AE 4080 Computer Project: Computations for Hypersonic Blunted-Cone Vehicle Using Matlab

Matt Altis

AE 4080

Summer Semester 2021

7-21-2021

Dr. Dec

Introduction/Methodology

Throughout the course, multiple different aspects of hypersonic flow have been studied and analyzed. This includes looking into elements such as shocks and analyzing the conditions both upstream and downstream. Other aspects of aerodynamics and vehicle performance were analyzed, such as the lift and drag forces acting on the vehicle of interest. Additionally, multiple methods for heat transfer analysis were studied in order to find the heat flux at locations of interest along the surface of our hypersonic vehicle. While these studies include a variety of different fields commonly used in engineering, including thermo/aerodynamics, and even a bit of chemistry, these seemingly different analyses are strongly interrelated when it comes to understanding the big picture analysis of hypersonic flows and flight.

Because of this strong relation between the varying topics covered in the class, it is a very worthwhile endeavor to create a computer program that would take in a set of inputs regarding the vehicle's geometry, flight speed, and altitude. Even with just a select few given conditions, a variety of aerodynamic and thermodynamic properties can be calculated extremely quickly provided a functioning code. A single Matlab function was written, called BluntedCone.m, that performed all of the assigned computations given function input parameters.

<u>Running The Matlab Code/Comments</u>

In order to run the code, first ensure that the attached file titled 'BluntedCone.m' is saved in the current directory within Matlab. Then, in order to run the code, the following line should be input into the command window, and then press enter:

```
[pinf,Tinf,rhoinf,p2,h2,rho2,T2,u2,eps,iterationHist,CL,CD,
qstag,qref] = BluntedCone(z,M1,x,dc,rn,rc)
```

In the above code, the inputs are as follows:

z	Geometric Altitude (km)
M1	Mach Number of Vehicle
x	x-location from cone section (For reference temperature method)
dc	Cone Angle in Degrees
rn	Nose radius in m
rc	Cone base radius in m

An example input, such as the one used for the heat flux results, would be the following:

```
[pinf,Tinf,rhoinf,p2,h2,rho2,T2,u2,eps,iterationHist,CL,CD,
qstag,qref] = BluntedCone(18.28,7,.3,35,.03048,.32892)
```

The output variables from the code are to be interpreted as the following:

pinf	Freestream pressure (Pa)
Tinf	Freestream temperature (K)
rhoinf	Freestream density (kg/m^3)
p2	Downstream pressure (Pa)
h2	Downstream enthalpy (J/kg)
rho2	Downstream density (kg/m^3)
T2	Downstream temperature (K)
u2	Downstream velocity (m/s)
eps	epsilon value for normal shock
iterationHist	Table of iteration values for shock calculations
CL	Vector of CL as function of Angle of Attack
CD	Vector of CD as function of Angle of Attack
qstag	Heat flux at stagnation point, in W/cm^2
qref	Heat flux at x-location input of interest, in W/cm^2

1b) Free Stream Atmospheric Conditions

The first step of the process involved computing the freestream conditions for pressure, temperature, and density. Given an input for geometric altitude, z, the atmospheric conditions could be obtained with use of the atmospheric tables provided in class. For this, two arrays were created, one for regions 1-6, and another for regions 7+. This array contained necessary values for calculation, such as T_{mi}, L_{Hi}, and h_i. These arrays are coded into the function, so no input is necessary for these baseline values. Then, for the given geometric altitude, the geopotential altitude (h) was computed.

With all inputs ready for the atmospheric condition computation, a series of conditional 'if' and 'elseif' statements were used to locate the region, based on the value of h (regions 1-6) or z (regions 7+). For each region, the equations used were similar to find p, rho, and T, with the only major variation being the values indexed from the atmospheric table arrays. Additional variation occurred in isothermal layers, where the temperature lapse rate was 0. Final adjustments (multipliers) to certain equations were added to ensure the end result for each layer contained the units for Pascals, Kelvin, and kg/m^3.

1c) Computing Conditions Downstream of Normal Shock

Before being able to compute the lift and drag coefficients for the blunted cone, as well as the heating of the stagnation point, the atmospheric conditions downstream of the normal shock are required. The first step in this process involved computing the upstream enthalpy using Cp = 1004.5 J/kg-K and the value for T_{inf}. Then, the speed of sound at the given altitude was found. With this, using the given input for the Mach Number of the vehicle, the upstream velocity u1 was computed.

For the input mach number, if the input was less than 3, the standard normal shock equations (eq 17-19) were used to find the downstream conditions, assuming a calorically perfect gas. However, for mach numbers greater than 3, the gas is assumed to be a chemically reacting gas, and an iterative equilibrium solution was utilized to obtain the downstream conditions.

The iterative process began by defining a few terms that were necessary, such as the upstream velocity, enthalpy, as well as the c_n array for gamma tilda, and the d_n array for the temperature curve fit. Then, starting values for downstream pressure (p2) and rho (rhostart) were computed using an initial guess for epsilon, eps = .1. With these guesses, X, Y, and Z values were computed to get values for gamma tilda constants. Again, a series of conditional 'if' and 'elseif' statements were used to pinpoint which row (n) of the large c_n array would be indexed. Once this numerical variable n was created, the curve fit equation could be created by indexing the row number 'n' of the array.

The next step in this solution procedure involved preparing two different while loops to drive the difference in epsilon to 0 by iterating through values of p2, h2, and rho2. The iteration would continue until a variable called Deps is greater than a threshold value close to 0. Deps was initially set to '1' in order for the loop to start. Within this first loop for the change in epsilon, values for p2, h2, and rho2 (called 'rhoguess') were computed with equations 14 and 15, using the current value for epsilon. An arbitrary value for h2j was also assigned in order for the nested loop to begin.

The inner loop was also a while loop, and its purpose was to converge this initial h2j value with the current value for h2 that is currently stored within the first loop. To do this, the loop would iterate until h2j and h2 converge. Within the loop, new values for X,Y, and Z were computed, using the current p2 and rho2 values. Then, gamma tilda was computed using the curve fit equation, using the index variable 'n' to extract values for the correct region. With this gamma value, a new value for h2j was computed with equation 20. Then, the variable for rho2 was changed by a small amount for the next iteration.

Once inner loop finishes iterating, a new value for Deps, or change in epsilon, is created from a vector that stores epsilon values for each iteration. By subtracting the most current value from the previous iteration's value, the change in epsilon after each iteration can be recorded and driven to 0.

After both loops have finished the iterations, downstream values for epsilon, p2, rho2, and h2 are obtained. Inside of the loop, to keep track of each iteration, a vector for each of the aforementioned variables was created, which concatenates/adds the current value to the vector of values from previous iterations. This is the same process that was done for the Deps vector. These vectors were then all combined into a table called 'iterationHist' that is output by the function to check the values after each iteration.

For temperature, a similar process was used, where a temperature region was assigned based on the values for X, Y, and Z. The array for temperature curve fit coefficients was then indexed according to a numerical value, m, which was dependent on the temperature region determined from the series of 'if' statements. This curve fit equation was used to obtain the temperature ratio T2/T0, which was then converted to T2 by multiplying by the value of 151.78. Finally, using the new value for epsilon, the downstream velocity u2 was computed. With all of these values calculated, it is now possible to analyze heat transfer at the stagnation point, as well as compute the lift and drag coefficients of the blunted cone.

1a) Lift and Drag Coefficient Calculations

For the computation of these aerodynamic coefficients, modified Newtonian theory was used for a blunted-cone vehicle, for which values of nose and cone base radius were given in meters. From the value of epsilon obtained in the previous downstream properties computation, a new value for maximum pressure coefficient was obtained using equation 27.

To obtain the coefficients at all the different angles of attack of interest, a for loop was created for a=0:10. Empty vectors entitled 'CL' and 'CD' were created and added to with each iteration. For each angle of attack, the normal and axial force coefficients were obtained using equations 36 and 38. Then, using equation 28, the lift and drag coefficients were computed, outputting a vector of values.

2a-Part 1) Stagnation Point Heat Flux

For the heat flux at the stagnation point downstream of the normal shock, the Fay-Riddell correlation was used for computation. Using this method requires knowledge of certain downstream conditions, which were computed in section 1c. However, transport properties, such as viscosity (mu) and conductivity (k), were also required. The sophisticated model used in class was integrated into the code to compute these transport properties, as the conditions behind the normal shock of interest were on the boundary of using the simple vs sophisticated model.

The gas composition behind the normal shock was provided as a 5 species model, consisting of N_2, O_2, N, O, and NO. The mole fractions and molecular weights for each species were provided and were each input as a vector. This allowed for convenient computation of new variables during the process of finding the transport properties, as element by element arithmetic operations between vectors allowed for minimal lines of code to be written.

Using data provided from the Bird document, sigma and epsilon/k values for each species were input as vectors. With these values, $\text{omega}_{mu,k}$, mu_i, and Xmu_i values could be computed for each species. The next challenge involved obtaining an array of $\text{phi}_{i,j}$ values, which are needed to compute the sum($X*\text{phi}_{i,j}$) vector. To do this, 2 'for' loops were used to iterate through species 'i' and then through all 'j' species, before moving on to the next i. An empty array was used to begin the iteration, where the outer loop would output a 5x5 vector of phi values. The inner loop was where the phi computation took place. For each of the 5 inner loop iterations, a 1x5 vector was created and added on to the array. After all iterations completed, a 5x5 array of $\text{phi}_{i,j}$ values were obtained.

A similar loop was used to obtain the sum of $X*\text{phi}_{i,j}$, where each subsequent loop would move on to the n+1 row of the 5x5 array of phi values. From here, the mu value for each species was computed, summed together to obtain a value for the entire mixture, and converted to units of kg/m-s. Similarly, conduction (k) was computed using values for x*k. Obtaining these values depended on whether the species was diatomic of the same element (ie H_2, N_2). Once x*k was computed for each species and converted to consistent units of W/m-k, the conductivity of the mixture was again found by summing the k value for each species.

For the stagnation point heat flux calculation, a cold-wall assumption of T_w=300 K was used to obtain the enthalpy, viscosity, conductivity, and density at the wall. Using the downstream conditions provided from the normal shock calculations, equation 89 was used, assuming a Lewis number of 1, to obtain the heat flux at the stagnation point.

2a-Part 2) Heat Flux At X=.3 m Along Cone Section

Because there was not a way to calculate the shock wave angle for the given conditions, conditions downstream of the oblique shock were input into the code, rather than using values that were computed previously. A new input into the function, X = .3 m , appears in this section as well. The process for reference temperature first involved computing the equivalent gamma value using a curve fit, which was used to find the mach number at the boundary layer edge. This then let us find the reference temperature, which was used to compute convection and conduction coefficients using the simple model, as T_{ref} was under 2000 K. Another curve fit equation was used for the cp_{ref}, and the vehicle geometry was used to compute the running length, s.

The final part of the reference temperature method involved computing the reference Reynolds number, which would indicate turbulent flow if it was over 10,000. An 'if' statement was used to compute m_f and the recovery factor depending on laminar or turbulent flow.

Finally, C_H and the recovery enthalpy, H_R, were used to compute the heat flux at the x-location of interest, using equation 101.

Results and Discussion

1b) Freestream Atmospheric Conditions Results

The two altitudes of interest in the test cases are z = 12.2 km and z = 18.28 km. Both of these altitudes fall well below z = 86 km, so the equations for regions 1-6 will be use in this test case. The results for the atmospheric conditions are shown in table 1 below.

Table 1. Freestream Atmospheric Calculations

	Z = 12.2 km	Z = 18.28 km
p (Pa)	18880	7240.3
rho (kg/m^3)	.3023	.1164
T (K)	216.65	216.65

From the table above, it is expected that there will be no change in temperature, as both altitudes correspond to region 1, which is an isothermal layer.

1c) Shock Downstream Conditions Results

For the normal shock computations, there were four different mach number and altitude combinations that were of interest for lift, drag, and heat flux calculations. The results for each of these cases is depicted in Table 2 below.

Table 2. Downstream Shock Conditions for Different Cases

	M = 5 Z = 12.2	M = 5 Z = 18.28	M = 7 Z = 12.2	M = 7 Z = 18.28
p2 (pa)	5.5466E05	2.1351E05	1.101E06	4.2402E05
Rho2 (kg/m^3)	1.627	.6253	1.882	.7233
h2 (J/kg)	1.268E06	1.268E06	2.295E06	2.295E06
T2 (K)	1170	1179	2021.3	2025
u2 (m/s)	274.09	274.689	331.74	332.4581
Ɛ	.1858	.1862	.1606	.1610

Additionally, for each of the above cases, the iteration history for the rho and enthalpies are shown in Tables 3-6:

Table 3. M = 5, Z = 12.2 km

i	eps	p2	h2	rho2	Deps
0	0.17196	6.1089e+05	1.2949e+06	1.758	0.071958
1	0.18355	5.6355e+05	1.2736e+06	1.647	0.011589
2	0.18546	5.5592e+05	1.2691e+06	1.63	0.0019143
3	0.1858	5.5466e+05	1.2683e+06	1.627	0.00034197

Table 4. M = 5, Z = 18.28 km

i	eps	p2	h2	rho2	Deps
0	0.17216	2.3528e+05	1.2949e+06	0.67626	0.072161
1	0.18385	2.1699e+05	1.2735e+06	0.63326	0.01169
2	0.18591	2.1403e+05	1.269e+06	0.62626	0.002055
3	0.1862	2.1351e+05	1.2681e+06	0.62526	0.00029732

Table 5. M = 7, Z = 12.2 km

i	eps	p2	h2	rho2	Deps
0	0.15191	1.1793e+06	2.329e+06	1.99	0.05191
1	0.15936	1.1124e+06	2.3011e+06	1.897	0.0074474
2	0.16046	1.1028e+06	2.2962e+06	1.884	0.0010996
3	0.16063	1.1013e+06	2.2954e+06	1.882	0.00017052

Table 6. M = 7, Z = 18.28 km

i	eps	p2	h2	rho2	Deps
0	0.15214	4.5419e+05	2.329e+06	0.76526	0.052139
1	0.15965	4.283e+05	2.301e+06	0.72926	0.0075103
2	0.16075	4.2457e+05	2.296e+06	0.72426	0.0011021
3	0.16097	4.2402e+05	2.2952e+06	0.72326	0.00022226

From the results in table 2, it is clear that some aspects of the downstream condition vary greatly depending on both altitude and flight speed, such as pressure and density. Other conditions, such as downstream temperature, enthalpy, and velocity, are much more dependent on the flight speed.

1a) Lift and Drag Coefficient Results

Using the epsilon values calculated from each of the four test cases above, lift and drag coefficients were then computed for a 35˚ blunted cone, with a nose radius of R_n = .03048 m, an R_C = .32893 m. For each case, the angle of attack was varied from 0-10 degrees, and the coefficients were computed as a function of this angle of attack. The results are shown in tables 7-10, as well as plotted in figures 1 and 2 below.

```
[pinf,Tinf,rhoinf,p2,h2,rho2,T2,u2,eps,iterationHist,CL,CD,qstag,qref] =
BluntedCone(12.2,5,3,35,.03048,.32892)
```

Table 7. M = 5, Z = 12.2 km

Alpha (Degrees)	CL	CD
0	0	.6004
1	.0106	.6006
2	.0212	.6015
3	.0317	.6029
4	.0421	.6048
5	.0524	.6073
6	.0624	.6104
7	.0721	.6139
8	.0816	.6180
9	.0907	.6225
10	.0995	.6275

Table 8. M = 5, Z = 18.28 km

Alpha (Degrees)	CL	CD
0	0	.6002
1	.0106	.6005
2	.0212	.6013
3	.0317	.6027
4	.0421	.6047
5	.0523	.6072
6	.0623	.6102
7	.0721	.6138
8	.0816	.6178
9	.0907	.6224
10	.0995	.6274

Table 9. M = 7, Z = 12.2 km

Alpha (Degrees)	CL	CD
0	0	.6087
1	.0108	.6090
2	.0215	.6098
3	.0322	.6112
4	.0427	.6132
5	.0531	.6158
6	.0632	.6188
7	.0731	.6224
8	.0827	.6265
9	.0920	.6311
10	.1009	.6362

Table 10. M = 7, Z = 18.28 KM

Alpha (Degrees)	CL	CD
0	0	.6086
1	.0108	.6089
2	.0215	.6097
3	.0322	.6111
4	.0427	.6131
5	.0531	.6156
6	.0632	.6187
7	.0731	.6223
8	.0827	.6264
9	.0920	.6310
10	.1009	.6361

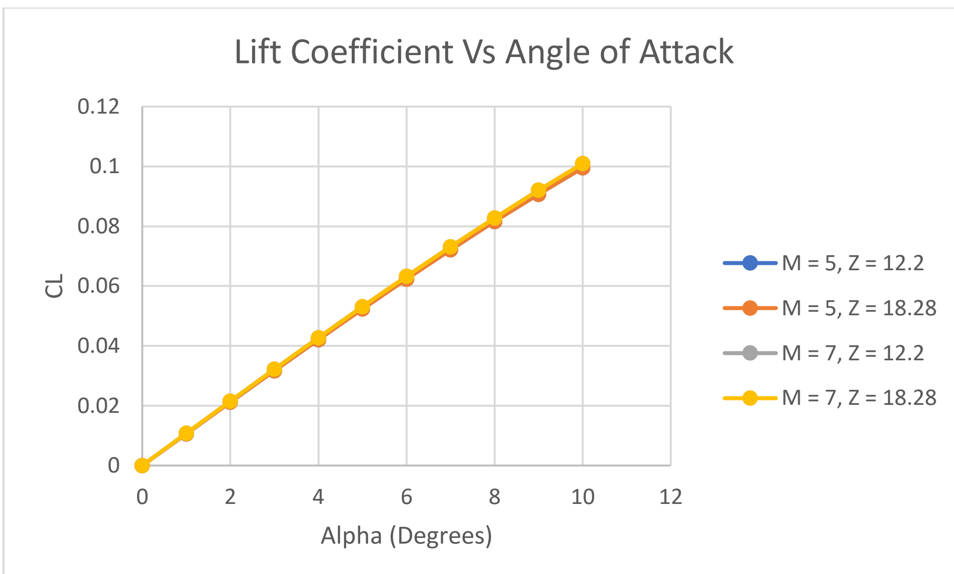

Figure 1. Lift Coefficient vs Angle of Attack

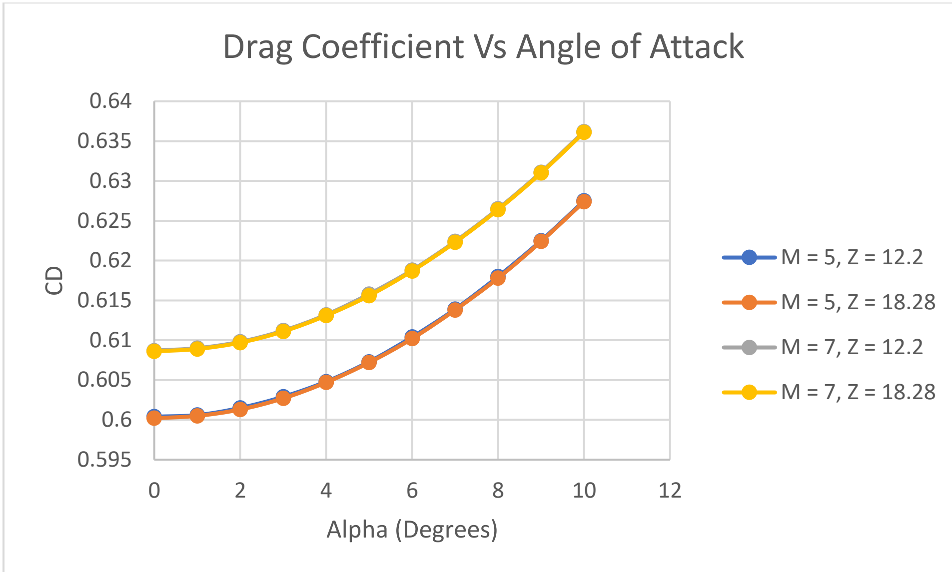

Figure 2. Drag Coefficient vs Angle of Attack

From the tables and figures above, we see that the blunt body produces a much higher drag coefficient than lift coefficient. The cone section of the vehicle would not be able to reliably produce a large amount of lift on its own, so other lifting surfaces would likely be necessary in the event that a desired lift force needs to be met. Variations in lift were minimal across the four test cases, mainly due to epsilon not changing considerably amongst the tests. There was slightly more variation among drag coefficient, where increasing the Mach number of the vehicle produced a larger drag force, which is expected.

2a- Heating Results

For the stagnation point heat flux, a Mach number of 7 and altitude of 18.28 km were used. At these conditions, the heat flux at the stagnation point behind the normal shock is estimated to be roughly 230.0558 W/cm^2, using the Fay-Riddell correlation. For the oblique shock, using the reference temperature method with the given downstream conditions, the heating at a location of x = .3 m along the cone came out to a much smaller value of 13.782 W/cm^2. It is expected that for a given shock, there would be less heating at a point farther down the cone behind an oblique shock, as the downstream conditions at this point (namely temperature and velocity) would result in a smaller heat flux. However, more certain results could be obtained by adding in an additional section for downstream oblique shock conditions, coupling the normal and oblique downstream conditions and reducing any relative error.