A Software Upgrade of the NASA Aeroheating Code "MINIVER"

by

Pierce Mathew Louderback

Bachelor of Science
Aerospace Engineering
Florida Institute of Technology
2008

A thesis
submitted to the College of Engineering at
Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Aerospace Engineering

Melbourne, Florida
July 2013

We the undersigned committee hereby recommend
that the attached document be accepted as fulfilling in
part the requirements for the degree of
Master of Science of Aerospace Engineering.

A Software Upgrade of the NASA Aeroheating Code "MINIVER"

a thesis by Pierce Mathew Louderback

Daniel R. Kirk, Ph.D.
Associate Professor and Associate Department Head
Aerospace Engineering
Thesis Advisor

Hector M. Gutierrez, Ph.D.
Associate Professor
Mechanical Engineering

Dr. Stephane Bucaille, Ph.D.
Assistant Professor
Electrical and Computer Engineering

Pei-feng Hsu, Ph.D.
Professor, Department Head
Mechanical and Aerospace Engineering

Abstract

A Software Upgrade of the NASA Aeroheating Code "MINIVER"

by

Pierce Mathew Louderback

Advisor: Daniel R. Kirk, Ph.D.

Computational Fluid Dynamics (CFD) is a powerful and versatile tool simulating fluid and thermal environments of launch and re-entry vehicles alike. Where it excels in power and accuracy, however, it lacks in speed. An alternative tool for this purpose is known as MINIVER, an aeroheating code widely used by NASA and within the aerospace industry. Capable of providing swift, reasonably accurate approximations of the fluid and thermal environment of launch vehicles, MINIVER is used where time is of the essence and accuracy need not be exact. However, MINIVER is an old, aging tool: running on a user-unfriendly, legacy command-line interface, it is difficult for it to keep pace with more modern software tools.

Florida Institute of Technology was tasked with the construction of a new Graphical User Interface (GUI) that implemented the legacy version's capabilities and enhanced them with new tools and utilities. This thesis provides background to the legacy version of the program, the progression and final version of a modern user interface, and benchmarks to demonstrate its usefulness.

Table of Contents

Abstract iii

List of Figures vii

List of Tables ix

List of Abbreviations x

List of Symbols xi

Acknowledgment xii

Chapter 1: Introduction 1

1.1 Background 1

1.1 Motivation 3

1.2 Objectives 4

1.3 Approach 4

1.4 Thesis Overview 4

Chapter 2: MINIVER Program Overview 6

2.1 Program Background 6

2.2 The Preprocessor 7

2.3 The Processor 11

2.4 EXITS Postprocessor 14

Chapter 3: MINIVER Upgrades 15

3.1 Upgrade Phases 15

3.2 Initial Migration 15

3.3 MINIVER Main Screen 19

3.3.1 New Projects and Cases 21

3.4 Project Tree and Information Window 22

3.4.1 Project Properties 23

3.4.2 Case Output Setup 24

3.4.3 Preprocessor Data 26

3.4.4 Output Timing 27
3.4.5 Trajectory 28
3.4.6 Atmosphere 28
3.4.7 Heat Transfer Models 29
3.4.8 Transition Models 30
3.4.9 Crossflow 31
3.4.10 Flowfield / Pressure Options 32
3.4.11 Mass Injection 33
3.4.12 Time-Dependent Geometry 34
3.4.13 Initial Conditions 35
3.5 Trajectory Editor 35
3.6 CAD Editor 37
3.6.1 Controls 38
3.6.2 Point Generation 40
3.6.3 Editing Properties: Line 41
3.6.4 Editing Properties: Global 42
3.6.5 Editing Properties: Local 43
3.7 Additional Tools and Utilities 45
3.7.1 Generalized Delimited Importer 45
3.7.2 Case Copier 46
3.7.3 Microsoft Excel Summary Output 48
3.8 Directory Structure 49
3.8.1 Program Organization 49
3.8.2 Save File Organization 50
Chapter 4: Benchmarks 52
4.1 Fortran Output vs. C# Output 52
4.2 OREX Flight Test 53

4.2.1 OREX Background 54
4.2.2 Creating the OREX MINIVER Project 55
4.2.3 OREX Results 61
Chapter 5: Conclusions and Future Work 64
5.1 Thesis Conclusion 64
5.2 Future Work 65
References 66
Appendix 68

List of Figures

Figure 2.1: Sample screenshot of a PREMIN menu 8
Figure 2.2: Sample PREMIN output (truncated trajectory for brevity) 10
Figure 2.3: LANMIN Summary Output example 11
Figure 2.4: LANMIN Detailed Output example (Input Summary) 13
Figure 2.5: LANMIN Detailed Output example (Detailed Data) 13
Figure 3.1: MINIVER Main Screen, Phase 1 (C++) 16
Figure 3.2: MINIVER Preprocessor, Phase 1 (C++) 16
Figure 3.3: Example Preprocessor submenu (C++) 17
Figure 3.4: A difficult "go to" statement collection. 19
Figure 3.5: MINIVER 2.0 Main Screen 20
Figure 3.6: New Project dialog 22
Figure 3.7: New Case dialog 22
Figure 3.8: Project Properties screen 23
Figure 3.9: Case Output Setup screen 25
Figure 3.10: Preprocessor Data summary, Modern Format 26
Figure 3.11: Preprocessor Data summary, Legacy Format 27
Figure 3.12: Output Timing screen 27
Figure 3.13: Trajectory Editor screen 28
Figure 3.14: Atmosphere screen 28
Figure 3.15: Heat Transfer Method screen example, Eckert / Spaulding-Chi Flat Plate option 30
Figure 3.16: Crossflow screen example 32
Figure 3.17: Flowfield / Pressure Options screen 32
Figure 3.18: Mass Injection screen example 34
Figure 3.19: Time-Dependent Geometry screen 34

Figure 3.20: Initial Conditions screen 35
Figure 3.21: Trajectory Editor 36
Figure 3.22: CAD Editor, using OREX geometry as an example 38
Figure 3.23: Point Generation screen 40
Figure 3.24: Line Properties window 42
Figure 3.25: Global Properties window 43
Figure 3.26: Local Properties window example 44
Figure 3.27: Generalized Delimited Importer, invoked via the CAD Editor 45
Figure 3.28: Case Copier window 47
Figure 3.29: Example Excel output using OREX data 48
Figure 3.30: Sample formatted graph generated by the Excel output type 49

List of Tables

Table 3.1: Memory-saving Fortran programming constructs 18

List of Abbreviations

API	Application Programming Interface
C/C	Carbon / Carbon
CFD	Computational Fluid Dynamics
CSV	Comma-Separated Values
FiCW	Finite Catalytic Wall
FLUINT	Fluid Integrator
GUI	Graphical User Interface
JAXA	Japan Aerospace Exploration Agency
KSC	Kennedy Space Center
LaRC	Langley Research Center
NASA	National Aeronautics and Space Administration
NCW	Non-Catalytic Wall
OREX	Orbital Re-entry Experiment
SINDA	Systems Improved Numerical Differencing Analyzer
TPS	Thermal Protection System
VSL	Viscous Shock Layer

List of Symbols

C_P	Coefficient of Pressure

Acknowledgment

// placeholder; acknowledge people / things / etc here.

Chapter 1: Introduction

1.1 Background

One of the most challenging design problems within aerospace engineering involves prediction and management of frictional heating during the motion of a body through the atmosphere. For higher speeds, this becomes more pronounced, particularly for cases such as takeoff and reentry of spacecraft. Supersonic speeds permit the generation of shock waves in the flowfield around the craft, and hypersonic speeds amplify real gas effects to the point where the application of ideal gas equations become unreliable for determining flowfield properties near the surface of a body. Certain reentry conditions and upper atmosphere maneuvers even contribute radiation as a primary heat transfer mechanism, providing additional complexity to the environment. Due to the hostile environment that such bodies must endure during high speed flight and reentry, a method must be available to provide the engineer with heat transfer data to certain sections of the body, allowing for accurate design of a heat shielding mechanism, generally known as a Thermal Protection System (TPS). As weight is a premium on any flight, whether going to space or not, a middle ground must be obtained that maximizes thermal protection for the body but minimizes weight that can otherwise be used for payload. An accurate heat transfer analysis will be able to do this, but with what speed?

Modern computing allows the engineer the use of Computational Fluid Dynamics (CFD) to calculate detailed and accurate information of the flow of a fluid over a body, but it is a cumbersome process. A computational domain, in the form of a grid of nodes where calculations take place, is made in the desired region around the body to be considered. This grid can be simple for basic shapes, but becomes increasingly complex to create for unusual 2D and 3D geometries. The grid must be generated carefully, ensuring that the domain is well-represented and

will not cause errors in the output, thus requiring additional time to build. Once the grid is designed, the problem is set up through the application of known boundary conditions, and calculations may then be performed. Computation time is at the mercy of grid size and complexity, causing extremely accurate 3D solutions to take days or weeks to reach convergence, sometimes even in the presence of a parallel processing environment. While it is undeniable that CFD is a valuable contributor to the field in terms of the raw computational insight it provides the engineer, it does not fit well into the early stages of project engineering which requires for estimations and design iterations to provide proof of concept for new vehicles.

As such, a faster program is necessary to bridge this gap. To demonstrate, consider a reentry vehicle that is in the early stages of development. Given a certain payload for reentry and a size and weight restriction based on the equipment that launches it in the first place, a design must be decided upon that will ensure the payload survives reentry. Many factors are taken into account here: size, shape, trajectory, and the material of the thermal shield itself. Since many of these can be modified to increase or decrease the heat that the vehicle receives during reentry, it is desirable to perform fast calculations that can roughly estimate the impact of these parameterizations on the heating environment of the vehicle.

MINIVER is one such program. Using theoretical and empirical correlations, the program will take in trajectory points, body information, and atmospheric data to formulate fluid property calculations in the flowfield at the body and provide heat loads and fluxes at the surface for use in thermal protection design. By setting up cases, MINIVER can provide time- and geometry-dependent data on heat, informing the engineer at what parts during the trajectory that the vehicle will take the greatest heat rates and what points on the vehicle should be the most shielded. In contrast to CFD, the setup and run times are on the order of minutes yet the accuracy is usually well within tolerance, allowing for quick parameterization of the project's design that can narrow down cases for a much more thorough, CFD-based analysis.

1.1 Motivation

MINIVER was developed in the 1960s using the state of the art computational resources of the time. This is not to suggest that the code is outdated; the engineering methods used in its routines are as relevant now as they were when the program was created. However, advances in computing technology have made significant leaps through the decades, and MINIVER's base functionality has needed to be upgraded. What started as punch card style interfacing was replaced with command line style input and output in a previous update, and now the time has come to modernize the interface.

MINIVER's most recent interface is based in Fortran, using a set of programs to operate its preprocessor (known as PREMIN), its processor (LANMIN), and its structural solver (EXITS). On program execution, a command line window opens which allows for a step-by-step sequence through the preprocessor, allowing the user to set up timing, trajectory, heat transfer methodology, and so on. Input errors force the user to start entire sections over, and sometimes the entire program as well, just to fix a value that was added incorrectly. Some input methods are rather cumbersome, particularly involving tabulated data such as trajectory and custom atmospheric data. There is little to no error handling in the preprocessor, allowing the user to input letters or negative values without any kind of dialog warning them of the mistake. Some options in the preprocessor have also exhibited bugs in their operation. Attempting to write the trajectory to a file, for example, forces the program into an unbreakable loop where the interface asks for the name of the file, but does not save the file or allow the user to continue.

With the presence of bugs and an interface that does not provide effective user interaction, it is deemed necessary to give the program an update to fix errors and make it easier to use. Additional features can also be implemented to increase the program's base functionality, making MINIVER a more desirable program to use in the presence of other, similar aerothermal heating software.

1.2 Objectives

To upgrade the MINIVER program as a whole, the following objectives are considered the primary goals of this thesis:

1) Translate the code from its Fortran language to a modern language
2) Become familiar with the Fortran version to isolate known bugs and correct them in the new version
3) Build a Graphical User Interface (GUI) that increases user-friendliness and streamlines preprocessor input for the user
4) Perform benchmark tests to ensure that the new version output does not deviate from the legacy version

1.3 Approach

The development language had to be chosen between proposed languages C++ and MATLAB. Initial investigation suggested that not all users may possess MATLAB, and that a C++-designed code would run on any Windows computer, C++ was decided as the development language. Later experience would demonstrate that C# would be a more apt language for the development of the GUI due to its strong Application Programming Interface (API) and the inclusion of the .NET Framework libraries with Windows Forms. The code would be developed by transferring the LANMIN processor environment into the modern language. This included the conversion of the processor framework into an object-oriented environment as well as removing archaic structures such as "go to" statements. Once the code translation was completed, the PREMIN preprocessor environment was regenerated in the modern language by duplicating the purpose of the Fortran subroutines and generating a GUI which could adequately handle the expected parameters.

1.4 Thesis Overview

This thesis begins by detailing the MINIVER program and its history. Chapter 2 explains the origins of the programs, known major updates made over the decades, and the routines performed by the three subprograms: PREMIN, LANMIN, and EXITS. In Chapter 3, upgrades to the program are detailed, starting with the original code translation from Fortran, then into the design of the GUI. Improvements made to certain subroutines are explained, along with fixes to bugs that were observed in the Fortran version. Chapter 4 discusses benchmarks that have been performed to verify that the results and predictions given by the new version do not deviate from the output of the old version. The thesis concludes with Chapter 5, marking conclusions of the work completed and detailing future work that can further improve the program.

Chapter 2: MINIVER Program Overview

2.1 Program Background

The origin of the MINIVER program dates back to the 1960s with a program known as the JA70 Aerodynamic Heating Code. This later evolved into "A Miniature Version of the JA70 Aerodynamic Heating Code," or MINIVER for short. The conversion to MINIVER is referenced in a paper written by D.R. Hender [1] in 1970. This paper is unfortunately unobtainable due to its proprietary nature, but is likely little more than the original code and a user's manual. The original input used punch cards, setting up an array with a maximum of 700 entries that defined what MINIVER needed to analyze. The entire code appeared to be condensed into a single program whose main routine was known as H800. Subroutines within the program were designed to perform flowfield calculations based on atmospheric data including shock wave effects, apply known engineering heat transfer methods for certain body shapes, and take transition effects into account. It appears as though the program has been disseminated to multiple agencies and upgraded/reprogrammed to the needs of their engineers, but this thesis will keep the discussion to the known iterations leading to this upgrade and its usage at NASA Kennedy Space Center (KSC).

In 1983, a major upgrade was performed by multiple engineers at REMTECH, Inc for NASA Langley Research Center, documented in a report broken into three volumes [2] [3] [4]. The upgrade included several software upgrades including new heat transfer routines, but the primary focus was on coupling MINIVER with the LaRC AVID system, which involved giving the program a user input system based on a command prompt, rather than having to use the previous system of punch cards. From this upgrade, MINIVER became three separate programs: PREMIN, LANMIN, and EXITS. PREMIN is the command-line preprocessor, allowing for interactive input from the user to build an input file

that sets up the 700 element array used in the LANMIN processor. LANMIN is Langley's version of MINIVER, performing the necessary computations to produce output files of varying detail as requested by the user. Previously, MINIVER had heat conduction subroutines to detail the effects of the calculated heat transfer upon any TPS structure used by the intended vehicle. The 1983 update separated this from the main routines and condensed it into the third program known as EXITS.

Throughout the years, branches of NASA have modified the source code to better serve their own needs, and it is unknown how much the MINIVER program has diverged as a result.Code comments in the program provided by NASA KSC indicate that small changes have been made over time to this particular version. As the three programs from the 1983 upgrade are core to MINIVER, the next few sections will briefly explain their functionality.

2.2 The Preprocessor

MINIVER's preprocessor is a simple program that allows text-based user input to set up a case file relevant to the vehicle being considered. Since it can be cumbersome to set up the 700 element array that LANMIN uses for calculations (known as the "W Array"), PREMIN streamlines this process into a set of menus tailored to certain aspects that will alter the heat transfer to the vehicle's body (see Figure 2.1 for a sample screenshot of MINIVER input). While the W array is capable of holding 700 elements, it never fills out every element; it is just a memory construct that is capable of holding data for any possible case setup. Major sections of the W array are composed of data tables such as trajectory information, which takes up a maximum of 200 entries (50 each for time, altitude, velocity, and angle of attack). To fill up this array with pertinent data, a sample case setup in the preprocessor will request the following information from the user:

1) Whether to use English or Metric for data input
2) What time intervals to use for data printout

3) Vehicle trajectory, input manually or from a file import
4) Which atmosphere model to use
5) Which heat transfer model to use
6) How to consider flow transition
7) Whether to consider crossflow or not
8) What type of shock wave and local pressure to consider
9) If the surface geometry changes over time
10) How the wall temperature should be set
11) How the user wants the output file to be generated

```
trajectory input is complete


    atmosphere data

options     1.  1962 u.s. standard atmosphere
            2.  wind tunnel option
            3.  input atmospheric data(alt,t-inf,p-inf)
            4.  1963 patrick air force base atmosphere
            5.  1971 vandenberg reference atmosphere
            6.  1973 vandenberg hot day atmosphere
            7.  1973 vandenberg cold day atmosphere
            8.  1971 kennedy hot day atmosphere
            9.  1971 kennedy cold day atmosphere
           10.  1976 u.s. standard atmosphere

 option selected ?
10

 1976 u.s. standard atmosphere

 is this option correct ?
y


 do you want to run a heating indicator ?
n


     heat transfer method

 options     1.  hemisphere stagnation point
             2.  cato/johnson swept cylinder
             3.  eckert ref. enthalpy flat plate method
             4.  eckert/spaulding-chi flat plate method
             5.  boeing rho-mu flat plate method
             6.  beckwith/gallagher swept cylinder method
             7.  boeing rho-mu swept cylinder method
             8.  lees/detra-hidalgo hemisphere distribution
             9.  leeside orbiter heating
            10.  flap reattachment heating
            11.  fin-plate peak interference heating
            12.  brake payload impingement heating
```

Figure 2.1: Sample screenshot of a PREMIN menu

Many options have deeper functions that will run if relevant to the task at hand. Not having a trajectory file, for instance, will force a line-by-line editor that allows the user to build a trajectory manually. Activating the Wind Tunnel Option in the atmosphere data section will allow the user to enter static temperature and pressure as a function of time, which is then saved into its own section in the W array. Each heat transfer option has its own subset of data inputs that are required to accommodate the chosen method. Overall, PREMIN is very thorough in ensuring that all necessary information is requested before preparing the output file.

Structurally, PREMIN is composed of a main routine and eighteen subroutines, with each subroutine focused on input parameters such as the trajectory input and choosing the heat transfer method. The MAIN subroutine directs the flow of the program, linking into subroutines to fill out data as necessary before returning back to MAIN. When PREMIN is finished with its data entry, is creates a file for use with LANMIN. An example can be seen below in Figure 2.2.

```
 1  miniver sample case
 2                0.000          25.000        2000.000
 3              200.000        5000.000           0.000
 4                0.000
 5               50.000
 6                0.300      396300.000       24570.000   41.130   0.000
 7               45.300      373800.000       24590.000   41.260   0.000
 8               90.300      351500.000       24620.000   41.210   0.000
 9              135.300      329500.000       24640.000   40.460   0.000
10              180.300      308000.000       24660.000   40.650   0.000
11              225.300      288000.000       24660.000   41.900   0.000
12              270.300      269700.000       24610.000   39.390   0.000
13              315.300      256300.000       24470.000   40.710   0.000
14              360.300      250200.000       24220.000   41.740   0.000
15              405.300      247000.000       23920.000   39.950   0.000
16              450.300      244600.000       23610.000   39.280   0.000
17              495.300      242500.000       23280.000   39.660   0.000
18              545.300      240000.000       22880.000   39.070   0.000
19              569.300      238800.000       22680.000   39.320   0.000
20              593.300      237700.000       22470.000   39.020   0.000
21              617.300      236600.000       22250.000   39.250   0.000
22              641.300      235300.000       22020.000   39.480   0.000
23              665.300      233700.000       21780.000   39.970   0.000
24              689.300      232000.000       21530.000   40.010   0.000
25              713.300      232300.000       21260.000   40.250   0.000
26              737.300      230300.000       20980.000   40.460   0.000
27              761.300      227900.000       20680.000   40.140   0.000
28              809.300      222300.000       20020.000   40.160   0.000
29    9  1.00    10  10.0    11  3.00    13  20.0    15  3.00
30   16  2.00    23 0.850    27  9.00    31  36.0    32  16.0
31   37  10.0    38  10.0   261  1.00   262  2.00   642  1.00
32  611  con20
33  643  1.00   647  1.00   641  2.00
34
```

Figure 2.2: Sample PREMIN output (truncated trajectory for brevity)

The PREMIN output file is broken up into three major sections: Title/Timing, Trajectory, and Remaining Options. The Title/Timing section contains the title ("miniver sample case" as in Figure 2.2) and the output timing which LANMIN will use to generate output data. In the example given, the output timing is 0 sec to 2000 sec with a 25 sec interval, followed by 2000 sec to 5000 sec with a 200 sec interval. The Trajectory section contains the trajectory that LANMIN will use for its flow calculations. The file first indicates how many trajectory points exist (50 points in the example) and then lists the trajectory line by line with time, altitude, velocity, angle of attack, and yaw angle as input data. The Remaining Options section serves to input the W Array values one by one. For instance, W(9) is set to 1.00, W(10) is set to 10.0, and so on. All unlisted options

are initialized to zero. With this file, the program flow now proceeds to the LANMIN processor to perform engineering calculations.

2.3 The Processor

The LANMIN Preprocessor performs all the major calculations to determine heating information at the point of interest. This is carried out through an extensive amount of code comprising a main routine and fifty-eight subroutines. These subroutines include the multiple heat transfer methods, atmospheric routines, transition capabilities, flowfield calculations, etc that are used in every case.

LANMIN's MAIN routine directs the flow of the program first by reading in PREMIN's output data file. It sets the W Array using these values, then initiates a calculation loop that iterates through the timing parameters. At each point in time, the program utilizes interpolation and atmospheric subroutines to determine the atmospheric data as dictated by the atmospheric model and the trajectory input. The program then takes this data and determines freestream properties of the flowfield through use of flow subroutines. Corrections are made for crossflow and virtual origin correction, then the data is sent to the heating method chosen by the user. Mass injection and time-dependent geometry is handled next, followed by flow transition modifications, and then finishing with output generation. Two types of output files can be generated by LANMIN, known as the "Summary" and "Detailed" printouts.

miniver sample case							con200.0				
0.0	396.3	24570.0	27.47	41.13	2.144E+00	2.731E-05	1.158E+04	475.9	3.132E-01	0.000E+00	1.474E-02	lam
25.0	384.0	24581.0	27.48	41.20	3.771E+00	3.437E-05	1.158E+04	531.0	3.903E-01	8.794E+00	2.360E-02	lam
50.0	371.5	24593.1	27.49	41.25	7.383E+00	4.520E-05	1.157E+04	600.7	5.123E-01	2.008E+01	4.104E-02	lam
75.0	359.1	24609.8	27.51	41.23	1.619E+01	6.201E-05	1.158E+04	687.4	7.019E-01	3.525E+01	7.836E-02	lam
100.0	346.8	24624.3	27.53	41.05	3.391E+01	8.516E-05	1.158E+04	781.4	9.620E-01	5.605E+01	1.501E-01	lam
125.0	334.5	24635.4	27.54	40.63	6.857E+01	1.178E-04	1.155E+04	884.8	1.325E+00	8.464E+01	2.850E-01	lam
150.0	322.5	24646.5	27.55	40.52	1.370E+02	1.644E-04	1.155E+04	1000.5	1.845E+00	1.243E+02	5.498E-01	lam
175.0	310.5	24657.6	27.57	40.63	2.708E+02	2.285E-04	1.157E+04	1125.2	2.561E+00	1.793E+02	1.071E+00	lam
200.0	299.2	24660.0	27.57	41.20	5.055E+02	3.079E-04	1.159E+04	1247.7	3.449E+00	2.545E+02	2.020E+00	lam
225.0	288.1	24660.0	27.57	41.89	9.231E+02	4.151E-04	1.162E+04	1379.5	4.645E+00	3.556E+02	3.762E+00	lam

Figure 2.3: LANMIN Summary Output example

The Summary Output, as shown in Figure 2.3, generates unlabeled columns of information that the user could quickly input into, for instance, an Excel spreadsheet for further manipulation. This data includes, from left to right: time, altitude, velocity, mach number, angle of attack, Reynolds number per unit length, heat coefficient, recovery enthalpy, radiation equilibrium temperature, heat rate, heat load, pressure, and flow type.

The Detailed Output provides significantly more data, but is more difficult to convert into a tabular format. This output file is split into three sections: Input Summary, Detailed Data, and Summary Data. Part of the Input Summary is shown in Figure 2.4, which is just a section that echoes the input data which was entered during with the PREMIN program. The Detailed Data section is shown in Figure 2.5, and provides a wealth of data to the user at every time point, including different classifications of property data (such as freestream, wall, edge, etc) and breakdowns of heat transfer (convection, radiation, net). The Summary Data section provides a reiteration of what would be in the Summary Output file, except it provides headers indicating what the entry is and what its units are.

```
                ***PC miniver 2 -- version lanmin91 (9/16/08)***


1

                          miniver sample case

 timing
      t1          =     0.000 sec.
      dt1         =    25.000 sec.
      t2          =  2000.000 sec.
      dt2         =   200.000 sec.
      t3          =  5000.000 sec.
      dt3         =     0.000 sec.
      t4          =     0.000 sec.

 control parameters
      w(641)=       2.000       w(642)=       1.000
      w(643)=       1.000       w(644)=       0.000
      w(645)=       0.000       w(646)=       0.000
      w(647)=       1.000       w(648)=       0.000
      w(649)=       0.000       w(650)=       0.000

 heat transfer
      ht method =     3.000
      rn        =     0.000 ft.
      l         =    20.000 ft.
      n sub l   =     3.000
      n sub t   =     2.000
      phi       =     0.000 deg.
      pitch pl  =     0.000
```

Figure 2.4: LANMIN Detailed Output example (Input Summary)

```
time =     0.000  m inf= 27.468  p inf = 0.497E-04  t inf =    205.31  h inf =    159.60  rho i = 0.393E-10  mu inf = 0.450E-06
z inf =  396300.  m u  = 27.468  p u   = 0.497E-04  t u   =    205.31  h u   =    159.60  rho u = 0.393E-10  mu u   = 0.450E-06
v inf =   24570.  m e  =  3.174  p e   = 0.147E-01  t e   =   6787.17  h e   =   8017.84  rho e = 0.772E-09  mu e   = 0.205E-05
a inf =     895.  l    = 20.000  p t   = 0.107E+01  t t   =   8767.16  h t   =  12208.47  rho t = 0.387E-07  mu t   = 0.251E-05
v e   =   14490.  rn   =  0.000  alpha =    51.130  t *   =   6068.71  h *   =   4847.75  rho * = 0.100E-08  mu *   = 0.188E-05
reinf =2.144E+00  pr   =  0.710  cf/2  = 0.600E-01  t r   =   -459.67  h r   =      0.00  rho r = 0.000E+00  mu r   = 0.000E+00
rel   =1.089E+02  ems*f=  0.850  tau w = 0.972E-02  t w   =      0.00  h w   =    110.35  rho w = 0.167E-07  mu w   = 0.338E-06
                                                     le  =      0.00  phi   =      0.00  para1 =.00000E+00  para2 =.00000E+00

                1         2         3         4         5         6         7         8         9
ff flag      36.       16.        0.        0.        0.        0.        0.        0.        0.
angle     51.13     51.13      0.00      0.00      0.00      0.00      0.00      0.00      0.00

nc l = 0.273E-04  hrecov l=  11579.88  qc l =  0.31E+00  k sub l 2 =  1.000  k sub l 3 =  1.000  k sub l =  1.000  para =     0.00
nc t = 0.211E-04  hrecov t=  11755.68  qc t =  0.25E+00  k sub t 2 =  1.000  k sub t 3 =  1.000  k sub t =  1.000  pct t=    0.000

n sub c= 0.2731E-04 h recov=  11579.88   q conv  = 0.3132E+00  q rad = 0.1806E-01  q net = 0.2952E+00
h ideal= 0.1549E+00 t recov=   7278.18   qc total= 0.0000E+00  qr tot= 0.0000E+00  qn tot= 0.0000E+00  t rad eq=     475.94

                                         qc cw   = 0.3162E+00  qc cwt= 0.0000E+00
```

Figure 2.5: LANMIN Detailed Output example (Detailed Data)

2.4 EXITS Postprocessor

The EXITS Postprocessor is a simple heat transfer program that allows the user to generate a TPS, apply the LANMIN output heat profile, and determine the heat distribution throughout the TPS. A library of materials is available to the user for use with EXITS, including common TPS materials and their heat transfer properties. The user can also specify certain behavior of the layers of the TPS, such as whether it behaves as a slab, thin skin, ablator, etc. NASA KSC prefers to utilize more modern software to perform this activity, such as SINDA/FLUINT and Thermal Desktop. As such, an upgrade of this section of MINIVER was omitted from the current work effort.

Chapter 3: MINIVER Upgrades

3.1 Upgrade Phases

The upgrade effort for the MINIVER upgrade was divided into two phases. The first phase saw the initial migration to a modern programming language and a demonstration that a GUI could be used to perform Legacy MINIVER's actions with increased ease. The second phase saw a design shift from C++ to C# based on employment experience, utilizing the strong Windows Forms and .NET Framework libraries available to C#. This phase included a major design shift for the Preprocessor and user interface as a whole, and additionally incorporated a standalone Trajectory Editor and a CAD Editor, as well as additional tools which improved user input and import capabilities. Phase 2 also included new atmospheric models, benchmark cases, and a robust User's Manual to help Legacy users and new users alike. This thesis comes after the completion of Phase 2. An Option phase is under consideration by NASA for additional improvements, with a focus on a connection to SINDA/FLUINT and Thermal Desktop in the interest of providing a swift connection to their most common thermal design tools.

3.2 Initial Migration

The initial migration phase was largely an explorative process, since initially there was little knowledge of how to program in C++ or how to build a GUI. This design started with a main screen (Figure 3.1) which allowed the creation of a "Project" which could contain multiple "Cases" within. In essence, a Project can be thought of as a re-entry vehicle, such as an Apollo module. A Case can be thought of as a point on that vehicle, such as the stagnation point at the front of the vehicle, or any point along the windward running length of the vehicle. After creating a project, the user then proceeds to create a case for the point of interest.

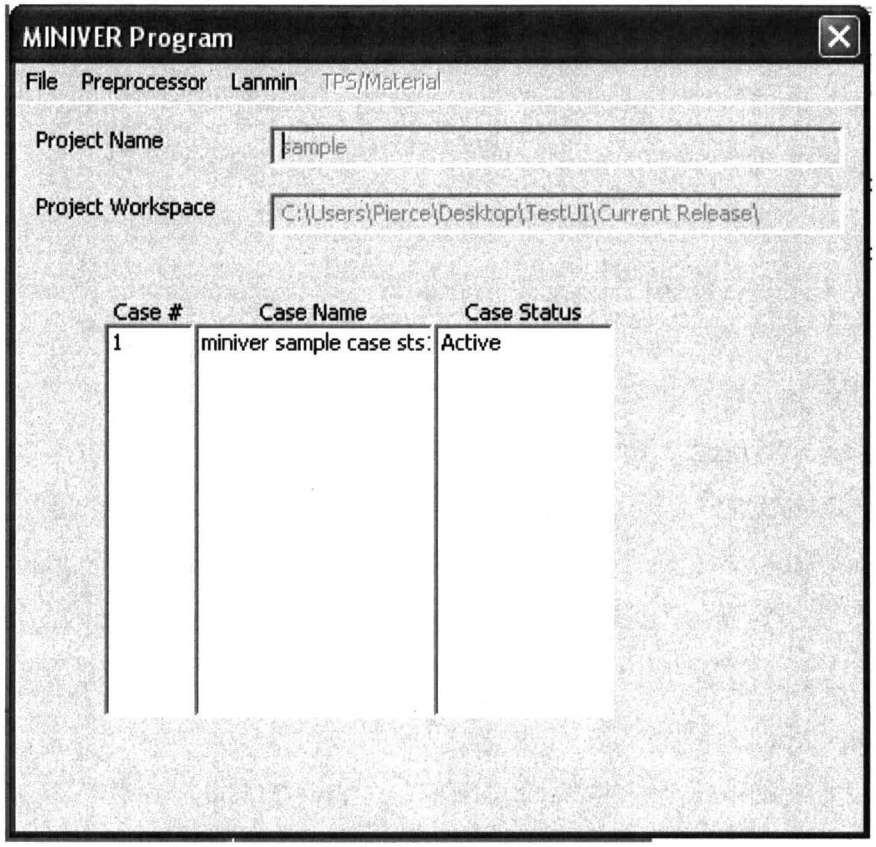

Figure 3.1: MINIVER Main Screen, Phase 1 (C++)

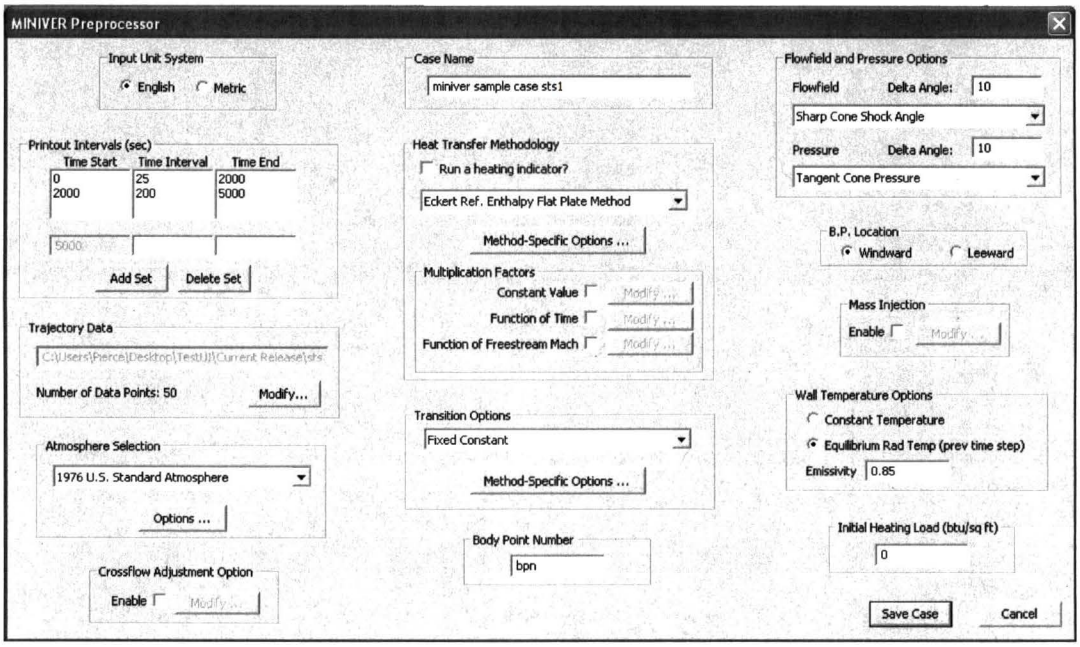

Figure 3.2: MINIVER Preprocessor, Phase 1 (C++)

The case dialog is shown above in Figure 3.2, which is an amalgam of all PREMIN input capabilities on one screen. It provided rudimentary error-checking, validating sections based on requirements of the section (e.g. values must be positive, values must be numeric, etc) and would throw a warning before attempting to save that the case was not yet ready for computation in the processor. Additional option menus would be invoked for more complex menus such as Heat Transfer Options and Transition Options, an example of which is shown below in Figure 3.3. By filling out the case dialog and returning to the main screen, the user could quickly send the data to the processor by choosing the Lanmin → Run Calculations menu item, and the LANMIN output files would be generated.

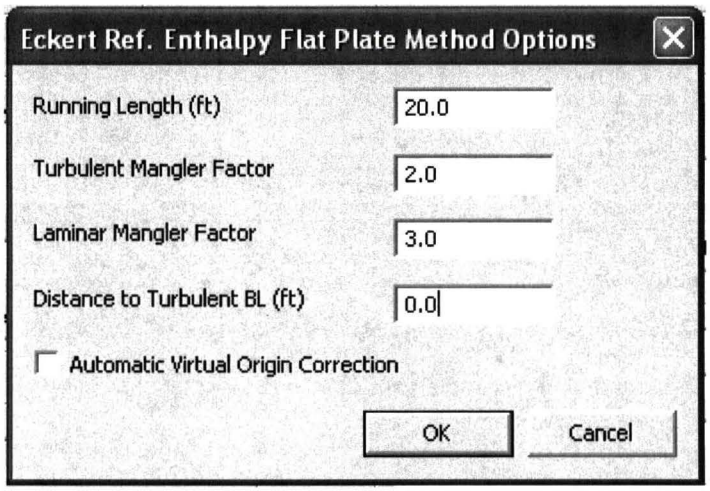

Figure 3.3: Example Preprocessor submenu (C++)

The processor itself posed a few interesting challenges in terms of conversion from Fortran to C++. While familiar with Fortran 90/95, Legacy MINIVER was written in a significantly older version of Fortran. In addition, it utilized programming constructs that were no longer standard and, in some cases, were considered unacceptable practice. Table 3.1 presents three of the most common of these constructs. At the time of the original programming, however, these constructs allowed less code to be written, helping to save memory back when computers did not possess much to begin with.

Table 3.1: Memory-saving Fortran programming constructs

Programming Construct	Example Code
Equivalence statement	`equivalence (w(11),htflag)`
Common memory block	`common/frstm/tfrstm,pfrstm,dfrstm`
Goto statement	`if(jfcp.le.0) go to 889`

Equivalence statements were easy to convert, simply by choosing a single variable to represent both. The main difficulty with performing the conversion was realizing that both variables represented the same memory block and thus, changing one changed the other. Observing this usage of memory provided some debugging difficulties at the beginning, trying to figure out how a variable ended up changing when not explicitly being called, instead being changed because the other variable was reassigned.

Common memory blocks required a greater amount of effort to convert because their declaration treated them as "global" variables, which could be altered at any point in the program. When performing the conversion, these were redesigned as properties of a data object that encompassed the intermediate properties of the case. As such, they lost their global status and required passing the object for manipulation.

The most difficult construct in the Legacy code was the prevalence of **go to** statements (with 506 uses in the LANMIN code alone) which resulted in some creative conversions. A particularly difficult example is shown in Figure 3.4, which is a small sample from a shock wave calculation subroutine. To perform this conversion, the entire subroutine was mapped out as a flowchart, and then re-condensed into an **if-else** statement structure that could be applied in C++ without the use of **go to** statements.

```
if( x .lt. 1.) go to  500
if(y .le. 0.) go to 510
if( y.gt.60. .and.((itable.eq. 2).or.(itable .eq.4)))go to 520
if ( y.gt.55. .and. ((itable .eq. 1).or.(itable .eq. 3)))go to 520
go to (10,30,50,70),itable
if((x.le. 1.7).and.(y.le.16.))go to 101
if((x.gt. 1.7).and.(x.le.2.8).and.(y.le. 32.))go to 102
if (x.le.2.8) go to 520
if(y.le. 16.) go to 103
if(y.le. 35.) go to 104
if((x.gt. 3.4) .and. (y.le. 45.))go to 105
if (x .gt. 8.) go to 106
go to 520
if((x.le. 1.5).and. (y.le. 28.))go to 101
if((x.le.2.8).and.(x.gt. 1.5).and.(y.le.48.))go to 102
```

Figure 3.4: A difficult "go to" statement collection.

With the LANMIN processor routines converted to C++ and the PREMIN preprocessor given a rudimentary GUI, the first phase of the project came to a close. While this phase produced a capable replacement to Legacy MINIVER, more professional experience was obtained between the two phases, including an introduction to the C# programming language. It was quickly ascertained that the program could benefit from a redesign.

3.3 MINIVER Main Screen

The first goal of the C# redesign was a new, more streamlined user interface. The clunky, grouped case dialog from the C++ version was too messy, ugly, and ill-centered. The new main screen, shown in Figure 3.5, was redesigned by adding a toolbar to quickly create new projects and cases. A tree view was added with the goal of organizing the project file, providing categorical separation of preprocessor input, and providing a fast method to edit properties of the case. The information screen, to the right of the tree view, is context-sensitive and updates its information based on the item selected in the tree view. The capabilities of the MINIVER main screen's menu system are as follows:

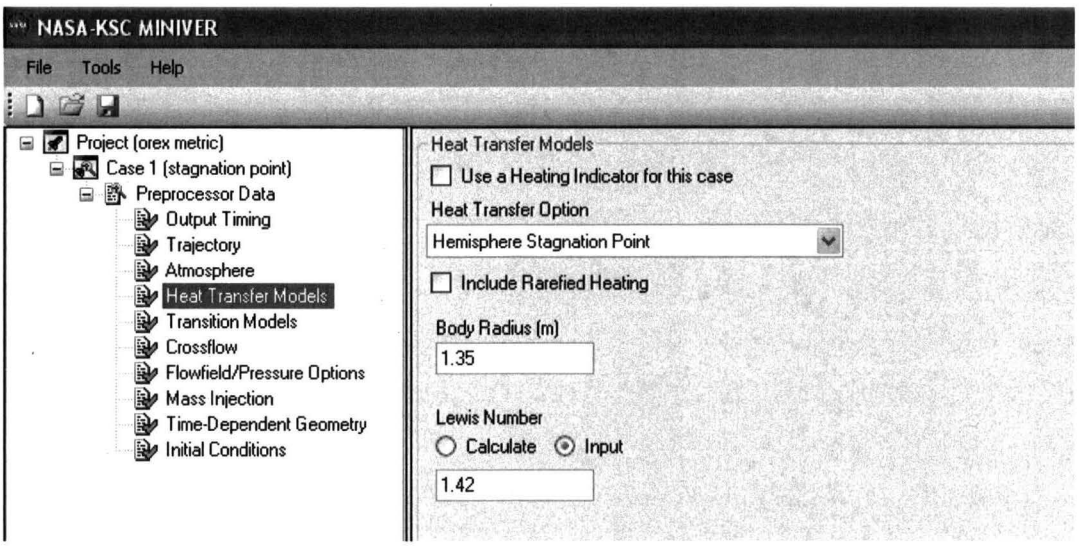

Figure 3.5: MINIVER 2.0 Main Screen

- File Menu
  - **New → Project…**: Creates a new Project for use with MINIVER.
  - **New → Case…**: Creates a new Case for use with the MINIVER Project. Cases cannot be generated until a Project is open.
  - **Open → Project…**: Opens a previously created Project for use with MINIVER.
  - **Open → Case…**: Opens a previously created Case for use with the current Project.
  - **Delete Case**: Deletes the case currently selected in the Project Tree.
  - **Close Project**: Closes the current Project.
  - **Save All**: Saves the Project file and all Case files.
  - **Exit**: Closes the program.
- Tools
  - **Trajectory Editor**: Opens the Trajectory Editor externally from the current Project and Cases. Useful for creating and editing Trajectories for later use.
  - **CAD Editor**: Opens the CAD Editor, which will import basic geometry for use in creating a MINIVER Project.

- Help
  - **About MINIVER...**: Opens an About page, containing version and date information regarding this version of MINIVER.
- Tool Strip
  - **New Project**: Creates a new Project.
  - **Open Project**: Opens a previously created Project.
  - **Save All**: Saves the Project file and all Case files.
- Project Tree
  - **Project Icon**: Indicates a Project, which can be abstractly thought of as a vehicle of some kind. Only one Project can be open at a time.
  - **Case Icon**: Indicates a Case, which contains information regarding a point on the vehicle.
  - **Preprocessor Data Icon**: A level which contains all Preprocessor data in sections underneath.
  - **Preprocessor Data Section Icons**: These icons represent Preprocessor Data Sections, which are categorized based on their importance to the Case being modeled. Green means a section has enough input, Red means it has not been started, and Yellow means it has information but is not quite complete.
- Information Box
  - Selecting nodes in the Project Tree will update the Information Box with context-sensitive information regarding the Project and its Cases.

3.3.1 New Projects and Cases

When a new project or case is requested, the program will create a dialog box to collect information required to create the requested item. Figure 3.6 shows the New Project dialog.

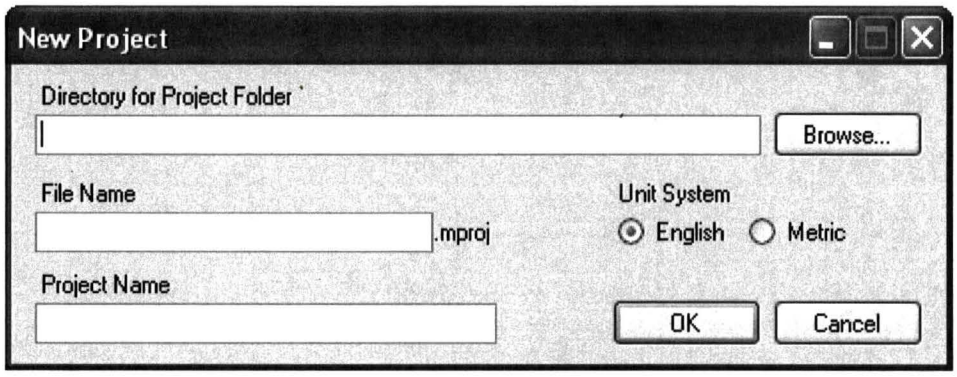

Figure 3.6: New Project dialog

- **Directory for Project Folder**: The location where the Project Root Directory will be saved.
- **File Name**: The filename of the project file. This will be the name of the Project Root Directory.
- **Project Name**: A user-specified identity for the project, which will be referenced in the Project Tree.
- **Unit System**: Selects whether to use English or Metric units for the Project. This unit system will determine input units; output units will be determined separately.

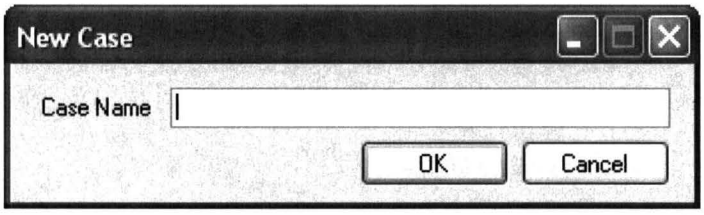

Figure 3.7: New Case dialog

- **Case Name**: Defines the name of the case and creates a file based off of this name in the Cases directory (\Cases\<Case Name>.mcase)

3.4 Project Tree and Information Window

This area of the main screen provides the ability to access project and case information and alter them as the user wishes. Items chosen in the Project Tree are displayed in the Information Window for review. Most settings will invoke a pair

of buttons in the lower right corner, indicating whether the user wishes to save or revert the changes made to the control in the Information Window.

3.4.1 Project Properties

The Project Properties screen (Figure 3.8) allows the user to change the name of the project, view its properties, and perform extended output operations. This screen is particularly helpful with running a large number of cases, as it provides a rundown of every case associated with the project, and provides controls with which to perform processor runs. One of the most important features here is the Global Comparison capability, which allows the user to plot data between cases. An example of this will be seen later with the OREX example.

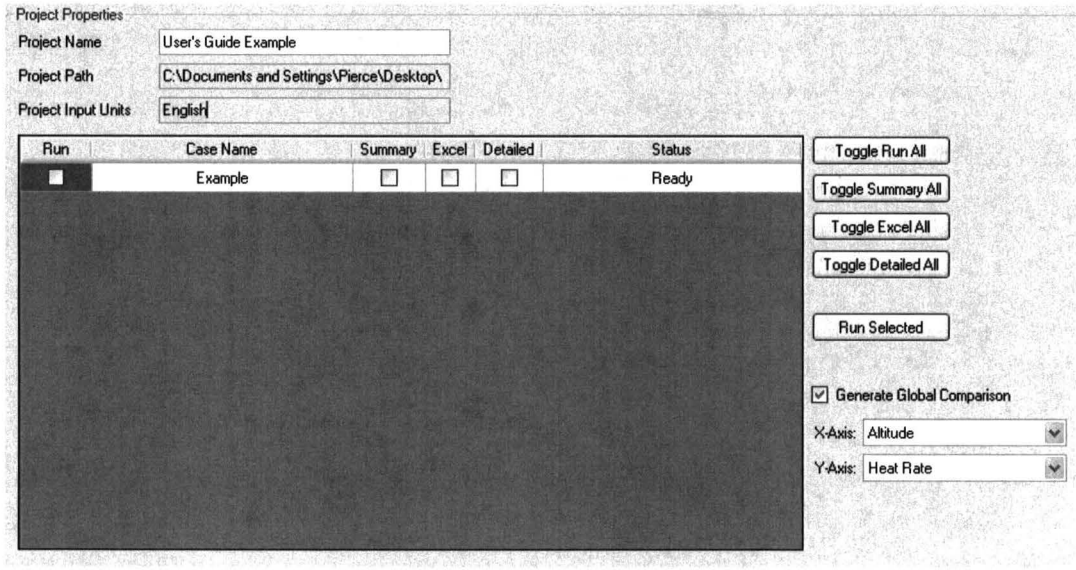

Figure 3.8: Project Properties screen

- **Project Name**: Allows the user to change the Project Name.
- **Project Path**: Indicates the location on the computer where the Project is stored.
- **Project Input Units**: Indicates the unit system that all input will be entered in.
- **Extended Output Table**
    - **Run Column**: Whether or not to run the case in that row.

- **Case Name Column**: The name of the case that is to be operated on.
- **Summary Column**: Whether or not to generate a Summary Output for the row.
- **Excel Column**: Whether or not to generate an Excel Output for the row.
- **Detailed Column**: Whether or not to generate a Detailed Output for the row.
- **Status**: Indicates if the case is awaiting runtime or if it is in the Processor.

- **Toggle Run All**: Toggles all checkboxes in the Run column on and off.
- **Toggle Summary All**: Toggles all checkboxes in the Summary column on and off.
- **Toggle Excel All**: Toggles all checkboxes in the Excel column on and off.
- **Toggle Detailed All**: Toggles all checkboxes in the Detailed column on and off.
- **Run Selected**: Sends all selected items to the Processor.
- **Generate Global Comparison**: Generates a Global Comparison CSV file. This will plot the Y-Axis of all selected cases against one X-Axis. It expects all cases to have identical timing parameters.

3.4.2 Case Output Setup

Alternative to the Project Properties send to processor capability, the Case Output Setup screen (Figure 3.9) allows the user to perform a single case run. The user can also set the output unit system, whether to generate a Summary Output, a Detailed Output, or an Excel Output. The Excel Output serves as an extension of the Summary Output, opening a COM interop channel with Microsoft Office Excel, then creating and formatting a table, complete with graphs. For examples of the Excel output, refer to the Additional Tools section.

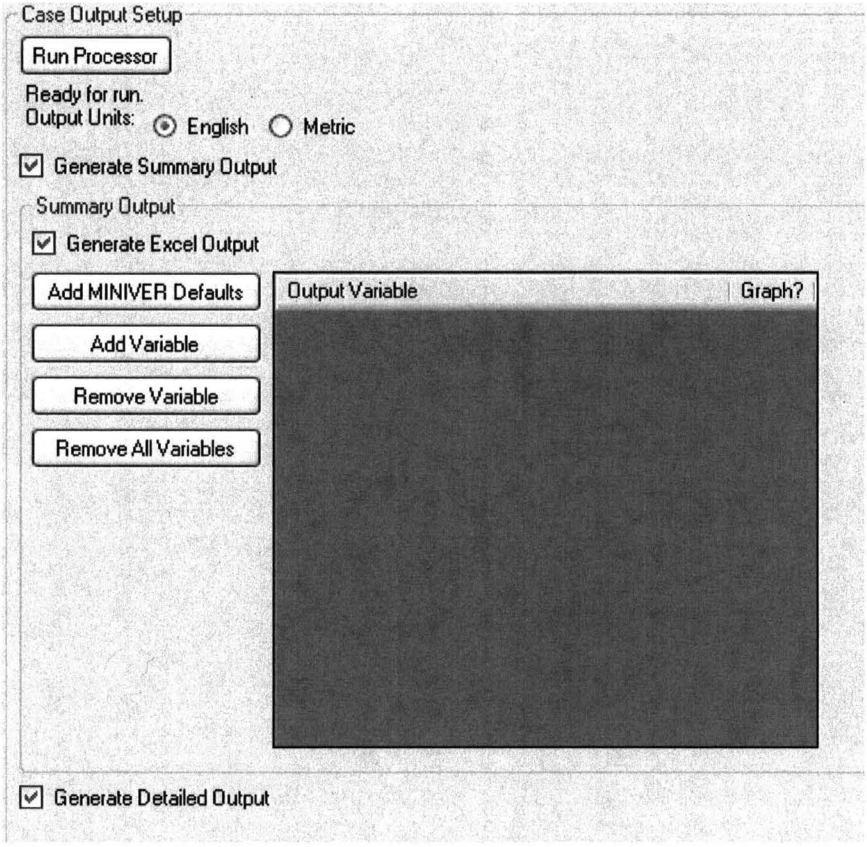

Figure 3.9: Case Output Setup screen

- **Run Processor**: Sends the case data to the Processor to generate output. The label underneath will update once the process is complete.
- **Output Units**: Specifies whether the Summary and Excel outputs are in English or Metric.
- **Generate Summary Output**: If selected, MINIVER will generate a summary of important variables over the timeframe indicated in Output Timing: Time, Altitude, Velocity, Mach Number, Angle of Attack, Reynolds Number per Length, Heat Coefficient, Recovery Enthalpy, Radiation Equilibrium Temperature, Heat Rate, Heat Load, Pressure, and Flow Type.
- **Generate Excel Output**: An extension of the Summary Output, this selection will use selected summary output variables (listed above) to format and plot an Excel spreadsheet.

- **Add MINIVER Defaults**: Adds the variables listed in the *Generate Summary Output* section above.
- **Add Variable**: Adds one variable to the list.
- **Remove Variable**: Removes one variable from the list.
- **Remove All Variables**: Clears the table entirely.

- **Generate Detailed Output**: Creates an output file with information on the case properties as well as detailed properties at every time step.

3.4.3 Preprocessor Data

The Preprocessor Data screen (Figure 3.10 and Figure 3.11) provides the user with a summary of the case. This is available as the Modern Summary and the Legacy Summary. This was a feature requested by many Legacy users who preferred to have a way to check the original W Array values that Legacy MINIVER used in its PREMIN output files. The Modern Summary was included as a more verbose, informative summary of the data entered for the case.

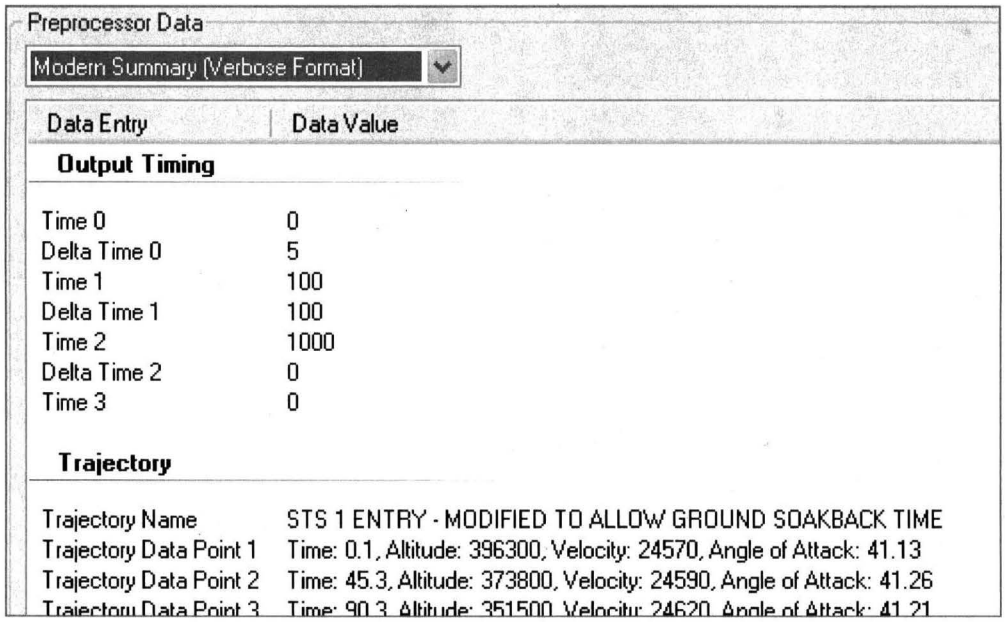

Figure 3.10: Preprocessor Data summary, Modern Format

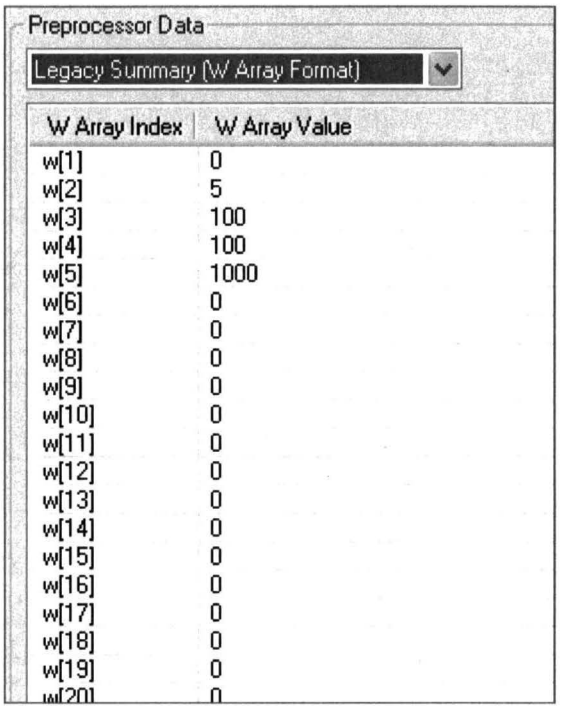

Figure 3.11: Preprocessor Data summary, Legacy Format

3.4.4 Output Timing

The Output Timing screen (Figure 3.12) provides the user with the output print value setup capability as in Legacy MINIVER. Instead of inputting values via command line, they have been replaced with a series of text boxes. A future enhancement may very well improve this with a list of timing values to increase the amount of intervals the user may specify.

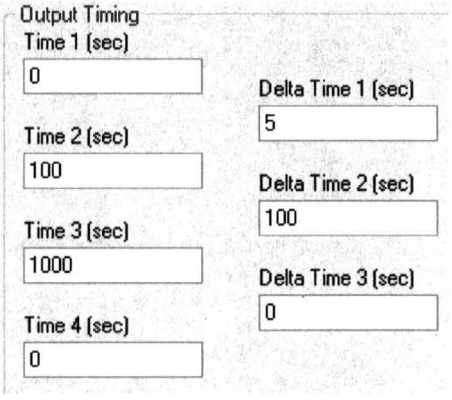

Figure 3.12: Output Timing screen

- **Time**: These fields provide time boundaries for reporting purposes.
- **Delta Time**: These fields provide time intervals to report at between two Time values.

3.4.5 Trajectory

The Trajectory screen (Figure 3.13) provides the user with verification that a trajectory has been loaded. From this screen, the Trajectory Editor may be launched, allowing the user to import a trajectory from Legacy MINIVER, or utilize a saved trajectory made with this version of MINIVER. The Trajectory Editor even operates in its own unit space, such that the user may use a trajectory in the Metric system in their English system project, and vice versa. The program will automatically detect the difference and perform a conversion.

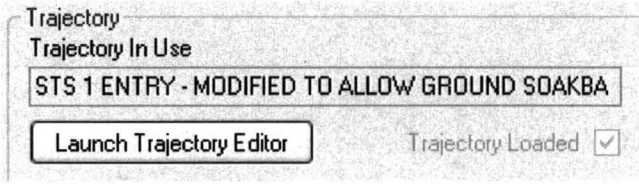

Figure 3.13: Trajectory Editor screen

3.4.6 Atmosphere

The Atmosphere screen (Figure 3.14) provides the user with an array of twelve common atmospheric models and two user-input models with which to draw flowfield properties from. In phase two of the project, the Range Reference Atmospheres were added to extend the model set available. Future enhancements to this section may include the ability to save user atmospheres as part of the drop down list.

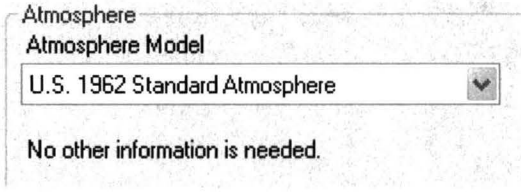

Figure 3.14: Atmosphere screen

- Atmosphere Models
    - **No Option Selected**
    - **U.S. 1962 Standard Atmosphere**
    - **Wind Tunnel Input**: Requires Temperature and Pressure input.
    - **Freestream T and P Input**: Requires Temperature, Pressure, and Altitude input.
    - **1963 Patrick Air Force Base Atmosphere**
    - **1971 Vandenburg Reference Atmosphere**
    - **1973 Vandenburg Cold Day Atmosphere**
    - **1973 Vandenburg Hot Day Atmosphere**
    - **1971 Kennedy Cold Day Atmosphere**
    - **1971 Kennedy Hot Day Atmosphere**
    - **1976 U.S. Standard Atmosphere**
    - **1990 – 2002 Range Reference Atmospheres**:
        - **Kwajalein, Marshall Islands**
        - **Vandenberg AFB, California**
        - **Wallops Island, Virginia**
        - **Cape Canaveral, Florida**

3.4.7 Heat Transfer Models

The Heat Transfer Models screen is a highly context-sensitive screen that updates based on the chosen model, of which there are several. This is, arguably, the core of the Preprocessor input types as it must be carefully chosen to reflect the behavior of the geometry at the point of interest. The choice of this option is therefore left to the discretion of the user based on his or her experience with the heat transfer methods. For detailed information on each method, the reader should refer to the Engel and Praharaj's MINIVER User's Guide [2]. A Legacy MINIVER user's guide PowerPoint file as created by Kathryn Wurster [5] provides an excellent summary of the capabilities of each method as MINIVER handles them.

Heat transfer methods include:

- Hemisphere Stagnation Point
- Cato-Johnson Swept Cylinder
- Eckert Reference Enthalpy Flat Plate
- Eckert / Spaulding-Chi Flat Plate
- Boeing Rho-Mu Flat Plate
- Beckwith / Gallagher Swept Cylinder
- Boeing Rho-Mu Swept Cylinder
- Lees-Detra-Hidalgo Hemisphere
- Leeside Orbiter Heating
- Flap Reattachment Heating
- Fin-Plate Peak Interference
- Brake Payload Impingement

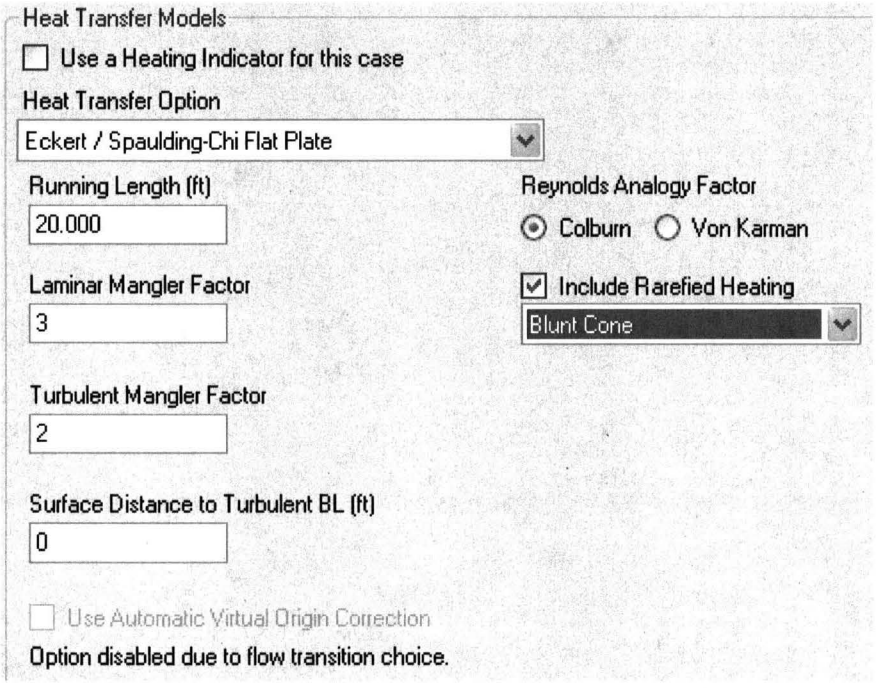

Figure 3.15: Heat Transfer Method screen example, Eckert / Spaulding-Chi Flat Plate option

3.4.8 Transition Models

The Transition Models screen provides the user with a way to describe the transition criteria at the case point. Each option varies in its input, but each option has zero, one, or two input values, leaving the screen a very simple implementation. Transition options include:

- Time-Dependent: Laminar to Turbulent
- Time-Dependent: Turbulent to Laminar
- Reynolds Number Dependent
- Re-Theta Dependent
- MDAC-E Transition
- MDAC-E Lookup
- NAR Re vs. Me Lookup
- Re-Theta / Me Dependent
- Fixed Constant

3.4.9 Crossflow

The Crossflow screen (example in Figure 3.16) provides the user with the ability to consider crossflow (divergent streamlines) at a case point. Five options are available:

- No Crossflow
- Constant Width Rectangle (Ideal Gas)
- Constant Width Rectangle (Real Gas)
- Sharp Edge Delta Wing (Ideal Gas)
- Sharp Edge Delta Wing (Real Gas)

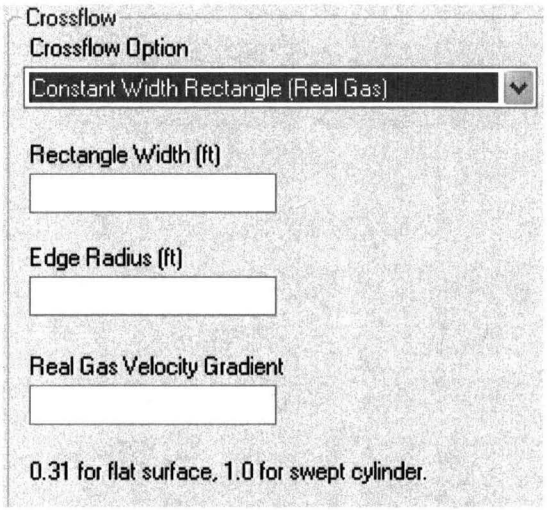

Figure 3.16: Crossflow screen example

3.4.10 Flowfield / Pressure Options

The Flowfield / Pressure Options screen (Figure 3.17) provides the user with the ability to configure the behavior of the flow and pressure fields at the case point of interest. These are inputted as pairs, along with corresponding angles. This option field is largely affected by the choice of heat transfer method (in fact, many heat transfer methods will force this screen to a certain configuration), but most utilize a single option pair. More complex geometries will utilize additional sets.

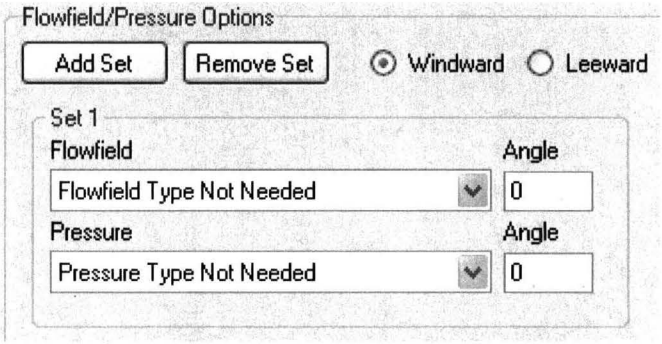

Figure 3.17: Flowfield / Pressure Options screen

- **Add Set**: Adds a new pair of Flowfield / Pressure Options to the case.
- **Remove Set**: Removes the last pair of Flowfield / Pressure Options from the case.
- **Windward / Leeward**: Some heat transfer options will request whether the point of interest is on the windward or leeward side of the vehicle.

It is left to the user to choose these options such that they are physically relevant to the geometry they are modeling. The flowfield and pressure types available to the program are as follows:

- Flowfield Types
  - Not Needed
  - Sharp Wedge Shock Angle
  - Sharp Cone Shock Angle
  - Oblique and Normal Shock
  - Parallel Shock
- Pressure Types
  - Not Needed
  - Input C_P vs. Mach
  - Tangent Wedge Pressure Coefficient
  - Tangent Cone Pressure Coefficient
  - Oblique Surface Pressure
  - Modified Newtonian Pressure – Oblique Shock
  - Prandtl-Meyer Expansion

3.4.11 Mass Injection

The Mass Injection screen (see Figure 3.18 for an example) allows the user to provide a mass injection solution to the case point of interest. This screen, like Crossflow, is entirely optional. The following types can be considered:

- No Mass Injection
- Specified Mass Injection Rate
- Blowing
- Suction

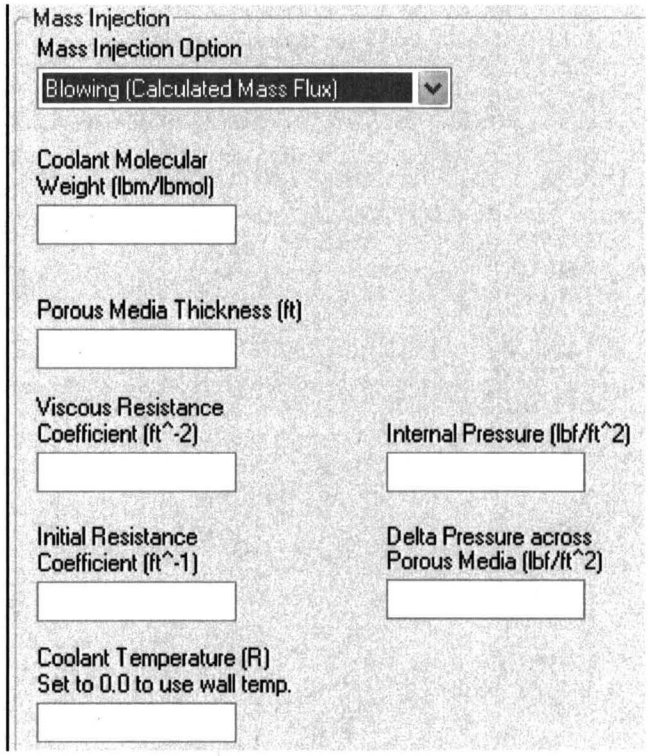

Figure 3.18: Mass Injection screen example

3.4.12 Time-Dependent Geometry

The Time-Dependent Geometry screen (Figure 3.19) is another optional set of parameters which allows the user to provide dynamic geometric behavior for the case point of interest. The primary control is a table (C# DataGridView) where the user can provide a list of time-dependent parameters, including body radius, running length, and local angle. A delimiter importer is available via the **Import…** button which allows the user to import a file delimited by commas, spaces, etc.

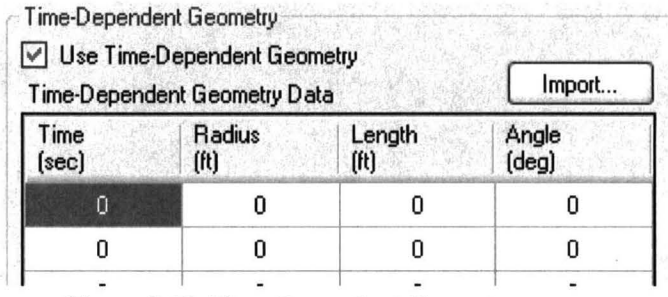

Figure 3.19: Time-Dependent Geometry screen

3.4.13 Initial Conditions

The Initial Conditions screen (Figure 3.20) allows the user to provide wall and initial conditions for the case point. The wall emissivity must always be specified (default value is 0.85). The user may choose between a constant wall temperature, which opens a text box to specify it, or the user may allow MINIVER to calculate the wall temperature by using the equilibrium radiation temperature of the last time point. In addition, the user may specify an initial heating load.

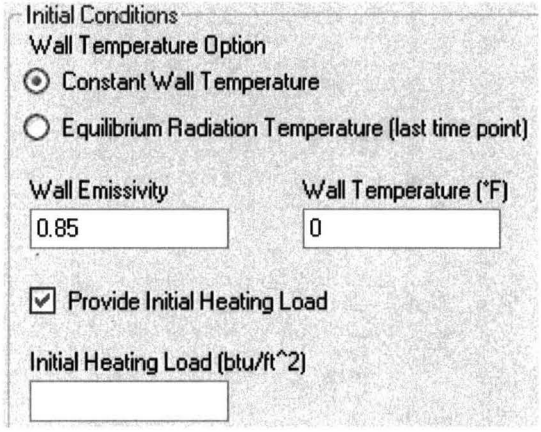

Figure 3.20: Initial Conditions screen

3.5 Trajectory Editor

One of the first major upgrades using C#, the Trajectory Editor utilizes the 2D Drawing libraries within the .NET Framework to plot a trajectory alongside an input table. The editor utilizes a DataGridView table to input data, and hotkeys (**Ins** and **Del**) are available to insert and delete lines from the table. The editor is capable of importing Legacy trajectories, parsing the original data and converting it into the new format for trajectory files. While the editor is capable of saving files, it does not actually utilize these files in the rest of the program. Instead, it copies the data to the case. In the event that the project is operating in a different unit system from the trajectory's unit system, it will perform a conversion before saving into the case. This prevents the user from having to manually convert the data if it is in a different system than the one they choose to work in with the project.

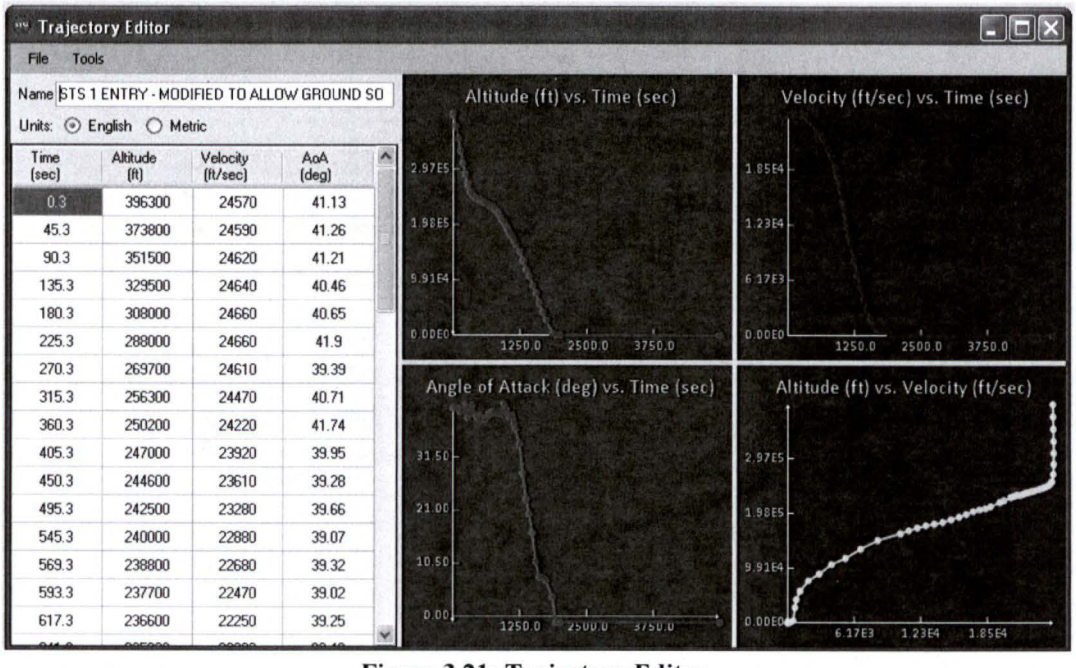

Figure 3.21: Trajectory Editor

The Legacy trajectory format included a column for yaw angle. At the request of NASA KSC and LaRC engineers, the yaw angle was removed for this version. To compensate for the fourth panel, the trajectory's altitude is plotted against its velocity, which is a commonly used plot in the engineering community. While trajectory files did not have any specific extension in Legacy MINIVER, the extension **.mtraj** was decided as the standard that MINIVER 2 will adhere to. It saves the trajectory data and view configuration into an XML document via serialization. In addition to being invoked via MINIVER directly, the user can invoke the Trajectory Editor as a stand-alone program for when they only need to create or edit a trajectory.

- **Trajectory Name**: This field specifies the name of the trajectory, which will be referenced in MINIVER as well as in the trajectory's save file.
- **Units**: Specifies whether the trajectory is in English or Metric units. Does not convert values that have already been entered.
- **Trajectory Table**: This table allows manual entry and modification of trajectory points.

- **Trajectory Graphs**: These graphs visualize the trajectory that is currently being used. A context menu can be invoked via right-clicking any of the four graphs.
    - **Change Color**...: Changes the color of the plot line.
    - **X-Axis**: Chooses the independent axis data.
    - **Y-Axis**: Chooses the dependent axis data.

The uppermost menu strip provides some additional tools for the user:

- **File Menu**
    - **New**: Initializes a new Trajectory, clearing all current data.
    - **Open**: Opens a previously saved MINIVER Trajectory, which possesses the file extension "*.mtraj".
    - **Save**: Saves the current Trajectory to a *.mtraj file. Updates a file if already bound to a save file.
    - **Save As**: Saves to a specific Trajectory (*.mtraj) file, or a new one.
    - **Save to Preprocessor**: When invoked via the Preprocessor Trajectory data section, pressing this button will copy the trajectory data into the MINIVER case it was invoked from. Units will be converted to the MINIVER Project unit system if it is different. The Trajectory file itself will not be altered.
    - **Exit Without Saving**: Exits the Trajectory Editor, discarding any active data.
- **Tools Menu**
    - **Delimited File Import...**: Activates the Generalized Delimited Importer to import any kind of delimited text file.
    - **Legacy Import**...: A Legacy MINIVER trajectory file can be imported for use through this capability.

3.6 CAD Editor

The CAD Editor (Figure 3.22) was the core upgrade of the second phase of the MINIVER upgrade. NASA KSC engineers requested a capability that would allow simple 2-D geometries to be used to generate a MINIVER project quickly.

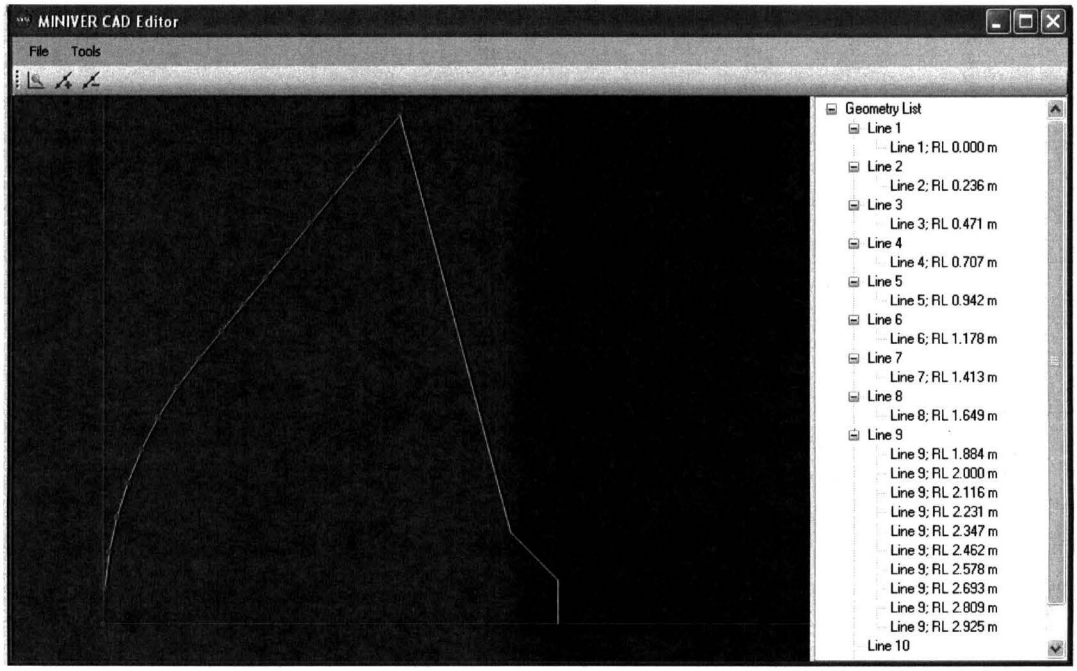

Figure 3.22: CAD Editor, using OREX geometry as an example

The CAD Editor is capable of reading AutoCAD geometry export data (via the **dataextraction** command in AutoCAD) as well as delimited data in order to create the geometry. In its current state, non-linear geometries must be approximated via linear segments. Handling arcs is being considered for a future upgrade phase. The save file of MINIVER CAD Editor projects use the extension **.mincad**. Its save format is an XML serialization of objects that make up the project as a whole.

3.6.1 Controls

The CAD Editor possesses multiple control mechanisms to speed up user input. Hotkeys include keyboard and mouse commands to manipulate the view window, the toolbar provides shortcuts to common actions, and the menu bar provides a link to more powerful tools such as saving, loading, and importing.

- **Hotkey Actions**
  - **Left-Click**: Selects points underneath the mouse pointer.
  - **Left-Click + Drag**: Performs a rectangle select which selects points within the region.
  - **Mouse Wheel**: Zooms in and out, centered on origin.
  - **Right-Click**: Opens a property editor to modify Local Data for selected points.
  - **Shift + Right-Click + Drag**: Pans the display in the direction of the mouse.
- **Toolbar**
  - **Zoom Center**: Resizes and pans the display to bring the origin and the geometry into view.
  - **Add Points**: Adds points to a selected line.
  - **Remove Points**: Removes selected points.
- **File Menu**
  - **New CAD Project**: Clears all stored data to make room for a new project.
  - **Open...**: Opens a previously saved CAD Editor Project.
  - **Save...**: Saves the currently open file to a CAD Editor Project.
  - **Export to Preprocessor**: Exports the current datapoints to the MINIVER Main Screen. The imported unit system determines the MINIVER Project's unit system.
  - **Exit**: Exits the CAD Editor.
- **Tools Menu**
  - **Import CAD Data**
    - **AutoCAD 2012 Format...**: Imports a file exported from AutoCAD 2012.
    - **Delimited Import...**: Imports a file using the Generalized Delimiter Importer.

3.6.2 Point Generation

Points can be created on lines by selecting the line via the Geometry Tree and selecting **Add Points to Line**. This invokes the interface shown in Figure 3.23. Users of Fluent and Gambit should instantly be familiar with the Grading Method types, as they use the exact same mathematical basis. In addition to auto-filling the running length for use with heat transfer options, the user can choose whether or not to use the geometry angle (with respect to the x-axis) to auto-fill into heat transfer options (where necessary) and the flowfield / pressure options screen.

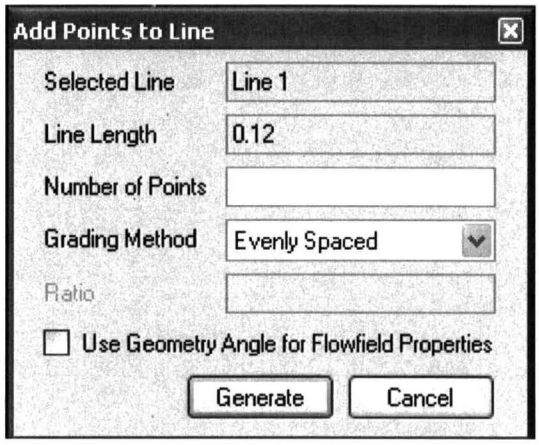

Figure 3.23: Point Generation screen

- **Selected Line**: This echoes the name of the line for reference.
- **Line Length**: This provides the length of the line based on its endpoints.
- **Number of Points**: The number of points to generate on this line. A minimum of one point can be added; this will generate a point at the start point, regardless of method.
- **Grading Method**: Determines how the points distribute themselves throughout the line.
    - **Evenly Spaced**: Each point is evenly spaced from one another along the line.
    - **Successive Ratio**: Each point is spaced at a distance that successively changes. For instance, a ratio of 2 would mean that each point interval would be two times larger than the last.

- **First Length**: Defines the first interval length on the line and adjusts all remaining intervals to compensate for the number of points using a successive ratio.
- **Last Length**: Defines the last interval length on the line and adjusts all remaining intervals to compensate for the number of points using a successive ratio.
- **First to Last Ratio**: Defines the ratio of the first interval to the last interval, then uses successive ratio to fill in the remaining points.
- **Last to First Ratio**: Defines the ratio of the last interval to the first interval, then uses successive ratio to fill in the remaining points.

- **Ratio / Length**: Defines the ratio or length for use with the Grading Method. Turned off for Evenly Spaced.
- **Use Geometry Angle for Flowfield Properties**: If selected, the CAD Editor will auto-fill case data with the geometry angle as calculated relative to the x-axis. This affects local body slope and flowfield / pressure options.

3.6.3 Editing Properties: Line

When case points are generated, they utilize the name of the line they are generated on in order to choose a name. The Line Name can be changed to something more physically relevant in order to aid in understanding its purpose once it is converted into a MINIVER project. In addition, the start running length of the line (through which points determine their local running lengths) is set here. After an import, start running lengths are automatically calculated by tracing backward from the origin point, but the user is welcome to adjust that value here if the calculation is incorrect (as may be the case in unusual or complex geometries).

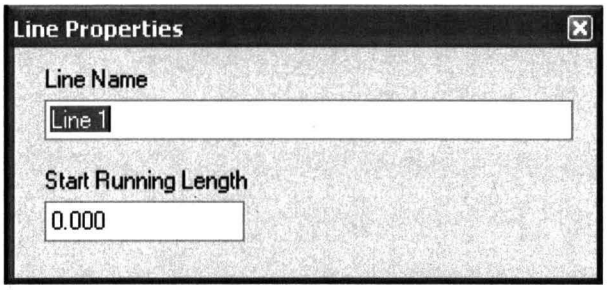

Figure 3.24: Line Properties window

3.6.4 Editing Properties: Global

The case properties, as one would be familiar with via a normal MINIVER project, are grouped slightly differently in the CAD Editor. As the project is a vehicle of some kind, there are some properties that are going to be the same for all points, regardless of location on the vehicle. The three preprocessor sections lumped together as "Global" properties are Timing, Trajectory, and Atmosphere. These properties can be edited by right-clicking the Geometry List node in the Geometry Tree and selecting **Edit Properties**. Their controls are identical to the ones used in the MINIVER information windows. When the CAD Editor exports to MINIVER, it copies the global properties to every case point.

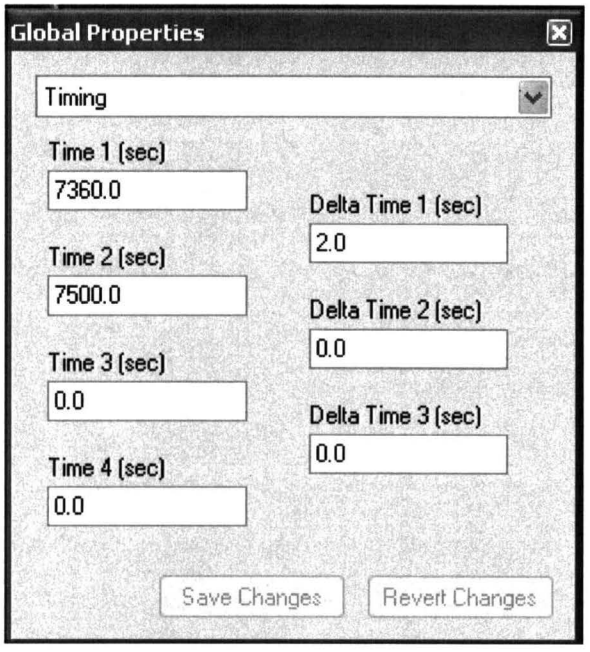

Figure 3.25: Global Properties window

3.6.5 Editing Properties: Local

The remaining MINIVER case properties (Heat Transfer Method, Transition Type, Crossflow, Flowfield / Pressure Options, Mass Injection, Time-Dependent Geometry, and Initial Conditions) are considered Local Properties, as they generally have the potential to change on a point-by-point basis. As with previous properties, these controls are identical to their main screen counterparts.

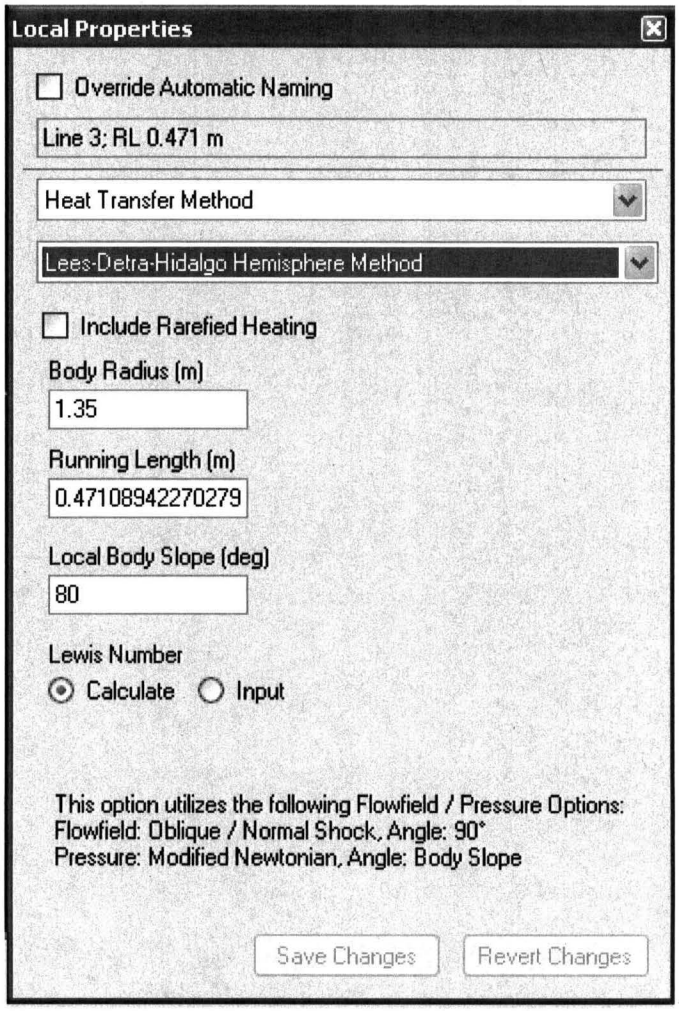

Figure 3.26: Local Properties window example

The Local Properties window includes an extra control that works alongside the property section control which allows the user to "override" the automatic naming convention. As mentioned previously, the CAD Editor is designed to provide a helpful service to the user by auto-filling running length values through the use of the imported geometry. However, there are some cases which this may not be relevant, such as the stagnation point of a body. In cases like these, the user need only check the override box and input the desired name. The automatic naming convention will format as **Line Name – RL x.xxx ft** (or m, if metric is chosen as the project unit system). As such, the user should change the line name early to reflect a physical description that makes sense for future reference.

3.7 Additional Tools and Utilities

To help with MINIVER workflow, a few additional tools were put together that aid the user with data entry and output formatting.

3.7.1 Generalized Delimited Importer

Many areas of the program require the user to fill in large tables. Examples of this include the Trajectory Editor, the CAD Editor, the Time-Dependent Geometry screen, the C_P vs. Mach input table, and so on. To reduce load on the user for this repetitious activity, and to allow for the use of files that can be passed amongst engineers, an importer was requested to assist in this goal. The Generalized Delimited Importer, shown in Figure 3.27, became the solution for this; a project in the Common namespace that can be easily instantiated anywhere in the program. The importer possesses a modular design, allowing the title and data columns to be set for each instantiation, allowing for implementation anywhere that a tabular input is requested.

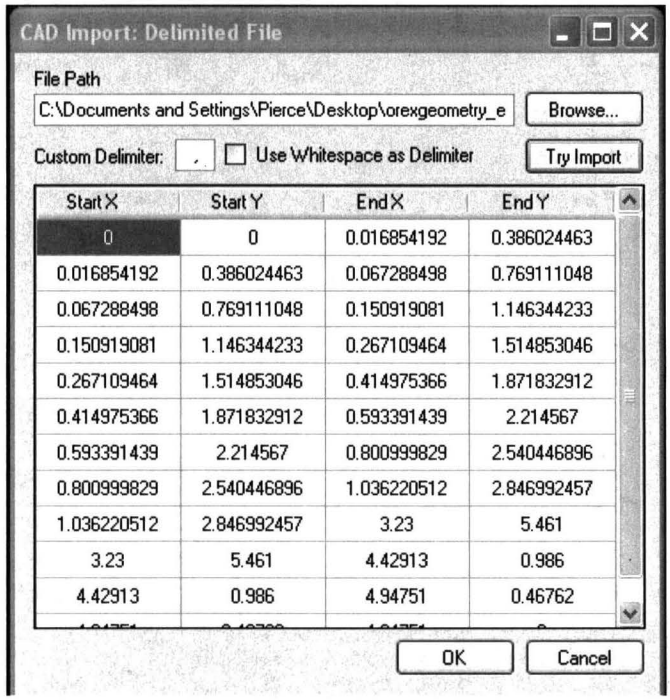

Figure 3.27: Generalized Delimited Importer, invoked via the CAD Editor

- **File Path**: Specifies the location of the file to import.

- **Custom Delimiter**: A character which specifies the delimiter that is used in the import file.
- **Use Whitespace as Delimiter**: If any combination of spaces and tabs (collectively known as "Whitespace" is used, this box should be checked.
- **Try Import**: Once the file path is set and the delimiter chosen, press this button to begin an import attempt. If successful, the table will fill with data. If not, an error will be presented informing the user of the issue.

A common import style is the Comma-Separated Values (CSV) format, which is standardized in the ***.csv** file extension. If this file extension is detected after a file is selected via the **Browse...** button, the importer will automatically set the delimiter to a comma. The importer is capable of performing the following error checks:

- Delimiter not present in read line
- Number of parsed values are less than the number of columns expected
- Non-numeric values detected in parse attempt

Once the parse is complete without errors, it will return to the importer with the collected values to fill the table. Pressing the **OK** button will then return to its calling location, and the table data is available to the calling function via a public property in the importer class.

3.7.2 Case Copier

The Case Copier utility provides the user with a quick way to copy sections between cases, as well as create new cases based on values from a chosen case. This is particularly useful for projects where the user wishes to use one set of Timing or Trajectory parameters for all cases, or to copy over a section like Heat Transfer to later modify a single value, such as running length. The Case Copier is invoked by right-clicking a case in the Project Tree and choosing **Copy Case**.

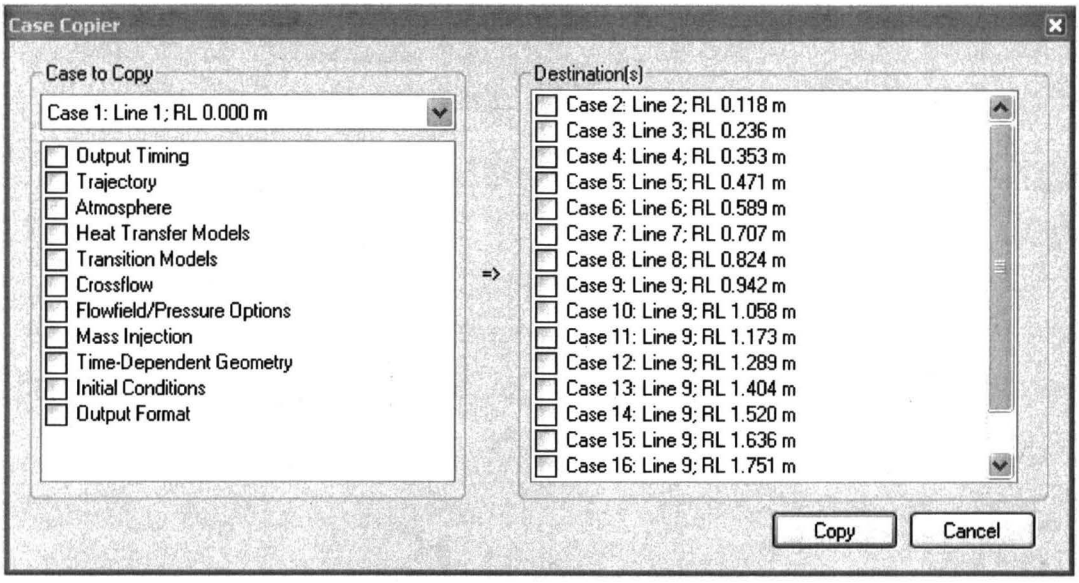

Figure 3.28: Case Copier window

The copier's left section selects the case to copy from by using the dropdown list. All cases are available for this purpose. From the listview beneath it, the sections to copy may be selected. A context menu exists, allowing the user to Select All and Select None to help expedite the process as necessary. The right section chooses the destination of the copy effort, and updates based on the case copy selection to make sure it is not included in the list. Similar to the case copy listview, a context menu exists to Select All and Select None from this list. In addition, the user can use the context menu to create a new case, which initializes a blank case and copies over the requested values.

Where the CAD Editor is available to create large numbers of cases through its point generation capability, the Case Copier can be seen as a way to create a large number of cases without the use of a geometry import. A future improvement is being considered where the copier can be expanded on with an importer-style capability that can create a large number of cases through the use of an import table. By parameterizing the import capability, it seems apparent that user input could be expedited even further by doing some form of a mass case generation action.

3.7.3 Microsoft Excel Summary Output

Since the Summary Output type is the most commonly used output type from Legacy MINIVER, it was thought that if an Excel spreadsheet could be generated from the output data, it would allow the user to quickly analyze output data by taking the formatting requirement out of the workflow. To this end, the Summary Output was extended with a COM Interop link to Microsoft Excel via Office Automation methods. Figure 3.29 shows an example Excel-formatted output, and Figure 3.30 shows a graph generated in Excel via this method.

Once output is created, if Excel output is selected, the program opens up an instance of Microsoft Excel in the background. Then, it formats the column headers and dumps the data in, row by row. Afterwards, it marches through the column types, and if Graph is checked, it will create a new tab in the workbook and plot the data versus time. A future improvement will be to extend this capability to plot versus any other parameter on the x-axis, as a common graph format is to plot versus altitude.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Time	Altitude	Velocity	Mach #	Angle of	Reynolds	Heat Coeff.	Rec. Enthalpy	Rad. Equilibrium	Heat Rate	Heat Load	Pressure	Flow Type
2	(sec)	(km)	(m/sec)		Attack	No/m	(kgm/m^2*s)	(J/kgm)	(deg K)	(W/m^2)	(J/m^2)	(N/m^2)	
3	7361.0	105.0	7451.0	27.3	0.00	1.260E+02	1.575E-04	2.794E+07	546.9	4.359E+03	0.000E+00	1.265E+01	lam
4	7363.0	104.2	7451.8	27.3	0.00	1.470E+02	1.863E-04	2.794E+07	570.2	5.101E+03	9.460E+03	1.462E+01	lam
5	7365.0	103.4	7452.5	27.3	0.00	1.714E+02	2.096E-04	2.794E+07	587.2	5.734E+03	2.029E+04	1.689E+01	lam
6	7367.0	102.6	7453.3	27.3	0.00	1.997E+02	2.366E-04	2.794E+07	605.2	6.470E+03	3.250E+04	1.952E+01	lam
7	7369.0	101.8	7454.0	27.3	0.00	2.326E+02	2.675E-04	2.795E+07	623.9	7.312E+03	4.628E+04	2.255E+01	lam
8	7371.0	100.9	7454.7	27.3	0.00	2.709E+02	3.032E-04	2.795E+07	643.7	8.281E+03	6.187E+04	2.608E+01	lam
9	7373.0	100.1	7455.0	27.3	0.00	3.164E+02	3.376E-04	2.795E+07	661.1	9.214E+03	7.937E+04	3.023E+01	lam
10	7375.0	99.3	7455.3	27.3	0.00	3.706E+02	3.471E-04	2.795E+07	665.6	9.467E+03	9.805E+04	3.517E+01	lam
11	7377.0	98.4	7455.7	27.3	0.00	4.341E+02	3.588E-04	2.795E+07	671.2	9.787E+03	1.173E+05	4.092E+01	lam
12	7379.0	97.6	7456.0	27.3	0.00	5.081E+02	3.728E-04	2.795E+07	677.6	1.017E+04	1.373E+05	4.761E+01	lam
13	7381.0	96.8	7456.3	27.3	0.00	5.943E+02	3.897E-04	2.796E+07	685.1	1.062E+04	1.580E+05	5.535E+01	lam
14	7383.0	96.0	7455.9	27.3	0.00	6.895E+02	4.093E-04	2.795E+07	693.5	1.115E+04	1.798E+05	6.386E+01	lam
15	7385.0	95.2	7455.4	27.3	0.00	7.975E+02	4.333E-04	2.795E+07	703.3	1.180E+04	2.028E+05	7.348E+01	lam
16	7387.0	94.4	7455.0	27.3	0.00	9.230E+02	4.633E-04	2.794E+07	715.1	1.261E+04	2.272E+05	8.461E+01	lam
17	7389.0	93.6	7454.5	27.3	0.00	1.067E+03	5.084E-04	2.794E+07	731.7	1.383E+04	2.536E+05	9.733E+01	lam
18	7391.0	92.8	7454.1	27.3	0.00	1.232E+03	5.585E-04	2.794E+07	749.0	1.518E+04	2.827E+05	1.118E+02	lam
19	7393.0	91.9	7452.1	27.3	0.00	1.442E+03	6.213E-04	2.792E+07	769.0	1.687E+04	3.147E+05	1.304E+02	lam
20	7395.0	91.1	7450.2	27.3	0.00	1.685E+03	6.965E-04	2.791E+07	790.9	1.889E+04	3.505E+05	1.520E+02	lam
21	7397.0	90.2	7448.2	27.3	0.00	1.968E+03	7.866E-04	2.789E+07	815.1	2.130E+04	3.906E+05	1.771E+02	lam
22	7399.0	89.3	7446.3	27.3	0.00	2.298E+03	8.954E-04	2.788E+07	841.6	2.421E+04	4.362E+05	2.065E+02	lam
23	7401.0	88.5	7444.3	27.3	0.00	2.683E+03	1.031E-03	2.786E+07	871.4	2.784E+04	4.882E+05	2.409E+02	lam
24	7403.0	87.6	7438.9	27.3	0.00	3.116E+03	1.197E-03	2.782E+07	903.9	3.223E+04	5.483E+05	2.794E+02	lam
25	7405.0	86.8	7433.5	27.3	0.00	3.620E+03	1.365E-03	2.778E+07	933.4	3.664E+04	6.172E+05	3.243E+02	lam
26	7407.0	85.9	7428.1	27.1	0.00	4.194E+03	1.397E-03	2.774E+07	938.4	3.739E+04	6.912E+05	3.758E+02	lam
27	7409.0	85.1	7422.7	27.0	0.00	4.788E+03	1.434E-03	2.770E+07	944.2	3.833E+04	7.669E+05	4.323E+02	lam

MINIVER Data | Altitude | Velocity | Mach No | AoA | Re # | Heat Coeff | Rec Enth | Rad Eq T | Heat Rate | Heat Load | Pressure

Figure 3.29: Example Excel output using OREX data

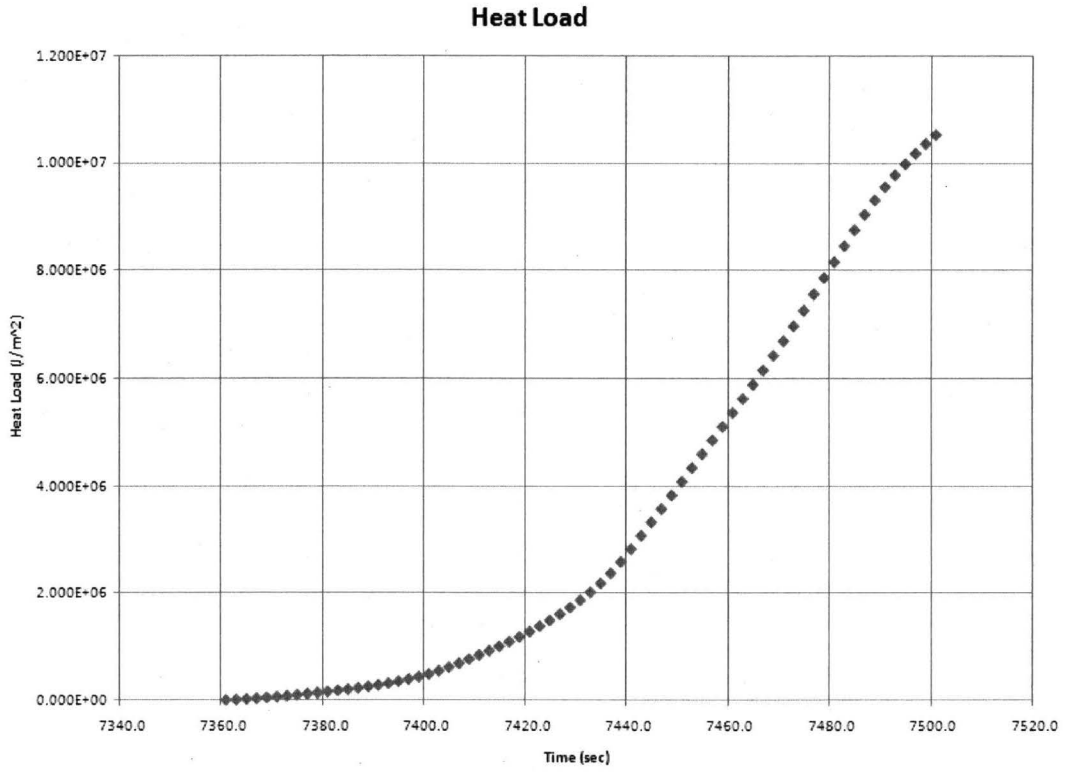

Figure 3.30: Sample formatted graph generated by the Excel output type

3.8 Directory Structure

All modern software standardizes the directory space it takes up on a computer so that the user can become familiar with it. The MINIVER directory structure has been organized as follows.

3.8.1 Program Organization

The MINIVER program itself is organized into a few sections within its root (install) directory. The bin folder contains binaries, or code libraries, which the program uses to function. The docs folder contains the User's Guide, which contains information for new and legacy users alike. To help with manipulating projects in MINIVER, two examples are provided in the docs directory for reference. All remaining executables are left in the root directory.

- MINIVER Root Directory\
  - bin\

        - **Common.dll**
        - **PreprocessorUIs.dll**
        - **Processor.dll**
    - docs\
        - \examples
            - Legacy MINIVER Example
            - CAD Editor Example
        - **MINIVER 2 User's Guide.pdf**
    - **CAD Editor.exe**
    - **MINIVER 2.exe**
    - **MINIVER 2.exe.config**
    - **TrajectoryEditor.exe**

The bin\ subdirectory contains Dynamic-Link Libraries (DLLs) that MINIVER references for normal operation. The docs\ subdirectory contains useful documentation for the program. The examples\ subdirectory includes two examples which are useful for the beginning user to get an understanding of the standard MINIVER workflow; details on how they are setup are included in the User's Guide. The root directory contains the executables that run the program and allow the user to invoke standalone versions of the CAD Editor and Trajectory Editor.

3.8.2 Save File Organization

- <projectname>\
    - Cases\
    - Results\
    - **<projectname.mproj>**

The Cases\ subdirectory stores all cases created within the scope of the project. The Results\ subdirectory stores all results generated by the processor, named using the case name and including **_summary**, **_detailed**, and **_excel** suffixes to indicate the type of output it contains.

Chapter 4: Benchmarks

4.1 Fortran Output vs. C# Output

The Legacy version of MINIVER was shipped with a sample log of MINIVER input using the STS-1 orbiter and its trajectory. This example provides a great demonstration for migrating from Legacy to Modern MINIVER versions as it is a fairly straightforward case. The importance of this test case is to demonstrate that the side-by-side output is identical, with error tolerance accounting for differences in data primitives and output presentation between Fortran and C#. The Legacy case sets the following information:

- **Input Data in English or Metric?** -> English
- **Specify Printout Intervals:** 0, 25, 2000, 200, 5000, 0
- **Trajectory Input File?** -> Yes
- **Atmosphere Data:** 1976 U.S. Standard Atmosphere
- **Run a Heating Indicator?** -> No
- **Heat Transfer Method:** Eckert Reference Enthalpy Flat Plate Method
- **Running Length:** 20 ft
- **Turbulent Mangler Factor:** 2
- **Laminar Mangler Factor:** 3
- **Surface Distance to Start of Turbulent B.L:** 0
- **Use Automatic Virtual Origin Correction?** -> No
- **Transition Option:** Fixed Constant, 0.0
- **Crossflow Adjustment?** -> No
- **Flowfield / Pressure Options:** Sharp Cone, 10.0°; Tangent Cone, 10.0°
- **Windward or Leeward?** -> Windward
- **Mass Injection?** -> No
- **Time-Dependent Geometry?** -> No

- **Wall Temperature Option:** Equilibrium Radiation Temperature
- **Emissivity:** 0.85
- **Initial Heating Load:** None
- **Printout Types?** -> Detailed and Summary
- **English or Metric Output Units?** -> English

After inputting the data into Modern MINIVER and generating data, the output files were compared. All comparisons came out near identically, with minor decimal shifts in exponentially-formatted data output. A snapshot of this data can be seen in Table 4.1 via the summary printout mode in MINIVER. Discrepancies are highlighted in yellow. These are rounding errors due to a difference in output format: the Legacy version formats as 0.00E+00 while the Modern version formats as .000E+00.

Table 4.1: STS-1 Summary Data Comparison Snapshot

Time (sec)	Rad Eq T (°R) [LEGACY]	Rad Eq T (°R) [MODERN]	Heat Rate (btu/ft^2-s) [LEGACY]	Heat Rate (btu/ft^2-s) [MODERN]	Heat Load (btu/ft^2) [LEGACY]	Heat Load (btu/ft^2) [MODERN]
0.0	475.9	475.9	3.13E-01	3.13E-01	0.00E+00	0.00E+00
25.0	531.0	531.0	3.90E-01	3.90E-01	8.79E+00	8.79E+00
50.0	600.7	600.7	5.12E-01	5.12E-01	2.01E+01	2.01E+01
75.0	687.4	687.4	7.02E-01	7.02E-01	3.53E+01	3.53E+01
100.0	781.4	781.4	9.62E-01	9.62E-01	5.64E+01	5.61E+01
125.0	884.8	884.8	1.33E+00	1.33E+00	8.46E+01	8.46E+01
150.0	1000.5	1000.5	1.85E+00	1.84E+00	1.24E+02	1.24E+02
175.0	1125.2	1125.2	2.56E+00	2.56E+00	1.79E+02	1.79E+02
200.0	1247.7	1247.7	3.45E+00	3.45E+00	2.55E+02	2.54E+02
225.0	1379.5	1379.5	4.65E+00	4.64E+00	3.56E+02	3.56E+02

4.2 OREX Flight Test

The CAD Editor was one of the most important upgrades of the second phase of the MINIVER upgrade project, so it would be ideal to have a solid test

case demonstrating its use with a real-world example. To this end, the Orbital Re-entry Experiment (OREX) test performed by the Japan Aerospace Exploration Agency (JAXA) possessed a wealth of data that could be used to build a test case and verify it with their results. In particular, the paper by Gupta, Moss, and Price (6) possesses a piece of the OREX re-entry trajectory, and then provides Viscous Shock Layer (VSL) analysis of the forebody. It is important to note that only the stagnation point was instrumented for thermal data collection; the VSL data in (6) is completely simulated. As such, the stagnation point data provides a MINIVER comparison to flight data, and the VSL data provides a MINIVER comparison to more computationally-intensive approach. In Hirschel and Weiland's book (7), they provide a larger plot of the OREX stagnation point data which this paper will reference.

4.2.1 OREX Background

The OREX re-entry vehicle (Figure 4.1) was designed as a demonstrator for thermal protection systems being developed at JAXA. It was launched on February 4, 1994 as the first of three experiments. To aid in research, they instrumented the vehicle with electrostatic and thermal probes. In particular, thermal data from the stagnation point at the forebody will be used to compare to MINIVER output data.

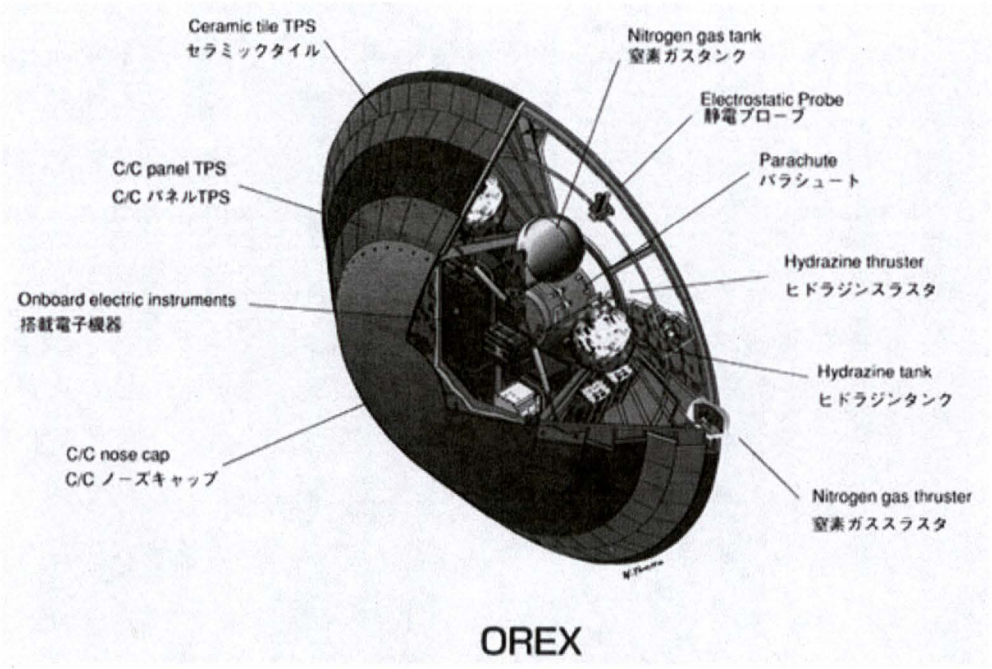

Figure 4.1: OREX vehicle, via (8)

4.2.2 Creating the OREX MINIVER Project

To accurately model the OREX vehicle's descent, MINIVER needs information regarding the vehicle geometry, its trajectory, atmospheric conditions, and what heating methods should be selected. Gupta, Moss, and Price's paper provides all of the above, and for reference they will be reiterated here. Figure 4.2 shows the geometry of the vehicle: a blunted cone with a 1.35 m radius carbon/carbon (C/C) nose cap followed by a ring of C/C tiles and finally by four rings comprised of ceramic tiles.

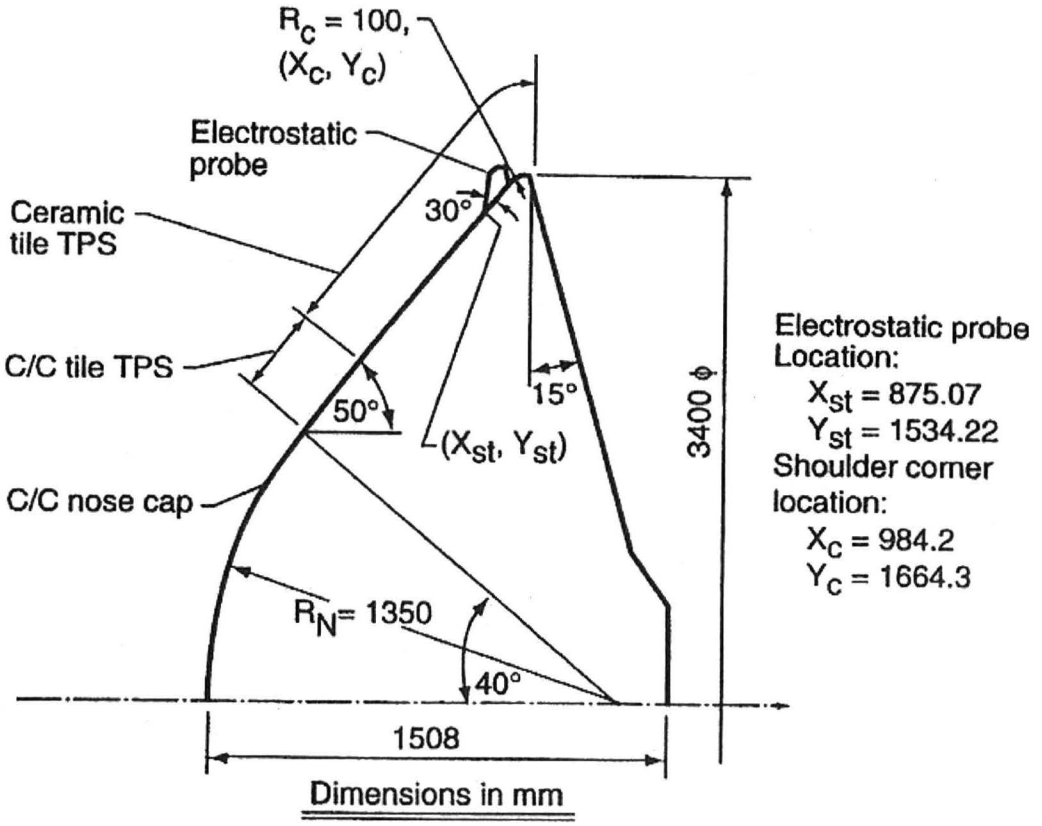

Figure 4.2: OREX geometry, via (6)

The trajectory is also included in Gupta, Moss, and Price and is reiterated in Table 4.2 for reference. The angle of attack was not included, but since the stagnation point is well-defined as the center of the cone, it is thusly implied that the angle of attack is consistently zero degrees, that is, the plane of the stagnation point is identical to the descent plane.

Table 4.2: OREX trajectory, via (6)

Time (sec)	Altitude (km)	Velocity (m/sec)	Angle of Attack (deg)
7361.0	105.00	7451.00	0.0
7370.6	101.10	7454.65	0.0
7381.0	96.77	7456.30	0.0
7391.0	92.82	7454.10	0.0
7401.0	88.45	7444.30	0.0
7411.5	84.01	7415.90	0.0
7421.5	79.90	7360.20	0.0
7431.5	75.81	7245.70	0.0

7441.5	71.73	7049.20	0.0
7451.5	67.66	6720.30	0.0
7461.5	63.60	6223.40	0.0
7471.5	59.60	5561.60	0.0
7481.5	55.74	4759.10	0.0
7491.5	51.99	3873.40	0.0
7501.5	48.40	3000.00	0.0

As far as atmosphere choice is concerned, Gupta, Moss, and Price's paper mentions that "The atmosphere values denoted as OREX … are similar to those for the 1962 U.S. Standard Atmosphere up to an altitude of about 90 km." (6). As such, for this comparison, the MINIVER atmospheric option will be set to the 1962 U.S. Standard Atmosphere as a close approximation. For the heat transfer method, there will be three choices: at the stagnation point, the Hemisphere Stagnation Point option will be selected. For the spherical cap of the nose cone, the Lees / Detra / Hidalgo Hemisphere option will be selected, Finally, the remainder of the forebody will utilize the Eckert Reference Enthalpy Flat Plate option as an approximation, including Mangler transformation values to convert from flat plate to cone.

To create the MINIVER project for this, the geometry was converted into X and Y coordinates as is the input style for the MINIVER CAD Editor. The results of this conversion are tabulated in Table 4.3. The CAD Editor's importer uses lines as its geometric base, and as such requires the start and end coordinates of each line. Note the use of lines to approximate the nose cap arc; this is done in increments of five degrees.

Table 4.3: OREX geometric data, based on geometry from Figure 4.2

Start X (m)	Start Y (m)	End X (m)	End Y (m)
0.000000	0.000000	0.005137	0.117660
0.005137	0.117660	0.020510	0.234425
0.020510	0.234425	0.046000	0.349406
0.046000	0.349406	0.081414	0.461727
0.081414	0.461727	0.126484	0.570535

0.126484	0.570535	0.180866	0.675000
0.180866	0.675000	0.244145	0.774328
0.244145	0.774328	0.315840	0.867763
0.315840	0.867763	0.984504	1.664513
0.984504	1.664513	1.350000	0.300533
1.350000	0.300533	1.508000	0.142531
1.508000	0.142531	1.508000	0.000000

To get started, the CAD Editor's importer is invoked and a file is imported, using the coordinate data as input. This is done in Figure 4.3.

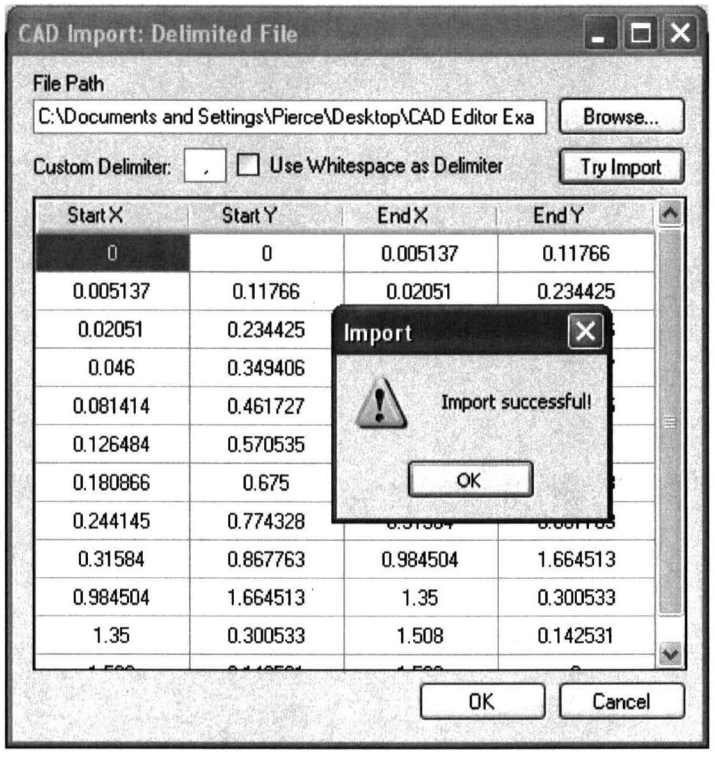

Figure 4.3: CAD Editor import of OREX geometry

Once the OREX geometry is imported, the lines are drawn in the CAD Editor screen as shown in Figure 4.4. To provide physical sense to the lines involved, each line is edited to change their names: the first eight lines get changed to "CC Nose Cap" and the ninth line changed to "Concentric Rings" in reference to the forebody geometry.

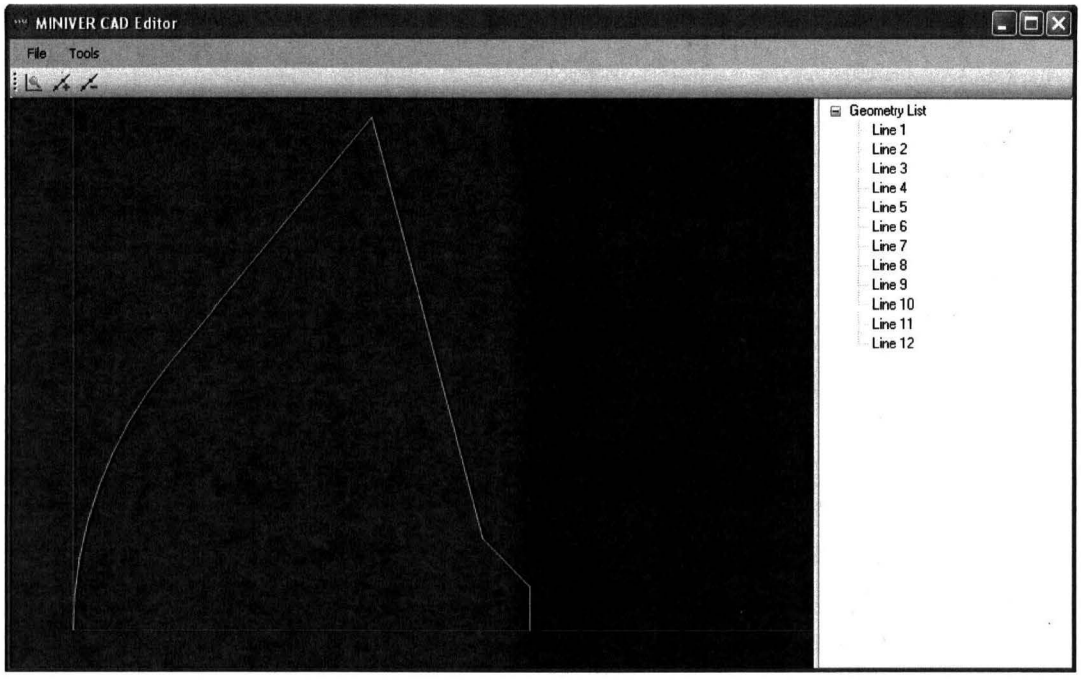

Figure 4.4: CAD Editor with imported OREX geometry

With the geometry imported, each line gets points added to them. In this case, each of the nose cap lines get one point added, which puts it at the start of each line. The ninth line has ten points added, distributing them evenly along the surface. Next, global properties are set for the project: for Timing, a start time of 7361.0 sec and an end time of 7501.0 sec is used along with a 2.0 sec interval. Trajectory is set by launching the Trajectory Editor and inputting the OREX trajectory values as tabulated above. The atmosphere is set to 1962 U.S. Standard Atmosphere as discussed above.

Local properties are set first by setting bulk data by using box select, then adjusting singular points as necessary. First, the Transition Type is set for all points using the MINIVER option NAR Re vs. Me Lookup option for simplicity. Since there is no crossflow, time-dependent geometry, or mass injection, these sections are skipped. Initial conditions are set to utilize the equilibrium radiation temperature with an emissivity of 0.85 as a generality and no initial heat load. Next, points are modified for heat transfer options; the stagnation point is set to Hemisphere Stagnation Point with a body radius of 1.35 m (known from geometry)

and a calculated Lewis number. Flowfield and pressure is set to Oblique / Normal Shock and Modified Newtonian Pressure at 90° angles.

The remaining nose cap points are set to utilize the Lees / Detra / Hidalgo Hemisphere Method with a body radius of 1.35 m and flowfield / pressure angles set to their local values (in the example, starting at 85° and decreasing by 5° per point.) Running length is automatically calculated thanks to the geometry. The rest of the forebody is set to Eckert Reference Enthalpy Flat Plate, running lengths are automatically calculated, and Mangler conversion factors are included for flat plate to cone. Flowfield / pressure is set to Sharp Cone and Tangent Cone, using the geometry to calculate the angles (known as 50° via geometry). With all the points set, the setup is then exported to the MINIVER main screen, as shown in the Project Properties window in Figure 4.5.

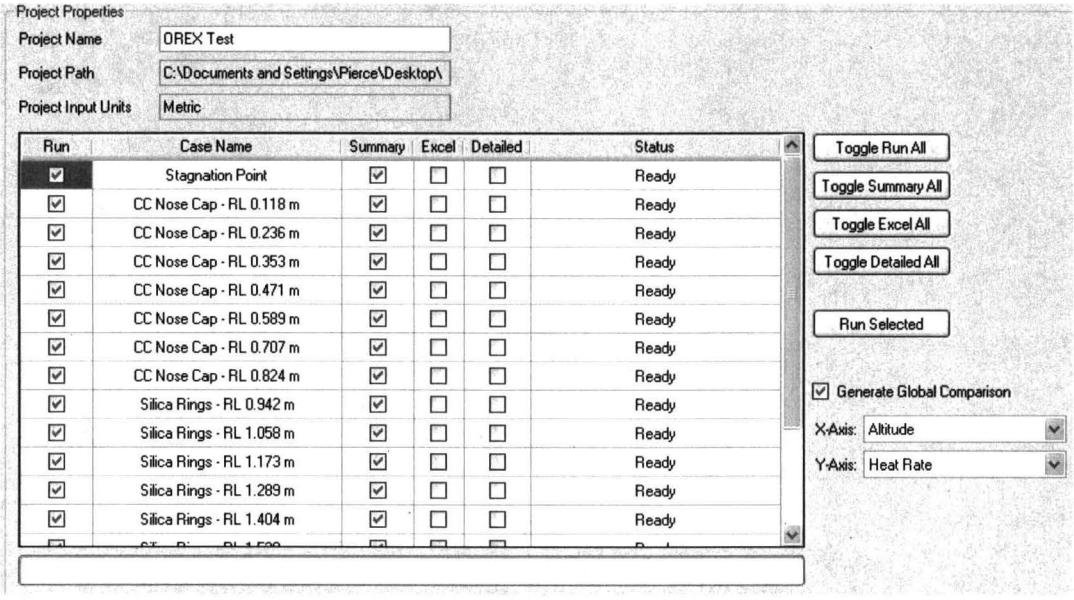

Figure 4.5: OREX CAD Editor data exported to the MINIVER main screen

Two important sets of data are obtained from output selections in this window: the Summary Printout, which will be used from the Stagnation Point case to compare with the plot in Hirschel and Weiland (7). The Global Comparison will generate heat rate vs. altitude plots for each output time point; in particular, the data

for 84.01 km and 59.60 km will be utilized to compare with plots in Gupta, Moss, and Price (6).

4.2.3 OREX Results

The resultant data from MINIVER for the OREX case is promising, especially at the higher and lower edges of the provided trajectory. This is consistent with the wall catalycity comparisons made in Hirschel and Weiland (shown in Figure 4.6,) which shows that wall catalytic effects are mostly present in the 60 km to 80 km altitude ranges, where MINIVER overpredicts. External to that window, MINIVER provides an accurate comparison to the OREX free flight data, where wall catalytic effects are negligible.

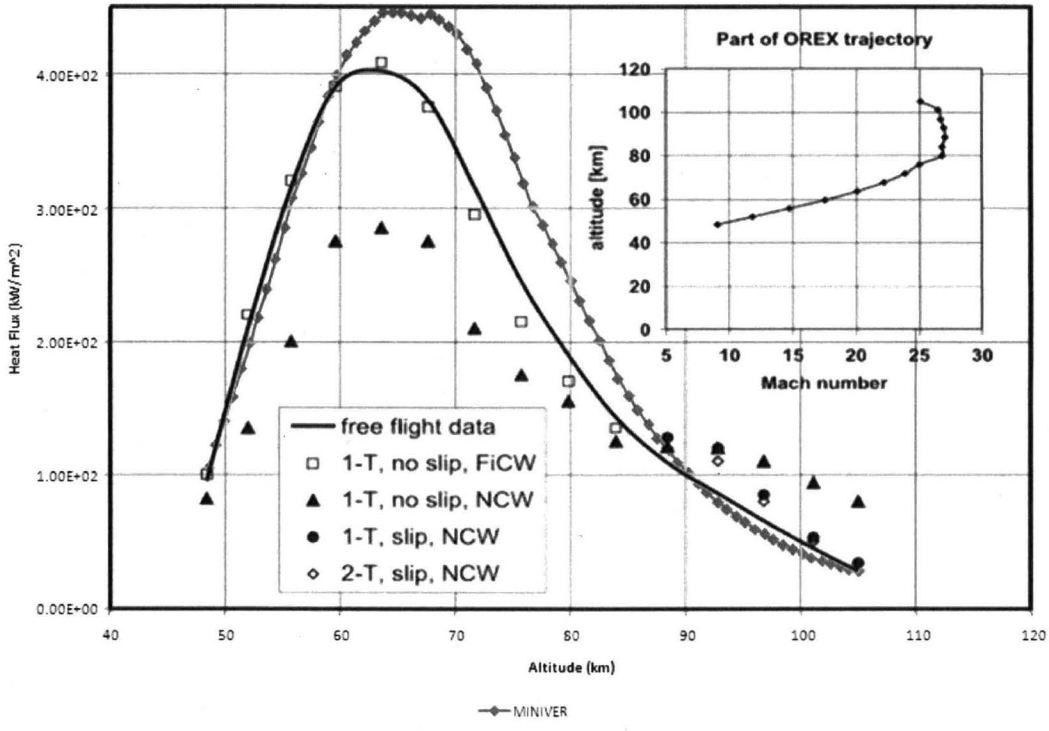

Figure 4.6: OREX stagnation point flight data (plot overlay via (7)) compared to MINIVER output and numerical simulations including catalytic wall considerations

Gupta, Moss, and Price provide two plots comparing to their own VSL calculations in (6) which are overlaid with MINIVER results in Figure 4.7 and Figure 4.8. At 84 km, there is a wide disparity between non-catalytic and fully-catalytic effects; MINIVER provides a middle-ground calculation for most of the

forebody. The jump in data at body point a (transition from nose cap to cone) is due to heat transfer choice: it is apparent that a better option could have been selected for the cone region of the forebody.

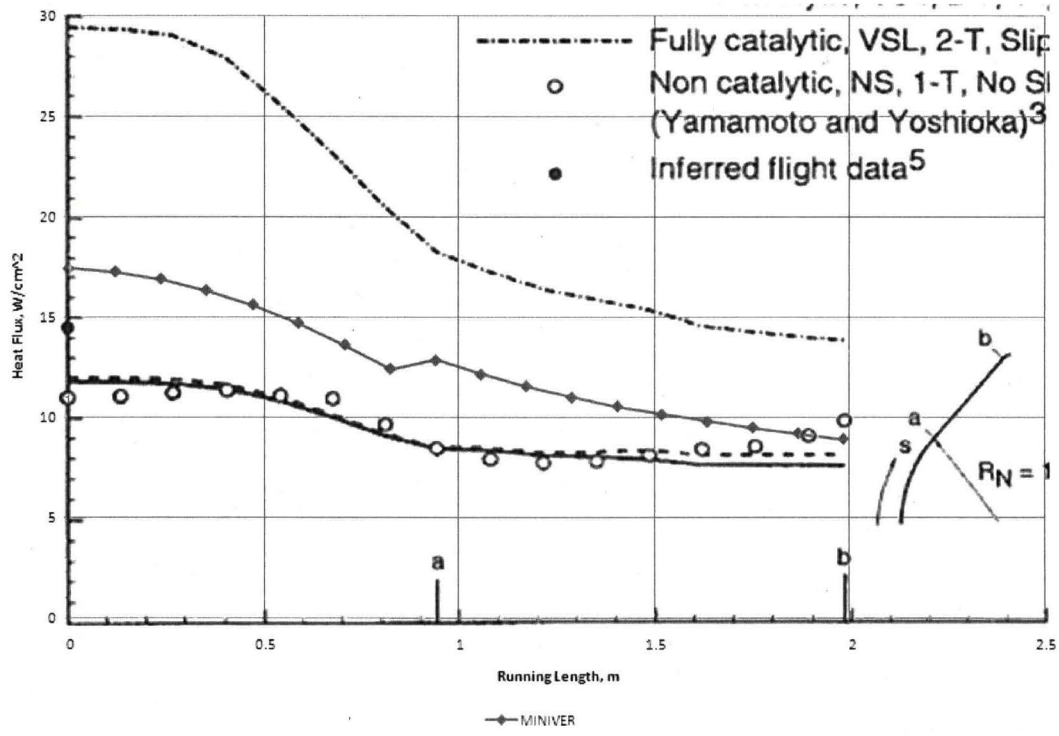

Figure 4.7: OREX forebody heat rate at 84.01 km; MINIVER comparison to VSL calculations (plot overlay via (6))

At 60 km, catalytic effects are much less prominent, and MINIVER provides a closer approximation to the finite catalytic calculation in Gupta, Moss, and Price. While the sources referenced do not provide any indication of simulation runtimes, the MINIVER test case took 13 seconds to calculate all 18 data points. Considering the relative accuracy and the time it took to set up the project and generate data, MINIVER demonstrates significant power in this regard.

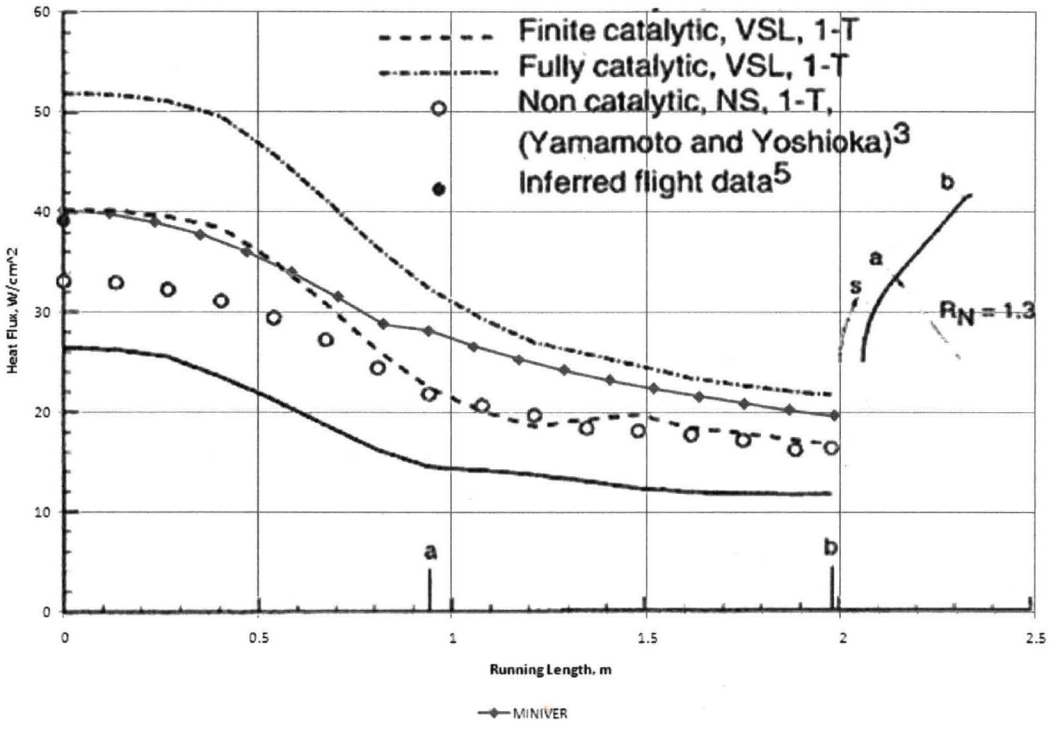

Figure 4.8: OREX forebody heat rate at 59.60 km; MINIVER comparison to VSL calculations (plot overlay via (6))

Chapter 5: Conclusions and Future Work

5.1 Conclusions

To recap from the objectives section: first, the aim was to transfer the code from the Fortran language to a modern language. While this initially was done with C++, it was improved significantly with the usage of C#. As demonstrated throughout this paper, the code was successfully transferred. The second objective was to isolate known bugs and correct them in the new version. Examples of this include a legacy bug where a trajectory file could not be saved because it became caught in an infinite loop. Trajectory files are core to the Trajectory Editor, so this bug is clearly resolved. Another example involves the preprocessor, where erroneous data entry (such as negative time values) could be inputted. Code checks are in place to prevent errors like this in the modern version.

The third objective was to build a GUI that increases user-friendliness and streamlines preprocessor input for the user. The figures demonstrated in this paper are testament to that conversion effort. Sections are clearly outlined and provide feedback to the user on completeness and prevent input errors. Many tools have been provided to ease the data entry task for the user as well. The final object was to perform benchmark tests to ensure that the new version output does not deviate from the legacy version. This is shown in Chapter 4 with the Legacy / Modern comparison, and the accuracy of MINIVER is displayed with the OREX test.

In conclusion, the new MINIVER user interface provides a greater degree of control over case generation and processor runtime. By including several quality-of-life improvements, engineers who use the latest version of MINIVER will find that they can focus less on data entry and more on analysis, and the advancement of file generation and saving can allow engineers to easily share and

collaborate with their tests in MINIVER. Overall, these new advancements provide a significant improvement to an already powerful software package.

5.2 Future Work

Further improvements have been considered for future work with the MINIVER program. A major addition would be the inclusion of the EXITS postprocessor that existed in the legacy version. NASA KSC engineers prefer to utilize third-party programs like SINDA/FLUINT and Thermal Desktop, which are indeed very powerful, but some MINIVER users may not have these software packages available. Providing the user with a choice between EXITS, SINDA/FLUINT, and Thermal Desktop would add a great degree of flexibility regarding MINIVER output.

Additional updates could further improve the CAD Editor, such as the inclusion of arcs and other geometric constructs, and adding the ability to edit point coordinates within the CAD Editor. Even further, the CAD Editor Project type could become centralized to the MINIVER Main Screen, allowing the ability to keep the geometry through the lifetime of the project. After postprocessor incorporation, the geometry could even be used to visualize thermal protection systems and heat distribution through them.

During the course of the study, feedback from NASA engineers was very valuable in optimizing the interface for their use, as well as identifying and eliminating bugs during the development process.

References

1. **Hender, D R.** *A Miniature Version of the JA70 Aerodynamic Heating Computer Program, H800 (MINIVER).* Huntington Beach, California : McDonnell Douglas Astronautics Co, 1970. MDC Report G-0462.

2. **Engel, Carl D. and Praharaj, Sarat C.** *MINIVER Upgrade for the AVID System, Vol 1: LANMIN User's Manual.* NASA CR-172212. August 1983.

3. **Engel, Carl D. and Schmitz, Craig P.** *MINIVER Upgrade for the AVID System, Vol 2: LANMIN Input Guide.* NASA CR-172213. August 1983.

4. **Pond, John E. and Schmitz, Craig P.** *MINIVER Upgrade for the AVID System, Vol 3: EXITS User's and Input Guide.* NASA CR-172214. August 1983.

5. **Wurster, Kathryn E.** *MINIVER - A Versatile Aerothermal Analysis / TPS Design Tool.* Hampton, Virginia, United States of America : s.n., October 2000. Powerpoint Presentation.

6. *Assessment of Thermochemical Nonequilibrium and Slip Effects for Orbital Re-Entry Experiment.* **Gupta, Roop N, Moss, James N and Price, Joseph M.** 4, Hampton, Virginia : s.n., 1997, Journal of Thermophysics and Heat Transfer, Vol. 11.

7. **Hirschel, Ernst Heinrich and Weiland, Claus.** *Selected Aerothermodynamic Design Problems of Hypersonic Flight Vehicles.*

8. **Japan Aerospace Exploration Agency.** OREX. *Japan Aerospace Exploration Agency.* [Online] 2007. [Cited: April 17, 2013.] http://www.rocket.jaxa.jp/fstrc/0c01.html#1.

9. **Engel, Carl D. and Schmitz, Craig P.** *MINIVER Upgrade for the AVID System, Volume 2: LANMIN Input Guide.* Huntsville : Remtech, Inc., 1983.

10. **Engel, Carl D. and Praharaj, Sarat C.** *MINIVER Upgrade for the AVID System, Volume 1: LANMIN User's Manual.* Huntsville : Remtech, Inch., 1983.

11. Pages - Publications. [Online] [Cited: January 03, 2012.] http://www.wsmr.army.mil/RCCSITE/Pages/Publications.aspx?RootFolder=%2fRCCsite%2fDocuments%2fRange%20Reference%20Atmospheres&FolderCTID=&View=%7B093DEFF5-C504-4518-B320-B3D05428987A%7D.

Appendix

// placeholder; any relevant appendices here