Visual C++ 프로그램을 이용한 Ramjet Simulation Code의 생성 방법

공창덕* · George Omollo Owino*

Use of Visual C++ program to generate Ramjet Simulation code

George Omollo Owino* · Changduk Kong**

ABSTRACT

This paper presents on research findings of how Visual C++ program can be used to generate codes capable of performing ramjet engine simulation an arbitrary ramjet model will be considered for which generated output values will be compared with those from a commercial program GASTURB 9.

Several governing thermodynamic equations will first be discussed in order that we understand the fundamental idea behind values printed out on the GUI. The program is designed that it generates its station input value. Similar results were realized compared to those produced by gasturb 9.

초 록


Key Word: Ramjet, Performance Simulation, Simulation code, Visual C++

† 2008 년 5월 9일 접수 ~ 2008 년 6월 9일 심사완료
* 정회원, 조선대학교 항공우주공학과
** 정회원, 조선대학교 항공우주공학과
공창덕, cdgong@chosun.ac.kr
광주광역시 동구 서석동 375

1. Introduction

It is a necessity in the design of a ramjet engine to know pressures, temperature,
velocities and flow areas at each point along the gas path as they are used to estimate stage performance of the given engine.

Various analytical methods of varying degree of accuracy are in common usage for these calculations. It is of general practice to assume that the flow we will consider here is one-dimensional across the passage to avoid complex flow patterns in actual flow regimes. We will consider the average specific heat method of calculation to try and limit errors most prevalent in other simpler methods like arbitrary and constant specific methods.

It would be more easy and accurate to evaluate defined dynamic model with given known values of \( K \) (pressure loss coefficient) \( C_d \) (nozzle discharge coefficient) and other parameters only determined after research. However though we shall assume ideal conditions where such values are needed.

### II. Model geometry

#### 2.1 station numbering

A convergent-divergent ramjet model with station numbering as used in this paper is represented in figure 1 below it should be noted that intake ramp were not considered.

Respective model areas of interest necessary for calculating air mass flow and nozzle area ratio are given in the following table 1 below.

<table>
<thead>
<tr>
<th>Area</th>
<th>1 (Inlet)</th>
<th>5 (Nozzle throat)</th>
<th>e (nozzle exit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M^2 )</td>
<td>0.0257</td>
<td>0.033</td>
<td>0.0615</td>
</tr>
</tbody>
</table>

#### 2.2 Operational condition

Flight and environmental conditions at 50000ft was selected to be the operational altitude which gives us the environmental ISA conditions values of pressure, temperature, and density \( P_T \rho \) respectively.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Temperature</th>
<th>Pressure</th>
<th>Mach no</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>216.65 (K)</td>
<td>12112 (N/M^3)</td>
<td>3</td>
</tr>
</tbody>
</table>

#### 2.3 Theory

We calculate total pressure, temperature and inlet air mass flow rate for Mach number 3 taken to be our flight mach number. These values are used to calculate different states of stage 1.

\[
\begin{align*}
\frac{T_1}{T_T} &= 1 + \frac{\gamma - 1}{2} M^2 \\
p_1 &= \left(1 + \frac{\gamma - 1}{2} M^2 \right)^{\frac{\gamma}{\gamma - 1}} \\
\dot{M} &= M \rho A
\end{align*}
\]

Similarly stage 2 diffuser downstream conditions calculated for static, total temperature and pressures. Diffuser pressure ratio and burner entry Mach number may also be determined at this stage.

\[
\begin{align*}
P_1 &= \frac{T_1}{T_T} \left(\frac{T_1}{T_T}\right)^{\frac{\gamma - 1}{2} M^2} \\
T_{1,2} &= \frac{T_1}{1 + 0.2 M_2^2} \\
M_2 &= \sqrt{1 + \frac{\gamma - 1}{2} M_1^2} \\
&\quad \sqrt{\frac{\gamma M_1^2}{\gamma M_1^2}}
\end{align*}
\]

Calculations for stage 3 commences with the assumption that momentum is conserved, although not practical, pressure and friction
losses are assumed so small to be ignored.

Flame holder drag $K$ was taken as 1. It should however be determined experimentally

$$P_{ch} = 1 - \frac{K \gamma M^2 \frac{\gamma - 1}{2}}{P_{in}}$$

This formula opens a way to calculate stage 3 total temperature. Further assumptions made at stage 4 (combustor) are that the combustor is of constant area passage and supplied with liquid fuel.

Given combustor total exit temperature, Mach number $M_{ch}$ is calculated, that also serves as reheat entry Mach number, although gamma value for air is ($\gamma = 1.4$) we will consider it to be $\gamma = 1.3$ due to temperature effect from the combustor inlet onwards.

Nozzle calculations for stage 5 start with the assumption that the nozzle is choked $M_{ch} = 1$

$$T_e = \frac{T_{ch}}{1 + \frac{\gamma - 1}{2} M_{ch}^2}, \quad V_e = M_{ch} \sqrt{R \frac{T_e}{\gamma}}$$

Using the area-mach relation exit Mach number is calculated

$$\frac{A_{in}}{A_{ch}} = \left( \frac{\gamma + 1}{\gamma} \right)^{\frac{\gamma+1}{2(\gamma-1)}} \left( \frac{M_{ch}}{(1 + \frac{\gamma - 1}{2} M_{ch}^2)^{\frac{\gamma+1}{\gamma-1}}} \right)$$

$$T_e = \frac{T_{ch}}{1 + \frac{\gamma - 1}{2} M_{ch}^2}$$

Once all stage 5 is done the final stage 6 calculations start with pressure at exit taken to be equal to ambient $P_e = P_{ambient}$

Different performance parameters are then determined.

All the above calculations and more form the fundamental backbone idea behind the programs working. It is at this stage that the constants and governing equations are edited into visual C++ compiler.

### III. Input values

<table>
<thead>
<tr>
<th>ITEM</th>
<th>Altitude</th>
<th>Mach no</th>
<th>Temperature</th>
<th>Pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>GASTURB 9</td>
<td>50000ft</td>
<td>3</td>
<td>216.65</td>
<td>12.045</td>
</tr>
<tr>
<td>C++ GUI</td>
<td>50000ft</td>
<td>3</td>
<td>216.65</td>
<td>12.045</td>
</tr>
</tbody>
</table>

Similar inputs were maintained for both cases to ensure output value comparison derived using the same preceding conditions.

3.1 GASTURB 9

This is a GUI type commercial program capable of performing steady state ramjet simulation; it involves entering the above table
3 inputs to produce thermodynamic calculation results of each stage

![GASTURB 9 input window](image)

It should be noted that geometric area cannot be edited in this program.

![GASTURB 9 output window](image)

3.2 Visual C++ GUI

This GUI is designed with the left hand side having editable input window with the output on the right side.

The reset button sets the default values at that pre chosen altitude and also used to clear the output window for an alternate command simulation.

![Visual C++ Output GUI](image)

The run button allows the program to start computing the given inputs through governing equations to generate the output values printed on the right hand side.

3.3 Output results

Table 4 below shows simulation results that were generated by both programs using environmental conditions at 50000ft and Mach number 3 as input

Reference should be made to figure 1 for station numbering, although similar, different numbering method are used in Gasturb 9 from station 4.

![Temperature Ratio against Mach number](image)

Respective values of temperature ratio at station two were generated by changing Mach number in the visual C++ GUI then plotted for values of gamma between 1.25 to 1.4

It can be concluded from the graphs that
constant increase in temperature occurs with increasing Mach number

Pressure ratio value for mach numbers between 0.5 to 4.0, plotted at varying values of gamma constant pressure is observed up to mach one where rapid increase is noted

Table 5 Performance output data

<table>
<thead>
<tr>
<th>Item</th>
<th>Thrust (kN)</th>
<th>Inlet Press Ratio</th>
<th>Reheat Mach no</th>
<th>Nozzle Exit Mach</th>
<th>Nozzle Area Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gasturb</td>
<td>9.12</td>
<td>1</td>
<td>0.433</td>
<td>2.029</td>
<td>1.862</td>
</tr>
<tr>
<td>C++</td>
<td>10.33</td>
<td>1</td>
<td>0.475</td>
<td>2.115</td>
<td>1.862</td>
</tr>
</tbody>
</table>

The difference in thrust may be attributed to the fact that Gasturb 9 considered pressure loss of 6.81% whereas we assumed ideal conditions and losses were ignored.

Table 6 Performance with varying Mach no

<table>
<thead>
<tr>
<th>Mach no</th>
<th>Units (KN)</th>
<th>(Gasturb 9)</th>
<th>(Visual C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>2</td>
<td>3.12</td>
<td>3.42</td>
</tr>
<tr>
<td>2</td>
<td>2.5</td>
<td>5.41</td>
<td>4.788</td>
</tr>
<tr>
<td>2.5</td>
<td>3</td>
<td>9.12</td>
<td>6.97</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>10.33</td>
</tr>
</tbody>
</table>

Since Gasturb 9 has no input provision for intake area which was considered in Visual C++ we may assume that they used the theory that the frontal area equals exit area. Hence explains great difference in thrust at low Mach numbers

3.4 Performance

One amongst the most importance performance defining parameters is the fuel flow rate.

Calculated herein as

\[
f = \frac{T_{14}}{T_{13}} - 1 = 0.029166 \quad (9)
\]

\[
M_f = 3600 \frac{fM_a}{F} \quad (10)
\]

\[
sfc = \frac{M_f}{F} = 0.04679667340 \quad (11)
\]
This graphs were plotted from performance results output of the simulation code program developed using visual C++ both graphs show expected trend

Fig. 10 Specific fuel consumption against thrust

The graph indicate that specific fuel consumption reduces with increasing thrust, this is so because specific fuel consumption is inversely related to thrust as defined in equation (11).

IV. Conclusion

From the analysis results we experienced close or similar values of temperature and pressure for the stations considered; it would be adequate hence to conclude that Visual C++ program is accurate as may be verified by the commercial program used.

Net thrust also reflects closeness the difference of which may be attributed to frictional pressure losses considered in Gasturb 9.

This code generated by C++ compiler may be suitable to simulate defined dynamic model as it allows editing of geometric data like intake area, combustor temperatures to match desired values

Visual compiler being readily available would be a suitable tool for making simulation codes at intellectual level with additional advantage of being used for commercial purposes due to its numerous interface capabilities with other programs.

References

1. High Speed Aerodynamics and Jet Propulsion Text by O. E Lancaster 1959 pp268
2. Ramjet Gas dynamics Lecture note Cranfield University
3. C++ Programming for Engineers Text Book
4. Fluid Dynamics Tables pp 301
5. NASA .Glenn research webpage http://www.nasa.gov/centers/glenn/home/index.html