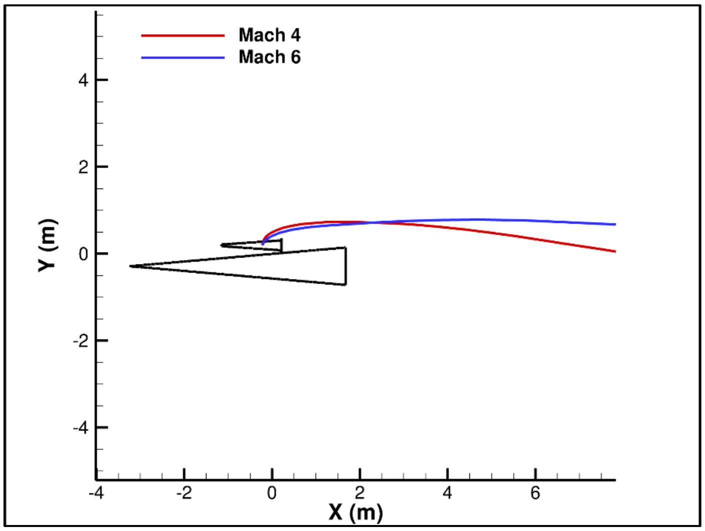


Figure 3.31 Center of mass trajectories for Cases C1 and C2

Although the drag disk configuration clears the vehicle as desired, an additional behavior arises in the store's trajectory that warrants investigation. As the store separates from the vehicle, its pitch orientation oscillates significantly. This effect is especially present in the Mach 6 case (C2). The simulation tracks the motion of the store about 70 meters beyond the end of the vehicle. For the Mach 4 case, the pitch of the store oscillates between +/- 7.5°. The Mach 6 case oscillates with a steadily increasing amplitude up to 41°. This oscillatory behavior in the store's attitude is shown in Figure 3.32 below for the Mach 4 and 6 cases. A comparative time history is shown for both cases in Figure 3.33. Comparing the Mach 4 and Mach 6 cases highlights two main effects. First, the Mach 6 case oscillates at a significantly higher frequency (about 3 times greater) than the Mach 4 case. Second, the Mach 6 pitch amplitude is not only greater, but it also steadily grows throughout the simulation.

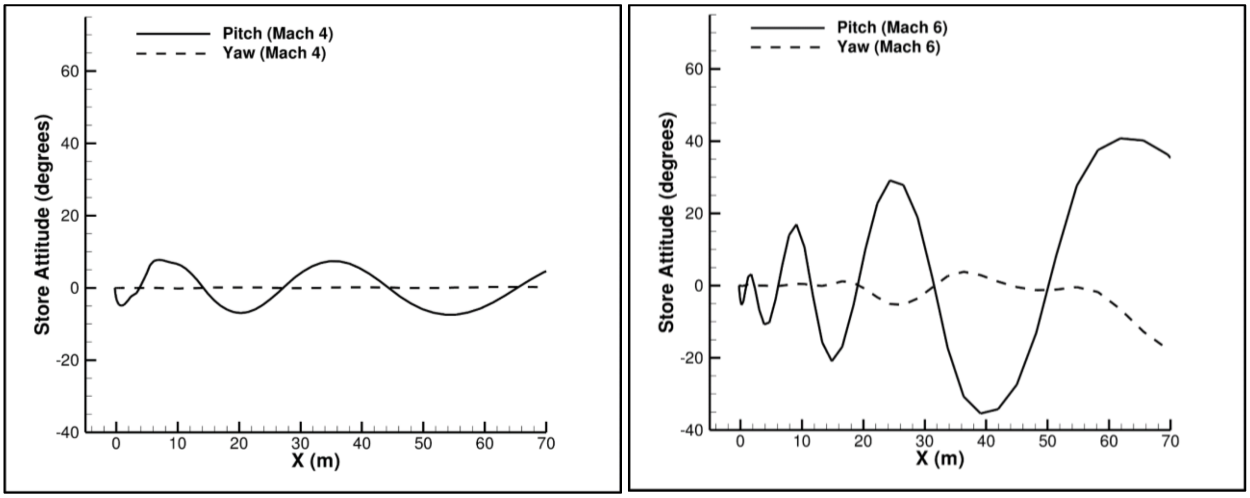


Figure 3.32 Store pitch and yaw versus streamwise "x" position for Cases C1 (left) and C2 (right)

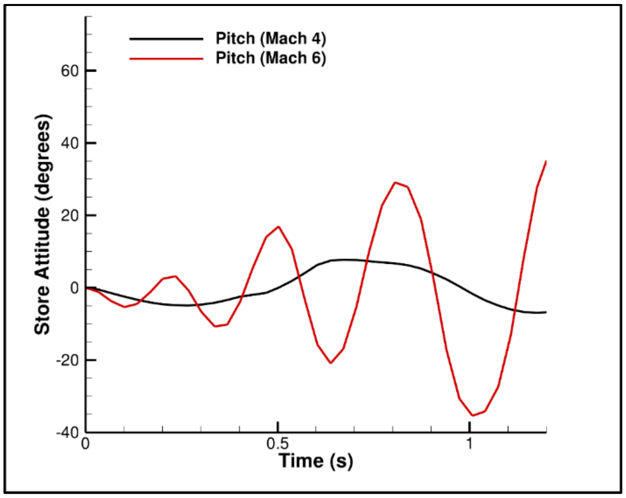


Figure 3.33 Store pitch time history for Cases C1 and C2

Perhaps, though, the main effects of this phenomenon are beyond the scope of ensuring safe store separation from the vehicle. One should note that these pitch oscillations primarily take place well beyond the end of the vehicle (where x ≈ 2m) – after separation and perhaps even after the jettison of the disk, which may eliminate the oscillations altogether. Although not critical to the question of safe store separation, this behavior is certainly of interest to efforts made in controlling the store trajectory after separation. At this point, interest is taken in whether the disk design can be modified to alleviate pitch oscillation, which leads the investigation to the next store design considered.

Modified Store – Ring Disk

The final store design analyzed is the ring disk configuration as shown in Figure 3. This design is identical to the solid disk store except for a 0.5” wide slot cut from the disk around the 9-inch diameter base of the cone. This design was simulated attempting to alleviate some of the angular oscillations present in the solid disk cases. Like the solid disk, the flow field around the ring disk is characterized by a strong shock at the base of the store, as shown in Figure 3.34. Figure 3.34 also shows a closer look at the flow behind the disk, with additional flow features due to the airflow through the added slot. This flow through the gap necessitates the use of very small element sizes as recorded in Table 3.3. On a coarser grid, the gap would not be resolved, and the simulation would treat the ring disk as if it were solid. Like the solid disk case, the ring disk configuration is simulated under Mach 4 and Mach 6 freestream conditions with a 3 m/s store ejection velocity. The store trajectories are shown in Figure 3.35 through Figure 3.37 for the Mach 4 and Mach 6 cases (D1 and D2).

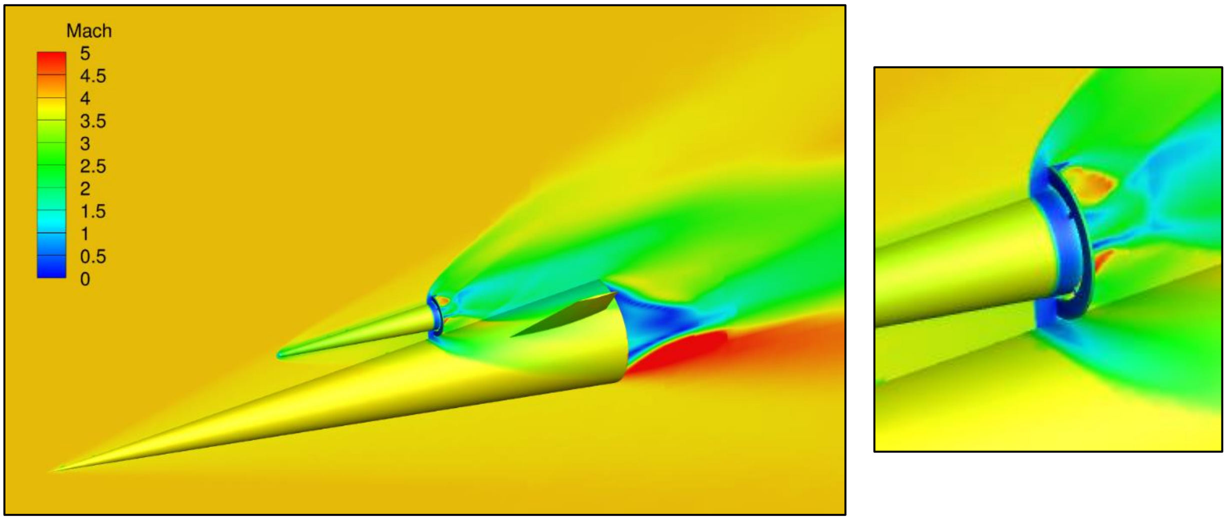


Figure 3.34 Mach number contours at time, t = 0, in Mach 4 freestream conditions

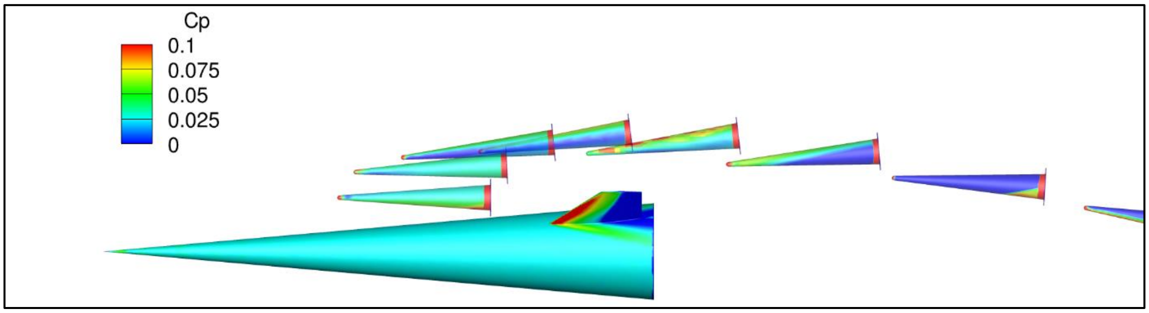


Figure 3.35 Store trajectory – Case D1: Mach 4, 3m/s

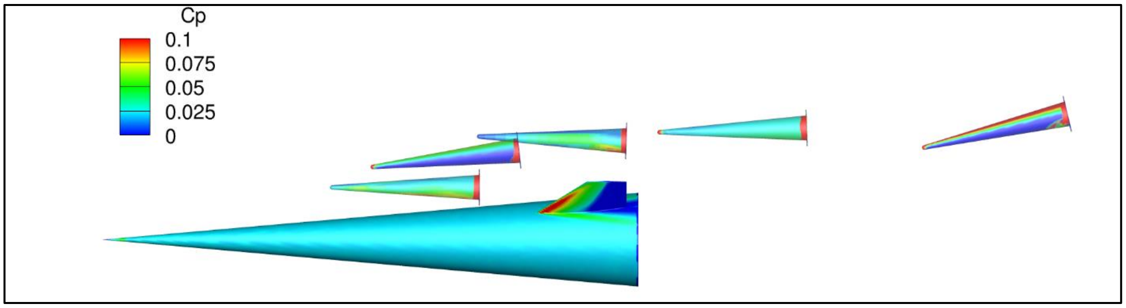


Figure 3.36 Store trajectory – Case D2: Mach 6, 3m/s

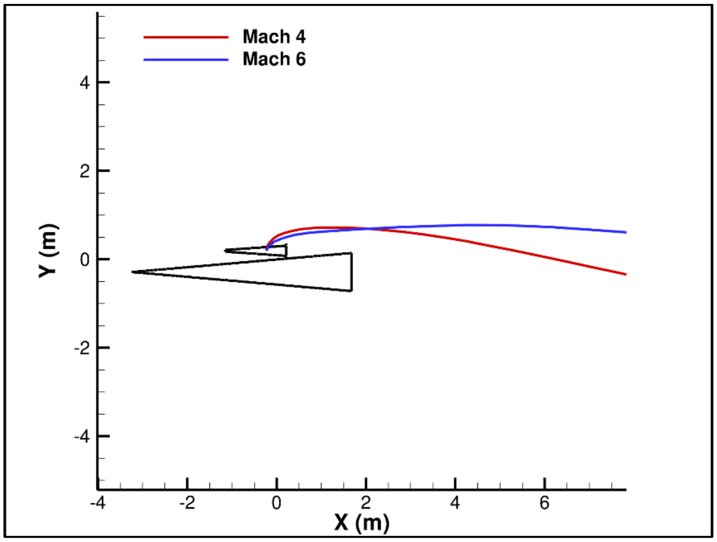


Figure 3.37 Center of mass trajectories for Cases D1 and D2

As desired, the ring disk configuration clears the carrier vehicle in both cases. In fact, there is very little difference between these trajectories and those of the solid disk store. The trajectories for the two designs (solid and ring disk) are compared side by side in Figure 3.38. In Mach 4 conditions, the effect of the reduced drag on the ring geometry is seen in a slightly lower trajectory. However, for the Mach 6 cases, the solid and ring disk trajectories are nearly identical within 5 meters of the end of the vehicle. So, like the solid disk store, the ring disk store successfully clears the vehicle with an ejection velocity of 3 m/s.

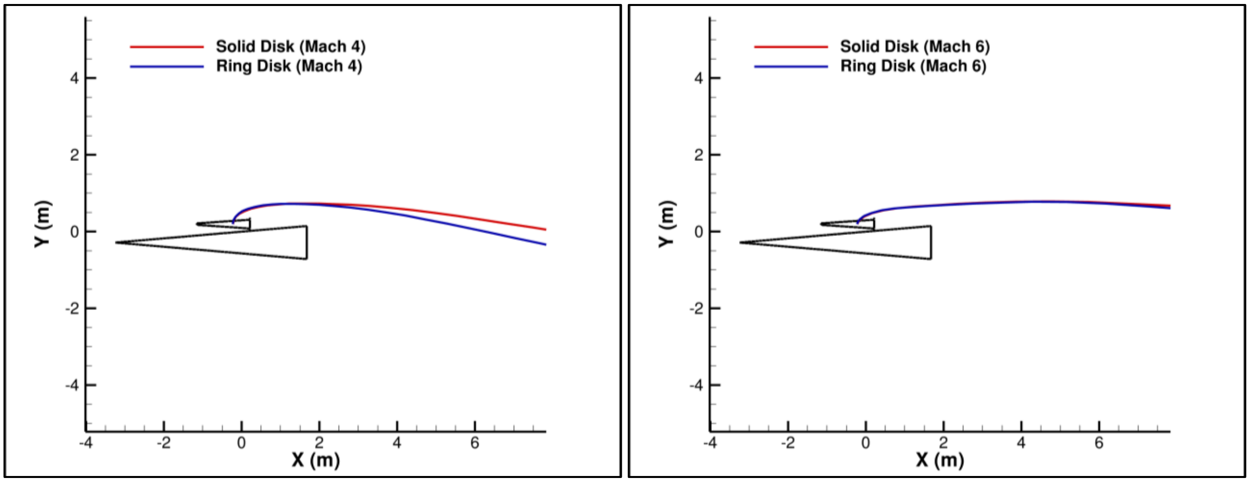


Figure 3.38 Center of mass trajectories for Cases C1 and D1 (left) and Cases C2 and D2 (right)

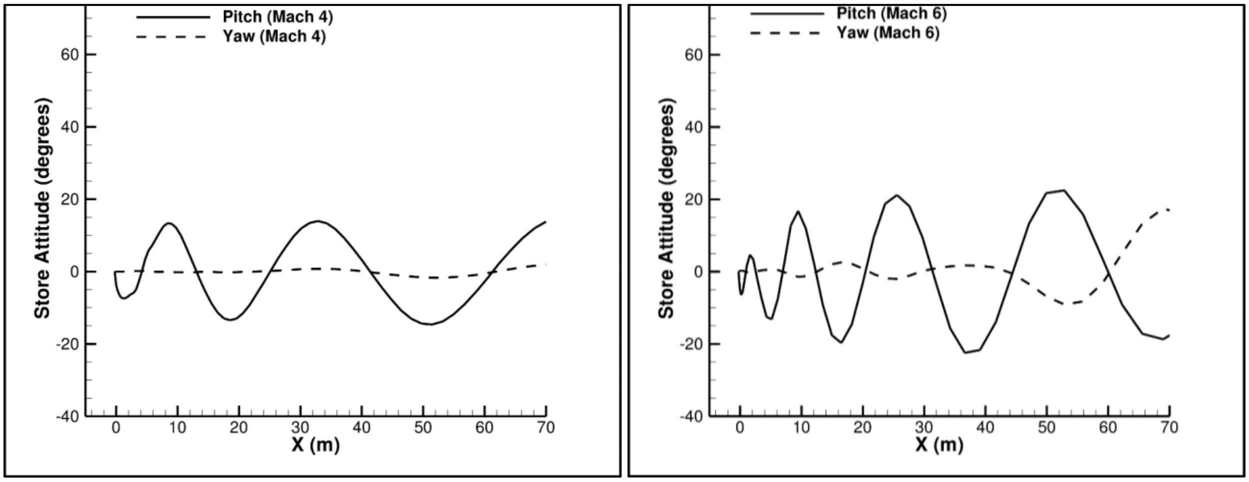


Figure 3.39 Store pitch and yaw versus streamwise "x" position for Cases D1 (left) and D2 (right)

Interest in the ring design was initially motivated by any effects that it may have on the oscillatory behavior of the store after clearing the vehicle. Figure 3.39 plots the angular position of the store as it travels about 70m behind the vehicle, for both the Mach 4 and Mach 6 cases (D1 and D2). For the Mach 4 case, the store oscillates between +/- 14.5°, which is 7° more than the solid disk. By this measure, it seems as if the incorporation of the slot design is not helpful. However, for the Mach 6 case, the oscillation amplitude only grows to 22.5°, which is 18.5° less than the solid disk. From this point of view, one can argue that the slots can help in certain scenarios. The store pitch time histories are plotted for both cases (Mach 4 and Mach 6) together in Figure 3.40 for comparison. As shown with the solid disk, the Mach 6 case for the ring disk oscillates with a higher frequency and with a larger (and growing) amplitude than the Mach 4 case. However, the disparity between the Mach 4 and Mach 6 amplitudes is now reduced. The effects of implementing the ringed slot can be observed in Figure 3.41 below, which compares the store pitch time histories for the solid and ring disk configurations.

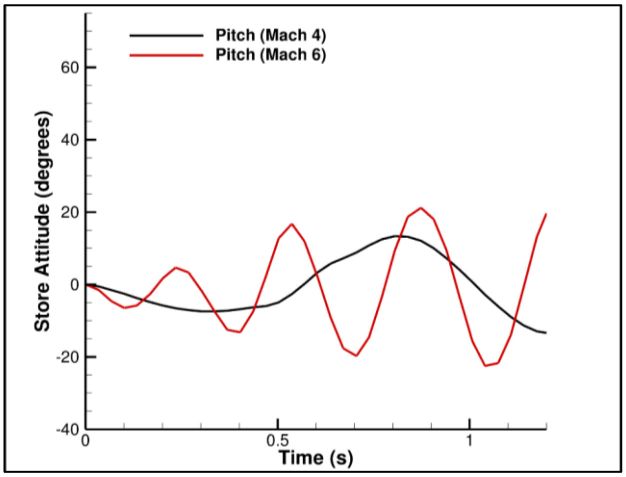


Figure 3.40 Store pitch time history for Cases D1 and D2

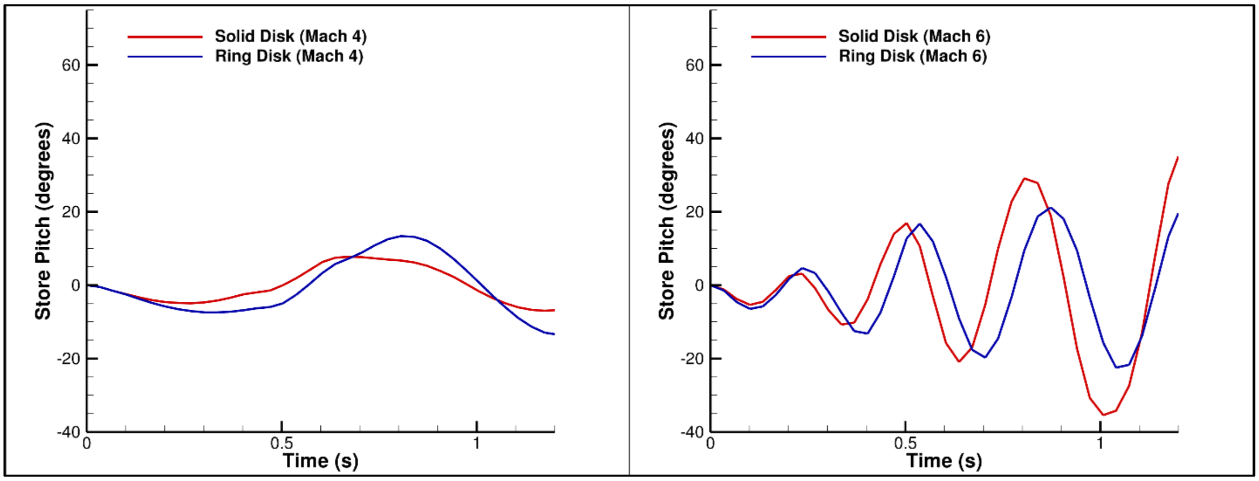


Figure 3.41 Store pitch time history for Cases C1 and D1 (left) and Cases C2 and D2 (right)

At the very least, the issue of angular oscillations has proven to be complex, and the mechanisms driving the phenomenon require further investigation. It is suggested that truly unsteady (rather than quasi-static) and viscous studies are pursued to gain a deeper understanding of the physics taking place.

6DoF Time Step Studies

A 6DoF time step study was conducted to validate the choice of the time step. Ensuring that a finer time step does not significantly alter the results is important to reinforce solution credibility. This is intentionally referred to as a "6DoF" time step study, reiterating that this pertains to the time step from one steady state solution to the next – each time step that the store is repositioned – not the pseudo-time step used within the context of a single steady state solution. To start, all cases were run with a dimensionless time step, Δt^* = 10. Non-dimensionalized by the freestream speed of sound (a_∞ = 298m/s for T_∞ = 221K), this correlates to a physical time step, Δt = 33.6 ms.

The simple cone (Cases A1 through A8) was the first configuration to be verified. By using a refined time step, Δt^* = 1 (Δt = 3.36 ms), the trajectories shown in Figure 3.42 are obtained for the four different ejection velocities (3 m/s up to 10 m/s), for both Mach 4 and Mach 6 freestream conditions. The trajectories at a refined time step are to be used as a baseline against which the larger time step (Δt^* = 10) solutions are compared. For the 3m/s and 5m/s cases, the trajectories of the large and small time step solutions are nearly identical – at least until the store collides with the vehicle, after which the solution is invalid anyway. The higher ejection velocity cases, though, show slightly more variation. The trajectories look similar when compared to each other, but when time is considered, there is a greater deviation. The time histories of the center of mass "x" and "y" positions are shown in Figure 3.44. Although the stores follow similar trajectories, they do so at different rates. With a larger time step, the store seems to accelerate ahead of its refined time step counterpart – their corresponding positions being multiple meters apart by the time the store reaches the end of the vehicle. Clearly, this is unacceptable, and an even smaller time step was studied to verify the Δt^* = 1 solution.

To this end, another set of solutions was obtained for the simple cone using the dimensionless time step, Δt^* = 0.1 (Δt = 0.334 ms). The trajectory comparison between the Δt^* = 1 and Δt^* = 0.1 solutions, as well as the “x” and “y” position time histories, are shown in Figure 3.43 and Figure 3.45 below. The results show that there is no significant change in the solution by refining the time step to 0.1. Any noticeable deviations are far downstream and are only on the order of a few centimeters. Because of this, Δt^* = 1 is considered an appropriate time step for the simple cone store configuration.

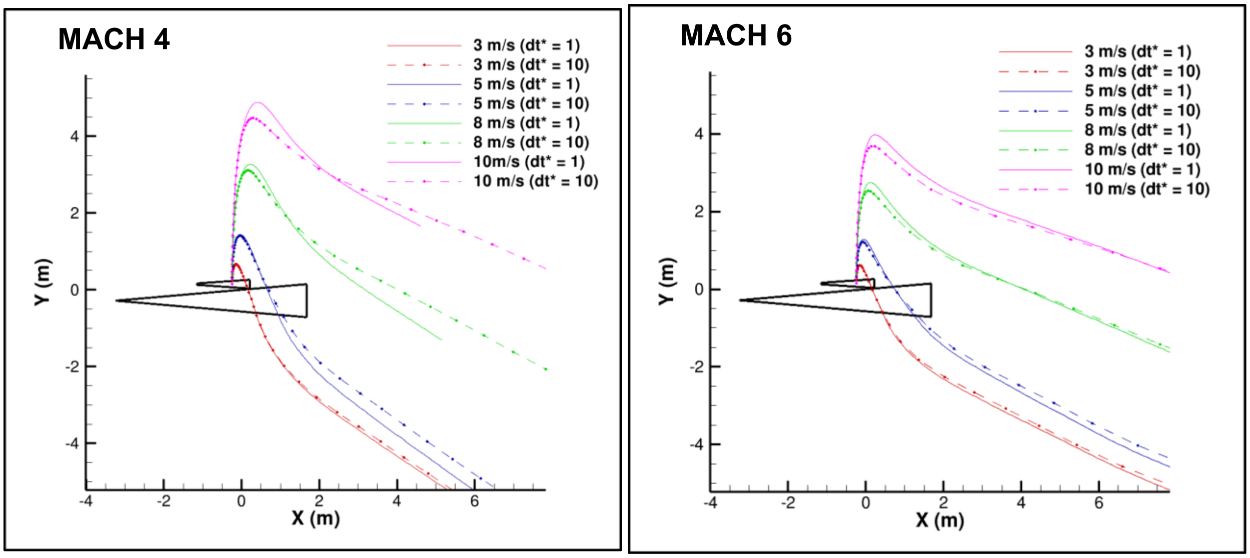


Figure 3.42 Simple cone trajectory comparisons using time steps Δt^* = 1 and Δt^* = 10

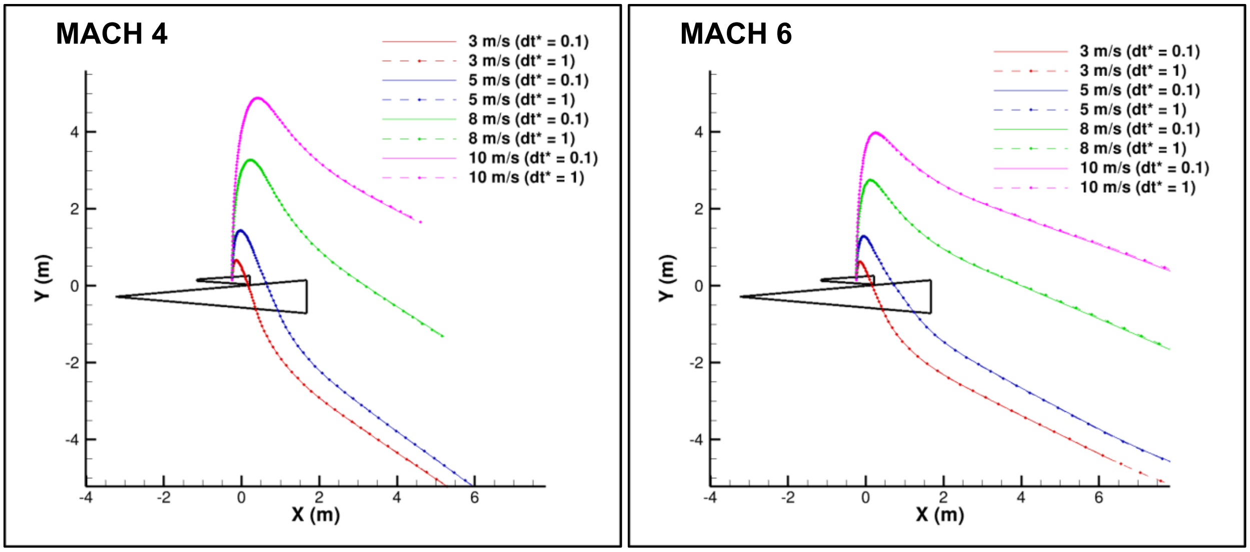
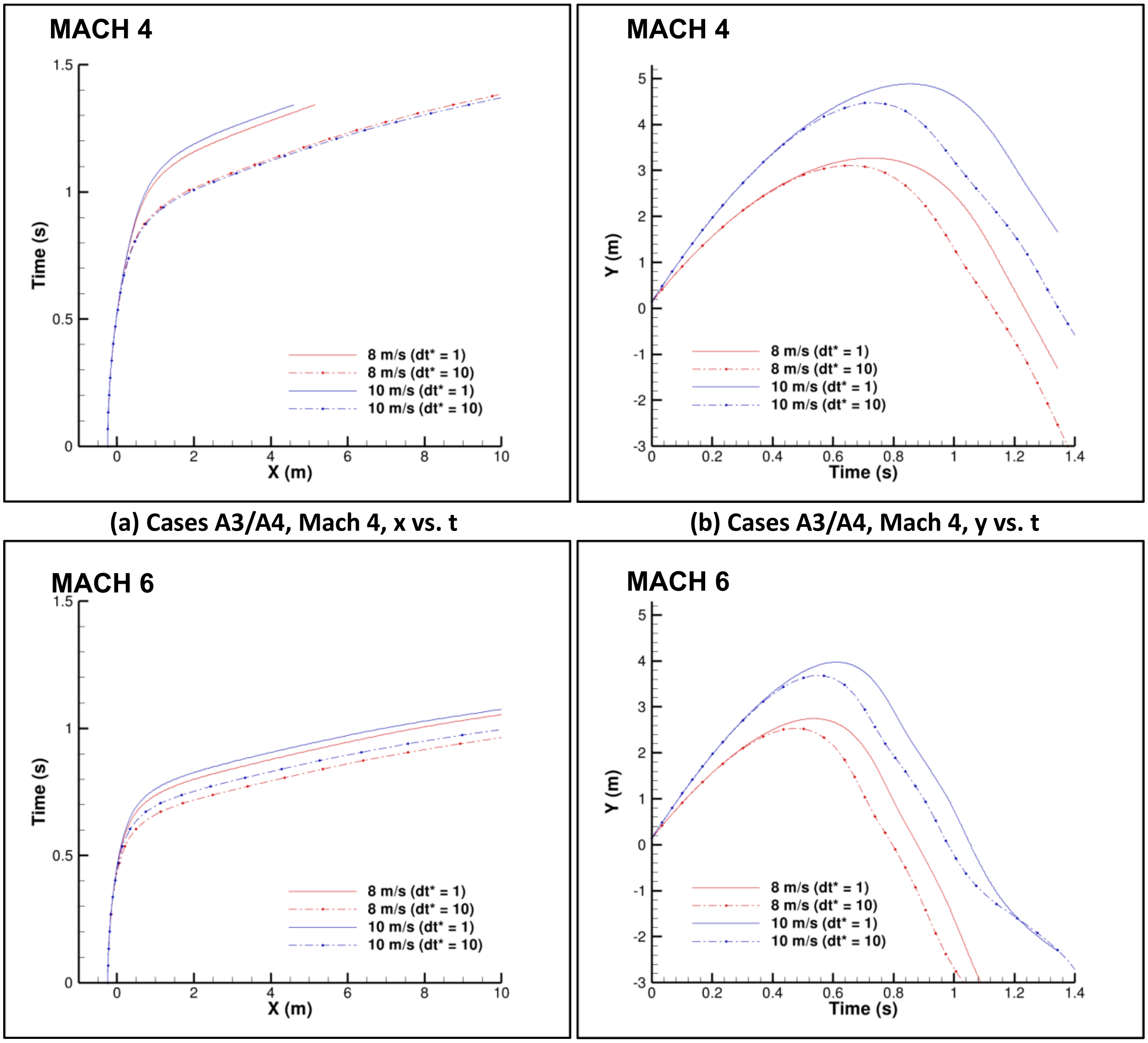


Figure 3.43 Simple cone trajectory comparisons using time steps *Δt** = 0.1 and *Δt** = 1



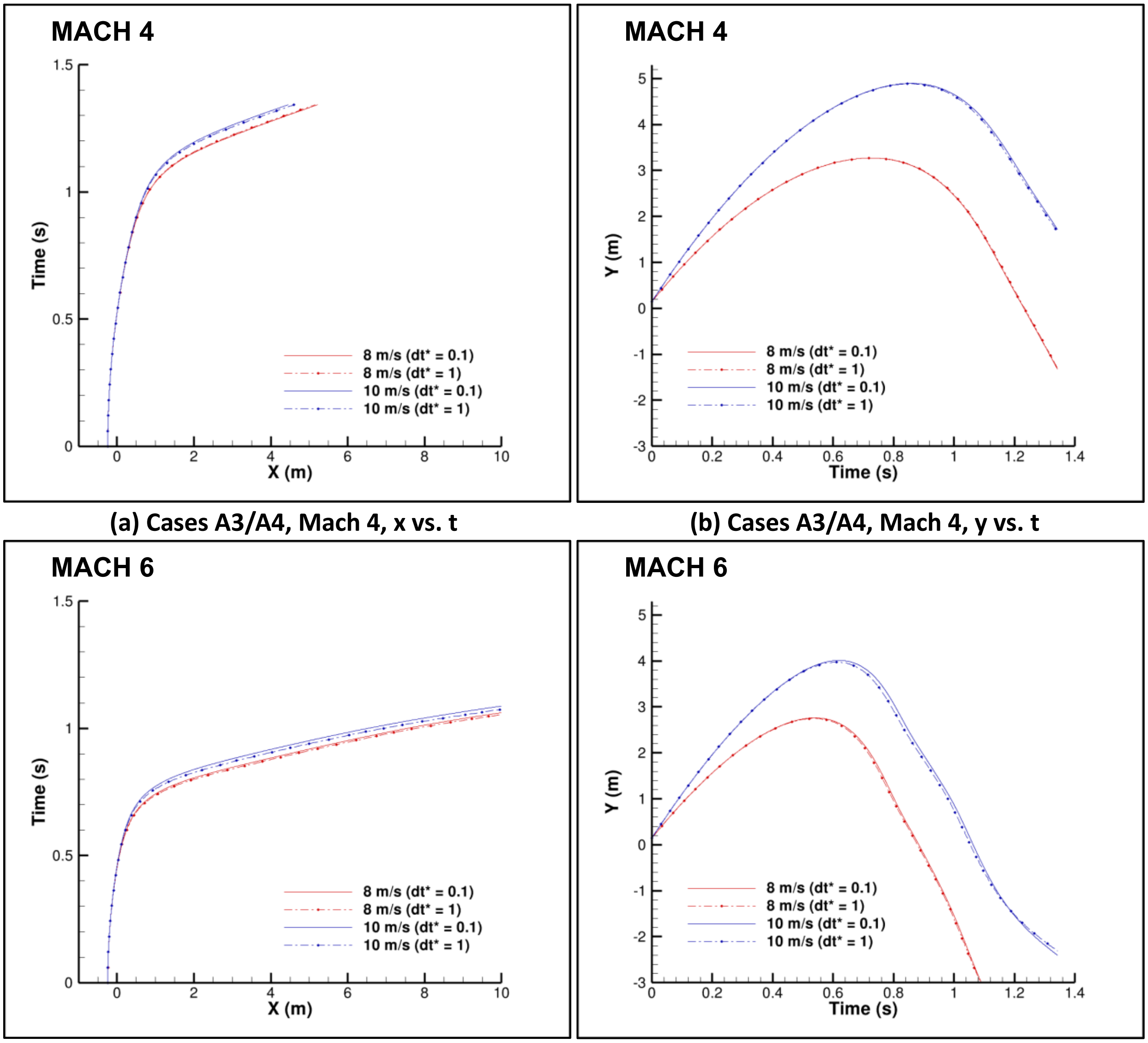
(a) Cases A3/A4, Mach 4, x vs. t

(b) Cases A3/A4, Mach 4, y vs. t

(c) Cases A7/A8, Mach 6, x vs. t

(d) Cases A7/A8, Mach 6, y vs. t

Figure 3.44 Simple cone “x” and “y” position time histories using time steps Δt^* = 1 and Δt^* = 10



(a) Cases A3/A4, Mach 4, x vs. t

(b) Cases A3/A4, Mach 4, y vs. t

(c) Cases A3/A4, Mach 6, x vs. t

(d) Cases A3/A4, Mach 6, y vs. t

Figure 3.45 Simple cone "x" and "y" position time histories using time steps Δt^* = 0.1 and Δt^* = 1

For all other store configurations (reversed cone, solid disk store, and ring disk store), solutions were generated using a time step of Δt^* = 1 and 10. Even though the Δt^* = 10 was found to be inadequate to resolve the 6DoF motion for the simple cone, it was considered here because the computational time savings would be significant if one could use a larger time step. Figure 3.46 and Figure 3.47 show the trajectory comparisons, x-position time history, and y-position time history for the reversed cone configurations (3 m/s, 5m/s, and 8 m/s ejection velocities in both Mach 4 and Mach 6 conditions). As can be seen from the figures, the deviation between the trajectories obtained using the two different time steps is not significant. Even where there is a slight deviation, one should note that the store has reached the end of the vehicle by about *x* = 2m, and up to this point, the variations are negligible in all cases. For these cases, it is decided that *Δt** = 10 is a sufficiently small 6DoF time step, as there is no significant change in the solutions by using a finer time step. So, for the investigation of safe store separation, *Δt** = 10 is certainly acceptable.

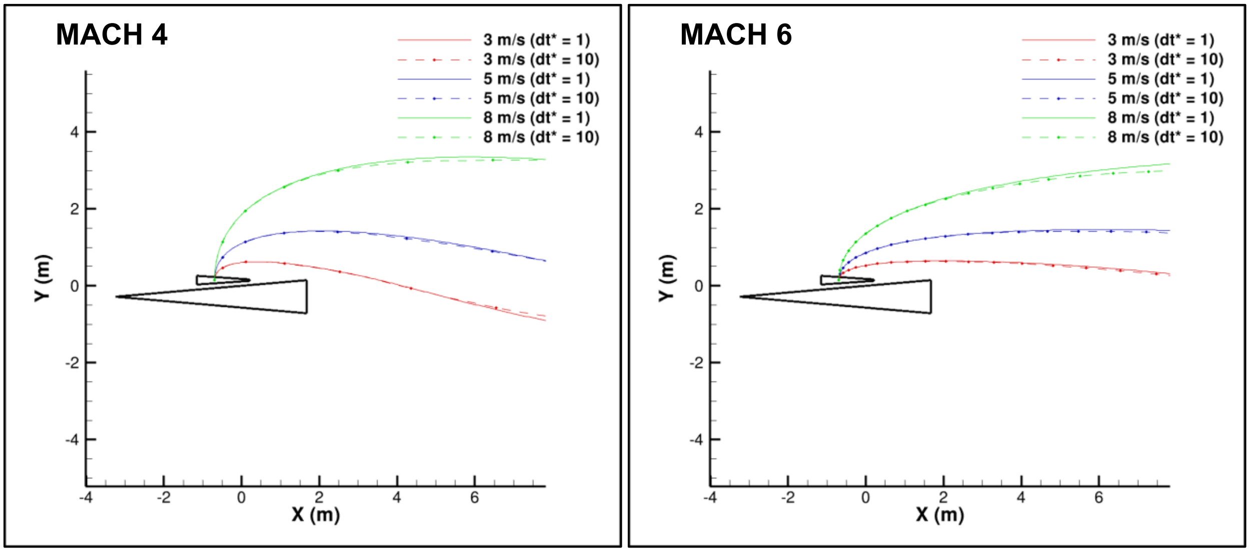
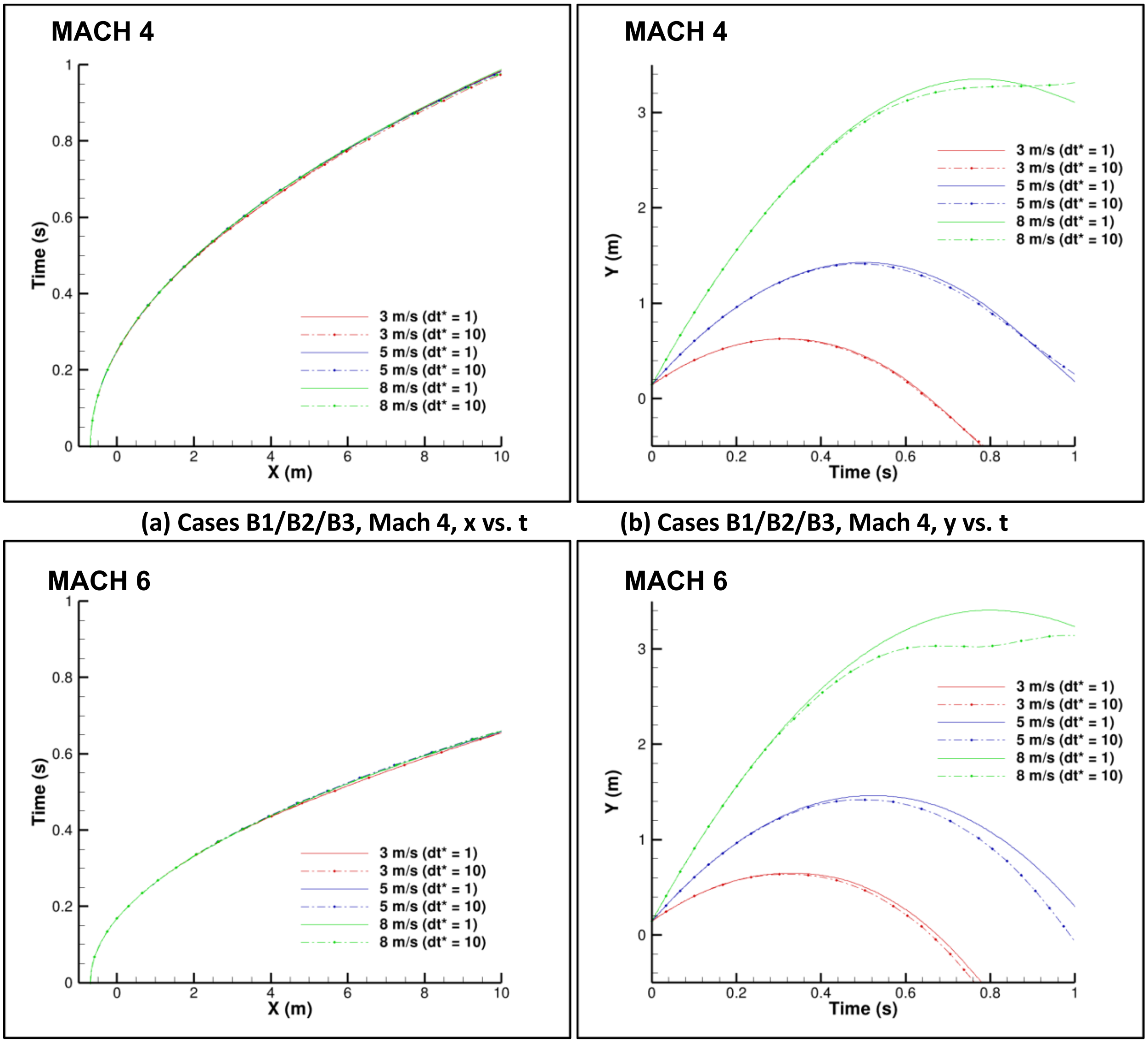


Figure 3.46 Reversed cone trajectory comparisons using time steps *Δt** = 1 and *Δt** = 10



(a) Cases B1/B2/B3, Mach 4, x vs. t **(b) Cases B1/B2/B3, Mach 4, y vs. t**

(a) Cases B4/B5/B6, Mach 6, x vs. t **(b) Cases B4/B5/B6, Mach 6, y vs. t**

Figure 3.47 Reversed cone "x" and "y" position time histories using time steps Δt^* = 1 and Δt^* = 10

A similar exercise was carried out for the store with the solid disk (Figure 3.48 and Figure 3.49) as well as the store with the ring disk (Figure 3.50 and Figure 3.51) at Mach 4 and Mach 6, both using an ejection velocity of 3 m/s. The time step study indicates that, in both cases, a Δt^* = 10 is sufficient to resolve the trajectory of the store.

It is important to note that the simple cone case requires a finer time step than the other store configurations. An initial conclusion can be drawn that the strong aerodynamic forces on the enhanced drag configurations greatly reduce the time-step sensitivity of store motion. The simple cone is sensitive to smaller pressure changes around the entire cone body while the other cases are dominated by a large aerodynamic force in a single direction, normal to the base surface. So, for this study as well as future simulations, a 6DoF time step of Δt^* = 10 is to be used for stores with drag features, and Δt^* = 1 is to be used for more streamlined bodies.

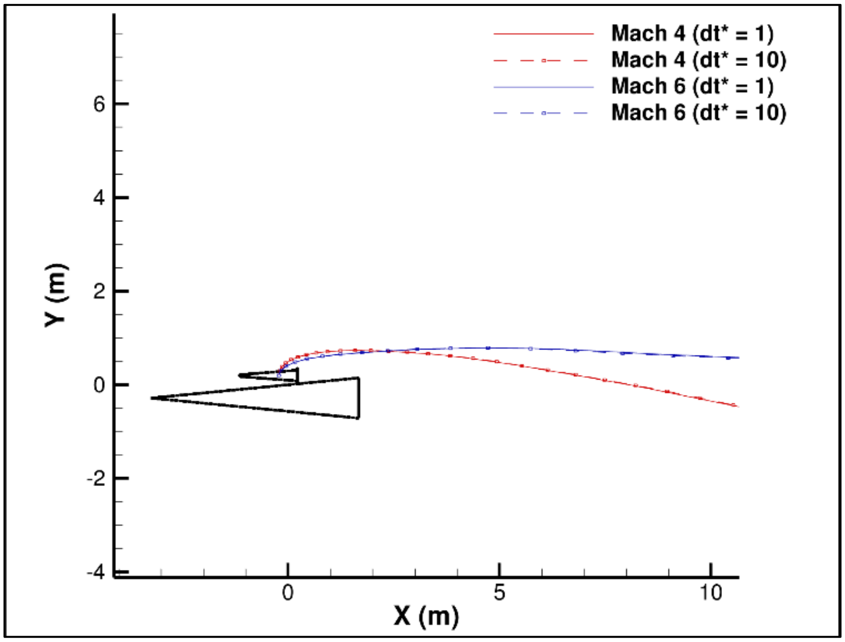


Figure 3.48 Solid disk store trajectory comparisons using time steps Δt^* = 1 and Δt^* = 10

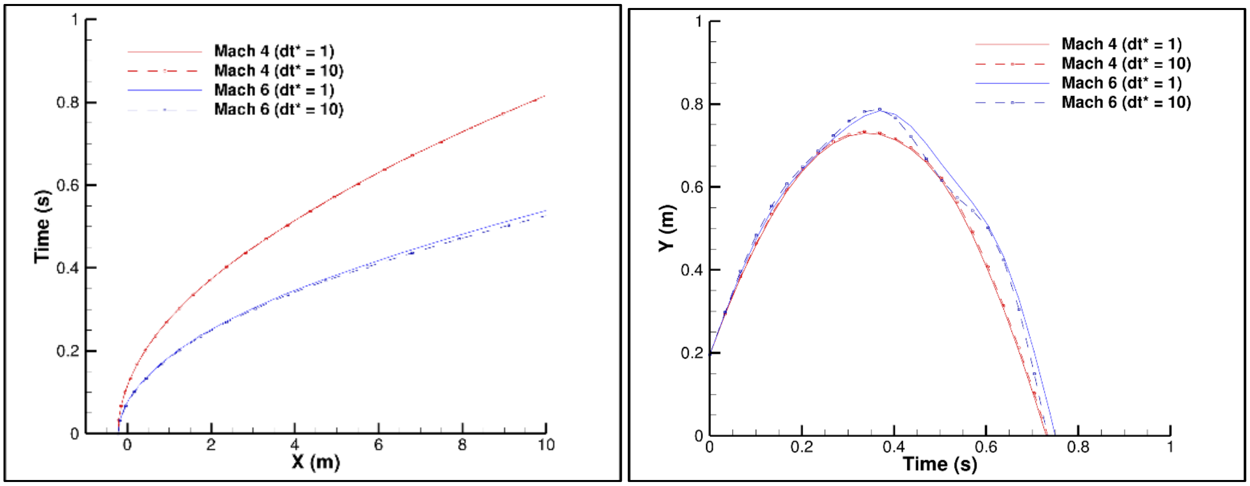


Figure 3.49 Solid disk “x” and “y” position time histories using time steps *Δt** = 1 and *Δt** = 10

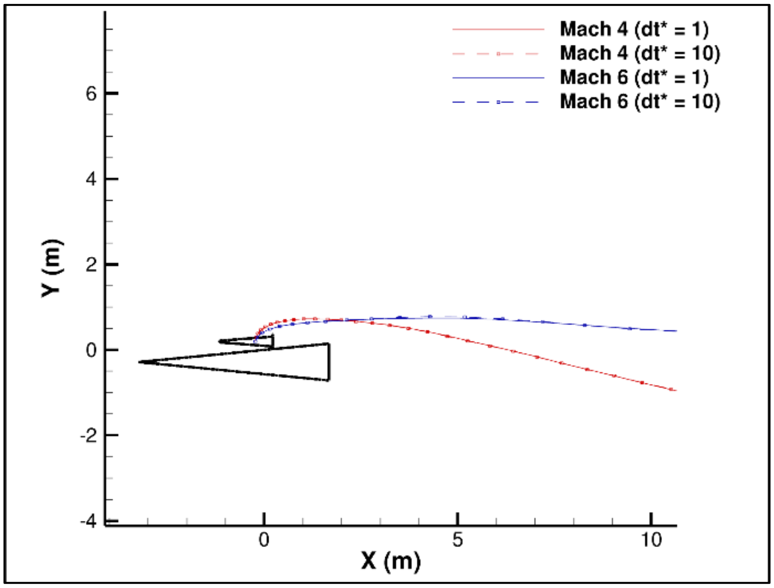


Figure 3.50 Ring disk store trajectory comparisons using time steps *Δt** = 1 and *Δt** = 10

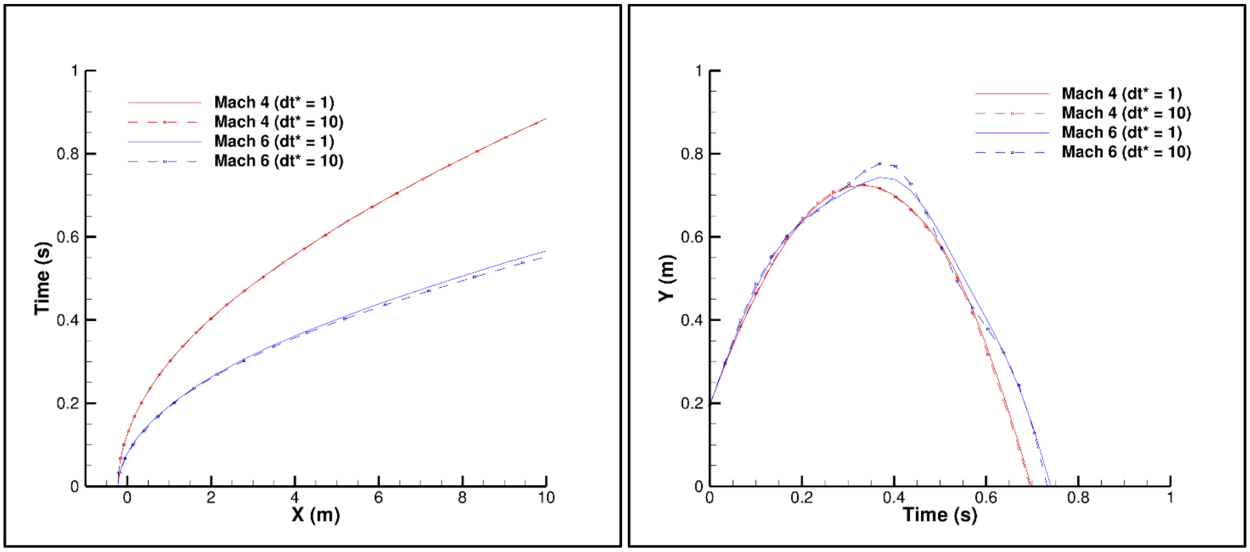


Figure 3.51 Ring disk store “x” and “y” position time histories using time steps Δt^* = 1 and Δt^* = 10

3.4 Concluding Remarks and Future Work

From this study, some important conclusions can be made regarding the separation of upward ejected stores into high-speed flows. First, one must take caution with streamlined store designs, as they may not induce sufficient drag to safely clear the carrier vehicle. This was the case with the simple cone, which required unfeasible ejection velocities to clear the vehicle. Second, one can conclude that safe separation is achievable by using drag features, such as the solid or ring disks implemented here. Finally, it is important to note that such drag features may also result in angular oscillations. These effects should be considered in drag feature design, perhaps designing such drag mechanisms to be jettisoned from the store after separation from the vehicle.

This investigation is merely the beginning of an exploration into upward store ejection into high-speed flows. Further study is certainly necessary to better understand similar design configurations. First, the design and simulation of more realistic drag feature designs, such as deployable drag fins, are recommended. This is currently underway at the University of Tennessee Chattanooga in cooperation with the University of Dayton Research Institute. The incorporation of additional flow physics is also recommended for a greater understanding of similar store separation problems. By nature of the quasi-static model, no momentum is imparted from the moving store to the surrounding fluid. Running time accurate CFD simulations could account for such effects neglected here. However, the released version of Cart3D does not include this capability, and hence, the impact of a quasi-static versus true time accurate simulation could not be ascertained. Also, full Navier-Stokes solutions should be considered to account for viscous effects which are neglected here. Finally, there are complicated, unsteady phenomena associated with high-speed flow over a cavity. Given that the store is ejected from within a cavity bay, future studies would benefit from modeling these effects and calculating the complicated unsteady forces of the store passing through the viscous shear layer over an open cavity. Chapter 4 begins this investigation. Nonetheless, the results presented here are a foundation, providing important initial insights into the separation of upward-ejected stores in high-speed flows.

CHAPTER 4

DYNAMIC CAVITY SIMULATIONS

The previous chapter investigated store separation trajectories under the physical assumption of inviscid flow and using a quasi-static method for moving the store through time. These assumptions and methods are generally adequate when the simulation starts with the store sufficiently separated from the vehicle body and away from the near-body viscous effects of the vehicle. The preceding trajectory analysis considers an initial condition with the store already above the vehicle with a given ejection velocity. Such an assumption for the initial velocity of the store is convenient for an exploratory investigation. However, finding a more accurate initial condition for the store requires the consideration of ejection from a realistic cavity.

When simulating store ejection from a cavity, the inviscid and quasi-static methods are insufficient. By the nature of cavity flow, both viscous and unsteady effects are important due to the viscous shear layer that causes strong cavity resonance. Several studies [12, 14, 30, 31] have investigated the release of stores from unsteady cavity flows. These demonstrate the importance of a viscous, time-accurate solution for ejecting a store from a cavity. However, these studies consider fully developed flow over an open cavity. They do not consider the state of the flow at the moment the cavity door is opened and the store is ejected, which may have very different flow properties than an open cavity that is given enough time for full flow development. This study seeks to take these effects into account by simulating a dynamic opening door.

Turpin's work [19] documents experimental results from a dynamically opening cavity, and Sheta et al. [21] develop computational solutions for moving doors, but neither consider the presence of the weapons store, given their more fundamental objectives. Loupy et al. [22] present computational results for the aeroelastic response of a store in a dynamically opening cavity. However, their study is conducted at relatively low speeds (Mach 0.85), and the doors are undersized relative to the cavity allowing cavity venting, which is not applicable at higher speeds where the door-cavity interface must be sealed. As shown herein, such close interaction between the door, store, and vehicle bodies proves to be a challenge. To the author's knowledge, this is the first study that presents a computational approach to the dynamic opening of a sealed cavity that carries a weapons store. This work anticipates that the results generated will be used as the initial condition for continued store ejection and trajectory analysis.

4.1 Problem Definition and Setup

Geometry Definition

The primary goal of this study is to develop satisfactory methods for the dynamic opening of a sealed cavity carrying a store. Focused on the methods developed, a simplified geometric case is presented for analysis. It consists of a simple vehicle, a flat rectangular door that slides open in the downstream direction, and a conical store, similar to the one used in Chapter 3. The carrier vehicle for this simplified setup is modeled as a simple box with leading and trailing angled faces, displayed in Figure 4.1.

The simplifications of this geometric model offer several advantages. First, the lack of surface curvature allows for coarser surface meshes, reducing the final node count and overall time for a computational solution. The inverted ramp at the leading edge results in a shock structure that keeps strong shocks beneath the vehicle, and the solution near the cavity resembles that of a flat plate. The aft wedge reduces flow expansion around corners, avoiding the solution challenges that come with extremely low density and pressure values at sharp expansion corners. The sliding door design also provides an element of simplicity. These geometric choices were made to provide a model that is as easy to work with as possible while providing the characteristics necessary to develop the tools and techniques required for the dynamic cavity and store separation simulations.

The vehicle itself is 5.2 meters long, 2 meters wide, and 0.5 meters high. The cavity is centered spanwise, and its leading edge is 2 meters aft of the leading edge of the vehicle. The cavity and door are 1 meter long by 0.25 meters wide. The cavity is 0.25 meters deep, and the door is 0.01 meters thick. The store model is a 4° half-angle cone with a base diameter of 0.148 meters, a nose radius of 0.013 meters, and an overall length of 0.888 meters.

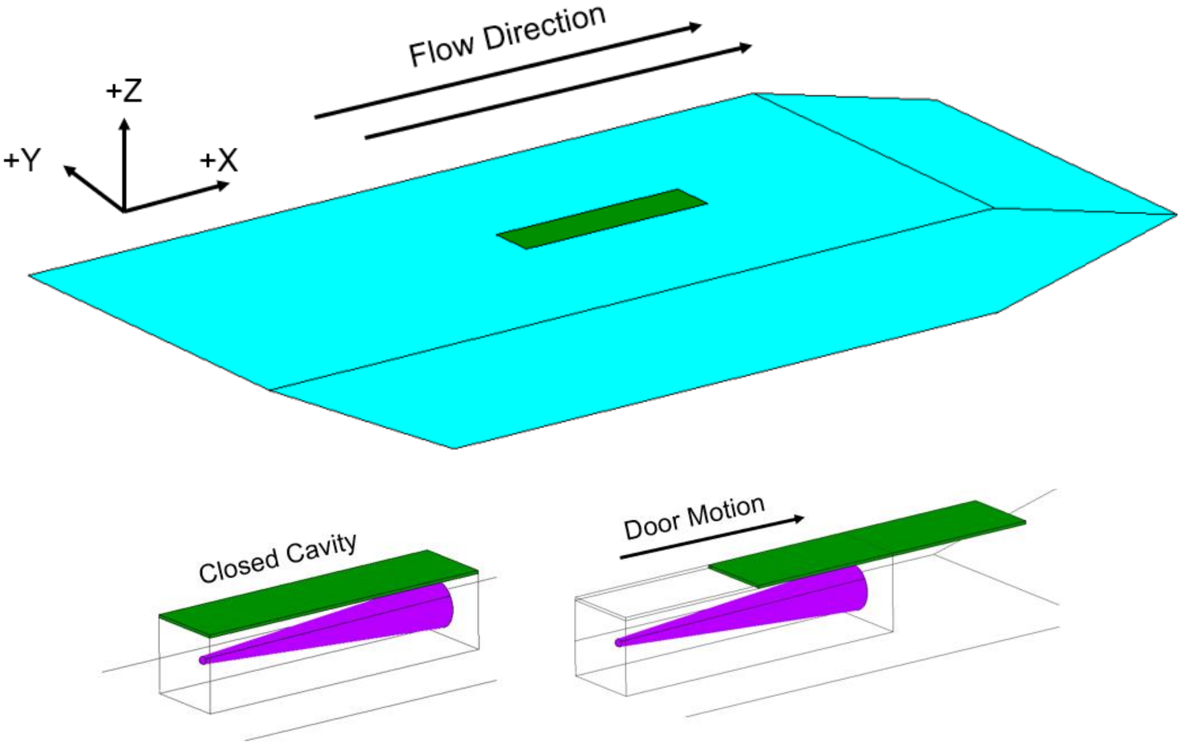


Figure 4.1 Vehicle and cavity (blue) with a downstream sliding door (green) and store (purple)

Freestream Conditions

For all computations, freestream conditions are specified for Mach 4 flight at an altitude of 80,000 feet (24,384 meters). At this altitude, freestream temperature, density, and dynamic viscosity are taken to be 221 K, 0.04610 kg/m^3, and 1.445x10^{-5} N·s/m^2, respectively. Accordingly, a Reynolds number of 3.8x10^6/m is used for all computations.

$$Re_x = \frac{\rho_\infty M_\infty \sqrt{\gamma R T_\infty}}{\mu_\infty}$$

Here, γ = 1.4, R = 287 J/kg·K.

Computational Methods

For all CFD computations herein, NASA's FUN3D flow solver is used for the solution of the conservation of mass, momentum (Navier-Stokes), and energy equations in three dimensions. The non-dimensionalization of the equations is demonstrated by the one-dimensional equations,

$$Mass\colon \quad \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} = 0$$

$$Momentum\colon \quad \frac{\partial(\rho u)}{\partial t} + \frac{\partial}{\partial x}\left[\rho u^2 + P - \frac{4}{3}\frac{M_\infty}{Re_x}\mu\frac{\partial u}{\partial x}\right] = 0$$

$$Energy\colon \quad \frac{\partial(\rho e_t)}{\partial t} + \frac{\partial}{\partial x}\left[(\rho e_t + P)u - \frac{4}{3}\frac{M_\infty}{Re_x}\mu u\frac{\partial u}{\partial x} - \frac{M_\infty}{Re_x \Pr(\gamma - 1)}\mu\frac{\partial T}{\partial x}\right] = 0$$

ρ, u, and e_t are local density, velocity, and total internal energy, respectively. P is local static pressure, M_∞ is freestream Mach number, and μ is dynamic viscosity. Re_x is the Reynolds number per unit length, and the Prandtl number, Pr = 0.72. For the equation of state, thermally and calorically perfect gas assumptions are implemented,

$$T = \frac{P}{\rho R}$$

T is local temperature, and R is the gas constant. The viscosity-temperature relationship abides by Sutherland's Law,

$$\mu = \frac{1 + C/T_\infty}{T + C/T_\infty} T^{\frac{3}{2}}$$

Here, T_∞ is freestream temperature, and Sutherland's constant, C = 110.33K.

Inviscid flux residuals are constructed using Roe flux difference splitting, and a stencil-based van Leer flux limiter is used. The van Leer flux limiter is augmented by a heuristic pressure limiter. A Spalart-Allmaras RANS turbulence model [32, 33] is used for the implementation of turbulent viscous terms. For steady computations, used primarily to generate initial conditions for dynamic simulations, a CFL number of 5 is used to calculate a local pseudo-time step. In time-accurate dynamic solutions, a second-order backward differencing method, BDF2 [34], is used for time integration. The choice of time step is specified in the following sections for each simulation.

Unless otherwise specified, all surface boundary conditions are viscous walls with an explicit no-slip condition. Farfield boundary conditions are set to the freestream condition, specified by fluxes.

4.2 Review of Overset Grid Assembly

The computational simulation of moving bodies provides an interesting challenge to domain discretization. For the quasi-static simulations in Chapter 3, the grid could be regenerated at each time step because the steady solutions at each time are independent of one another. In the time-accurate simulations here, solutions in time depend on one another, meaning a time history of the solution at every node in the mesh is required, and the grid cannot be regenerated between time steps. This poses questions as to how the grid will change over time, conforming to moving surfaces. If body motion is relatively small, as in aeroelastic deformations, computational grids can be elastically deformed without significant detriment to mesh quality. However, when body motion is too large for mesh deformation, other techniques are required. Overset grid assembly, also known as the Chimera method [35], is often employed for dynamic store separation cases. The overset methodology is also well-suited for the simulation of dynamic cavity doors.

In overset grid assembly, rather than building one continuous mesh that must conform to the surfaces of all bodies (vehicle, door, and store), individual grids are generated for each body. These are called “component grids”. The separate component grids are overlaid, or “overset”, onto one another to form what is called a “composite grid”. Component grids for moving bodies are allowed to translate and rotate relative to other grids, and the overset assembly routine passes updated information to the flow solver to account for the new grid positions. Overset assembly is a pre-processing step that takes place once before a steady-state computation or before every time step in a dynamic simulation. Naturally, the composite grid needs to be reassembled each time bodies move in the domain.

A basic understanding of the overset process and its associated terminology is useful for understanding this study. First, nodes in one component grid which are located inside the body of another component grid must be effectively turned off in the composite grid. One could imagine this process, which is called “hole cutting”, as each body cutting a hole in the other component grids. These nodes are tagged with the status, “out”. They are effectively, “out” of the solution. Ultimately, all nodes in component grids are tagged with one of the following statuses – “in”, “out”, “fringe”, or “orphan”. “In” nodes are treated normally, as they are “in” the solution, and the flow solver solves the governing equations at each of them.

Because the “in” nodes, in general, do not extend to the outer boundaries of the component grid, the flow solver needs some nodes, on the fringe of the “in” nodes to be assigned solution values which can be treated as boundary conditions for the “in” portion of the grid. These nodes, at the boundary of the “in” nodes, are tagged as “fringe”. The solution value at each fringe node is an interpolated value from nodes, called “donors”, in other components’ grids that are nearby. Effectively, one could imagine the overset composition as a set of quilt patches, the “in” portions of each component grid stitched together by fringe nodes where the solution is interpolated between adjacent grids.

When a fringe node cannot find suitable donors in its neighborhood from the other component grids, its status is updated to “orphan”. Orphan nodes are often problematic because the solution values assigned to them which subsequently propagate into the flow field have no guarantee of being accurate.

Sometimes, as demonstrated in this study, unreasonable values can result at orphan locations. It is the responsibility of the user to construct component grids such that orphan occurrences are minimized.

For the sliding door simulated here, several geometric conditions are atypical for overset methods. These include bodies in flush contact (door and vehicle) and intersecting orthogonal boundary layers. Also, two flow regions (inside and outside the cavity) must be initially separate without flow between them but later contiguous as the door opens. These conditions pose challenges to the overset process, some of which leave unavoidable orphan nodes in the solution.

Of final note for overset grid assembly is how information is passed from the overset program to the flow solver. The collection of information that the solver needs from the overset assembly, most notably the node statuses, is called “domain connectivity information” (DCI). The DCI is the output of the overset assembler and an input to the solver. In dynamic simulations, when the overset assembler is called from the solver between each time step, this is referred to as computing DCI “on the fly”. Alternatively, if the body motion is known *a priori* to solution computation, the DCI for each time step can be precomputed. Precomputed DCI may result in significant time savings if its independence from the solver can allow the domain assembler and solver to each take full advantage of computational parallelism.

Two overset assembly programs are used in this study. First, NASA’s Yoga domain assembly tools are utilized. They are distributed with FUN3D and implemented directly into the solver, which is demonstrated in Reference [36]. Facing many challenges with Yoga in this specific application, Suggar++, a commercial overset assembly code, is also utilized [37]. Its results are compared to those using Yoga. FUN3D has the option to use DCI from Suggar++ as well. Suggar++ has the added benefit of the ability to precompute the DCI while Yoga’s implementation requires that DCI is generated “on the fly”. The following section provides additional detail for the individual component grids used for the remainder of the study.

4.3 Domain Discretization

Flow simulations of the ramped box are primarily conducted using the overset mesh technique described in the previous section. This setup uses four component meshes to build each composite mesh – one for each of the vehicle, store, and door bodies, and an additional background mesh extending far beyond the vehicle. Throughout the results recorded in this study, these grids are incrementally modified to address the challenges faced and lessons learned. However, details are provided for a sample of each component mesh, describing the primary characteristics of the grids which generally remain unchanged. The significant modifications from these grids will be highlighted in subsequent sections as they arise.

All component grids are generated for a viscous Navier-Stokes solution. Therefore, small wall-normal spacing is used on all body surfaces to appropriately capture near-wall solution gradients. Each of the following component grids uses a first-cell wall spacing of $5x10^{-6}$ meters. Triangular surface elements are generated for all components, primarily using advancing front algorithms. In each grid, triangular prisms are grown from the surfaces until geometric isotropy is reached, and a Delaunay algorithm is used to fill the rest of the volume with tetrahedra.

Vehicle Grid

The component grid for the vehicle is built to maximize solution accuracy near the cavity and, otherwise, keep the overall node count minimal. The surface spacing over most of the vehicle is 0.05m. As the leading oblique shock is directed under the vehicle, there are no significant flow features upstream of the cavity that would warrant a finer surface grid. To improve the solution near the cavity, the cavity wall surfaces and the volume inside and just above the cavity use a refined spacing of 0.01m.

The presence of the door places an additional constraint on the vehicle mesh. It must be refined further within the volume that encompasses the door in the closed position. This refinement is necessary because without it, the solution may effectively “leak” through the closed door. Cells in this refined region must be smaller than the thickness of the door. This volume surrounding the door is refined to an average spacing of 0.002m.

It is best practice for different component meshes to have similar element sizes where they are expected to interpolate their solutions. The outer boundaries of the vehicle grid have an average spacing of 0.625m so that element sizes are close to those in the background mesh. A mid-span section view of the grid is shown in Figure 4.2, with a closer view of the refined cavity volume in Figure 4.3. Although grids for different cases may differ, this particular mesh contains 5,616,444 nodes, which is largely driven by the refinement around the door.

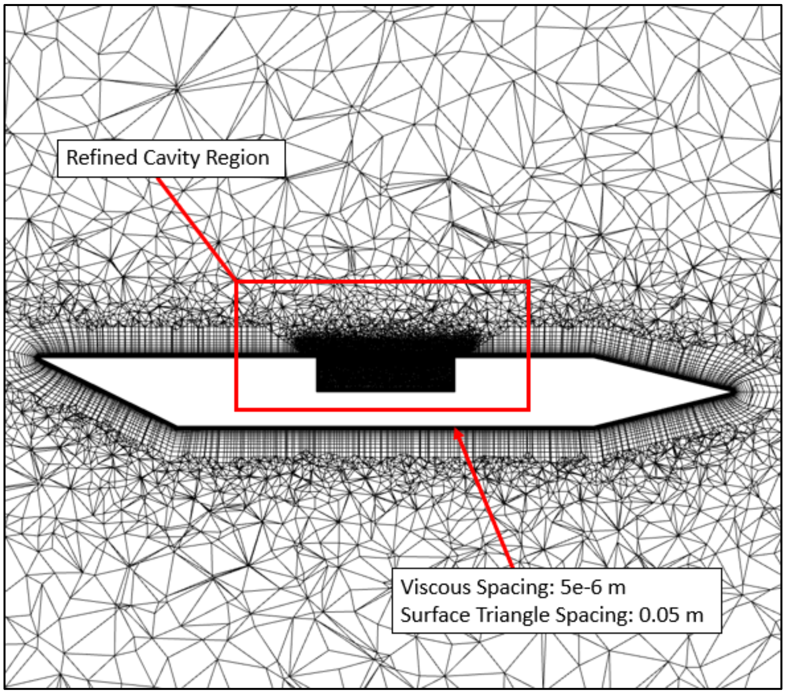


Figure 4.2 Mid-span section view of the vehicle component grid

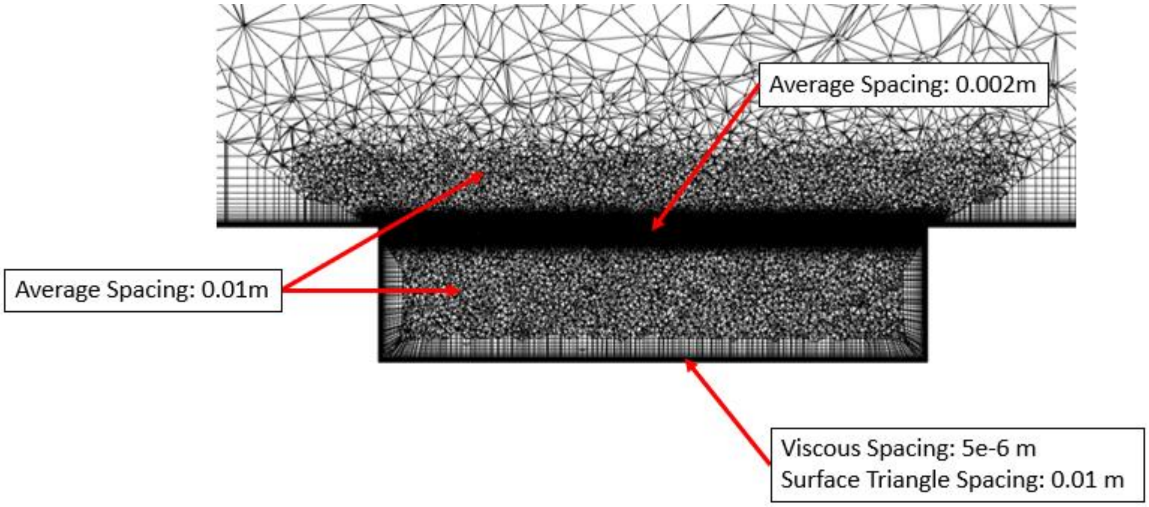


Figure 4.3 A closer view of the refined cavity volume in the vehicle component grid

Door Grid

Despite its geometric simplicity, the surface mesh for the door is required to be much more refined than the vehicle. Not only is the door located where high solution accuracy is needed, but it also requires additional refinement to help “seal” the door when it is in the closed position. The sides of the door are flush with the sides of the cavity, and the simulation relies on the no-slip boundary condition to keep flow from entering the cavity while the door is closed. To best ensure this sealed behavior, several cells must span the thickness of the door on its side surfaces. With the door being 0.01 meters thick, a point spacing of 0.002 meters was chosen on the edges to ensure sealing.

The requirement for grids to have similar spacings at fringe cells for interpolation drives the spacing decisions for the rest of the door grid. As mentioned previously, the vehicle grid was required to have 0.002 meter spacing near the door so that flow cannot “leak” through the surface of the door by interpolation. Therefore, the volume in the door grid around the door surfaces must also be refined to 0.002 meters so that the domain assembly process interacts well with the vehicle grid. The outer region of the door volume grid has an average spacing of 0.01 meters to match the cavity refinement spacing in the vehicle grid. Finally, the portion of the grid upstream of the door is also refined to 0.002 meters so that as the door opens, this portion of the grid interacts properly with the vehicle grid. A mid-span section of this grid is shown in Figure 4.4. Its total node count is 8,926,379.

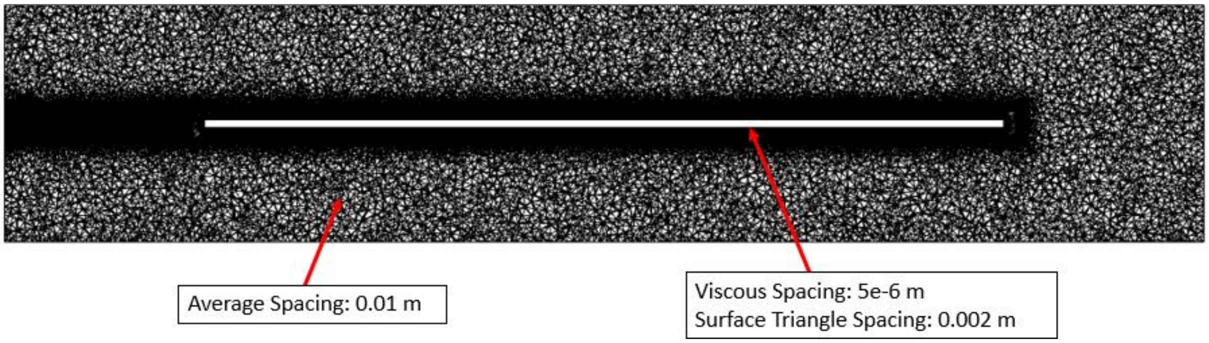


Figure 4.4 Mid-span section view of the door component grid

Store Grid

Accurate force calculations on the store are of primary importance for this study. It is critical to sufficiently refine the store grid to maximize solution accuracy. The surface spacing on the store is 0.005 meters. The nose surface of the store is further refined to 0.0025 meters, primarily for geometric resolution. To match volume spacings with the vehicle grid, including the refined cavity volume, the immediate neighborhood of the store has an average point spacing of 0.01 meters, and the remainder of the volume grid is spaced at 0.02 meters. A mid-span section of the store grid is shown in Figure 4.5. This store component grid contains 1,098,013 nodes. This particular grid is used for all simulations using the Suggar++ domain assembler. Simulations using Yoga will require an additional refinement zone in the area that overlaps the closed door, increasing the total number of nodes to 3,197,365. This modification is described in Section 4.5.

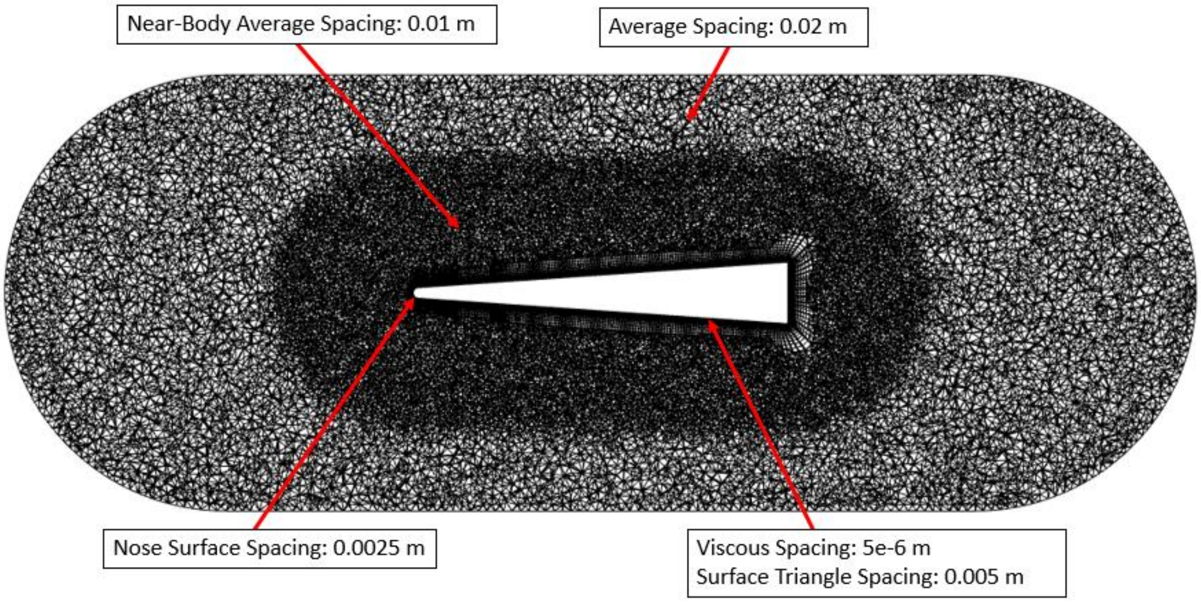

Near-Body Average Spacing: 0.01 m
Average Spacing: 0.02 m
Nose Surface Spacing: 0.0025 m
Viscous Spacing: 5e-6 m
Surface Triangle Spacing: 0.005 m

Figure 4.5 Mid-span cut view of the store component grid

Background Grid

In addition to the three grids for the vehicle, store, and door bodies, an additional background mesh is included for the farfield. One could hypothetically use one component grid for both the vehicle body and the farfield, as the vehicle is static. However, in this case, the vehicle component grid and background grid are left separate as a standard practice. The background grid is a 150-meter cube. The point spacing at the farfield boundary is 15 meters. There is an inner volume zone which is refined to a 1.5-meter spacing so that overset assembly works well with the vehicle grid. A section of the background grid is shown in Figure 4.6. It has a total of 27,896 nodes.

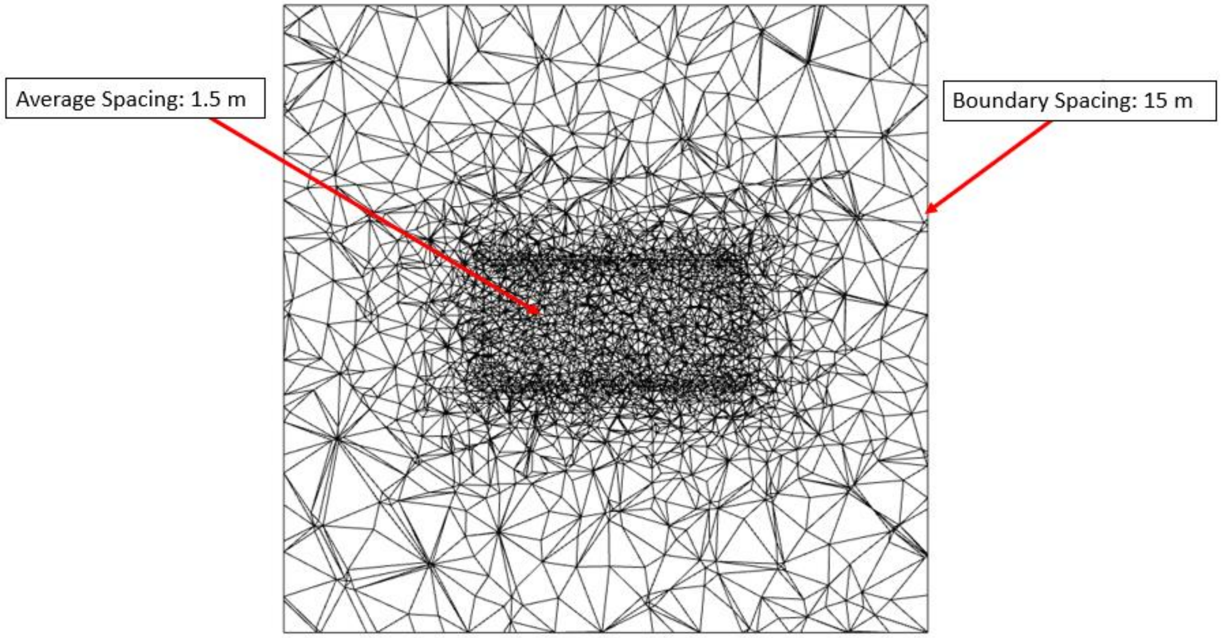


Figure 4.6 Mid-span cut view of the background grid

4.4 Lessons Learned with an Empty Cavity

Steady-State Quiescent Cavity Solution

Before any dynamic moving body simulations can be run, a steady-state solution must be obtained with the door closed. Part of the validation for the steady solution is ensuring that the flow remains quiescent inside the closed cavity. Using multiple bodies from separate overset component grids to create an air-tight sealed cavity is not a standard use case for overset grid assembly, and the author is unaware of other applications that have done so. After much trial-and-error to develop a setup that maintains a quiescent cavity, two practical techniques were identified which allow for an acceptable steady solution. For simplicity, the store is omitted from the composite grid, simulating an empty cavity.

The first technique to maintain a quiescent cavity requires a modification of the initial conditions. By default, all solution variables are initialized to freestream conditions throughout the entire domain. This includes inside the closed cavity. Given the Mach 4 freestream condition, this means that the flow inside the closed cavity is initialized at high speeds with nowhere to go. A quiescent cavity could only be achieved here if the solver were left to run long enough for viscous and numerical dissipation to sufficiently slow the flow, which would take an impractical amount of time. Instead, a rectangular volume is specified which encapsulates the entire cavity, and its initial velocity condition is set to zero. To illustrate this specification, Mach contours are shown in Figure 4.7 for the initial solution before computation.

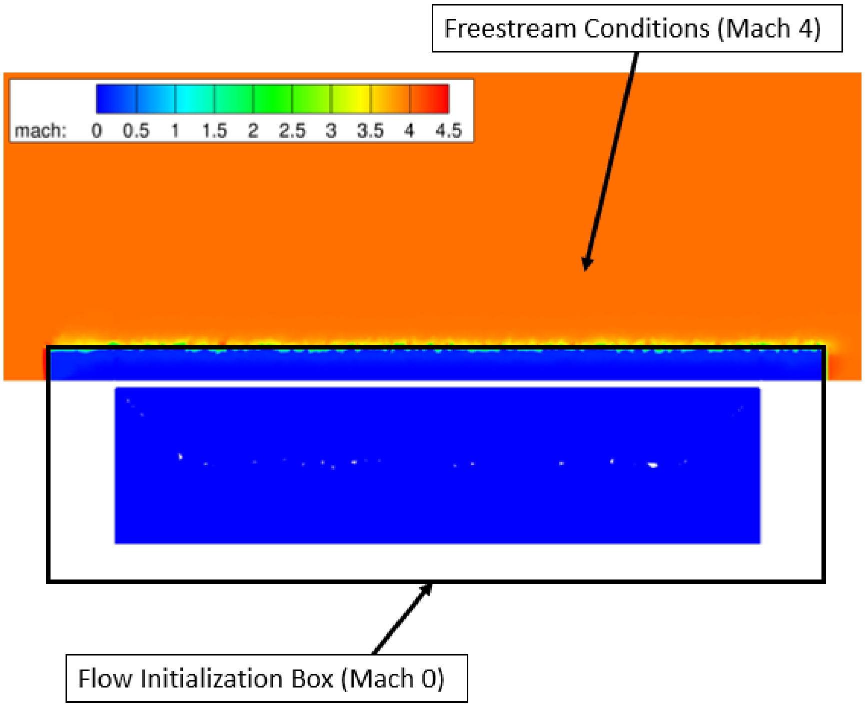


Figure 4.7 Mach number contours for solution initialization with zero velocity around the cavity

First attempts for a steady solution did not result in quiescence inside the cavity, even after implementing the zero-velocity initialization box. It was originally thought that flow was “leaking” around the edges of the door into the cavity, and numerous efforts were made to alleviate the issue. Some of these attempted strategies included grid refinement around the door’s edge, extending the door outward so it physically overlaps with the vehicle instead of abutting flush, and using a revised vehicle grid constructed such that no cells cross the surfaces of the door in the closed position. An alternate “frozen” boundary condition was also applied to the edges of the door and the cavity surfaces in contact with the door, attempting to solve the apparent leaks in the steady solution. As its name implies, the “frozen” boundary condition prevents boundary solutions from being updated, which are initialized at zero velocity around the door. Unfortunately, none of these strategies solved the mystery of the leaking cavity door.

Each of these solution strategies attempted to address the issue as if the flow was physically leaking through the door, by modifying the component grids, geometry, and velocity boundary conditions. One effect that had not been considered yet was the potential for natural convection due to the temperature of the wall. Sure enough, the flow was not “leaking” into the cavity, but rather, improper temperature boundary conditions were causing convective flow in the cavity.

Natural convection is caused by a temperature difference between a surface and the surrounding fluid. So, to eliminate convection inside the cavity using isothermal wall conditions, the cavity wall temperature should be set to the same temperature at which the cavity volume is initialized. In this case, freestream temperature ($T = T_\infty$, $T^* = 1$) was used as the isothermal wall boundary condition and volume initial condition. It is assumed that the vehicle has not been flying long enough for the walls to achieve an adiabatic condition at this point in the flight. By default, FUN3D sets an isothermal wall temperature parameter to 1.0, as desired, which was understood while other methods to seal the cavity were tested. However, it was unknown that FUN3D does not use this parameter unless specified by another input flag. By default, an isothermal boundary condition is applied using the recovery temperature,

$$T^* = \frac{T_{wall}}{T_{ref}} = 1 + \frac{\sqrt{Pr}(\gamma - 1)M^2}{2}$$

Unknowingly, the cavity walls were being held at an absolute temperature 3.7 times greater than the air in the cavity, causing convective currents.

Once the wall temperature boundary conditions were corrected by setting all walls to T^* = 1.0, then cavity quiescence was achieved for the closed-door steady solution. In the end, the only requirements were to set the volume initial conditions to zero velocity within the cavity, employ viscous wall boundary conditions, and model the door edges to be flush with the cavity edges. The steady solution is shown in Figure 4.8. The largest Mach number values inside the cavity are on the order of 10^{-14}. This solution was computed using Yoga for domain assembly. The Suggar++ implementation has additional challenges which are addressed in Section 4.6.

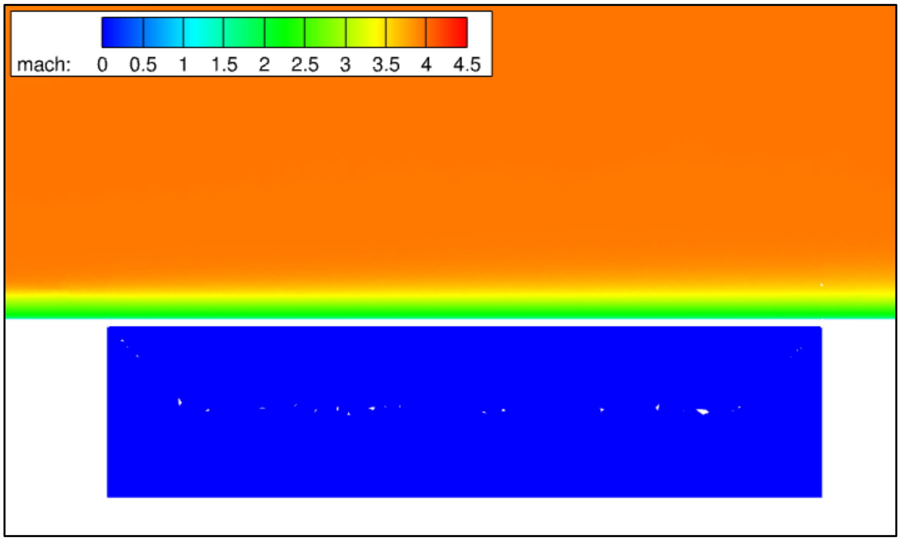


Figure 4.8 Mid-span Mach number contours for the steady-state solution

The sufficiency of flush door edges and viscous walls to properly seal the cavity deserves further explanation. Viscous walls enforce zero velocity on the wall surface. This helps keep flow out of the cavity as air cannot flow along the door edge surfaces or the cavity walls. However, where flush faces meet to seal the cavity, one must ensure that even one cell away from the wall, which should be inside an adjacent body, nodes are assigned an “off” status during overset assembly. Otherwise, flow can effectively leak into the cavity. One could summarize these requirements by stating that in any given component’s grid, all nodes that are adjacent to, but not inside, the closed cavity volume must either be on a viscous wall or turned “off” in overset assembly.

Figure 4.9 shows that this is indeed the case if the door edge and cavity wall surfaces are approximately flush (relative to the height of the first cell off the wall). The overset assembly process recognizes that the nodes just off the wall from one body are inside another body, and they are set to the “out” status. There are no fringe nodes between the “in” nodes on the wall and the “out” nodes just off the wall. Because the only relevant “in” nodes are on the viscous wall and there are no “fringe” nodes, flow cannot enter the cavity. It is properly sealed. In Figure 4.9, yellow nodes are "in", green nodes are "out", and blue nodes are "fringe".

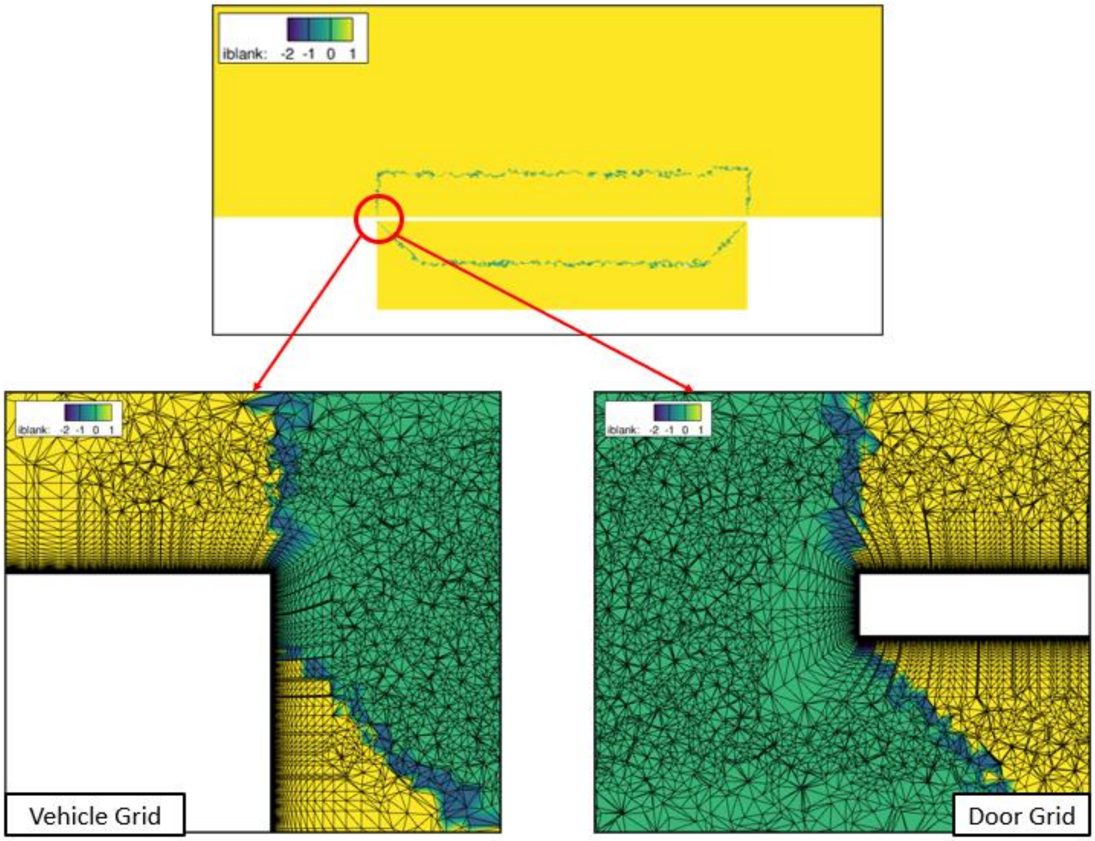


Figure 4.9 Overset node statuses where vehicle and door component grids are in contact

The only other opportunity for flow to enter the cavity is by way of orphan nodes. These are the nodes assigned a “fringe” status but have no suitable donors. These orphan nodes are unavoidable just inside the top edges of the cavity where the viscous spacings of the door and vehicle grids meet perpendicularly. See Figure 4.10. As discussed in more detail later, the Yoga and Suggar++ implementations appear to handle these orphans differently. The Yoga implementation properly seals the cavity while the Suggar++ case allows some minor leaks.

It should be noted that one in-house modification to the Yoga/FUN3D source code was required to successfully obtain solutions using Yoga for overset assembly. When interpolation weights are calculated, a least squares algorithm is employed. The weights that are returned for a given node must abide by two requirements. They should all be positive, and they should sum to one. Given this unusual application for overset assembly with bodies in sliding contact and potentially coincident nodes, the least squares algorithm does not always abide by these requirements. Occasionally, the weights do not sum to one, or if they do, some of them are negative, causing the solver to fail. The in-house code modification checks for these conditions, and if an invalid set of weights is found, the weights are re-calculated using an inverse distance method. This fixes the problem, allowing for a computational solution.

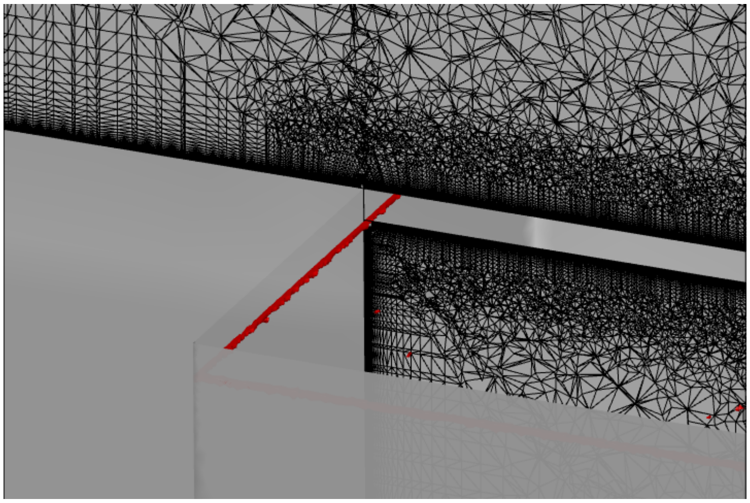


Figure 4.10 Orphan nodes (red) occur where viscous vehicle and door grids intersect

Dynamic Door Opening Time Study

Now that a steady-state solution has been obtained with a quiescent cavity, the next logical step is to simulate opening the door. Throughout the dynamic portion of the study, it is important to choose the time step carefully. This time step is not only used in solving the governing equations, but it also governs how far the door travels between each solution step. If the time step is too large, critical transient flow features may be missed. If the time step is unnecessarily small, then the time to solution may be much larger than acceptable. One must remember that at each time, the door moves, and the overset assembly process must be recalculated. Often, overset assembly can take significantly more time at each time step than the solution of the governing equations.

A study was conducted to decide how small of a time step is required for a satisfactory solution. To simplify the study, the store was removed from the cavity. The door slides open at a rate of 1m/s. The total duration of the simulation is 1 second. Three time steps were chosen for this study – 1x10^{-3}s, 2x10^{-4}s, and 2x10^{-5}s. These are herein referred to as the "1K", "5K", and "50K" simulations, respectively, as their corresponding number of time steps to open the door are 1,000, 5,000, and 50,000. Given that the flow-through time (the time it takes for the flow to travel the length of the cavity at freestream velocity) in this case is about 8.4x10^{-4}s, one can see that the 1K case is quite coarse, and only gross flow features are expected to be captured. The 5K time step seems more reasonable, at least capturing multiple time steps per flow-through period. The 50K simulation has a much higher time resolution, but it takes a long time to run, making it difficult to iterate and apply additional learnings. Also, both the 5K and 50K simulations use 15 Newton steps per time step for improved solution accuracy, whereas the 1K simulation only uses 5 Newton steps per time step.

The total run times for the three simulations are recorded in Table 4.1 below. It is also shown that for higher resolution studies, more time is spent on average per step. This effect is due to the increased number of Newton iterations for the 5K and 50K simulations. Notably, increasing the number of Newton iterations by a factor of three does not linearly increase the time per solution step. This is because a significant portion of the solution time is spent computing DCI using Yoga. Also, because of time limits for jobs on HPC resources, the larger runs were broken down into shorter simulations. From the data available, the total job time is not clear for the restarted simulations, but reasonable estimates can be made and are recorded in Table 4.1.

Table 4.1 Dynamic Simulation Time Step Data

Simulation	**Number of Steps**	**Time Step, Δt (s)**	**Total Time (d:h:m:s)**	**Time per Step (s)**
1K	1,000	1×10^{-3}	0d : 19h : 25m : 16s	69.9
5K	5,000	2×10^{-4}	6d : 3h*	105.8*
50K	50,000	2×10^{-5}	47d : 2h*	81.4*

*Estimates based on available data from multiple restarts.

Figure 4.11 shows the steady-state solution for this study. Each of the three time step variations begin with this solution at the initial time, t = 0. One can note that the region inside the sealed cavity is quiescent, and the flat plate boundary layer on the outside of the cavity door has fully developed. All other major flow features (i.e., shocks and expansions) are below the vehicle or aft of the cavity. Shocks and expansions are not well-resolved, but the grid is left coarse in these regions understanding that they have little effect on the flow inside the cavity, saving computational expense. Figure 4.12 shows that all residuals have converged, including the Spalart-Allmaras turbulence model, indicating that a steady state has been reached.

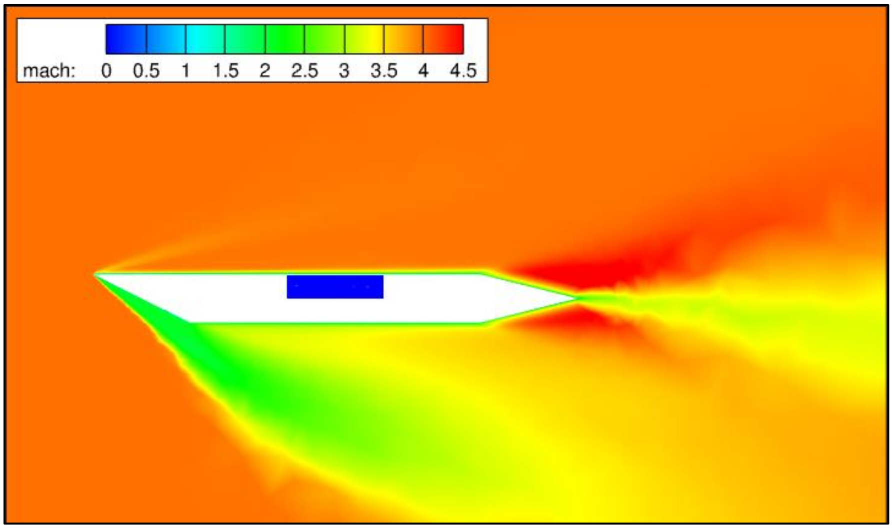


Figure 4.11 Mach number contours for the steady-state solution with a closed cavity door

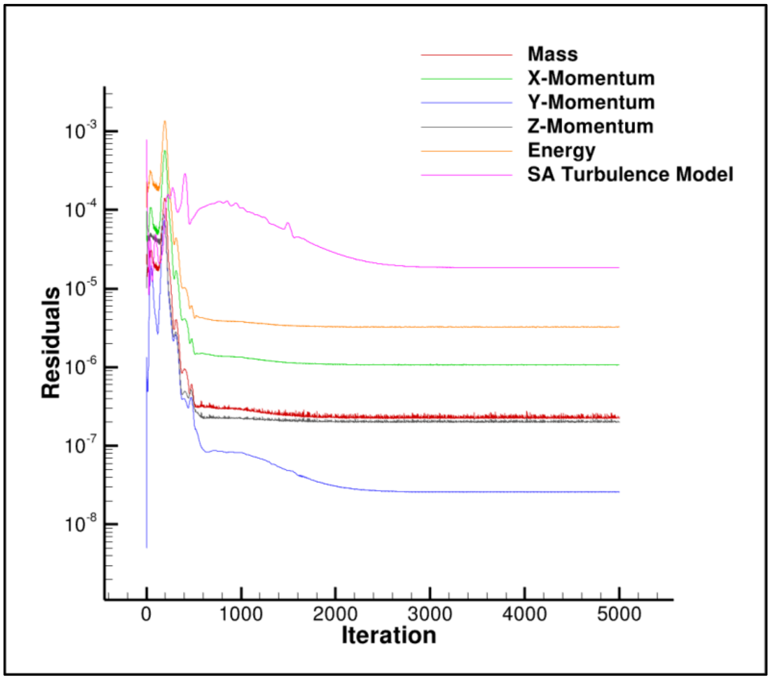


Figure 4.12 Plot of residuals for the steady-state solution with a closed cavity door

To compare the solutions, pressure data is recorded over time at one point in the center of the rear wall of the cavity, corresponding with the (x, y, z) coordinates (1.0, 0.0, -0.125) meters. Regardless of the solution time step, data is recorded every 0.01 seconds, resulting in 100 total samples for the simulation. Evidence of resonant behavior in the cavity is of primary interest. We seek to find a time step that is sufficiently small to capture fluctuations in the flow that are strong enough to have a significant impact on total forces. This data is plotted for the 1K, 5K, and 50K simulations in Figure 4.13.

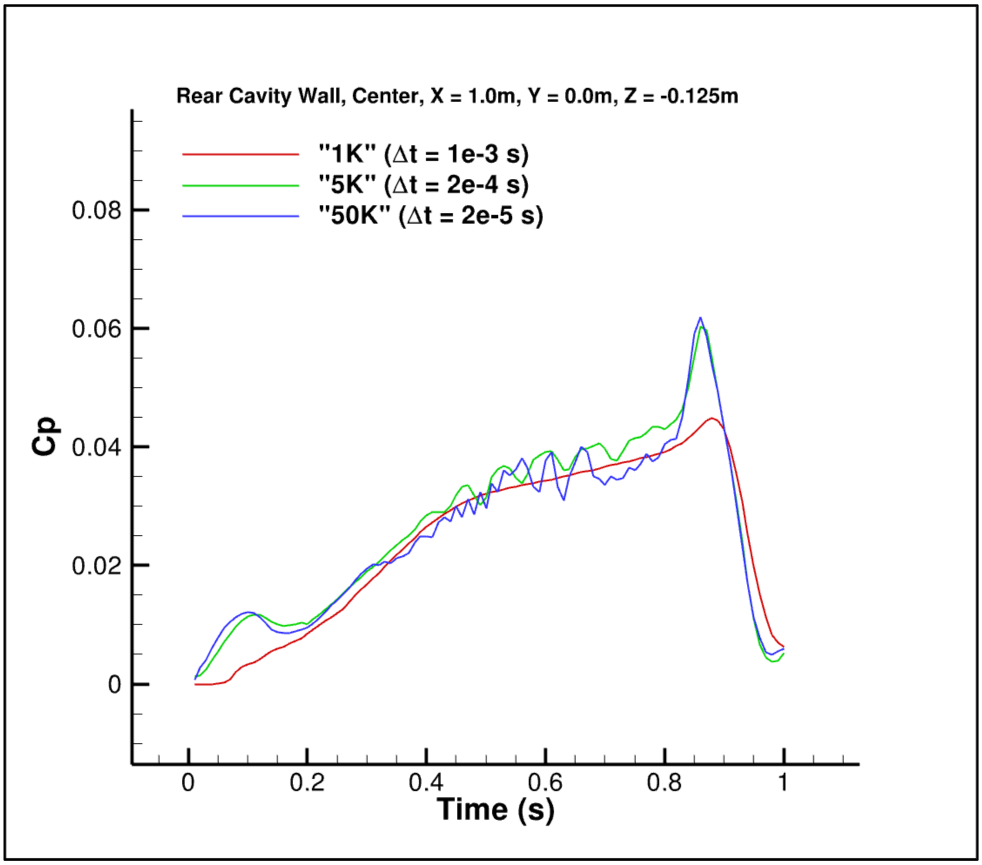


Figure 4.13 Coefficient of pressure on the rear cavity wall for the "1K", "5K", and "50K" resolutions

It is shown that the 1K simulation (largest time step, Δt = 1x10^{-3}s) does not capture the solution with sufficient accuracy. The virtual pressure probe at the back of the cavity shows a smooth pressure variation without the expected oscillatory behavior due to cavity resonance. The 1K solution does not capture the initial pressure rise when the door first opens (t = 0s – 0.2s) nor does it capture the sharp rise in pressure around t = 0.86 s.

The pressure at the highest time resolution (50K) fluctuates at a frequency that is 2 to 3 times greater than what is seen in the 5K data. Despite the difference in frequency, the mean pressure values are consistent between the two simulations. This indicates that higher resolution is required to capture more accurate frequency data, but it is not necessary to compute a reasonable solution for large-scale forces. The significant additional computational expense may or may not be deemed necessary depending on the goal of the study. If frequency-driven phenomena are the engineering focus, as in fatigue analysis, then additional computation time is justified as it is required to get sufficiently resolved frequencies. In the case of this study, larger-scale integrated forces with the potential to impact store ejection and trajectory are most important. From here, it is assumed that finer frequency analysis is not required to study the main effects, and simulations will be limited to use time steps on the order of the 5K study ($\Delta t = 2\text{x}10^{-4}$s). It is expected that this will yield meaningful results while avoiding a degree of computational expense that inhibits iterative learning.

4.5 Challenges to Overset Methodology with a Store in Carriage

Having generated solutions for both a steady, closed cavity as well as a dynamic opening door, the study continues with the addition of a weapons store inside the cavity. Ideally, this would be as simple as adding a store component grid to the same overset composite grid as the empty cavity simulation. The overset assembler would then consider the store surfaces in the domain assembly and include the store grid in the domain connectivity information passed to FUN3D. In these cases, the Yoga overset assembler is used. Unfortunately, the addition of the store grid into the composite mesh system presented several challenges. Those challenges are described here, as well as the consequences which were imposed on the study.

The Yoga domain assembly code has several advantageous features, particularly in its parallel scalability for large problems. However, one significant limitation was found in how Yoga's implementation handles composite grid systems where meshes of three or more bodies overlap – in this case, the vehicle, the door, and the store. It appears that where the solution must interpolate between two grids at a fringe location, the interpolated solution includes information from the third body's mesh, including "out" nodes which should not impact the solution at all. Yoga's integration into FUN3D does not update "out" nodes at all, leaving them at initial conditions. Thus, where the solution seeks to interpolate between two grids, it may use initial condition values as part of that interpolation. In this specific case, depending on location, those initial condition values may be freestream (outside the cavity) or zero velocity (inside the cavity). Either way, this is problematic for even a reasonable solution, as shown in the steady results when the store grid is added to the composite system.

Steady-State Solutions Using Yoga

To obtain a steady-state solution with the store in carriage, an additional modification to the store component grid was required. The modified store mesh is depicted in Figure 4.14, with a closer view in Figure 4.15. There is a refined region in the store's mesh where the door sits. This helps stop leaks into the cavity during the steady solution. Without the refinement zone, store grid cells near the door would be about the same size as the thickness of the door. This change increases the total number of nodes in the component grid from 1,098,013 to 3,197,365. The cause of the leaks and the necessity of this refinement merits further study, as its elimination would significantly help reduce the node count and computational expense.

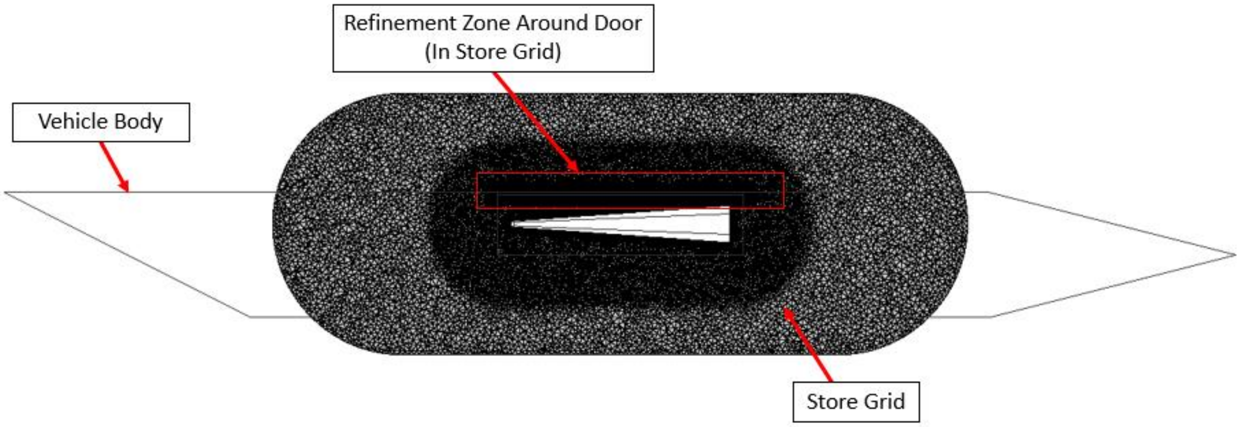


Figure 4.14 Store component grid which has been added to the composite assembly

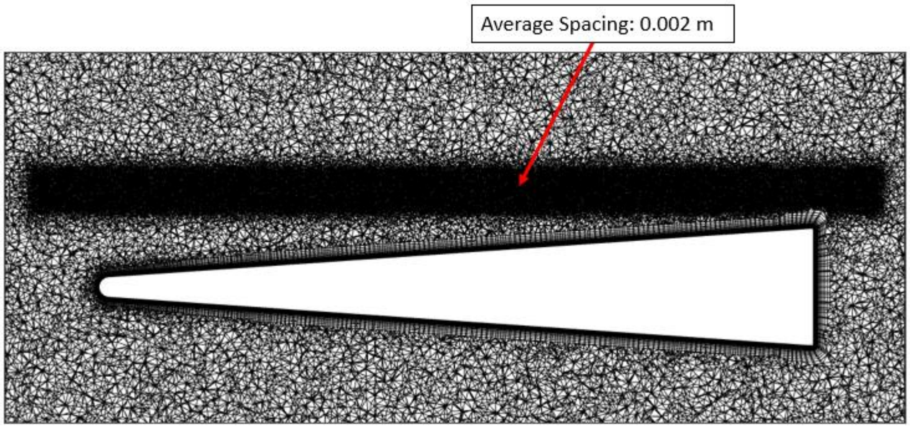


Figure 4.15 A closer view of the refinement zone in the store component grid

For this steady solution, the problems coming from the close interaction of many component grids are manifested just above the edges of the door. In the two-dimensional section shown in Figure 4.16, attention is brought just above the leading edge of the door. Here, the grids for all three bodies overlap. Ideally, this should be a “fringe” location for both the vehicle and door grids, prompting the solver to interpolate the solution between the two grids. The store grid, here, should be disregarded entirely as its nodes are considered “out”. Yoga correctly flags “fringe” nodes in the vehicle and door meshes, and it does call store nodes “out”. However, when the solution interpolates between the vehicle and door solutions, it also includes the Mach 4 initial conditions from the store grid, despite the “out” status of the nodes.

Figure 4.16 shows the steady solution that is computed using this set of meshes. With the door closed, the solution at the leading edge should resemble a typical flat plate boundary layer solution. Instead, the solution abruptly changes where the door and vehicle grids interact, and the Mach 4 initial conditions effectively “leak” into the flow. This is clearly incorrect, and the problem requires a solution before moving forward with dynamic simulations opening the door.

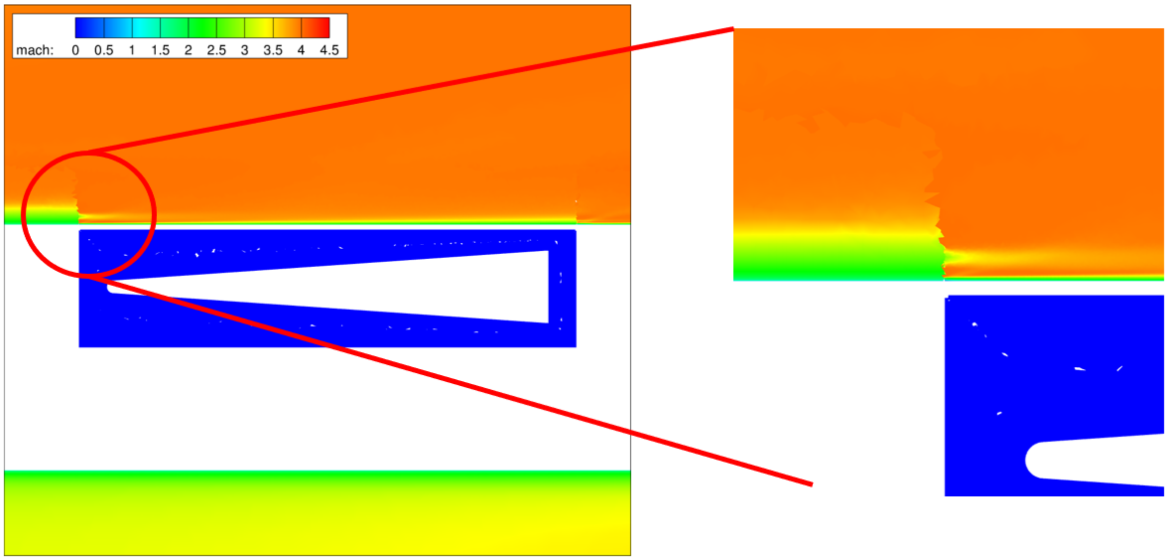


Figure 4.16 Mach number contours of a steady-state solution with a closed cavity door

To both verify the cause of the issue and provide a remedy, a modified store grid was generated. Because the cause of the issue is believed to be the overlap of the store grid where the other grids are producing interpolated solutions, the store grid was reduced such that its outer boundaries are within the confines of the cavity. This eliminates any “out” nodes from the store’s grid which may be impacting the solution interpolation above the door. This modified store grid is shown in Figure 4.17. Admittedly, this decision to limit the grid comes at the cost of solution accuracy later in the simulation. It forces interpolated solution values near the store body throughout the entire simulation, which is known to be poor practice for overset methods. Nonetheless, this limitation on the store was deemed necessary to compute an acceptable steady solution. For completeness, it is also noted that the door grid was also modified as a preventative measure. Its bottom surface was extended downward so that it is below the cavity, rather than passing through the store body.

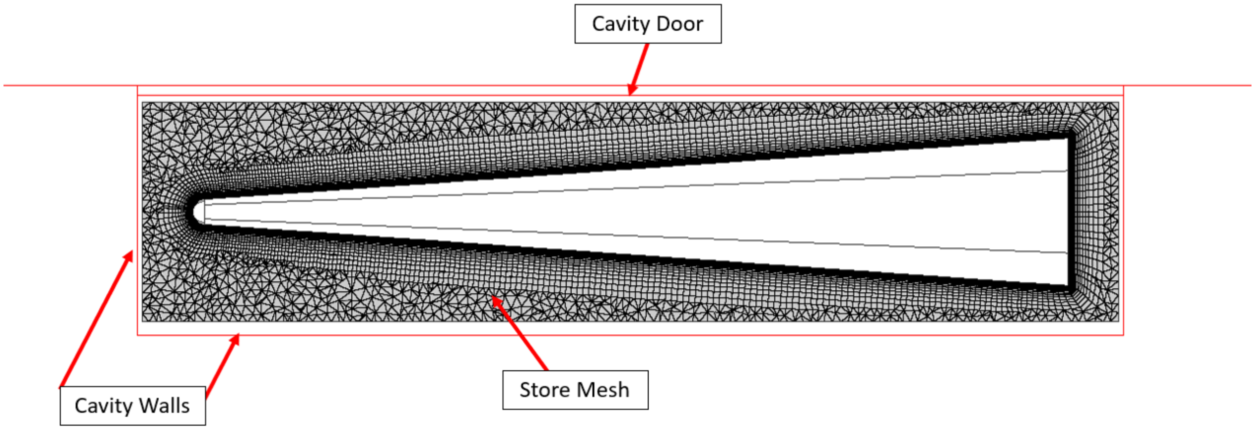


Figure 4.17 Modified store grid such that grid boundaries are confined to the cavity space

The steady solution using the modified store mesh corrects the issue. This solution is shown side-by-side with the previous results, before and after the store grid modification, to show the impact of restricting the store grid to the confines of the cavity. See Figure 4.18. A smooth boundary layer solution is displayed as expected.

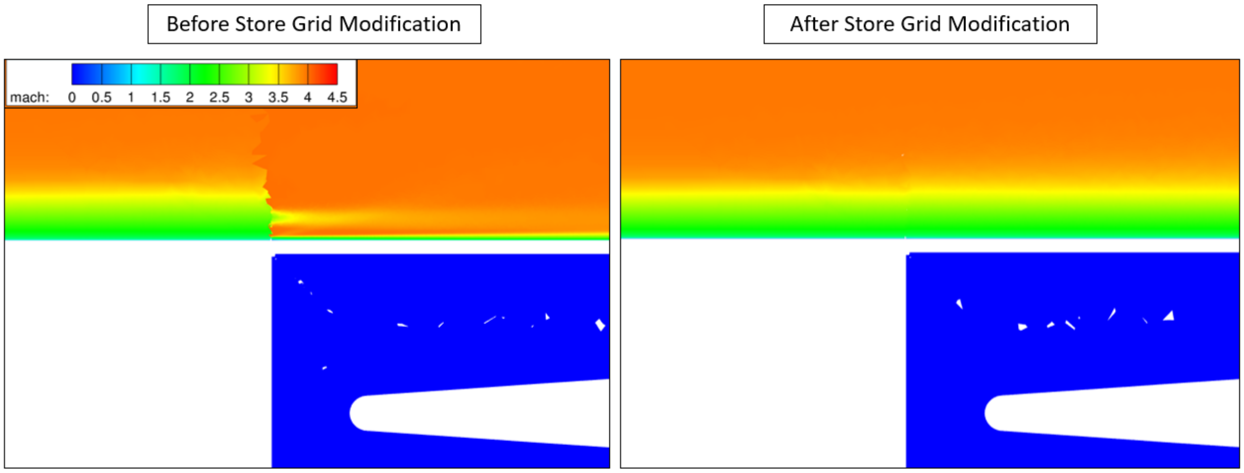


Figure 4.18 Steady-state solution before and after modifying the store component grid

Additional Overset Challenges with a Dynamically Opening Door

Unfortunately, it becomes apparent during the dynamic door-opening simulation that this challenge has not been addressed entirely. It should be noted that while these dynamic computations are still a “proof-of-concept”, a coarse time step is used, Δt = 0.001s, and the door opens in only 1000 time steps. As such, the overall accuracy of the solution is not yet of particular interest. The goal is to produce a reasonable solution, validating the grid composition and moving body mechanics before spending the computational time required for more accurate solutions. The problems addressed in the steady solution were caused by the interaction of more than two component grids at one location. The presence of a third grid behind the interpolated solution of the other two grids caused unexpected results. A similar effect is apparent as the door opens.

For the dynamic door opening simulation, the interaction between grids for all three bodies is more complex. It remains, though, that their interaction is problematic, resulting in high-velocity “leaks” into the solution. First, it can be noted that the domain assembly process does not handle the door grid properly. One would expect that the door grid nodes are considered “out” anywhere beneath the store. This is not the case. Figure 4.19 shows the composition of the grid where each color represents the “in” space of each mesh, taken at time t = 0.56s. Note that there is a region beneath the store where the door grid is considered "in", unexpectedly. Also, there is a region where the vehicle "in" space encroaches very close to the store, taking place where the door grid's upstream boundary passes through the cavity. This is also unexpected. Evidently, the conditions of this study challenge the overset assembly process for more than two bodies in proximity. In Figure 4.19, the "in" region of the store grid is yellow, the door grid is blue, and the vehicle grid is green.

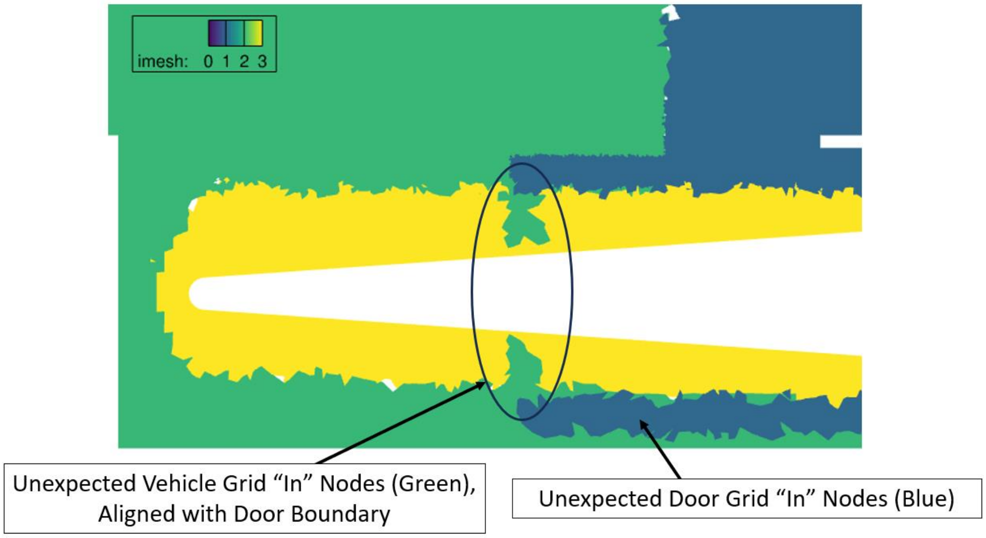

imesh: 0 1 2 3
Unexpected Vehicle Grid "In" Nodes (Green), Aligned with Door Boundary
Unexpected Door Grid "In" Nodes (Blue)

Figure 4.19 Composite grid composition at time, t = 0.56s, colored by component grid ID

When the solution is initialized prior to the steady computation, the door grid nodes located upstream of the cavity (outside the cavity, but inside the vehicle body) are set to Mach 4 freestream conditions. As these nodes are considered "out" in overset assembly, they are not updated during the steady solution. As the door slides open, these "out" nodes, with Mach 4 solution values attached to them, enter the cavity. When these nodes interact with the solution interpolation in the areas mentioned above (and highlighted in Figure 4.19), spurious high-velocity flow "leaks" into the cavity, invalidating the remainder of the solution. These high-speed leaks are shown in Figure 4.20, introducing supersonic spikes in velocity where there should not be. Due to these challenges, additional grid modifications were required.

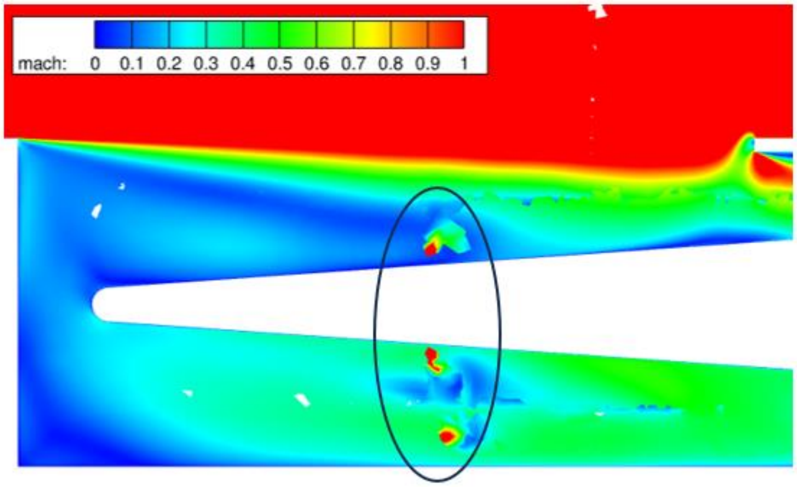


Figure 4.20 False velocity spikes in the dynamic simulation

Like the restrictions placed on the store mesh, the door grid was also modified. Because a reasonable solution could not be obtained with three interacting grids (excluding the background grid), the door and store grids were restricted so that they never interact. The outer boundaries of the door grid are constricted significantly so that the bottom boundary does not overlap with the store grid. Both the reconstructed door and store grids are shown in Figure 4.21. Again, it is admitted that this has significant consequences for solution accuracy. Forcing interpolated solution values so close to the door surface is not recommended. For the sake of obtaining even an approximate solution, though, the modifications are accepted to further explore the dynamic simulation. With these changes, Mach number contour plots are shown in Figure 4.22 for several time steps during the dynamic simulation of the opening door. Note that the contour scale is restricted to Mach 0 through Mach 1, highlighting subsonic flow features. The freestream Mach number is 4. As the door opens, there are no longer any unexpected velocity spikes as before. The mesh modifications fix the problem, but they do so at the expense of solution accuracy which will prove costly as the store is ejected from the cavity in the following section.

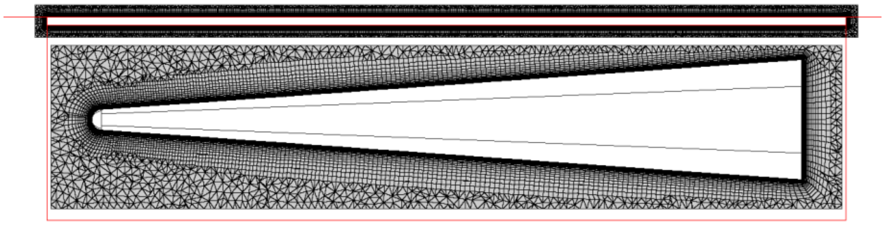


Figure 4.21 Modified door and store components grids

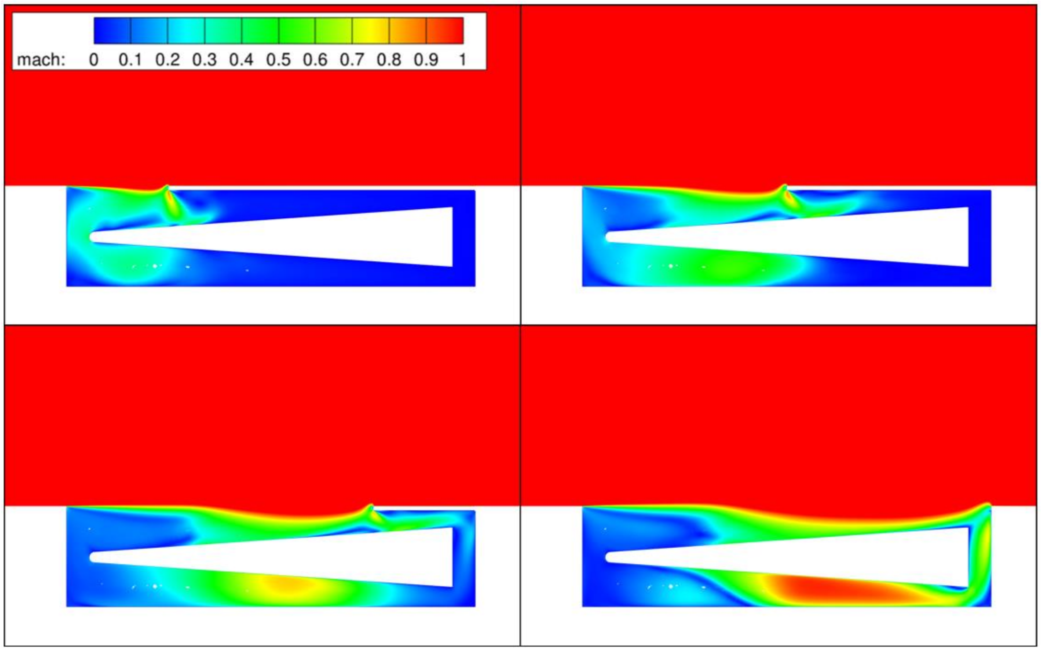


Figure 4.22 Mach contours in a dynamic simulation with the door 25, 50, 75, and 100% open

Dynamic Solution of an Ejecting Store

Accepting the consequences of placing restrictions on the boundaries of the component grids, the study continues by investigating the initial ejection of the store, after the door has been fully opened. The primary goal of this portion of the study is to verify the ability to simulate store ejection from a cavity while the store is still under prescribed motion. Prescribed motion is appropriate for the store only while it is still being pushed upward by the ejector mechanism. Once the store is free from the ejector, its motion is subject to the aerodynamic forces imposed on it, and it must be simulated utilizing a 6DOF driver to predict the motion. The ejector stroke length, and thus, the appropriate portion of the store motion to be prescribed, is generally only a few inches. However, for more robust validation of the methods used and the interaction between dynamic grid components, the store is prescribed to translate 0.5 meters upward, well outside the cavity.

The first lesson learned in attempting this dynamic store simulation required that the vehicle component grid be modified. The modified vehicle grid is shown in Figure 4.23, which has been refined in the volume above the cavity into which the store will travel. This region has been refined to an average spacing of 0.01 meters. Without this refinement, many orphan nodes are tagged once the refined store grid moves up into the coarse region of the vehicle mesh, and the solution fails. This reinforces the good practice when using overset grids to ensure that solution interpolation between grids takes place where the cell sizes of those grids are roughly equivalent. In this case, there is a significant difference between the 0.01-meter spaced nodes of the store grid and the 0.05-meter spaced nodes above the refined cavity in the vehicle grid. To avoid this interaction between dissimilar cell sizes, the vehicle grid is refined in the path where the store will pass. This allows the solution interpolation between the two grids. This strategy to refine the vehicle grid is acceptable for this prescribed motion case, but one could imagine the challenges that would arise when the store switches to 6DOF motion for its trajectory, and its flight path is unknown *a priori*.

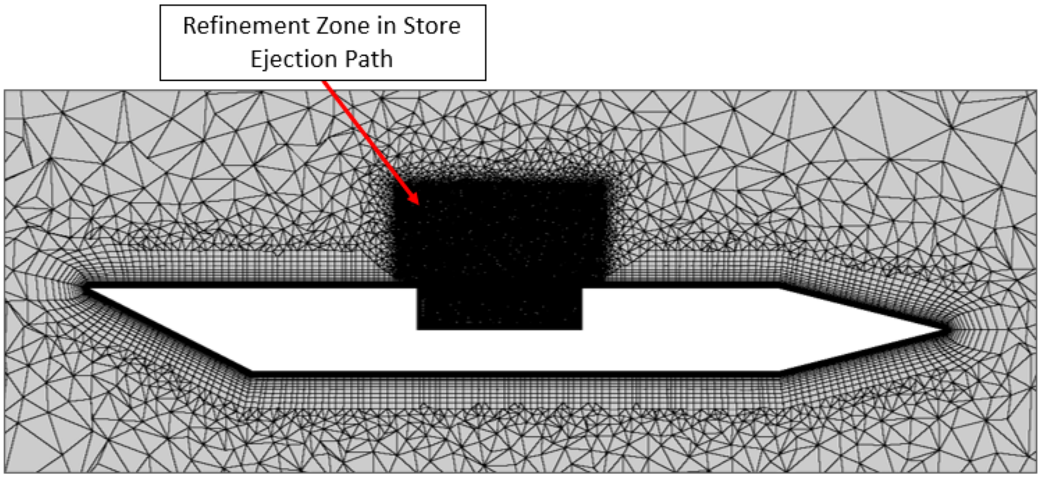


Figure 4.23 Modified vehicle component grid with extended cavity refinement zone

The store translates upward at a velocity of 3m/s with a coarse time step, Δt = 3.33x10^{-4}s. Again, in this portion of the study, the emphasis is placed on the validation of methods. Mach contour plots for several time steps are shown in Figure 4.24 as the store is ejected from the cavity.

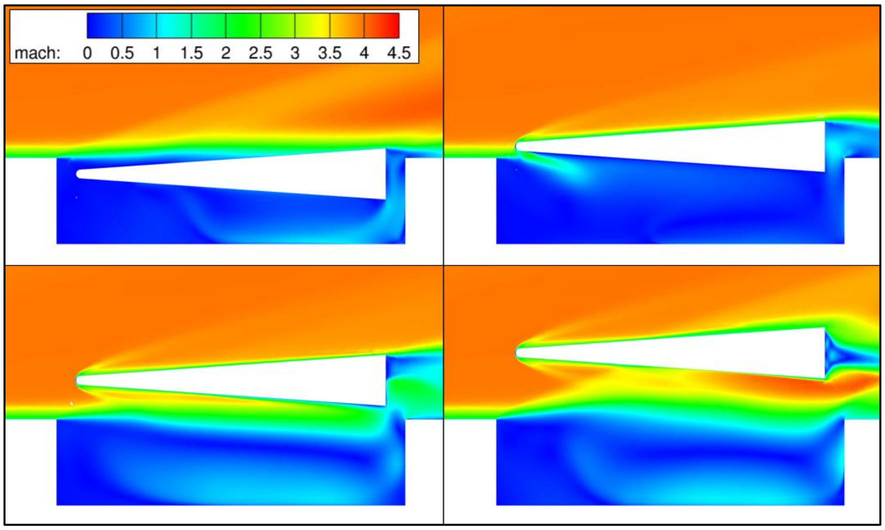


Figure 4.24 Mach contours as the store is ejected out of the cavity in a dynamic simulation

However, the store does not reach its full 0.5-meter translation before the flow solver can no longer proceed. Throughout the dynamic solution, low values of density (typically less than 10^{-3}) are computed just downstream of the store. Very low, or even negative, values of pressure or density are not only non-physical, but they also cause numerical solution failure in calculations of the local speed of sound. Consider that $a = \sqrt(\gamma\ P/\rho)$. After 324 steps ($t$ = 0.108s), the solution fails due to these low values. FUN3D attempts to apply a density fix to keep the computation going, but at some point, the values are too extreme to continue. Figure 4.25 shows the location of these low-density values behind the store.

This is as far as dynamic simulations have been taken using Yoga as the overset domain assembler. The solution failure here is attributed to how close the boundaries of the store mesh are to the surfaces of the store body itself. By forcing the flow solver to interpolate solution values so close to the bodies, solution accuracy is sacrificed. These limitations and inaccuracies motivated the search for an alternate domain assembly strategy. This is pursued further in the following section which employs the commercial domain assembly software, Suggar++, and compares its results to those computed using Yoga.

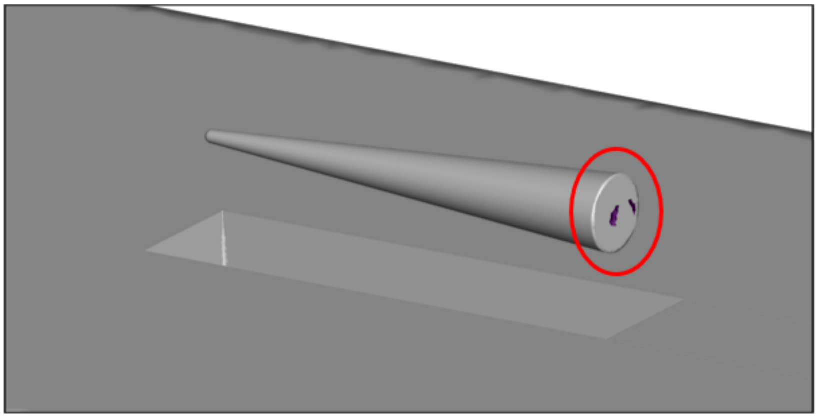


Figure 4.25 Location of low-density solution failure behind the store as it is ejected

4.6 Cavity and Door Simulations Using Suggar++

Until this point in the study, all domain connectivity information has been generated using the Yoga overset assembler that is distributed with FUN3D. This study pushes any overset algorithm to its limits. Having three bodies in close proximity, two of which have coincident sliding surfaces, is far from the standard use case for overset assembly tools. The requirement to have a sealed cavity that is bound by multiple bodies is also unique to this problem. Unfortunately, Yoga's assembly tools must be set aside for this study. In cases where the component grids of three bodies interact, the challenges presented forced modifications to the grids which are not acceptable for accurate solutions. An overset assembly package that can handle three bodies is desired.

Suggar++ is a well-established commercially available overset assembly program. This study revisits the previous cases to see if Suggar++ yields better results. FUN3D is built to integrate with either Yoga or Suggar++ for domain connectivity information, so the following CFD solutions come from the same solver using similar inputs as the cases built using Yoga, only now obtaining domain connectivity information from Suggar++. That is not to state that any differences shown are a result of what information is generated in Yoga or Suggar++, as it is possible that FUN3D handles DCI data differently from each overset assembler. The integration of each assembler into FUN3D would be the subject of a deeper study, but here, an emphasis is placed on the solution results that come from each, guiding the study toward a set of methods that successfully solve the problem at hand.

Before analyzing results from Suggar++, it is best to understand one of the key differences between the Yoga and Suggar++ implementations in FUN3D. Particularly, the way that "out" nodes are handled has a significant impact on the results displayed using each grid assembly code. When using Yoga, nodes that are flagged "out" do not have their solution values updated during the computation. The Suggar++ implementation, however, uses an average of nearby nodes to update the solution at "out" nodes. One motivation behind this feature is that, during a dynamic simulation, when nodes move into the flow field and have their statuses updated from “out” to “in”, they need to have a reasonable value. The initial condition values are usually not acceptable.

To demonstrate this difference, the steady solution for the door grid alone is shown in Figure 4.26, from both Yoga and Suggar++. This computation included the vehicle and store grids, but the solution on the door grid is isolated for visualization. Note that the entire grid is displayed, including “fringe” and “out” nodes. These are full size door grids, not the constrained door grid from Figure 4.21. In the solution using Yoga, it is clear which locations in the grid are considered “out” as they maintain the initial solution, Mach 4 freestream values outside the cavity, and zero-velocity inside the cavity. The solution from Suggar++, however, shows that the out nodes are not held at their initial values. They are updated over the course of the solution using average values from their neighboring nodes. This difference between the overset assembly implementations has an impact on both the steady and dynamic results.

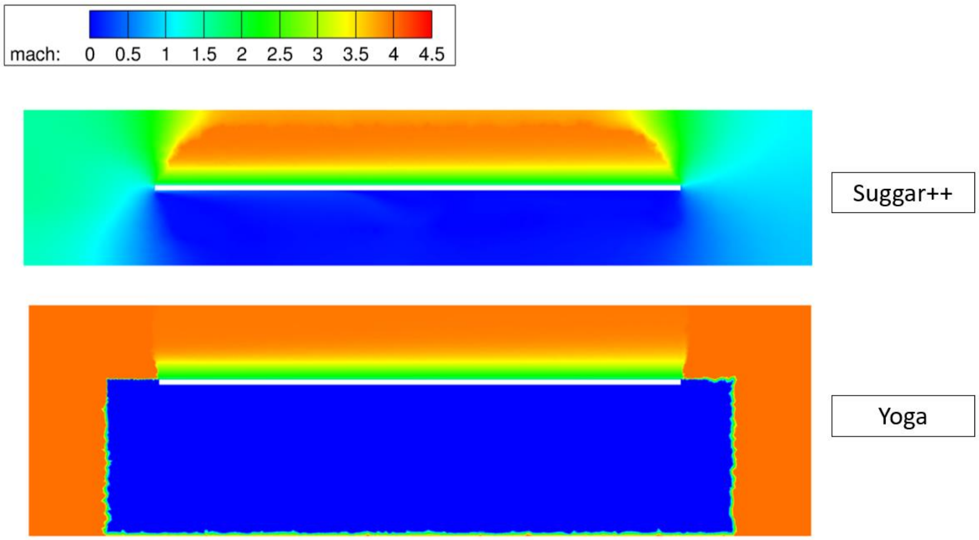


Figure 4.26 Mach contours for the steady-state solution shown on the door component grid only

Steady-State Solution Using Suggar++

Switching to Suggar++ for overset assembly, this study returns to the use of the "full" grids where the meshes for all three bodies (vehicle, door, and store) overlap each other significantly. The original door, store, and background grids are used, shown in Figure 4.4, Figure 4.5, and Figure 4.6, respectively. The modified vehicle grid with an extended cavity refinement zone is used, as shown in Figure 4.23. Before computing a solution, the domain connectivity is assessed from Suggar++, and the results are compared to Yoga. Figure 4.27 and Figure 4.28 show the grid composition using each domain assembler. Separate regions represent the "in" nodes from each component grid. The images display how the component grids are effectively "stitched" together.

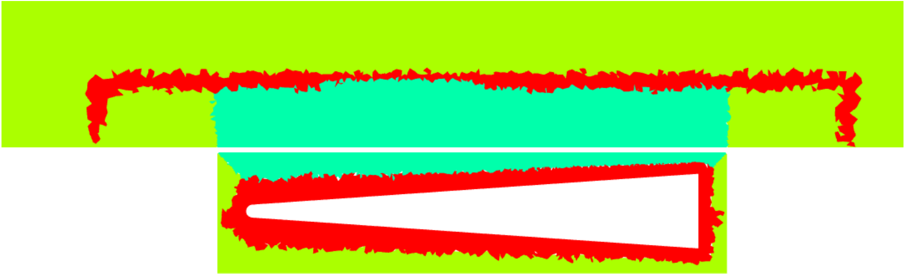


Figure 4.27 Grid composition using Yoga with a closed door, colored by component grid ID

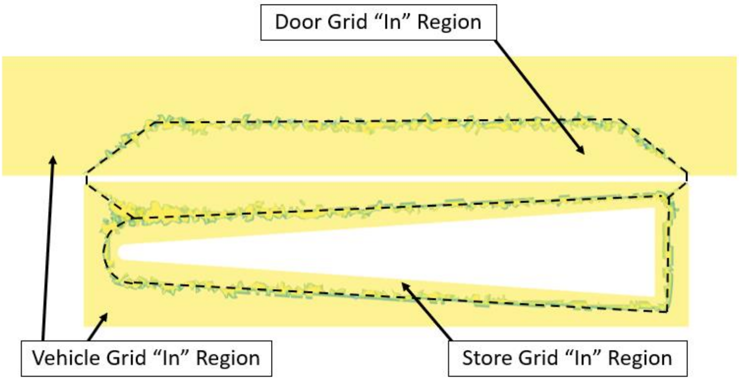


Figure 4.28 Grid composition using Suggar++ with a closed door

When the store is inside a closed cavity, one would expect that nodes from the store grid are only considered "in" inside the cavity. This is not the case when Yoga is used for domain assembly. There is an unexpected region of "in" nodes from the store grid well outside the cavity (Figure 4.27). This region, although it is part of the store grid, coincides with the outer boundaries of the door grid. However, when Suggar++ is used for domain connectivity in this case, the domain assembly appears as expected, as shown in Figure 4.28. The "in" nodes of the store are limited to within the cavity.

The steady solution using Suggar++ and the fully overlapping grids is shown in Figure 4.29. These results use similar component grids as the computation which produced the unsatisfactory results shown in Figure 4.16. It is noted that the boundary layer is appropriately captured above the edges of the door where the door and vehicle grids interact. Where the solution is interpolated, at the leading edge of the door, for instance, the presence of an overlapping third grid does not result in an erroneous solution. Previously, using Yoga, this set of meshes resulted in the "leak" of freestream values into the flow from the store grid into the solution. Fortunately, that is not the case here. This means that there is no need to modify the store grid, limiting its boundaries to the confines of the cavity. Keeping the grid boundaries far from the body surface is still viable, which is a better practice for accurate solutions.

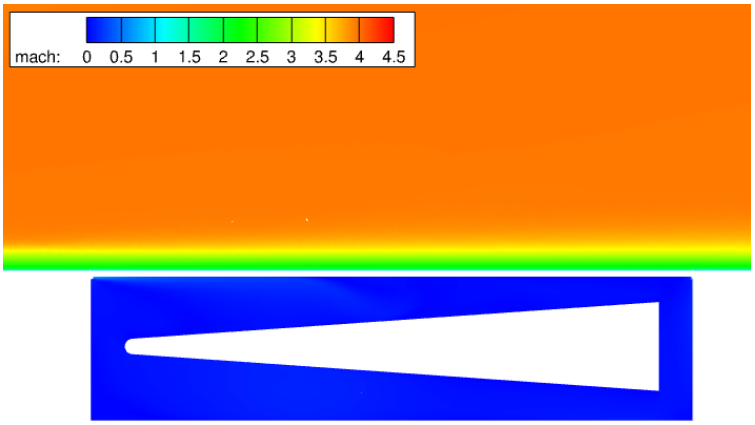


Figure 4.29 Steady-state solution using Suggar++ and fully overlapping component grids

One drawback of the Suggar++ implementation is how orphan nodes manifest themselves in the solution. These orphan nodes are present regardless of whether Yoga or Suggar++ is used for domain assembly, and they occur at the intersection of boundary layers in different component grids. This takes place, unavoidably, around the edges of the bottom surface of the door, where the boundary layer of the door intersects with the boundary layer of the cavity side walls, similar to Figure 4.10. In these boundary layers, the wall-normal grid spacing is orders of magnitude smaller than the surface or wall-tangential spacing. Fringe nodes are marked at this intersection, and the difference in point spacing causes the domain assembler to not find sufficient donor cells. Without sufficient donor information, the cells are tagged as orphans. This turns out to be more problematic for Suggar++ than Yoga in the case of this study.

One of the primary goals for the steady solution was to maintain a quiescent state in the cavity. This was achieved using Yoga. However, there are additional challenges presented using Suggar++. Figure 4.30 shows a Mach number contour plot again for the steady solution using Suggar++ on a modified contour scale from Mach 0 to Mach 1. This scale highlights subsonic flow features that were imperceptible in Figure 4.29. Here it is evident that there is a significant solution leak in the cavity, especially at the leading edge of the door. This leak originates at the orphan nodes where the door and front cavity wall meet. Maximum Mach number at the leak location is 0.7. Typical Mach number values in the cavity range from 0.05 to 0.15. The orphan nodes must be assigned some solution value so that a solution can be computed on their corresponding component grids, and it appears as if the values assigned to these orphans are different for the Yoga and Suggar++ implementations.

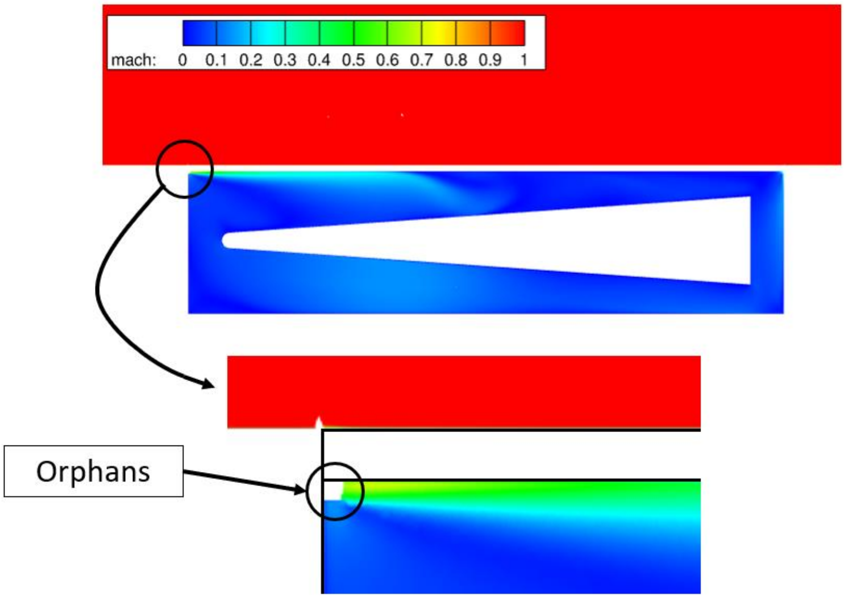


Figure 4.30 Mach number contours for the steady-state solution using Suggar++

In the absence of suitable donor information, orphans are generally assigned solution values according to an average solution of the nodes around them. It is suspected that the difference here between Yoga and Suggar++ is a result of how each implementation handles the nodes tagged as "out". When using Yoga, the "out" nodes around these orphans were not updated (refer to Figure 4.26) and still held the initial zero-velocity solution. Thus, these orphan nodes also remained at zero velocity as they averaged the solution of their neighbors. In the Suggar++ implementation, however, "out" nodes are updated through the computation. As stated before, this is generally a benefit, but in this case, it introduces solution values near the orphan locations that have non-zero velocities. This results in non-zero velocity values being assigned to the orphans, and flow is introduced inside the "sealed" cavity. This leak problem is yet to be solved, but it is assumed for the time being that it does not have a significant impact on the solution as the cavity door opens. This allows the study to continue using Suggar++ for domain assembly, but the impact of this leak should be studied further, and if possible, corrected. At this time, there is ongoing communication between the research team and the Suggar++ developer to address these issues.

Discussion of Results for a Dynamically Opening Door

Fortunately, as the door opens in a dynamic simulation, the Suggar++ implementation does not encounter the same challenges that were presented using Yoga. As the door opens, the composite mesh is built as expected, and the erroneous flow spikes (as in Figure 4.20) are not present in the solution. To demonstrate successful domain assembly using Suggar++, Figure 4.31 shows the results of the domain composition with the door partially open. These results appear as expected, in contrast to the unexpected "in" region from the door grid previously shown in Figure 4.19.

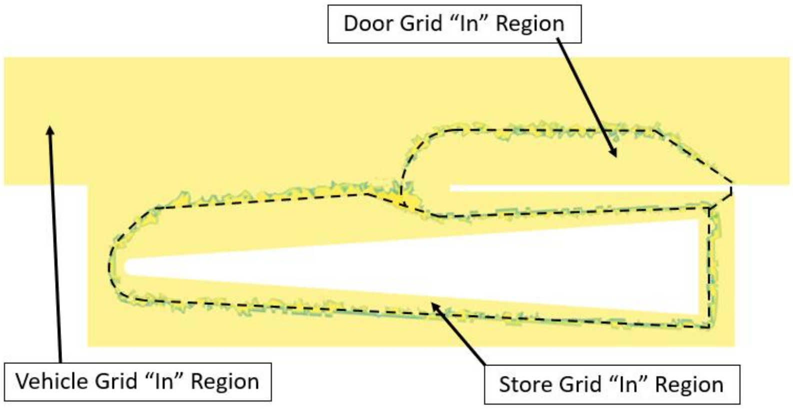


Figure 4.31 Grid composition using Suggar++ at t = 0.56s

When discussing the results of opening door simulations, a couple of key questions come to mind. The first question posed here is, "To what degree does the overset nature of the grid scheme impact the solution?". To address this question, static solutions using both Yoga and Suggar++ assemblers will be compared to solutions coming from a single continuous mesh model. In the continuous mesh, herein referred to as the “single mesh”, there are no individual grid components, but a single grid entity represents the entire domain. The single mesh is not suitable for time-accurate dynamic simulations, so only steady solutions with the door in various positions will be considered for this comparison between single mesh and overset solutions. The Mach number distribution in the cavity will be considered in a qualitative sense, and the resulting forces on the store will be considered quantitatively.

A second question of interest is, “How do the results of the time-accurate dynamic solution compare to a quasi-static approach?” To answer this question, the steady solutions obtained with the cavity door in various positions will be compared to the results of the corresponding times during a time-accurate, dynamic computation. Mach number and temperature distributions will be used to compare the results qualitatively. The resulting forces on the store and data from pressure and temperature probes provide a more quantitative comparison.

Comparison of Overset and Single Mesh Static Solutions

To support the validity of solutions using an overset grid scheme, results are compared with the store in several static positions using a single mesh, an overset composite grid using Suggar++, and an overset composite grid using Yoga. Because the dynamic simulations using Yoga required that the door and store mesh never overlap, the modified non-overlapping grids (shown in Figure 4.21) are used for the Yoga solutions here. The Suggar++ solutions use fully overlapping component grids. The single mesh grid has comparable grid resolution to the overset grids. For this comparison, steady-state solutions were obtained with the door in the 25% open, 50% open, 75% open, and 100% open positions, using each of the three grid schemes.

Figure 4.32 shows Mach number contour plots for each grid scheme with the door in the 50% open and 100% open positions. Because the impact of the overset method is being investigated here, the single mesh solution is considered the baseline for comparison. The Suggar++ implementation provides results much closer to the single mesh solution than does the Yoga implementation. Some minor differences can still be seen between the Suggar++ and single mesh solutions because of the solution interpolations that take place between component grids. This comparison shows that the Suggar++ solution resembles the single mesh solution well, but it is a somewhat subjective comparison.

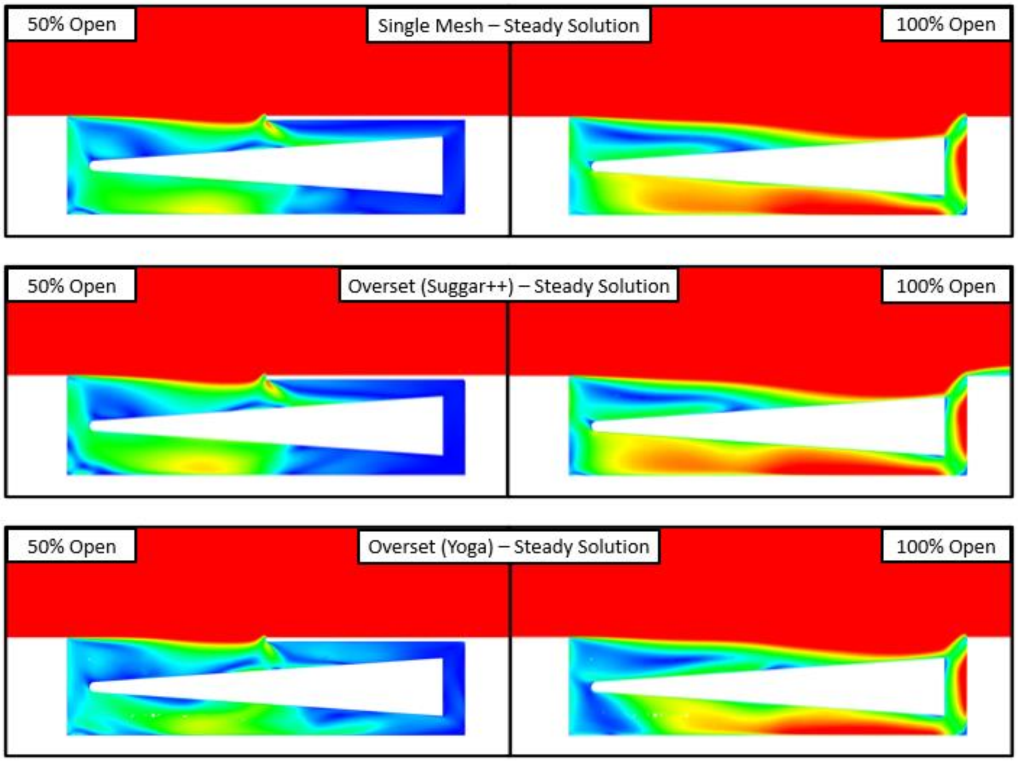


Figure 4.32 Mach contours for steady-state solutions using each of the three grid configurations

Comparing the aerodynamic forces on the store provides a more quantitative analysis. Force coefficients are plotted in Figure 4.33 for each grid configuration at each static door position. The force coefficient is defined,

$$C_{X,Y,Z} = \frac{F_{X,Y,Z}}{\frac{1}{2}\rho_\infty U_\infty^2 L_{ref}^2}$$

The store forces from the Suggar++ case are very similar to the single mesh results. Any minor differences are accepted, understanding that the solution interpolations in an overset method will always have some error. The results from Yoga are not consistent with the single mesh baseline results, especially in the spanwise "y" direction. A set of results is also provided in Figure 4.33 using Suggar++ and the modified "non-overlapping" component grids, like the Yoga solutions. These results are also similar to the baseline single mesh forces, indicating that the deviation of the Yoga results is related to the implementation of the overset assembler, rather than a result of using a different set of component grids.

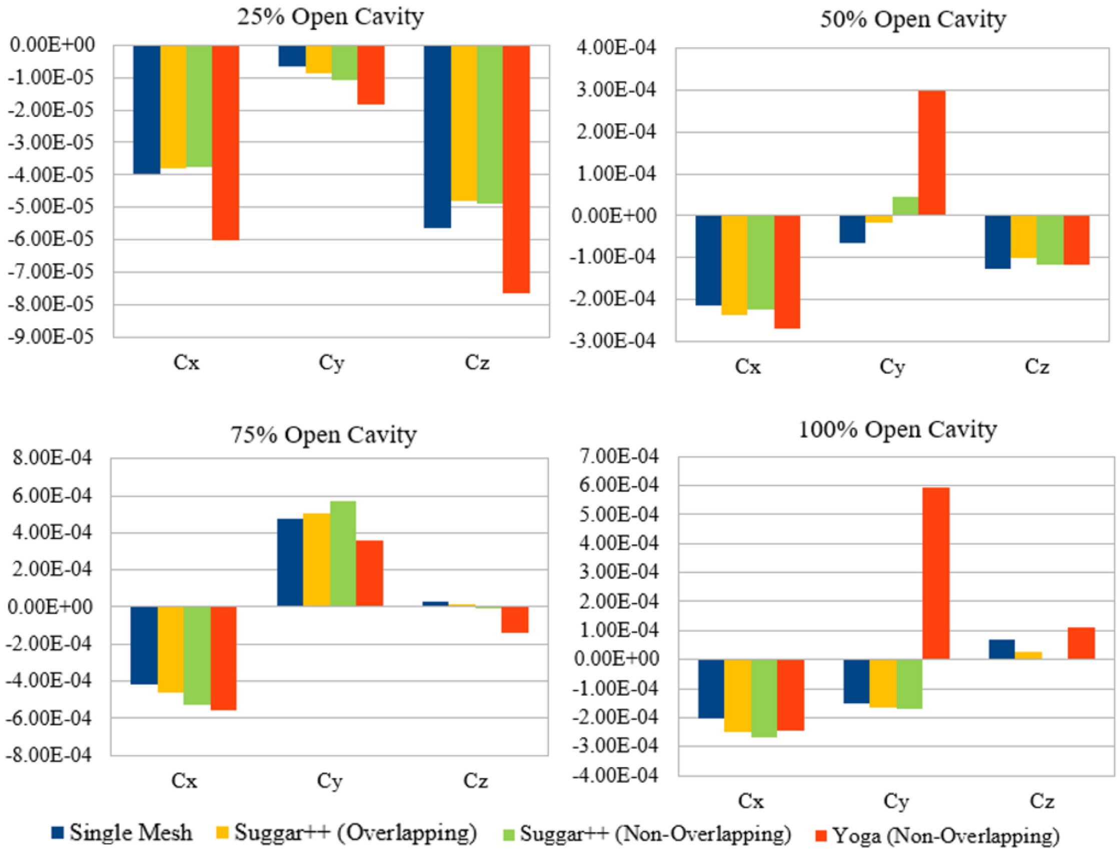


Figure 4.33 Store force coefficients from steady-state solutions using different grid configurations

The results validate the use of an overset grid method with Suggar++ for the simulation of an opening door with a store in carriage. The following analyses are limited to the use of overset grids using Suggar++. The impact of a dynamic solution approach is to be assessed.

Comparison of Dynamic and Quasi-Static Solutions

Now that an overset method has been established for door-opening simulations with a store in carriage, dynamic simulations are carried out, emphasizing that the door position is updated at each solution time step. This is unlike the quasi-static approach, which obtains a series of steady-state solutions with the door in various positions. Here, the question is addressed as to how these dynamic and quasi-static approaches compare. The importance of this comparison is two-fold. First, if there are transient phenomena as the door opens that are only captured in a dynamic approach, then the associated pressure and temperature loads should be accounted for in mechanical design. Second, if the state of the cavity flow after the door opens dynamically is significantly different from a static open cavity simulation, then a dynamic simulation approach would be required to appropriately capture loads on the store that impact separation trajectory analysis.

The following analyses all compare a single computational simulation of a dynamic door opening to a series of steady-state solutions with the door at various open positions. The dynamic simulation uses the same freestream conditions and computational methods described in Section 4.1, and it uses the same component grids described in Section 4.3, except for the vehicle grid. Anticipating ejecting the store from the cavity in future simulations, the vehicle grid with an extended refinement region is used as shown in Figure 4.23. Consistent with the time study presented in Section 4.4, a fixed time step of 2x10^{-4}s is used. This is the "5K" time resolution as the door opens in 5,000 steps, sliding downstream with a velocity of 1m/s for a duration of 1 second. The time study in Section 4.4 was conducted without a store in carriage. It is assumed here that, like the time study, a finer time step will capture higher frequency pressure oscillations but not significantly change the magnitude of the mean pressure loads in the cavity. There was no store present during the time study, so further study is warranted to verify that this assumption is valid when a store is in carriage. However, the assumption is accepted for now. The steady-state solutions for the quasi-static door opening utilize the same set of component grids, using Suggar++ for domain assembly.

Figure 4.34 shows Mach contour plots at the mid-span section as the door opens for both the quasi-static and dynamic cases. Both cases show similar results when the door is 25% and 50% open, with slightly higher Mach numbers in the quasi-static case where the flow has had time to fully develop. Significant differences between the quasi-static and dynamics cases are apparent when the door is 75% and 100% open. By these opening times, simulating a dynamic door yields different results than the quasi-static approach.

Examining the instant that the door is fully open, these differences are clarified with streamlines plotted on the mid-span section as well as a cavity half-depth section. These are shown in Figure 4.35. In the quasi-static open cavity solution, the flow is characterized by two symmetric counter-rotating vortices on either side of the store. However, the flow pattern is much more complex in the dynamic door-opening solution. As shown in Figure 4.35(d), the twin vortices have shifted upstream and are no longer spanwise-symmetric. Small, asymmetric vortical regions have also developed just downstream of the primary vortices. These flow features that are only captured in a dynamic simulation should be considered in cavity design and may even have an impact on subsequent store trajectory analysis.

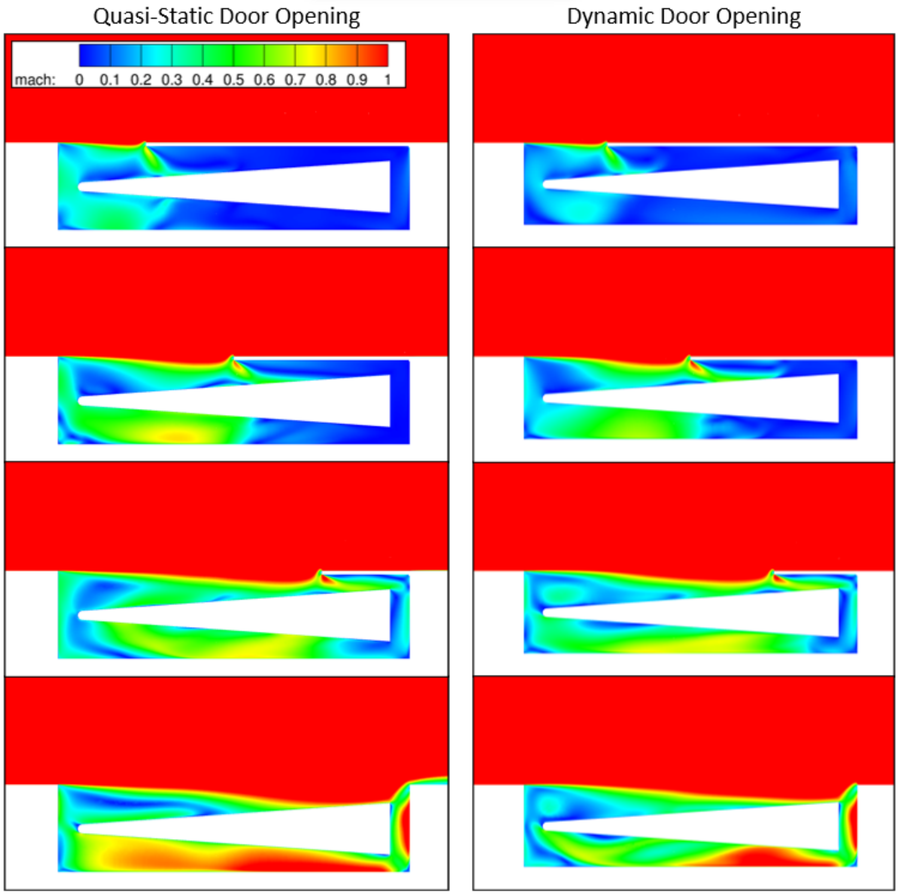
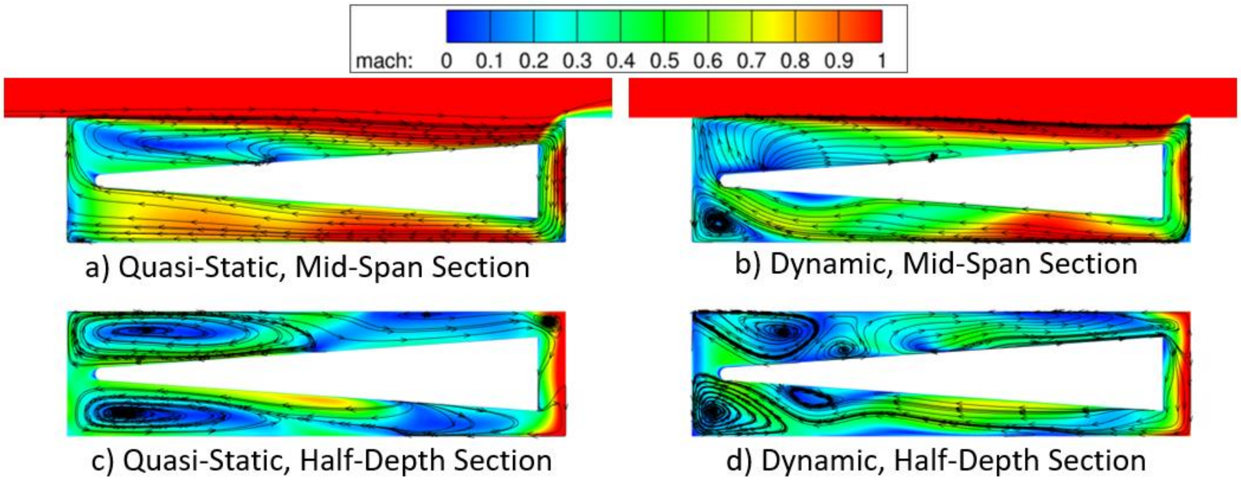


Figure 4.34 Mach number contours for quasi-static and dynamic door opening simulations



a) Quasi-Static, Mid-Span Section

b) Dynamic, Mid-Span Section

c) Quasi-Static, Half-Depth Section

d) Dynamic, Half-Depth Section

Figure 4.35 Mach number contours for quasi-static and dynamic simulations with 100% open door

This comparison between quasi-static and dynamic solutions for an opening cavity door is analyzed more quantitatively by comparing surface pressure data using virtual pressure probes. At three select locations on the cavity walls, surface pressure data is stored for the entire dynamic simulation and compared to the pressures calculated during the quasi-static runs. These probe locations are centered on the cavity front wall, rear wall, and floor. These locations are depicted in Figure 4.36, and coordinates are recorded in Table 4.2. The pressure history over the course of the door opening simulation is plotted in Figure 4.37, using a coefficient of pressure, defined as,

$$C_P = \frac{P - P_\infty}{\frac{1}{2}\rho_\infty U_\infty^2}$$

Alongside the dynamic pressure data, Figure 4.37 contains the quasi-static results using both an overset grid scheme (Suggar++) and the single mesh for reference. Each quasi-static data point is plotted at the time where the door position corresponds with the dynamic simulation. The surface pressure data from the quasi-static solutions align with the dynamic data generally, but not closely. Perhaps of most interest to the design engineer, surface pressure maxima and minima are not captured by the quasi-static solutions.

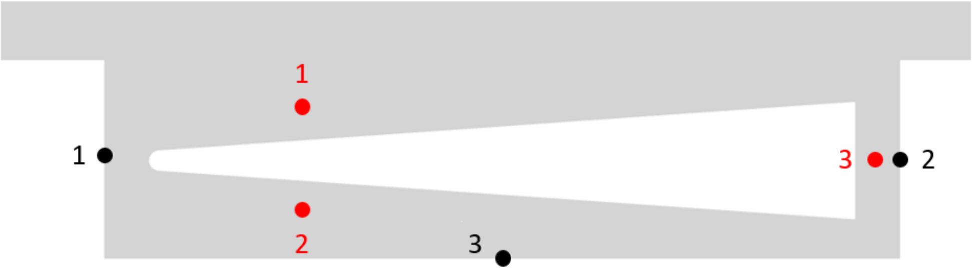


Figure 4.36 Numbered locations for pressure (black) and temperature (red) data probes

Table 4.2 Pressure and Temperature Data Probe Positions

Probe Type/Number	X-Position (mm)	Y-Position (mm)	Z-Position (mm)
Pressure 1	0	0	-125
Pressure 2	1000	0	-125
Pressure 3	500	0	-250
Temperature 1	250	0	-62.5
Temperature 2	250	0	-187.5
Temperature 3	972.5	0	-125

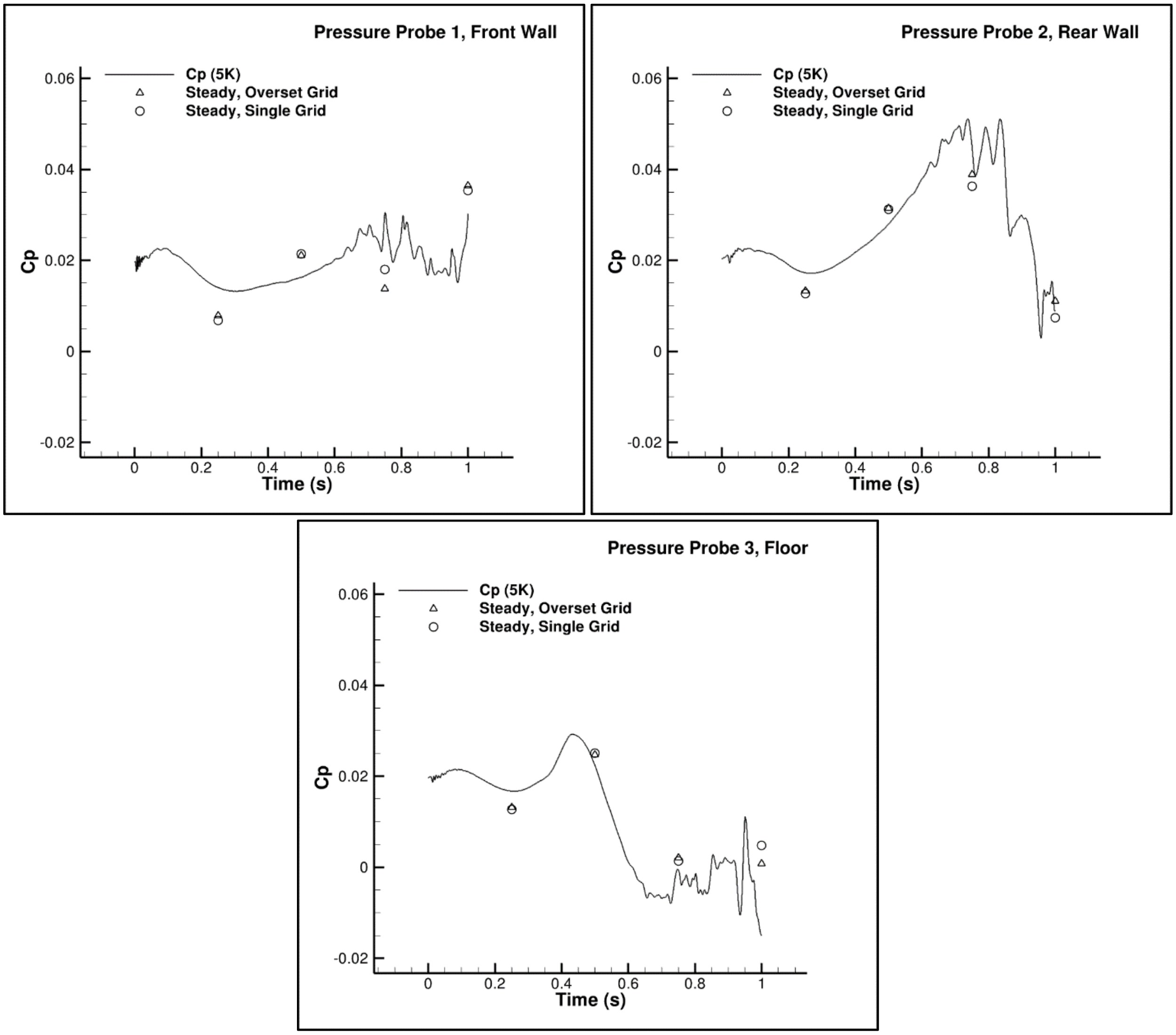


Figure 4.37 Dynamic pressure history and quasi-static data points at probe locations

The aerodynamic forces on the store are also of interest for this comparison. Despite the differences shown thus far between the solutions, the quasi-static and dynamic simulations record very similar net forces on the store. These forces are plotted in Figure 4.38. Again, the quasi-static data points are plotted at times where the door positions correspond. Although the quasi-static forces agree with the dynamic simulation, the additional transient data provided by the dynamic simulation provides insight into the force variations on the store over time. Like the pressure data, the store force values become highly oscillatory after the door is about 60% open. Notably, this coincides with the time that Turpin et al. [19] estimate for cavity resonance startup. Also, one can see that maximum forces on the store occur when the door is about 82% open, not fully open, which is important for any design decisions based on store loads.

Although the quasi-static and dynamic force results agree, this does not indicate that the loads on the store are equivalent for the two opening methods. The distributions of aerodynamic pressure loads on the store differ. This is exhibited in Figure 4.39, where the coefficient of pressure on the surface of the store is plotted with the door fully open, using both a quasi-static and dynamic approach. The pressure distribution data is taken from a cross-section of the store at the half-depth of the cavity. Clearly, the pressure distributions on the store are quite different. The pressure distribution for the dynamic simulation shows larger differences in pressure values on either side of the store, consistent with the flow asymmetries that were observed in Figure 4.35. Thus, it is shown that simulating the door opening dynamically has an impact on store loads in the cavity. Whether the agreement in the net force values is meaningful or not warrants further study.

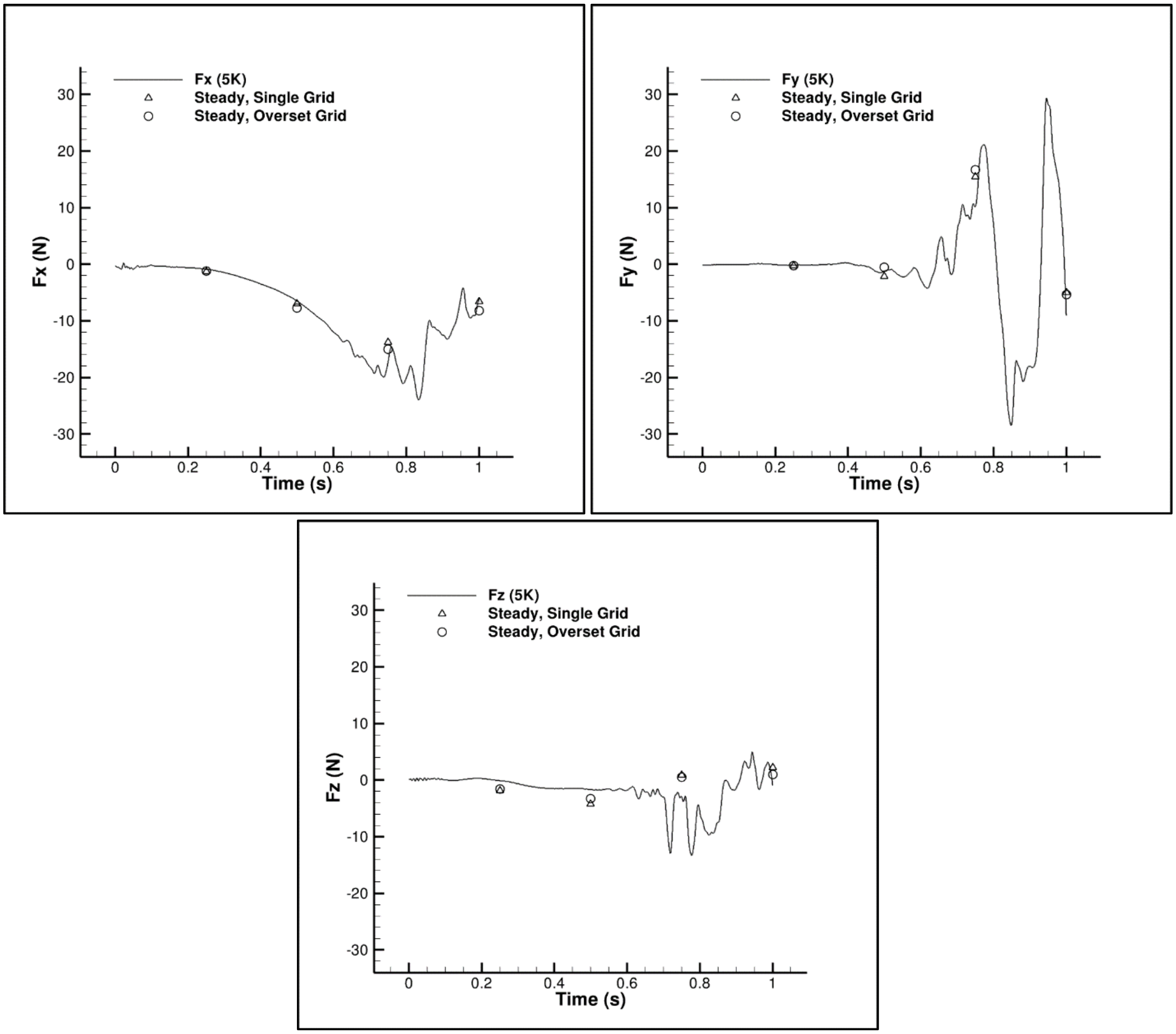


Figure 4.38 Dynamic store force history and quasi-static data points in the x-, y-, and z-directions

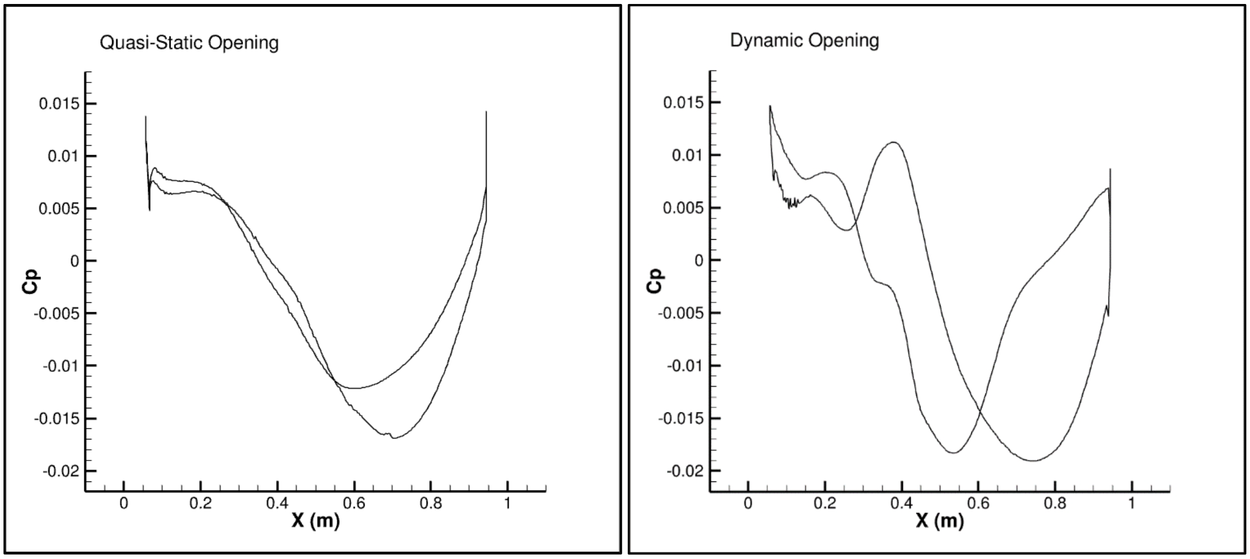


Figure 4.39 Pressure distribution on the store when the door is 100% open

Additionally, cavity temperature data are compared between the quasi-static and dynamic simulations. Figure 4.40 shows temperature contour plots for both simulations with the door in various positions. Especially when the door is only 25% or 50% open, the temperature rise goes deeper into the cavity for the quasi-static solutions. This effect is attributed to the fact that the quasi-static method computes a steady solution at each door position, allowing as much as time necessary for the temperature distribution to permeate the cavity space.

Like the surface pressure point data that was recorded, temperature data was probed at three locations and recorded as the door opened. These locations are depicted in Figure 4.36 and their coordinates are recorded in Table 4.2. The first two locations are toward the front of the cavity, just above and below the store. The third location is halfway between the rear face of the store and the rear cavity wall. Figure 4.41 plots the dynamic temperature history at each point, and the temperature values from the quasi-static solutions are plotted for comparison. Also like the pressure data, the quasi-steady temperature values only roughly agree with the data from the dynamic simulation. Local maxima and minima are not adequately captured. If cavity temperatures are of interest in mechanical design, a dynamic simulation approach may be required to accurately calculate the temperature extremes that impact design decisions.

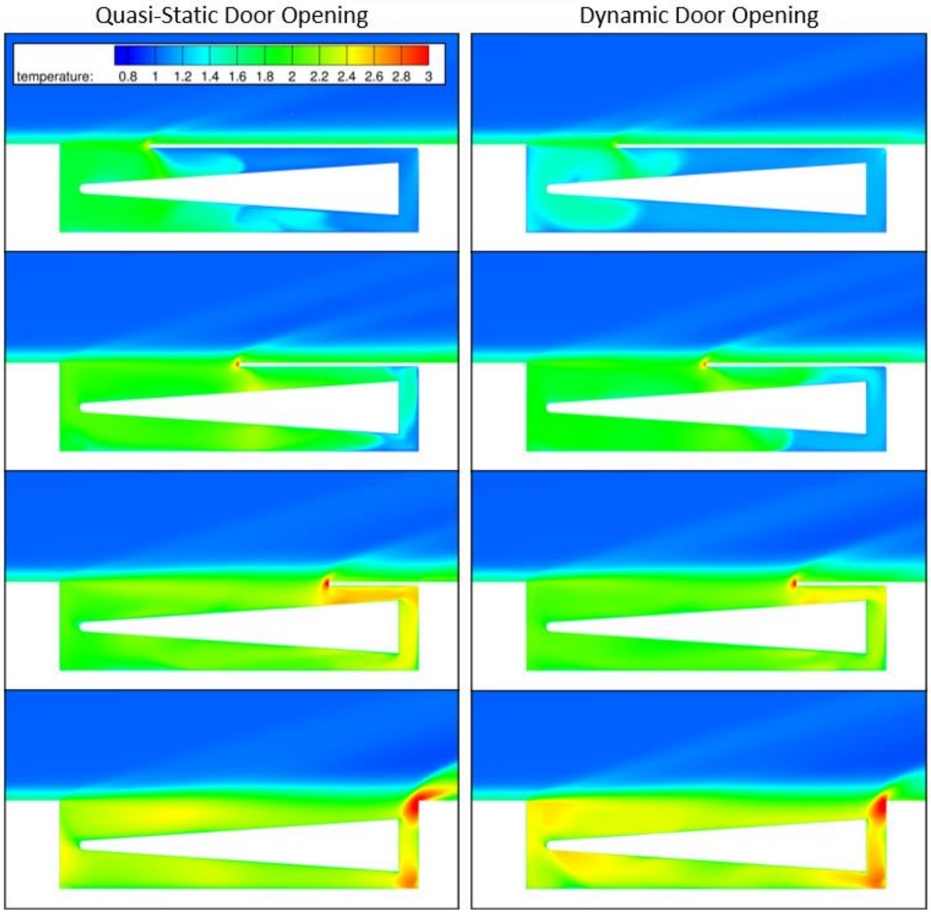


Figure 4.40 Temperature contours for quasi-static and dynamic door opening simulations

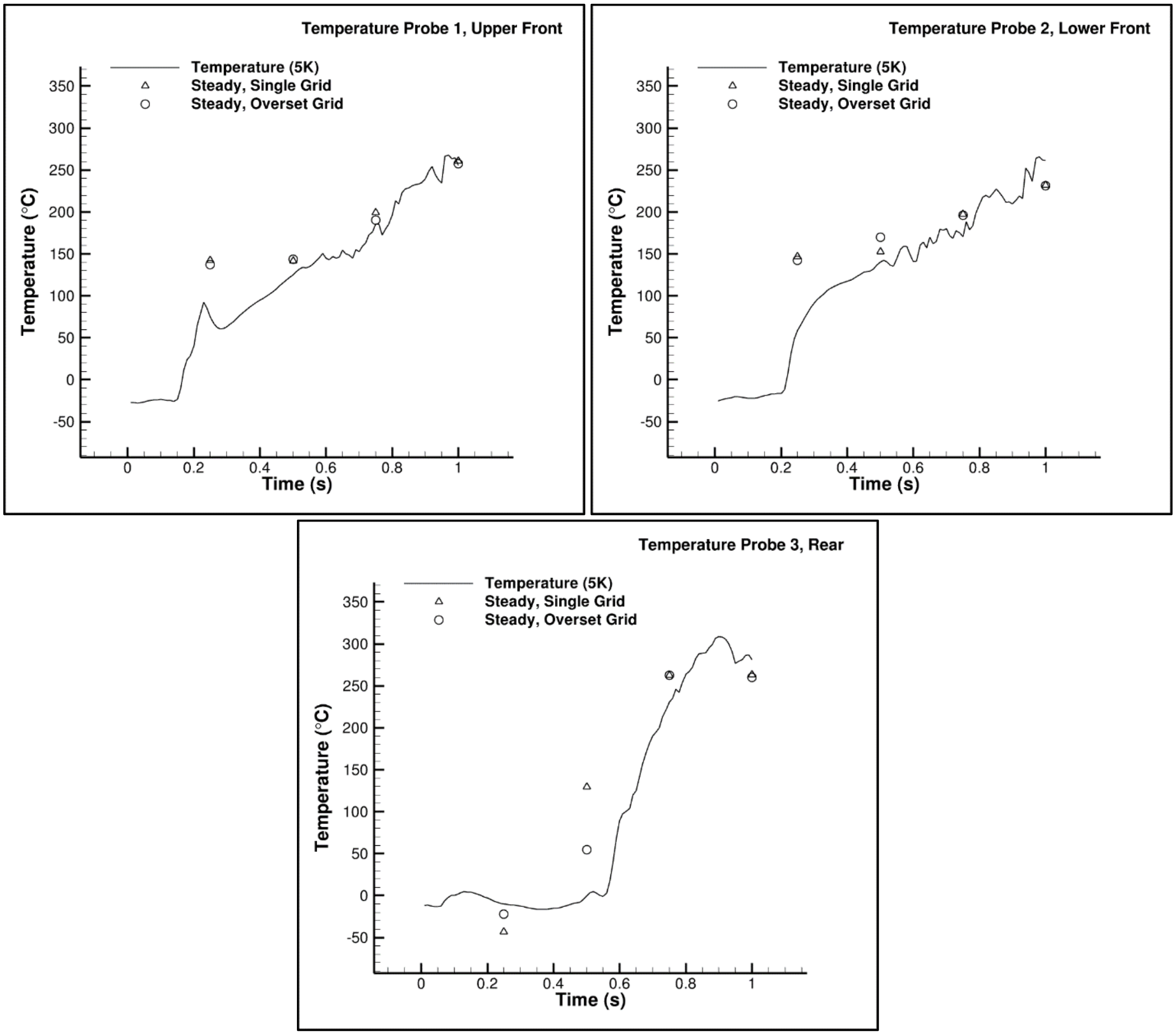


Figure 4.41 Dynamic temperature history and quasi-static data points at probe locations

All in all, these comparisons demonstrate the importance of a dynamic simulation approach to cavity opening, moving the door at every solution time step rather than computing a series of steady solutions with the door in various positions. Flow visualization using Mach contours and streamline plots show asymmetric flow features which are present in the dynamic simulation but not using the quasi-static approach. This has an impact on store loads, changing the pressure distribution. However, net forces seemed unaffected, which requires further investigation. Future work will assign a center of mass reference point to the store body so that integrated store moments are calculated, which may help clarify the impact on store loads. Temperature results also varied between the quasi-static and dynamic approaches.

This study lays the groundwork for future studies in store ejection and trajectory analysis. The trajectory predictions of Chapter 3 considered initial conditions where the store had already left the viscous effects of the cavity flow. By simulating a dynamic opening door, more realistic initial conditions can be calculated for a cavity-ejected store. An added benefit of the dynamic simulation method is that it provides a more accurate analysis of the pressure and temperature loads experienced by the cavity and the store in carriage, which may be of critical importance for cavity design. It is anticipated that research will continue from this study, simulating an ejecting store and analyzing its trajectory.

CHAPTER 5

CONCLUSIONS

5.1 Summary of Results

As a field of study, store separation analysis has a long history of developing its techniques and methods to adapt to evolving aircraft technology. As the primary means of store carriage has shifted from externally mounted stores to internal weapons bays, store separation analysis has adapted to account for highly unsteady cavity flows. The work presented here looks toward a future including upward-ejected stores and investigates how CFD solutions can begin to assess the main effects of such trajectory simulations. Inviscid, quasi-static trajectory analysis is presented with an emphasis on the inclusion of drag-enhancing features in store design to improve separation safety. Recognizing the need to account for the viscous, unsteady nature of cavity ejection, methods were developed and presented for time-accurate, dynamic door opening simulations. Such simulations provide more realistic initial conditions for trajectory simulations and provide critical transient information for cavity and store design.

Chapter 3 presents trajectory predictions for stores ejected upward into Mach 4 and Mach 6 flow. The results show that streamlined bodies may not induce sufficient drag for safe separation, simply falling back onto the vehicle. Modifications to the store, such as reversing its orientation and the addition of an aft-mounted disk, show that store designs with enhanced drag have potential for safe separation. They are pushed downstream quickly enough prior to falling back toward the vehicle. Oscillations in store pitch were noted for stores modified with a drag-inducing disk. Such behavior should be considered for store control after ejection. These trajectory predictions are a preliminary analysis, not accounting for viscous and unsteady cavity effects.

Chapter 4 starts an investigation to include such cavity phenomena into store separation analysis. The novelty of the work presented here is a time-accurate, dynamic simulation of opening a sealed cavity with a store in carriage. Such a simulation includes various bodies in close proximity, a configuration that challenges the conventional use of overset grid methodology. This work addresses those challenges and successfully simulates the dynamic cavity. The dynamic approach, moving the door's position at each solution time step, shows some important differences in comparison to a quasi-static approach that approximates the dynamic solution by a series of steady-state computations. By the time the door opens, the dynamic cavity simulation shows significant spanwise asymmetries in the cavity flow that do not appear using a quasi-static method. This impacts the pressure distribution on the store the moment before it is ejected, potentially influencing its trajectory. Also, more accurate transient pressure and temperature data inside the cavity may be important for cavity and store design decisions. These studies are foundational to continued work in store separation analysis from cavities that are modeled with moving doors.

5.2 Future Work

The work presented here investigates new topics in store separation. As such, many of these studies are exploratory, introducing new questions. Also, engineering assumptions were made to produce these initial results, which should be addressed to assess their impact on solution accuracy. In the analyses presented, various suggestions for future work have been made, which are summarized here to effectively guide the continued success of these developments.

Following the trajectory predictions of Chapter 3, it is a natural progression to investigate the impact of the inviscid and quasi-static assumptions that were made. Time-accurate, full Navier-Stokes solutions should be coupled with a 6DoF routine to predict store trajectories, comparing their results to those presented. Also, due to the success of drag-enhancing store design features to improve separation safety, deployable drag fins that rotate open from the store should be studied. Such a design can minimize the store's profile in carriage and may be a more realistic design than the simple geometries assessed here. This investigation is already underway, utilizing overset grid methods to model drag fin motion.

Future study is also suggested to continue the investigation of opening a cavity with simulated door motion. Following these analyses of an opening door, it is natural to continue by simulating the ejection of the store through the viscous shear layer. Subsequently, one could predict more accurate store trajectories based on these more realistic initial conditions. Additionally, there are some lingering questions which need to be addressed from the cavity studies. First, the flow leaks into a closed cavity when using Suggar++ should be addressed, eliminating them if possible, or at least understanding their impact on the dynamic solution. Also, the choice of time-step should be reevaluated with a store in carriage as the presented time study only considers an empty cavity. Finally, when comparing the forces on the store in quasi-static and dynamic door-opening simulations, the integrated net forces agree while the pressure distributions on the store show significant differences. Such an effect deserves more understanding, which can be addressed in the future with integrated moment calculations on the store.

As presented, both the trajectory predictions and the dynamic cavity simulations provide important contributions to early studies in upward-ejected stores. With much continued work ahead, there is great potential for computational methods to provide critical insights into designs for safe store separation in evolving aircraft technologies.

REFERENCES

[1] "HORIZON Tennessee Aerothermodynamics Laboratory (TALon)," University of Tennessee Space Institute, accessed Sept 12, 2023.
http://www.utsi.edu/horizon/talon

[2] "Index of /~AAE519/BAM6QT-Mach-6-tunnel," Purdue University, accessed Sept 12, 2023.
http://www.engineering.purdue.edu/~aae519/BAM6QT-Mach-6-tunnel

[3] Cenko, A., "Store Separation Lessons Learned During the Last 30 Years," Defense Technical Information Center, ADA538155, 2010.

[4] Cenko, A., "Store Separation Overview," North Atlantic Treaty Organization: Science and Technology Organization, EN-SCI-277-04, 2016.

[5] Bamber, M. J., "Two Methods of Obtaining Aircraft Trajectories from Wind Tunnel Investigations," Navy Dept., 1960.

[6] Cenko, A., Tinoco, E. N., Dyer, R. D., and Dejongh, J., "PAN AIR Applications to Weapons Carriage and Separation," *Journal of Aircraft,* Vol. 18, No. 2, 1981, pp. 128-134.
doi: 10.2514/3.57473

[7] Madson, M., Moyer, S., and Cenko, A., "TranAir computations of the flow about a generic wing/pylon/finned-store configuration," AIAA-1994-155, 32nd Aerospace Sciences Meeting and Exhibit, 1994.
doi: 10.2514/6.1994-155.

[8] Cenko, A., and Lutton, M., "ACFD applications to store separation - status report," *Aeronautical Journal,* Vol. 104, No. 1040, 2000, pp. 459-466.

[9] Madson, M., and Talbot, M., "F-16/generic store carriage load predictions at transonic Mach numbers using TranAir," AIAA-1996-2454, 14th Applied Aerodynamics Conference, 1996.
doi: 10.2514/6.1996-2454.

[10] Fox, J. H., "Generic wing, pylon, and moving finned store," Defense Technical Information Center, ADP010735, Arnold Engineering Development Complex, 2000.

[11] Spinetti, R., and Jolly, B., "Time-Accurate Numerical Simulation of GBU-38s Separating from the B-1B Aircraft with Various Ejector Forces, Store Properties, and Load-Out Configurations - IHAAA Store Separation Cavity (SSC) Project," AIAA-2008-187, 46th AIAA Aerospace Sciences Meeting and Exhibit, 2008.
doi: 10.2514/6.2008-187.

[12] Sickles, W., Hand, T., Morgret, C., Masters, J., and Denny, A., "High-Fidelity, Time-Accurate CFD Store Separation Simulations from a B-1B Bay with Comparisons to Quasi-Steady Engineering Methods," AIAA-2008-186, 46th AIAA Aerospace Sciences Meeting and Exhibit, 2008.
doi: 10.2514/6.2008-186.

[13] Atkins, D., "Flight Test Results of a GBU-38 Separating from the B-1B Aircraft," AIAA-2008-184, 46th AIAA Aerospace Sciences Meeting and Exhibit, 2008.
doi: 10.2514/6.2008-184.

[14] Lee, J., and Cenko, A., "Evaluation of the GBU-38 Store Separation from B-1 Aft Bay," AIAA-2008-185, 46th AIAA Aerospace Sciences Meeting and Exhibit, 2008.
doi: 10.2514/6.2008-185.

[15] Cenko, A., Deslandes, R., Dillenius, M., and Stanek, M., "Unsteady Weapon Bay Aerodynamics - Urban Legend or Flight Clearance Nightmare," AIAA-2008-189, 46th AIAA Aerospace Sciences Meeting and Exhibit, 2008.
doi: 10.2514/6.2008-189.

[16] Kim, D. H., Choi, J. H., and Kwon, O. J., "Detached eddy simulation of weapons bay flows and store separation," *Computers & Fluids,* Vol. 121, 2015, pp. 1-10.
doi: 10.1016/j.compfluid.2015.07.022

[17] Murman, S., Aftosmis, M., and Berger, M., "Simulations of 6-DOF Motion with a Cartesian Method," AIAA-2003-1246, 41st Aerospace Sciences Meeting and Exhibit, 2003.
doi: 10.2514/6.2003-1246.

[18] Murman, S., Chan, W., Aftosmis, M., and Meakin, R., "An Interface for Specifying Rigid-Body Motions for CFD Applications," AIAA-2003-1237, 41st Aerospace Sciences Meeting and Exhibit, 2003.
doi: 10.2514/6.2003-1237.

[19] Turpin, A. M., Granlund, K. O., Hayashi, T., and Sakaue, H., "Supersonic cavity flow with a downstream-sliding door," *Experiments in Fluids,* Vol. 62, No. 12, 2021.
doi: 10.1007/s00348-021-03338-w

[20] Baugher, S. K., and Gaitonde, D. V., "3D Effects of Doors on Cavity Bay Dynamics," AIAA-2023-1003, AIAA SCITECH 2023 Forum, 2023.
doi: 10.2514/6.2023-1003.

[21] Sheta, E. F., Harris, R. E., George, B., Ukeiley, L., and Luke, E., "Loads and Acoustics Prediction on Deployed Weapons Bay Doors," *Journal of Vibration and Acoustics-Transactions of the Asme,* Vol. 139, No. 3, 2017.
doi: 10.1115/1.4035701

[22] Loupy, G. J. M., Barakos, G. N., and Taylor, N. J., "Multi-disciplinary simulations of stores in weapon bays using scale adaptive simulation," *Journal of Fluids and Structures,* Vol. 81, 2018, pp. 437-465.
doi: 10.1016/j.jfluidstructs.2018.05.012

[23] Rossiter, J. E., "Wind tunnel experiments on the flow over rectangular cavities at subsonic and transonic speeds," 1964.

[24] Heller, H. H., Holmes, D. G., and Covert, E. E., "Flow-Induced Pressure Oscillations in Shallow Cavities," *Journal of Sound and Vibration,* Vol. 18, No. 4, 1971, pp. 545-&.
doi: 10.1016/0022-460x(71)90105-2

[25] Unalmis, O. H., Clemens, N. T., and Dolling, D. S., "Cavity oscillation mechanisms in high-speed flows," *AIAA Journal,* Vol. 42, No. 10, 2004, pp. 2035-2041.
doi: 10.2514/1.1000

[26] Bartel, H. W., and McAvoy, J. M., "Cavity Oscillation in Cruise Missile Carrier Aircraft," Defense Technical Information Center, ADA108610, 1981.

[27] Subbareddy, P. K., and Candler, G. V., "A fully discrete, kinetic energy consistent finite-volume scheme for compressible flows," *Journal of Computational Physics,* Vol. 228, No. 5, 2009, pp. 1347-1364.
doi: 10.1016/j.jcp.2008.10.026

[28] Menter, F. R., and Egorov, Y., "The Scale-Adaptive Simulation Method for Unsteady Turbulent Flow Predictions. Part 1: Theory and Model Description," *Flow Turbulence and Combustion,* Vol. 85, No. 1, 2010, pp. 113-138.
doi: 10.1007/s10494-010-9264-5

[29] Babu, S. V., Zografakis, G., Barakos, G. N., and Kusyumov, A., "Evaluation of scale-adaptive simulation for transonic cavity flows," *International Journal of Engineering Systems Modelling and Simulation,* Vol. 8, No. 2, 2016, pp. 106-124.
doi: 10.1504/ijesms.2016.075510

[30] Johnson, R., Stanek, M., and Grove, J., "Store Separation Trajectory Deviations due to Unsteady Weapons Bay Aerodynamics," AIAA-2008-188, 46th AIAA Aerospace Sciences Meeting and Exhibit, 2008.
doi: 10.2514/6.2008-188.

[31] Loupy, G. J. M., Barakos, G. N., and Taylor, N. J., "Store Release Trajectory Variability from Weapon Bays Using Scale-Adaptive Simulations," *AIAA Journal,* Vol. 56, No. 2, 2018, pp. 752-764.
doi: 10.2514/1.J056485

[32] Spalart, P. R., and Allmaras, S. R., "A one-equation turbulence model for aerodynamic flow," *Recherche Aerospatiale*, No. 1, 1994, pp. 5-21.

[33] Rumsey, C., "The Spalart-Allmaras Turbulence Model," NASA Langley Research Center, accessed Sept 17, 2023.
https://turbmodels.larc.nasa.gov/spalart.html#sa

[34] Carpenter, M., Viken, S., and Nielsen, E., "The Temporal Efficiency of Higher Order Schemes," AIAA-2003-86, 41st Aerospace Sciences Meeting and Exhibit, 2003.
doi: 10.2514/6.2003-86.

[35] Benek, J., Steger, J., and Dougherty, F. C., "A flexible grid embedding technique with application to the Euler equations," AIAA-1983-1944, 6th Computational Fluid Dynamics Conference Danvers, 1983.
doi: 10.2514/6.1983-1944.

[36] Druyor, C. T., "Enhancing Scalability for FUN3D Rotorcraft Simulations with Yoga: an Overset Grid Assembler," AIAA-2021-2746, AIAA AVIATION 2021 Forum, 2021.
doi: 10.2514/6.2021-2746.

[37] Noack, R., Boger, D., Kunz, R., and Carrica, P., "Suggar++: An Improved General Overset Grid Assembly Capability," AIAA-2009-3992, 19th AIAA Computational Fluid Dynamics, 2009.
doi: 10.2514/6.2009-3992.

VITA

Tim Wolfe was born in Amsterdam, New York to Patrick and Kathleen Wolfe. Growing up in rural upstate New York, Tim and his two older sisters attended Fort Plain High School, each of them graduating with high academic distinction. He then went on to study at Grove City College in Grove City, Pennsylvania where he obtained a bachelor's degree in mechanical engineering in 2017. Following his undergraduate studies, Tim then joined his family in a southbound migration to Cleveland, Tennessee where he met his wife, Rebekah. He began his engineering career working for a gas appliance manufacturer in Dayton, Tennessee, gaining experience in design for manufacturing and engineering project management. After five years working in industry, Tim returned to academia in 2022 to pursue a master's degree in engineering with a special focus on computational fluid dynamics. Attending the University of Tennessee at Chattanooga, Tim worked as a graduate research assistant in the mechanical engineering department's hypersonics program, gaining valuable research experience as he studied. Completing his graduate degree, Tim now looks forward to pursuing a career in computational engineering solutions.