REDUCING SKIN FRICTION AND HEAT TRANSFER OVER A HYPERSONIC CRUISING VEHICLE BY MASS INJECTION

By

YOSHIFUMI NOZAKI

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2007

To my parents, Yoshikazu Nozaki and Keiko Nozaki, and my brother, Toshihiro Nozaki. You have inspired me to become who I am today. Thank you for always supporting and believing in me. I dedicate this to you.

ACKNOWLEDGMENTS

I gratefully acknowledge the support provided by Dr. Pasquale M. Sforza, Professor of Mechanical and Aerospace Engineering at the University of Florida.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

REDUCING SKIN FRICTION AND HEAT TRANSFER OVER A HYPERSONIC CRUISING VEHICLE BY MASS INJECTION

By

Yoshifumi Nozaki

August 2007

Chair: Pasquale M. Sforza
Major: Aerospace Engineering

Demonstrating technologies for hypersonic aircraft that cruise at speeds greater than Mach 5 is one of the long-term visions of many agencies, like NASA. Reducing skin friction and heat transfer on the surface of hypersonic cruising vehicles has been a focus of constant attention.General methods for estimating the aerodynamic forces and heat transfer around a hypersonic vehicle are used to evaluate the reduction in skin friction and heat transfer on the surface of a hypersonic vehicle by mass injection. Particular attention was paid to the X-24C configuration because of the existence of experimental data of X-24C performance against which the predictions can be compared. The local surface inclination method and the flat plate reference enthalpy methods for laminar and turbulent flow were used to find aerodynamic forces and heat transfer. High temperature effects were included by using a classical approximation of thermodynamic properties. Although this analysis is based on many approximations, these methods worked well and flow properties were reasonably predicted. Reducing skin friction and protecting surfaces from heating by injecting mass did result in a penalty in the form of decreased flight time of the vehicle, and therefore flight range. These penalties were often very light. Also, as more mass is injected, the effect of mass injection grows, but more slowly.

CHAPTER 1
INVISCID ANALYSIS

Introduction

Hypersonic flow is complicated because of physical aspects of hypersonics such as high-temperature chemically reacting, thin shock layer, etc. Such complex phenomena cannot be described by a simple linear system. Even without these phenomena, the basic theory of inviscid compressible flow, when the Mach number is very large, does not yield aerodynamic theories which are mathematically linear. By using supersonic thin airfoil theory, the pressure coefficient on the surface is obtained from

$$c_p = \frac{2\theta_i}{\sqrt{M_\infty{}^2 - 1}} \tag{1-1}$$

where M_∞ is the free stream Mach number, and θ_i is flow inclination. Eq.(1-1) is a classical result from inviscid, linearized, two-dimensional, supersonic flow theory.[1] This method is called the local surface inclination method, and it is very simple and easy to use to predict c_p. This method does not need a detailed solution of the complete flowfield. This simplicity is very useful, but unfortunately, it is not valid for hypersonic speeds since nonlinear effects become important at high Mach number. However, there are other valid local surface inclination methods, and some of those are presented, which are applied to our hypersonic bodies.

Local Surface Inclination Method

As M_∞ approaches ∞ and γ approaches 1, the shock layer becomes coincident with the body surface. This is because the density ratio across the shock approaches zero, and since the density behind the shock is so high the shock layer becomes thinner and thinner. Therefore, it looks as if the incoming flow is directly impinging on the wedge surface, and then is running parallel to the surface downstream. Under these conditions, Newtonian theory is used to find c_p.

For blunt bodies the modified Newtonian theory should be used, and such results usually produce acceptable accuracy. In contrast, Anderson[2] suggests that Newtonian results for slender bodies should use the straight Newtonian theory. Newtonian theory works reasonably well alone and lends itself to application to arbitrary slender body shapes. In the Newtonian model of fluid flow, the particles in the free-stream impact only on the frontal area of the body; they cannot curl around the body and impact on the back surface. Hence, for that portion of a body which is in the "shadow" of the incident flow, no impact pressure is left, so over this "shaded" region the Newtonian theory is inaccurate. Therefore the Newtonian theory is described by the following equations:

$$c_p = 2 \cdot \frac{\left(\vec{V}_\infty \cdot \hat{n}\right)^2}{\vec{V}_\infty \cdot \vec{V}_\infty} \text{ for } \vec{V}_\infty \cdot \hat{n} < 0 \tag{1-2}$$

$$c_p = 0 \text{ for } \vec{V}_\infty \cdot \hat{n} \geq 0 \tag{1-3}$$

where $\hat{n}$ is unit outward normal vector on the body surface. The above equations are locally applicable to every surface panel on a smooth body. In order to have more accurate results, Prandtl-Meyer expansion theory should be applied for the surface panel at which the Newtonian method is inaccurate. In this study, we use Newtonian theory for all surface panels except for the part in the "shadow", at which we applied Prandtl-Meyer expansion theory. In practice, the body surface of a vehicle is subdivided into a number of individual panels and each panel is treated separately to determine the pressure force acting. In this study, the surface of the X-24C vehicle is divided into 284 panels as described subsequently. The normal vector of each panel can be found by the cross-product of the *P* and *Q* vector as shown in Figure 1-1.

$$\vec{N} = \vec{P} \times \vec{Q} \tag{1-4}$$

$$\hat{n} = \frac{\vec{N}}{\sqrt{\vec{N} \cdot \vec{N}}} \tag{1-5}$$

$$dA = \frac{1}{2}\left|\vec{P} \times \vec{Q}\right| = \frac{1}{2}\left|\vec{N}\right| \tag{1-6}$$

where dA is differential area of individual panel. Once having the unit normal to the surface element, the pressure force acting on each panel can be found by:

$$d\vec{F}_{press} = -\left(c_p q + P_\infty\right) \cdot dA \cdot \hat{n} \tag{1-7}$$

$$q = \frac{1}{2}\rho_\infty {V_\infty}^2 \tag{1-8}$$

where $d\vec{F}_{press}$ is the pressure force acting on individual panel and q is the dynamic pressure.

Inviscid Aerodynamic Forces and Moments

Before determining differential lift(dL) and drag(dD) on each panel, the angle of attack must be accounted for the flow condition. In this analysis, the angle of attack is defined as the angle between the free stream velocity vector (or x-axis) and the fuselage reference line of the vehicle. Therefore, each node point must be rotated with respect to y-axis as the angle of attack increases (the origin of the coordinate is located at the nose of the vehicle). The following equations are used to change each node on the surface according to variable angle of attacks:

$$x' = \sqrt{x^2 + z^2}\cos\left(\tan^{-1}\left(\frac{z}{x}\right) - \alpha\right) \tag{1-9}$$

$$y' = y \tag{1-10}$$

$$z' = \sqrt{x^2 + z^2}\sin\left(\tan^{-1}\left(\frac{z}{x}\right) - \alpha\right) \tag{1-11}$$

(x, y, z): a node at $\alpha = 0$
(x', y', z'): a node at nonzero α

Having the differential pressure force acting on each panel with an accounting for the angle of attack, differential lift(dL) and drag(dD) on each panel are found as:

$$dL_{invisc} = dF_{press,z} \tag{1-12}$$

$$dD_{invisc} = dF_{press,x} \tag{1-13}$$

$$d\vec{F} = dF_{press,x}\hat{i} + dF_{press,y}\hat{j} + dF_{press,z}\hat{k} \tag{1-14}$$

$$d\vec{F} = d\vec{P} \tag{1-15}$$

where dL_{invisc} and dD_{invisc} are differential lift and drag caused by only the pressure acting on the vehicle. Note that the free stream is always directed in the positive x-direction, and the vehicle rotates with respect to y-axis and the free stream flow direction never changes even though the angle of attack changes. Therefore, lift (L) and drag (D) are always directed in the z-direction and x-direction, respectively. However, the angle of attack changes the normal vector on each surface since the P and Q vectors of each surface change with angle of attack.

The differential moment of each panel about a reference point, say the center of mass, due to the differential force on a particular body surface element is given by:

$$d\vec{M}_0 = \vec{r} \times d\vec{F}' = \left[(x'' - x_0)\hat{i} + (y'' - y_0)\hat{j} + (z'' - z_0)\hat{k}\right] \times d\vec{F}' \tag{1-16}$$

$$d\vec{M}_0 = dl\hat{i} + dm\hat{j} + dn\hat{k} \tag{1-17}$$

(l, m, n): (rolling moment, pitching moment, yawing moment)

(x_0, y_0, z_0): reference point at zero angle of attack (e.g. center of gravity)

The vehicle in free flight has three rotational degrees of freedom. Three rotational disturbances must be originated at a reference point of the vehicle. However, the coordinate system is not set to the vehicle body, so a vector transformation is necessary as following:

$$x'' = x_c \cos\alpha - z_c \sin\alpha \tag{1-18}$$

$$y'' = y_c \tag{1-19}$$

$$z'' = x_c \sin\alpha + z\cos\alpha \tag{1-20}$$

$$d\vec{F}' = (dF_x \cos\alpha - dF_z \sin\alpha)\hat{i} + dF_y\,\hat{j} + (dF_z \cos\alpha + dF_x \sin\alpha)\hat{k} \tag{1-21}$$

(x'', y'', z''): center of each panel in the transformed coordinate system

(x_c, y_c, z_c): center of each panel computed by Eq.(1-9) – (1-11)

The differential force $d\vec{F}'$ calculated by Eq.(1-21) is used in Eq.(1-16). In order to have more accurate pressure force, Prandtl-Meyer expansion theory should be applied for the surface in the "shadow region," in which the Newtonian theory is inaccurate. This theory will be discussed in the next chapter since Prandtl-Meyer expansion theory is used to aid in determinning thermodynamic properties such as viscosity on the shaded surface.

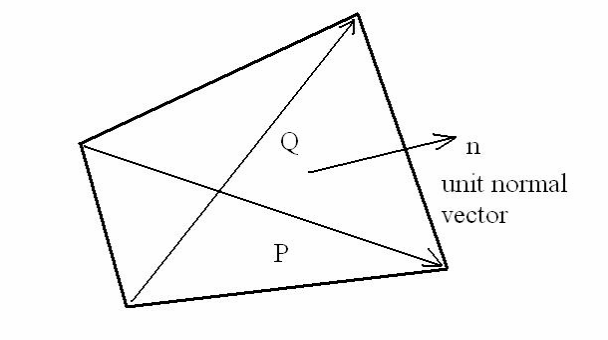

Figure 1-1. General body surface panel showing the unit normal vector along with the locations of the four corner points.

CHAPTER 2
VISCOUS AND HIGH TEMPERATURE CONSIDERATIONS

Introduction

In the preceding analysis the fluid dynamic effects of high Mach number is emphasized, without the added complications of viscous and high temperature effects. However, the matter of friction and thermal conduction should not be neglected since high speed flow is slowed by viscous effects within the boundary layer, and lost kinetic energy is transformed in part into internal energy. This extreme viscous effect can create very high temperatures – high enough to excite vibrational energy within molecules, and to cause dissociation and even ionization within the gas. The geometric layout of the body surface panels, the pressure on each panel, and the determination of the velocity component tangential to the body surface panels all are used in an approximate analysis of the skin friction and heat loads experienced by the vehicle during hypersonic flight. In order to calculate skin friction for both laminar and turbulent flows, the flat plate reference enthalpy method and Reynolds' analogy with heat transfer are used. For very large hypersonic Mach number, the assumption that the pressure is constant in the normal direction through a boundary layer is not always valid. However, for vehicles designed to fly at around $M = 6$ a constant pressure in the normal direction is the case.

Local Reynolds Number

One of the major parameters used in the analysis is the Reynolds number based on the local tangential velocity, temperature, and distance, s, from the stagnation point, that is

$$\mathrm{Re}_{s,e} = \frac{\rho_e u_e s}{\mu_e} \tag{2-1}$$

The variable s denotes the distance along the surface of the vehicle measured from the relevant stagnation point or stagnation line, while the subscript e indicates that these variables

are conditions at the outer edge of the boundary layer. The tangential component of the free stream velocity is denoted by u_e, and that is found by the preceding inviscid analysis as

$$\vec{u}_e = \vec{V}_\infty - \left(\vec{V}_\infty \cdot \hat{n}\right)\hat{n} \tag{2-2}$$

If the x component of the unit vector is positive, and the panel of a body is in the "shadow" of the incident flow, the Prandtl-Meyer expansion relations are applied locally since the Newtonian formula on merely has $c_p = 0$ everywhere in the shadow region. Now consider the centered Prandtl-Meyer expansion around a corner of deflection angle θ, as sketched in Figure 2-1. Upstream of the wave is the windward area of a body and the downstream is the shadow area of a body. The Mach numbers upstream and downstream of the wave are M_1 and M_2, respectively. From basic compressible flow, the relation between θ, M_1 and M_2 is given by:

$$\theta = \nu(M_2) - \nu(M_1) \tag{2-3}$$

where ν is the Prandtl-Meyer function:

$$\nu(M) = \sqrt{\frac{\gamma+1}{\gamma-1}}\tan^{-1}\sqrt{\frac{\gamma-1}{\gamma+1}\left(M^2-1\right)} - \tan^{-1}\sqrt{M^2-1} \tag{2-4}$$

From the equation above, the value of ν corresponding to M_1, and M_2 is found by the tangential components of the free stream velocity u_e at the outer edge of boundary layer. Figure 2-2 shows a 3-dimensional sketch of the flow around the corner. The deflection angle θ is found from the following relation:

$$\theta = \sin^{-1}\frac{u_{e1}\times u_{e2}}{\left|u_{e1}\right|\left|u_{e2}\right|} = \sin^{-1}(\hat{u}_{e1}\times\hat{u}_{e2}) \tag{2-5}$$

where u_{e1} and u_{e2} are the tangential component of the free stream velocity u_e at the outer edge of boundary layer of upstream and downstream respectively. Both u_{e1} and u_{e2} are found by Eq.(2-2) although u_{e2} obtained by Eq.(2-2) is not real outer edge velocity. In order to find the

deflection angle θ, only the directions of u_{e1} and u_{e2} must be found, so Eq.(2-2) is used only to find the unit vector of u_{e2}. Now θ and M_1 are known, so M_2 can be found by using Eq.(2-3) and Eq.(2-4), and the real u_{e2} and downstream thermodynamic properties are given by:[3]

$$u_{e2} = M_2\sqrt{\gamma R_{air} T_2} \tag{2-6}$$

$$T_2 = T_1 \frac{1+\left(\frac{\gamma-1}{2}\right)M_1^2}{1+\left(\frac{\gamma-1}{2}\right)M_2^2} \tag{2-7}$$

$$P_2 = P_1\left(\frac{T_2}{T_1}\right)^{\frac{\gamma}{\gamma-1}} \tag{2-8}$$

$$\rho_2 = \rho_1\left(\frac{T_2}{T_1}\right)^{\frac{1}{\gamma-1}} \tag{2-9}$$

In order to find the downstream thermodynamic properties, the upstream thermodynamic properties must be known.

For a specific value of γ, the Prandtl-Meyer function ν asymptotically approaches the maximum value $\nu_{\max}$ as Mach number increases. Thus, if $\nu(M_2) > \nu_{\max}$, an infinite Mach number is generated and the pressure falls to zero. Expansion at such condition would, according to the pressure theory, lead to a vacuum adjacent to the wall. Of course, in reality, the continuum and ideal gas assumption become invalid long before this situation is reached. However, panels whose ν may be greater than $\nu_{\max}$ won't have high heat exchange or skin friction. What we need for this study is identifying those surface panels whose total heat transfer is not negligible, so an accurate analysis for these special cases of large expansion angles is not necessary.

In order to evaluate the density and viscosity, the pressure and temperature are required. Since the pressure is assumed to be constant in the direction normal to the panel, the pressure on

the panels may be obtained from the pressure coefficients obtained in chapter 1 or Eq.(2-8) for the shaded panels, and that is

$$P_w = P_e = c_p q + P_\infty$$
$$\text{or } P_w = P_e = P_2 \text{ by Eq.(2-8) for shaded panels} \tag{2-10}$$

Temperature may be obtained by using the energy equation along a streamline

$$h_{total} = h_\infty + \frac{1}{2} V_\infty{}^2 = h_e + \frac{1}{2} u_e{}^2 \tag{2-11}$$

Therefore,

$$h_e = h_\infty + \frac{1}{2} V_\infty{}^2 - \frac{1}{2} u_e{}^2 = C_{p,\infty} T_\infty + \frac{1}{2} V_\infty{}^2 - \frac{1}{2} u_e{}^2 \tag{2-12}$$

where $C_{p,\infty}$ is constant pressure specific heat at temperature of T_∞. The assumption here is that kinetic energy carried in the normal component of velocity is transformed into internal energy by adiabatic compression. In order to find the temperature from enthalpy, it is necessary to use tables or models for the thermodynamic and transport properties of high temperature air. Here Hansen's "Approximations for the Thermodynamic and Transport Properties of High-Temperature Air"[4] is used to evaluate the behavior of thermodynamic and transport properties at high temperature.

Numerical codes are used to predict temperature, compressibility, density, viscosity, and Prandtl number. Prandtl number is not used to find Reynolds number, but will be used to find skin friction later, so the code computing Prandtl number is introduced here. The code for computing the temperature needs only inputs of enthalpy and pressure which are already known. The other codes computing compressibility, density, viscosity, and Prandtl number need inputs of pressure and temperature which is obtained from the code computing temperature. However, in order to find the temperature from pressure and enthalpy, the compressibility must be known since temperature is defined as a function of enthalpy, pressure, and compressibility in Hansen,

and inversely compressibility is obtained from pressure and temperature. Pressure and enthalpy are known, and therefore the temperature is guessed, and an iteration is carried out until the given enthalpy matches the enthalpy computed from guessed temperature. This approach establishes the following five functions.

$$T = T(P, h) \tag{2-13}$$
$$Z = Z(P, T) \tag{2-14}$$
$$\rho = \rho(P, T) \tag{2-15}$$
$$\mu = \mu(P, T) \tag{2-16}$$
$$\mathrm{Pr} = \mathrm{Pr}(P, T) \tag{2-17}$$

To find the distance s, the stagnation point (stagnation line for the wing) must be specified. Here it is assumed that for typical vehicles, like the X-24C that will be studied in this paper, there is one stagnation point for the fuselage while the two-dimensional cross section of the wings (airfoils) have stagnation lines. We can set the stagnation point at the center of the panel whose outward normal vector is the closest to the opposite vector of free stream. In this study, the vehicle has only low angles of attack, so the stagnation points are always located at the most windward panel. The most windward panel is so small that s of other panels do not change much even if the stagnation point and line are shifted to the nose of the fuselage and leading edge of the wing. Therefore, the stagnation point and line are set to the nose point of the fuselage and leading edge of the wings, respectively for convenience. The distance s is assumed to be the distance from the stagnation point to the center of the panel under consideration.

Local Skin Friction

The Van Driest II method for turbulent boundary layers is probably the most accurate generally applicable equation for skin friction, but it is too complicated for the entire surface of a vehicle. Another simpler method uses a reference value of temperature at which the density and physical properties of the fluid are evaluated and used in the available constant density, constant

property boundary layer solutions to provide an adequate approximation to the actual, variable density, variable property flow. That value of the temperature is called the reference temperature, T^*.[5] In this study, the flat plate reference enthalpy method is utilized to determine local surface heat transfer and Reynolds' analogy with heat transfer for laminar and turbulent flow is used to determine friction on panels. The flat plate panels considered here have approximately constant pressure over the skin surface, and thus permits this approach. The Nusselt number is given by

$$Nu = A\left(\frac{\rho^*}{\rho_e}\right)^a \left(\frac{\mu^*}{\mu_e}\right)^b \mathrm{Re}_{s,e}{}^c \left(\sqrt{3}\right)^j \tag{2-18}$$

The coefficients in Eq.(2-18) are listed in Table 2-1. The reference enthalpy and the adiabatic wall enthalpy are given by

$$h^* = 0.28h_e + 0.5h_w + 0.22h_{aw} \tag{2-19}$$

$$h_{aw} = h_e + \mathrm{Pr}_e{}^m \frac{u_e{}^2}{2} \tag{2-20}$$

where h_e and h_w are the enthalpies at the edge of the boundary layer and at the wall, respectively, and Pr_e is the Prandtl number evaluated at the edge of boundary layer. The quantity m is $1/2$ for laminar flow and $1/3$ for turbulent flow. It is noted that the skin friction coefficient and the Nusselt number are related. This observation can be formalized an generalized for non-slip condition, $u_w = 0$, and the relation is

$$Nu = \mathrm{Re}\cdot \mathrm{Pr}^{\frac{1}{3}} \frac{c_f}{2} \tag{2-21}$$

or

$$c_f(s) = \frac{2A}{\mathrm{Pr}^{1/3}}\left(\frac{\rho^*}{\rho_e}\right)^a \left(\frac{\mu^*}{\mu_e}\right)^b \mathrm{Re}_{s,e}{}^{c-1} \left(\sqrt{3}\right)^j \tag{2-22}$$

where $c_f(s)$ is the local skin friction coefficient.

Local Heat Transfer

From Eq.(2-21) and Eq.(2-22), the local heat transfer can be found since the definition of the Nusselt number is

$$Nu = \frac{q_{c,w} s}{k_e (T_w - T_{aw})} \tag{2-23}$$

where $q_{c,w}$ is the convective heat transfer at the wall, and is obtained from the relation

$$q_{c,w} = \frac{c_f(s)\,\mathrm{Pr}_e^{1/3}\,\mathrm{Re}_{s,e}\,k_e(T_w - T_{aw})}{2s} \tag{2-24}$$

The subscripts w and e denote conditions at the wall and the edge of the boundary layer. Eq.(2-24) provides reasonable values for the heat transfer, except for the extremely high value at the region near the stagnation point, since the distance from the stagnation point, s is very small around the stagnation point. Thus, the blunt body heat transfer method is applied to the region near the stagnation point. Fay and Riddell [6, a] first carried out a rigorous study of stagnation point convective heat transfer at hypersonic speeds and provided the following result

$$q_{c,s} = 0.76\,\mathrm{Pr}^{-0.6}(\rho_e\mu_e)^{0.4}(\rho_w\mu_w)^{0.1}\sqrt{\left(\frac{du_e}{ds}\right)_s}(h_{s,e} - h_w)\left[1 - (Le^{0.52} - 1)\left(\frac{h_D}{h_{s,e}}\right)\right] \tag{2-25}$$

In Eq.(2-25), the term in square brackets represents the effects of equilibrium chemical reactions occurring in the stagnation region and

$$Le = \frac{\rho D_{12} c_p}{k} \tag{2-26}$$

$$h_D = \sum_{i=1}^{n} c_i \Delta h_{f,i} \tag{2-27}$$

The gas considered here is air, which can be considered to be a binary mixture. This mixture is made up of two species: oxygen and nitrogen atoms (O and N) and molecules (O2 and

[a] Taken from "Space Access Vehicle Design Handbook" (Sforza, P. M.)

N2). The quantity D_{12} is the binary diffusion coefficient. The quantities c_i and $\Delta h_{f,i}$ are the molar concentrations of the individual species (O, O2, N, and N2) and the chemical heat of formation of each species, respectively. The Lewis number for an air-like mixture given by Eq.(2-26) is close to unity, Le~1.4, so that the quantity $(Le^{0.52} - 1) \sim 0.19$, and the contribution of the chemical reaction term can be often be safely neglected.

The velocity gradient at the stagnation point in Eq.(2-25) may be found by

$$\frac{du_e}{ds} = \frac{1}{R_b}\sqrt{\frac{2(P_e - P_\infty)}{\rho_e}} \tag{2-28}$$

In the Newtonian approximation this becomes

$$\frac{du_e}{ds} \approx \sqrt{2\varepsilon}\frac{V_\infty}{R_b} \tag{2-29}$$

Then Eq.(2-25) is simplified to

$$q_{c,s} = \frac{0.9038}{\varepsilon^{1/4}}\left(\frac{C_w}{\mathrm{Pr}}\right)^{0.1}\sqrt{\frac{\rho_\infty V_\infty \mu_s}{R_b \,\mathrm{Pr}}}(h_{s,e} - h_w) \tag{2-30}$$

In Eq.(2-30) the variable $C_w = \rho_w \mu_w / \rho_s \,\mu_s$ is the Chapman-Rubesin factor and the Prandtl number is calculated at the stagnation conditions at the edge of the boundary layer. The stagnation enthalpy behind the shock as well as the density ratio across the shock can be determined from the shock relations for equilibrium air chemistry. A schematic diagram of stagnation region is shown in Figure 2-3.

A stationary normal shock wave is considered here. The shock is so strong that the temperature behind the shock is high enough to ensure that vibrational excitation and chemical reactions occur behind the shock front. It is assumed that local thermodynamic and chemical equilibrium conditions hold behind the shock, and all conditions ahead of the shock wave are known. The governing equations for the flow across a normal shock are

Continuity $\rho_1 u_1 = \rho_2 u_2$ (2-31)

Momentum $P_2 = P_1 + \rho_1 u_1^2 \left(1 - \frac{\rho_1}{\rho_2}\right)$ (2-32)

Energy $h_2 = h_1 + \frac{u_1^2}{2}\left[1 - \left(\frac{\rho_1}{\rho_2}\right)^2\right]$ (2-33)

In addition, the equilibrium thermodynamic properties for the high-temperature gas are known from the numerical techniques introduced by Eq. (2-13) and Eq.(2-15). The codes used here are

$T_2 = T(P_2, h_2)$ (2-34)

$\rho_2 = \rho(P_2, T_2)$ (2-35)

Since all the upstream conditions, ρ_1, u_1, P_1, h_1, etc., are known, Eq.(2-32) and Eq.(2-33) express P_2 and h_2, respectively, in terms of only one unknown ρ_1/ρ_2. This establishes the basis for an iterative numerical solutions introduced by Anderson[2], as follows

1. Assume a value for ρ_1/ρ_2. (A value of 0.1 is usually good first guess.)
2. Compute P_2 from Eq.(2-32) and h_2 from Eq.(2-33)
3. Using the values of P_2 and h_2 obtained, compute T_2 from Eq.(2-34) and ρ_2 from Eq.(2-35).
4. Form a new value of ρ_1/ρ_2 from the value of ρ_2 obtained in step 3.
5. Use this new value of ρ_1/ρ_2 in Eq.(2-32) and Eq.(2-33) to obtain new values of P_2 and h_2, respectively. Then repeat step 2 through 5 until convergence is obtained, i.e., until there is only a negligible change in ρ_1/ρ_2 from one iteration to the next.
6. At this stage, the correct values of P_2, h_2, T_2, and ρ_2 are obtained. Using Eq.(2-31), obtain the correct value of u_2.

By means of step 1 through 6 above, all properties behind the shock wave are found for given properties in front of the wave.

Now the stagnation point heat transfer is obtained by Eq.(2-30) and thermodynamic properties behind the shock. For hypersonic flow conditions over blunt bodies, it is considered that the flow along an inviscid streamline emerging from the stagnation region as if that streamline were everywhere a local surface of the vehicle. The following is good approximation of the local heat transfer on the surface of blunt bodies derived by Lees.[7, b]

$$q_c = q_{c,s} \cdot \frac{1}{2^{\frac{n+1}{2}} \left(\frac{du_e}{dx}\right)_s^{\frac{1}{2}}} \frac{\left(\frac{\rho_e \mu_e}{\rho_s \mu_s}\right) u_e {r_0}^n}{\left[\int_0^s \left(\frac{\rho_e \mu_e}{\rho_s \mu_s}\right) u_e {r_0}^{2n} ds\right]^{\frac{1}{2}}} \tag{2-36}$$

where r_0 is radius of cross section of bodies of revolution and n is 1 for bodies which are like bodies of revolution and 0 for two dimensional bodies. The subscript s indicates computation at the stagnation point and should not be confused with the variable s, the distance from the stagnation point.

Total Forces Acting on the Vehicle

Now that the pressure and skin friction distribution around the body is known, the forces acting on the vehicle may be determined. We already know the inviscid aerodynamic forces, L_{invisc} and D_{invisc} by summing the contributions of all the panel differential inviscid forces dL_{invisc} and dD_{invisc} over all the panels. However, these differential forces do not include expansion effects for the shaded panels. In order to recalculate the accurate inviscid aerodynamic forces, the pressure computed in this chapter (Eq.(2-8) or Eq.(2-10)) should be used as follows,

$$d\vec{F}_{press} = dF_{press,x}\hat{i} + dF_{press,y}\hat{j} + dF_{press,z}\hat{k} = -P_e \cdot dA \cdot \hat{n} \tag{2-37}$$

$$dL_{invisc} = dF_{press,z} \tag{2-38}$$

$$dD_{invisc} = dF_{press,x} \tag{2-39}$$

[b] Taken from "Space Access Vehicle Design Handbook" (Sforza, P. M.)

Inviscid aerodynamic forces mean the forces caused by only the pressure acting on the vehicle. So far we know the local skin friction coefficients, so we can find the aerodynamic forces including viscous effects. The differential skin friction forces for turbulent flow on the plates can be found from,

$$dF_{fric,x}\hat{i} = c_{f,tub} \cdot q \cdot dA \cdot \hat{u}_{e,x} \tag{2-40}$$

$$dF_{fric,y}\hat{j} = c_{f,tub} \cdot q \cdot dA \cdot \hat{u}_{e,y} \tag{2-41}$$

$$dF_{fric,z}\hat{k} = c_{f,tub} \cdot q \cdot dA \cdot \hat{u}_{e,z} \tag{2-42}$$

where $\hat{u}_{e,x}$, $\hat{u}_{e,y}$, and $\hat{u}_{e,z}$ are the x, y, and z component of the unit vectors of the outer edge flow velocity of the panel. The dynamic pressure of the free stream is q, the differential area of the panel is *dA*, and $c_{f,tub}$ is the local skin friction coefficient of the turbulent boundary layer. When the laminar boundary layer is considered, $c_{f,tub}$ is simply replaced by $c_{f,lam}$. Aerodynamic forces including skin friction are found from the following:

$$dL = dL_{invisc} + dF_{fric,z} \tag{2-43}$$

$$dD = dD_{invisc} + dF_{fric,x} \tag{2-44}$$

$$L = \sum dL \tag{2-45}$$

$$D = \sum dD \tag{2-46}$$

Discussion of Results without any Cooling Methods

We have just established a method to compute the total forces acting on the body surface as well as the heat transfer. Aerodynamic forces can be found by summing the contributions of all the panel differential forces (*L, D*) and moments can be found by summing the moment contributions of all the panel differential data (dL, dD, and dM_0). The aerodynamic forces and moments are expressed in term of dimensionless force and moment coefficients:[8]

$$C_D = \frac{D}{\frac{1}{2}\rho_\infty V_\infty{}^2 S_{ref}} \text{ :drag coefficient} \tag{2-47}$$

$$C_L = \frac{L}{\frac{1}{2}\rho_\infty {V_\infty}^2 S_{ref}} \text{ :lift coefficient} \tag{2-48}$$

$$C_m = \frac{m}{\frac{1}{2}\rho_\infty {V_\infty}^2 S_{ref} c} \text{ :pitching moment coefficient} \tag{2-49}$$

where S_{ref} is the reference area, which is the planform area in this study and c is the fuselage length. Unlike hypersonic gliders like the Space Shuttle Orbiter, hypersonic cruising vehicles have relatively slender bodies and fly at low angles of attack. Therefore, here we analyze only low angles of attack, in which skin friction effects on flight performance are more pronounced than in high angles of attack.

The X-24C configuration is chosen for study here and is illustrated in Figure 2-4 with the surface features being approximated by 284 panels. Table 2-2 shows comparisons of computed lift and drag coefficients and L/D with experimental data. The flight condition of the X-24C studied here is a Mach number of 5.95, an angle of attack of 6º, a characteristic unit Reynolds number of $1.64\times10^7 / m$, and a turbulent boundary layer, which are the same conditions as the available experimental data.[9-11, c] The base pressure, P_{base} is set to the same value as the ambient pressure, which is probably an optimistic choice for the cruising vehicle. In this study, we confine our attention to the X-24C vehicle. The configuration details are shown in table 2-3.

The results for the aerodynamic forces are slightly underpredicted, but they agree reasonably well with experimental data. Figure 2-5 and 2-6 show L/D and aerodynamic forces coefficients for various α, respectively. Viscous effects on L/D and C_D are obvious, but not for C_L because at low angles of attack friction doesn't affect normal forces. Figure 2-7 shows the

[c] Reference 9 and 10 are taken from Reference 11

pitching moment coefficient. The reference point is located at the center of gravity of the vehicle $(x = 9.706, y = 0, z = 0)$.

Table 2-4 shows the effect of base pressure on the aerodynamic forces. C_L does not change with P_{base} since the base pressure does not have an effect on the lift much when the angle of attack is relatively low. On the other hand, the predicted C_D depends on P_{base} and therefore so does L/D. At $P_{base} = 0$, C_D is the closest to the experimental data, while at $P_{base} = 0.2P_\infty$, the predicted L/D gives the best agreement with the experiment. In order to have good agreement with the experimental data, we should set P_{base} to 0 or $0.2P_\infty$. However, this value is too small for cruising vehicle, and C_L has the best agreement with the experiment when $P_{base} = P_\infty$, so it is not easy to find the appropriate P_{base} here. For the present study, the latter assumption $P_{base} = P_\infty$ is used.

We have introduced two methods to calculate the local heat transfer, the flat plate reference enthalpy method and the blunt body solution method. At stagnation points, the flat plate reference enthalpy method provides extremely high values of the heat transfer, so the blunt body solution should be used for panels around the stagnation point or line. Although the flat plate reference enthalpy method and the blunt body solution method give different values of heat transfer, at least both solutions have similar behavior. In order to find the location where we may start to use the flat plate reference enthalpy method, a heat transfer comparison is carried out for the Space Shuttle Orbiter since there are more experimental data for that configuration than for the X-24C. Figure 2-8 shows comparison of heating distribution along windward symmetry plane of the Orbiter between the present method (the flat plate reference enthalpy method and the blunt body solution method) and actual flight data.[12] In this figure, s/L is the distance from the stagnation point normalized by total length of the windward symmetry plane line. The flat plate

reference enthalpy method agrees with actual flight data except for the region around the stagnation point. Therefore, around the stagnation point or line, the blunt body heat transfer method is to be used, and the heat transfer calculated by both methods should be analyzed for each stream line. There is a panel at which two methods provide the same or very close heat transfer, and there we must switch the methods to use. In Figure 2-8, for example, the blunt body heat transfer method should be used for the panels at which $s/L = 0$ and 0.0173. For the panels downstream from $s/L = 0.0864$, the flat plate reference enthalpy method should be used. Mass injection through the surface is used for cooling, which causes additional mixing in the boundary layer. Since the boundary layer is approximated to be turbulent, we use the flat plate reference enthalpy method for turbulent flow to find the heat transfer.

Since most of the experimental data[9, 10, 12] for the X-24C was measured at several streamwise stations, the format of comparison is constructed accordingly. Typical results are given by abscissas in the form of normalized arc length. This length is calculated from the leeward symmetry (top) plane toward the windward (bottom) counterpart and scaled by the total arc length of each individual cross section. Surface pressure and the Stanton number distribution comparisons with experimental measurements at the body furthest downstream station ($x/r_n = 104.75$) of the X-24C are shown in Figures 2-9 and 2-10, respectively. Our vehicle configuration does not use very many panels, so we cannot show the results at exactly the same stations. Thus surface pressure and Stanton number are computed at the furthest downstream station ($x/r_n = 90.2$) in our vehicle model. In Figure 2-9, we define the Stanton number as:[11]

$$St = \frac{q_c}{\rho_\infty V_\infty (h_s - h_w)} \tag{2-50}$$

The agreement in surface pressure and heat transfer distribution is reasonable. The windward and leeward pressure distributions along the symmetry plane are given in Figure 2-11. The present results slightly underpredict the experimental data at most points, but the theory seems applicable for analyzing the flow properties around the X-24C. In order to find aerodynamic forces and heat transfer over the X-24C vehicle, therefore it is reasonable to use the methods introduced in Chapters 1 and 2.

Table 2-1. Coefficients in Eq.(2-18)

Type of flow	A	a	b	c	j
laminar	$0.3320\,\mathrm{Pr}^{1/3}$	0.5000	0.5000	0.5000	-
turbulent	$0.0296\,\mathrm{Pr}^{1/3}$	0.8000	0.2000	0.8000	-
flat plate	-	-	-	-	0.0000
axisymmetric	-	-	-	-	1.0000

Table 2-2. Comparison of aerodynamic forces and L/D.

	C_L	C_D	L/D
Experimental data	0.0368	0.0317	1.16
Present results	0.0332	0.0253	1.31
Percent error	-9.78	-20.2	13.1

Table 2-3. Configuration of X24C studied.

Parameter	Value
Reference area (m^2)	57.2
Fuselage (m)	14.7
Wingspan (m)	7.38
Surface Temperature (K)	317.0

Table 2-4. The effect of the base pressure on aerodynamic forces and L/D.

P_{base}/P_∞	C_L	error in C_L	C_D	error in C_D	L/D	error(%) in L/D
0	0.0327	-11.0	0.0294	-7.21	1.11	-4.00
0.2	0.0328	-10.8	0.0286	-9.82	1.15	-1.00
0.4	0.0329	-10.6	0.0278	-12.4	1.18	2.21
0.6	0.0330	-10.3	0.0270	-15.0	1.22	5.62
0.8	0.0331	-10.1	0.0261	-17.6	1.27	9.25
1.0	0.0332	-9.82	0.0253	-20.2	1.31	13.1

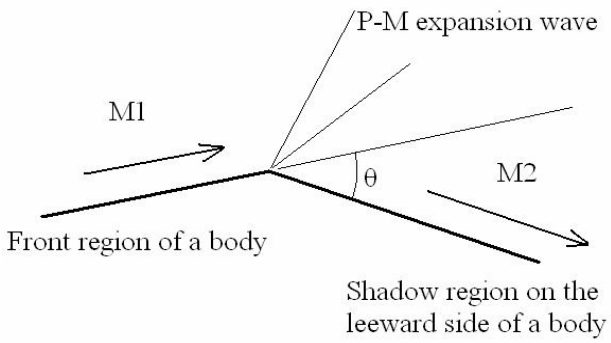

Figure 2-1. Expansion wave.

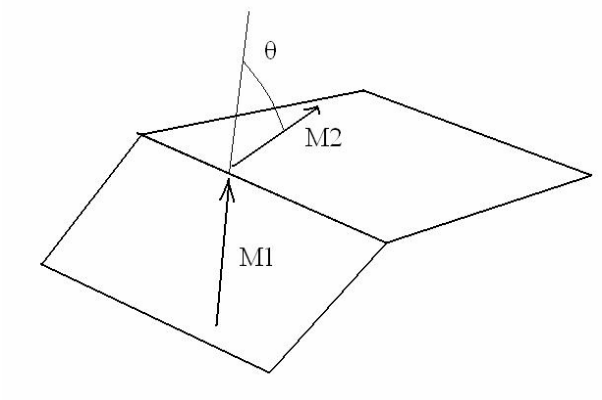

Figure 2-2. Deflection angle θ at the corner.

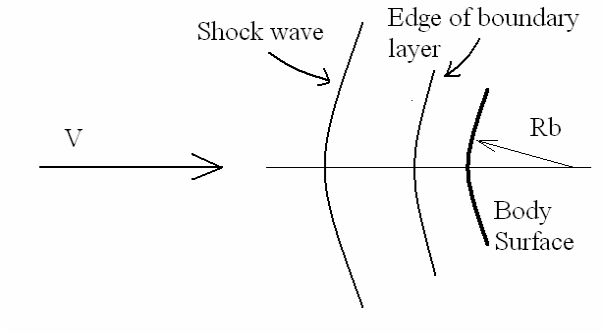

Figure 2-3. Schematic diagram of stagnation region.

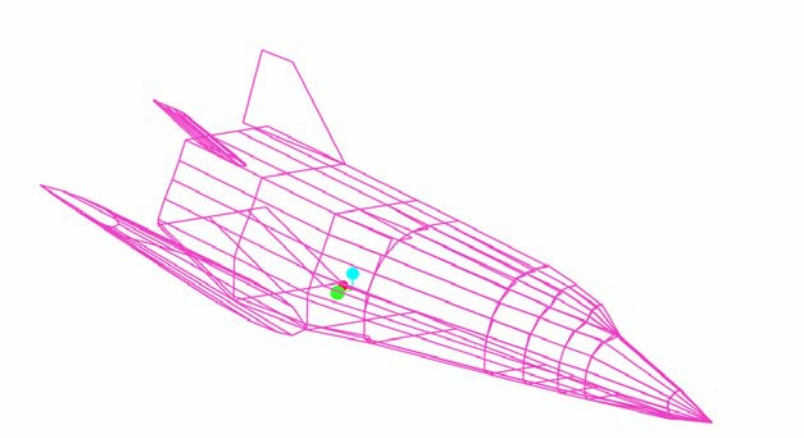

Figure 2-4. X-24C configuration. Note that this is a representation of the right half of the aircraft.

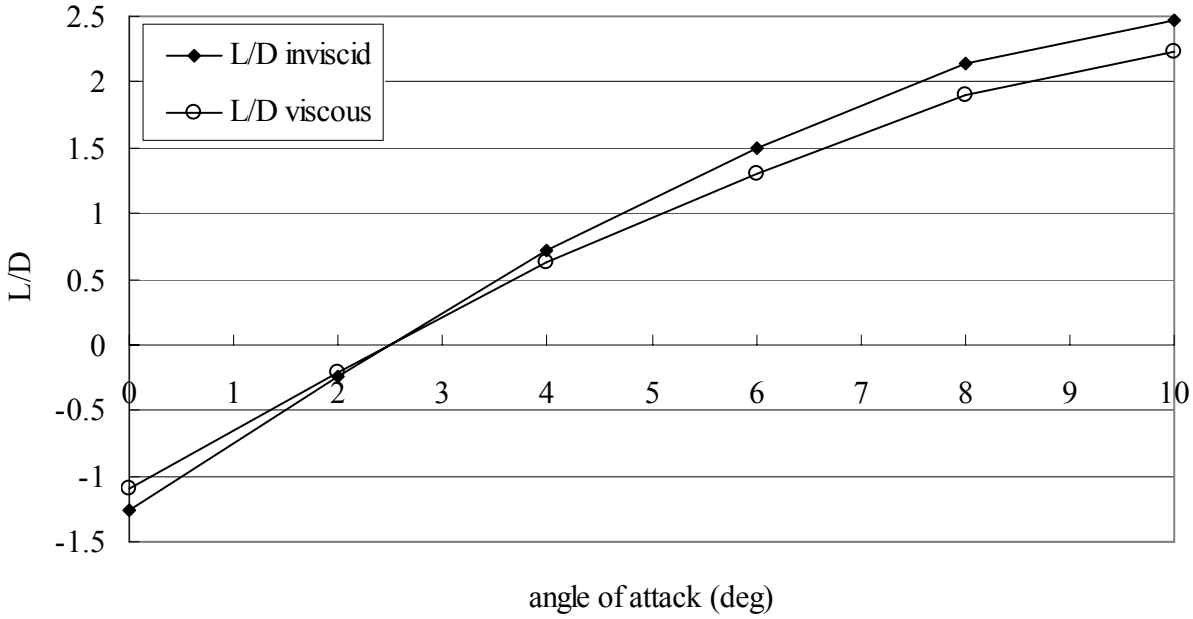

Figure 2-5. X-24C's L/D as a function of angle of attack.

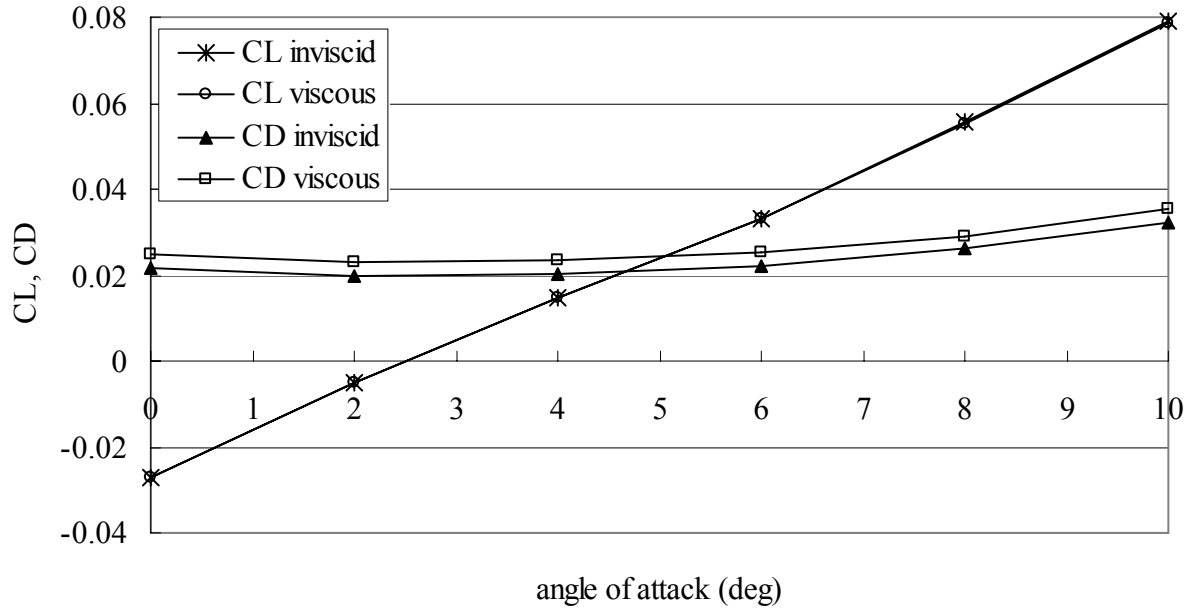

Figure 2-6. X-24C's aerodynamic force coefficients as a function of angle of attack.

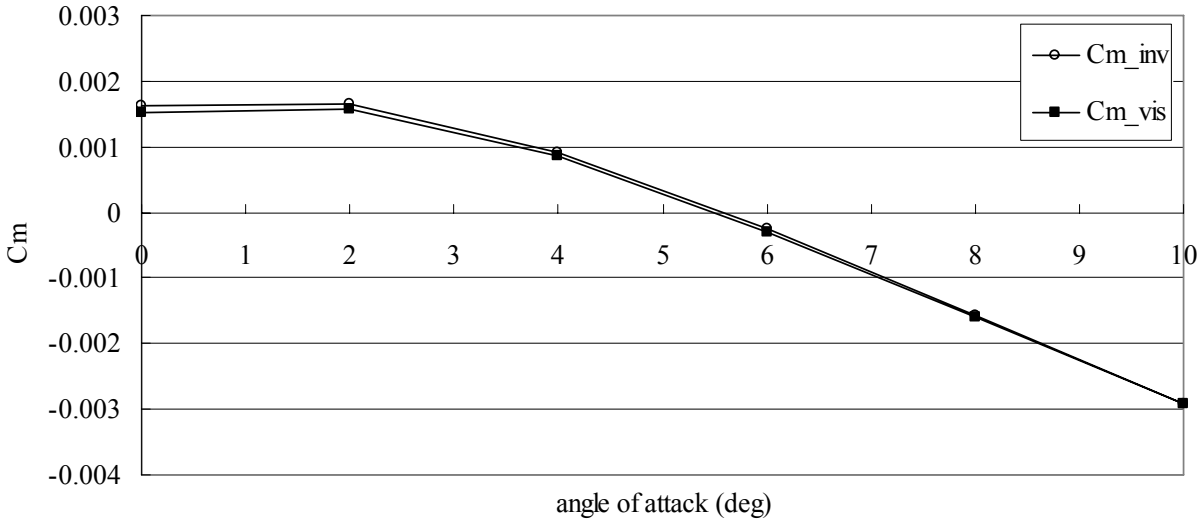

Figure 2-7. X-24C's pitching moment coefficients as a function of angle of attack.

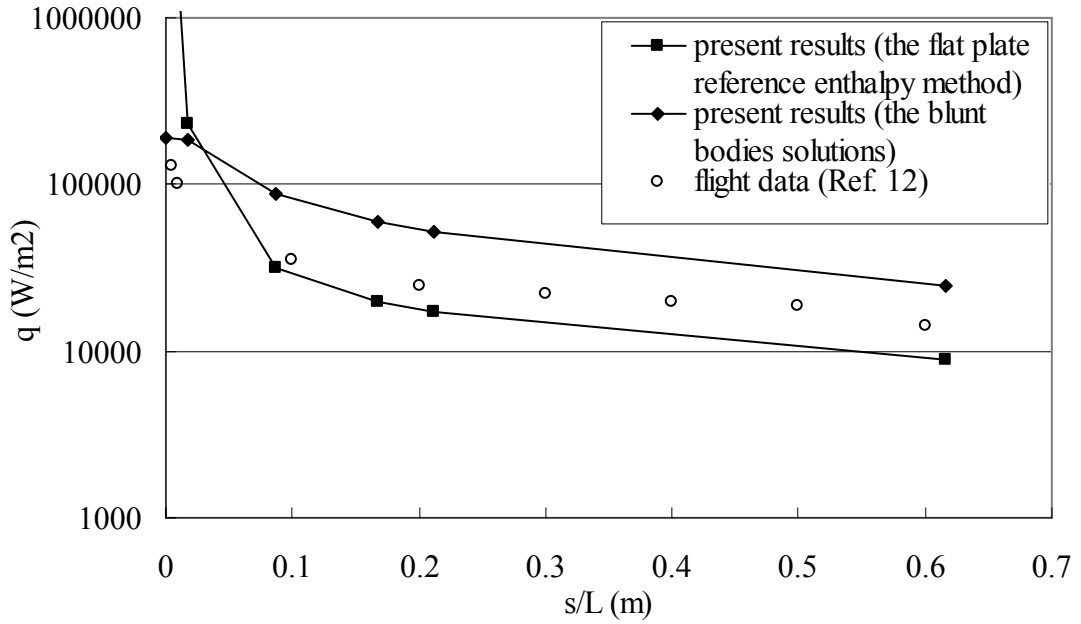

Figure 2-8. Heating distribution along windward symmetry plane of Space Shuttle Orbiter at 34.8° of angle of attack, $M = 9.15$, and 47.7*km* altitude.

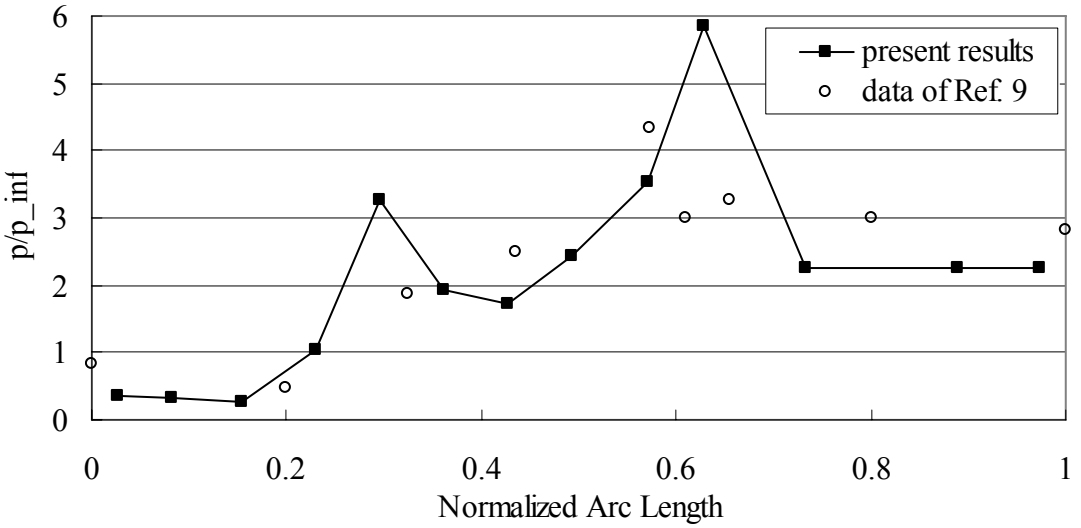

Figure 2-9. Comparison of surface pressure distributions around the X-24C fuselage at the farthest downstream station.

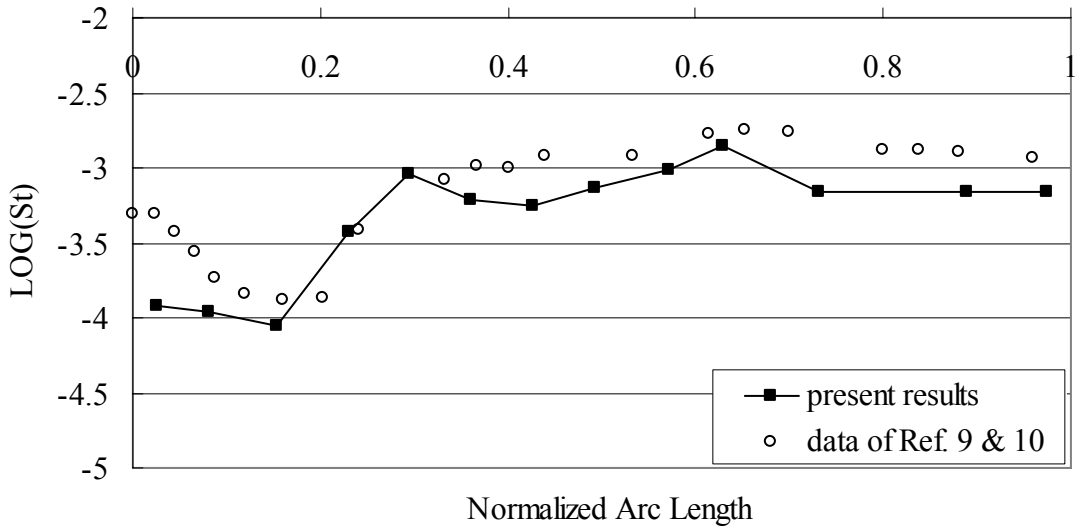

Figure 2-10. Comparison of heat transfer around the X-24C fuselage at the farthest downstream station.

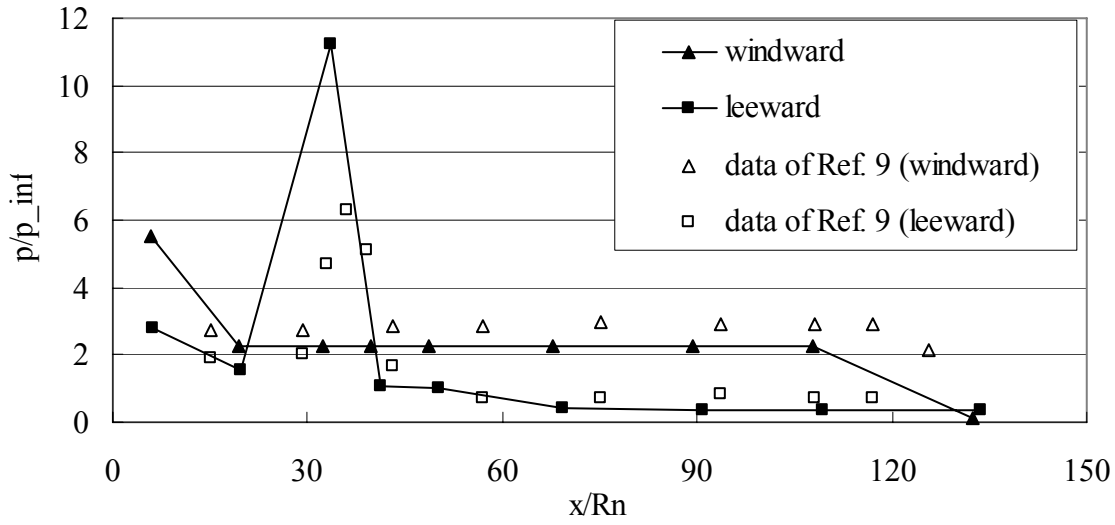

Figure 2-11. Comparison of streamwise surface pressure distribution along the windward and leeward symmetry lines of the X-24C.

CHAPTER 3
MASS INJECTION EFFECTS

Introduction

Reducing skin friction and heat transfer is one of the most challenging areas of research in hypersonic aerodynamics. Attention has been focused on surface suction to delay transition to the turbulent flow region which produces relatively large skin friction. However, even a very small protuberance can cause laminar flow to transition into turbulent flow. Also, laminar flow is more susceptible to flow separation than turbulent flow. Therefore this technique of using suction still remains in the research stage. One of the drag reduction methods that has been ignored is surface mass injection because researchers believed that the penalty associated with mass injection is very large because of the susceptibility of flow separation. Despite this shortcoming, many experiments were conducted in 1970's on mass injection on a flat plate with no pressure gradient. It was well established that mass injection significantly reduced skin friction with respect to the skin friction of the same porous plate with no injection.[13]

Thus far in the present study we have discussed only boundary layers for which the normal component of velocity at the surface, v_w is zero. Nonzero values of v_w can occur if the wall is porous and mass is injected into the boundary layer. Mass injection from the surface is one of several cooling methods for protecting the surface from an extremely high-temperature stream. It is assumed that the boundary layer over the body is turbulent everywhere in order to maintain a conservative evaluation of the heating load. In addition, the transition location is not known even for the zero injection case, so to be consistent we have focused on purely turbulent flow. Dominant parameters are angle of attack and ratio of the mass flow rate of the cooling gas to air of free stream.

Effects of Mass Injection on the Boundary Layer

In order to examine the effect of injecting gas into the boundary layer, we use reviews of the effects of transpiration on the turbulent boundary layer presented by Kays and Crawford.[14] They introduced a simple algebraic Couette flow solution which has the virtue of fitting the additional experimental data. The effect of porous surface injection on Stanton number and skin friction coefficient at the same Reynolds number based on distance s are described as follows:

$$\frac{St}{St_0} = \frac{\ln(1+B_h)}{B_h} \tag{3-1}$$

$$\frac{c_f}{c_{f,0}} = \frac{\ln(1+B_f)}{B_f} \tag{3-2}$$

where

$$B_h = \frac{\rho_w v_w / \rho_e u_e}{St} \tag{3-3}$$

$$B_f = \frac{\rho_w v_w / \rho_e u_e}{c_f/2} \tag{3-4}$$

In the above equations, ρ_w is the density of injected gas, and v_w is the normal velocity component at the surface. The subscript 0 refers to the case with zero injection, that is, $B_h = 0$ and $B_f = 0$. Because of the implicit nature of such equations, it is frequently more convenient to use other blowing parameters than B_h and B_f, such as:

$$b_h = \frac{\rho_w v_w / \rho_e u_e}{St_0} \tag{3-5}$$

$$b_f = \frac{\rho_w v_w / \rho_e u_e}{c_{f,0}/2} \tag{3-6}$$

We find that $b_h = \ln(1+B_h)$ and $b_f = \ln(1+B_f)$, and then

$$\frac{St}{St_0} = \frac{b_h}{e^{b_h}-1} \tag{3-7}$$

$$\frac{c_f}{c_{f,0}} = \frac{b_f}{e^{b_f}-1} \tag{3-8}$$

If comparison is made at the same Reynolds number based on streamwise distance s for the case of constant free-stream velocity, the above equations fit the experimental data remarkably well. The Nusselt number is related to the Stanton number as follows:

$$St = \frac{\mathrm{Pr}_e}{\mathrm{Re}_{s,e}} Nu \tag{3-9}$$

where Pr_e and $\mathrm{Re}_{s,e}$ are constant at the same location (or in our case panel) for any values of B_h and b_h, so the Couette flow analysis above can be developed in the same manner for the Nusselt number as for the Stanton number.

$$\frac{Nu}{Nu_0} = \frac{\ln(1+B_h)}{B_h} = \frac{b_h}{e^{b_h}-1} \tag{3-10}$$

From Eq.(2-23) and Eq.(2-24), the convective heat transfer at the wall can be found to be

$$q_{c,w} = \frac{b_h}{e^{b_h}-1} q_{c,w,0} \tag{3-11}$$

Eq.(3-11) is to be evaluated at the same k_e, T_w, and T_{aw} for injected and non-injected cases.

A limiting case occurs for large values of blowing when the friction coefficient tends to zero and boundary layer is literally blown off the surface, an occurrence similar to the separation of a boundary layer in an adverse pressure gradient. Two commonly used rules of thumb for "blow-off" are as follows: if $\rho_w v_w / \rho_e u_e = 0.01$ and/or $b_f = 4.0$, then is safe to assume that "blow-off" has occurred.

General Behavior of Reducing Skin Friction and Heating

Unlike hypersonic gliders such as Space Shuttle Orbiter, hypersonic cruising vehicles have relatively slender bodies and fly at low angles of attack. Therefore, here we consider only low angles of attack in which skin friction effects on flight performance are more pronounced than at

high angles of attack. In order to illustrate the general effects of mass injection, the case of a flat plate in hypersonic flight is briefly discussed, in which, without loss in generality, we neglect forces acting on the upper surface. Figure 3-1 shows the effects of mass injection on the lift to drag ratio by plotting $(L/D)_{cooled}/(L/D)_0$ with different angle of attack. $(L/D)_{cooled}$ is L/D with some mass injection, and $(L/D)_0$ is the L/D ratio without any mass injection. Figure 3-2 shows the effects of mass injection on the heat transfer on the surface by show q_{cooled}/q_0 as a function of mass injection for different angles of attack here. q_{cooled} is heat transfer with some mass injection, and q_0 is the heat transfer without any mass injection.

In general, the lower α is, the more mass injection reduces skin friction and protects the surface from heating, and even a small amount of injected mass is effective to improve flight performance. Figure 3-1 and 3-2 show that merely increasing the amount of injection for reducing skin friction and heat transfer is not very effective because the improvement of L/D and the reduction in q are more pronounced for small amounts of injection. Also, the beneficial effects of injection are larger for lower α than for higher α.

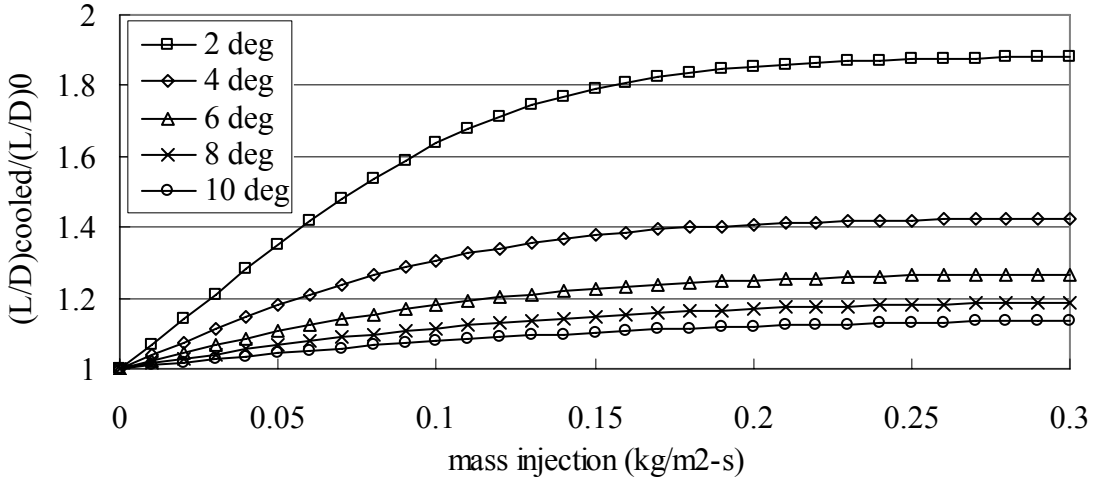

Figure 3-1. Effects of mass injection on L/D with different angle of attack for a flat plate ($6m \times 6m$ square plate) at a Mach number of 6.

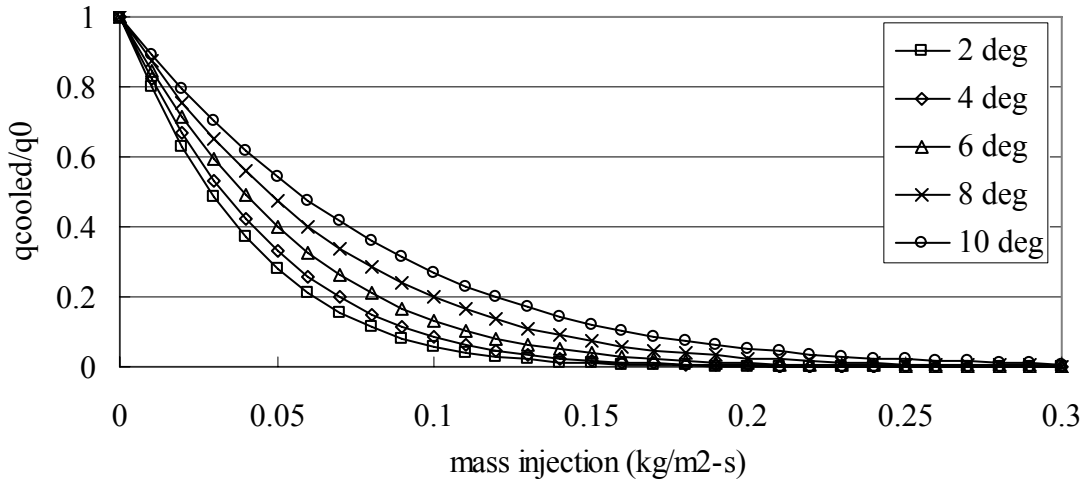

Figure 3-2. Effects of mass injection on heat transfer with angle of attack for a flat plate ($6m \times 6m$ square plate) at a Mach number of 6.

CHAPTER 4
RESULTS

Introduction

In the preceding chapters, we discussed inviscid flow, viscous flow, and reducing skin friction and heat transfer on a hypersonic cruising vehicle by using mass injection and established a theory for finding flow properties whether or not we use mass injection. In this chapter, we show the advantages and penalties associated with the mass injection cooling method. The coolant injected may be a part of the total fuel load or different material (but a gas) from the fuel. In both cases, the critical parameter in analyzing the reduction of skin friction and heat transfer as discussed in the previous chapter is the injected mass flux $\rho_w v_w$ and therefore the density of the coolant (ρ_w) and the velocity of the coolant (v_w) are not individually important. In this chapter, "fuel" is defined as the fuel used only for producing thrust, and "coolant" is defined as the material (although it may be the same material as the fuel) to be injected through portions the vehicle's surface to reduce viscous and heating effects. In order to evaluate the advantages and penalties of mass injection, the total mass of fuel and coolant consumed is kept constant no matter how much coolant is injected. For instance, if the total mass of the fuel and the coolant consumed is set to 100*kg* and the mass of the coolant is 10*kg*, then the mass of the fuel is 90*kg*. If the mass of the coolant is increased to 20*kg*, then the mass of the fuel is 80*kg*. Our perspective is that injecting mass (coolant) from the vehicle's surface reduces viscous effects, reducing the heat transfer on the surface (advantage) even though it may decrease the flight range (or flight time) since the amount of the fuel used for the thrust is decreased (penalty).

The Range Equation

One of important performance metrics for cruising aircraft is the flight range, the total distance that an aircraft can travel on a given amount of fuel. Simple flight behavior is assumed

so that the thrust vector is assumed to be aligned with the flight path and the flight path angle, the angle between the flight trajectory and the local horizontal, is very small. For equilibrium flight we have the following relations:

$$F_T = D \tag{4-1}$$
$$L = W \tag{4-2}$$

where F_T is thrust and W is weight of the vehicle. The rate at which the fuel and the coolant are consumed in quasi-steady flight is the summation of the weight flow rate of fuel used to produce the needed thrust and the weight flow rate of the coolant injected from the surface.[16] This may be written as follows:

$$\frac{dW}{dt} = \frac{dW_{consumed}}{dt} = \frac{d\left(W_{fuel} + W_{coolant}\right)}{dt} = -\frac{C_j}{3600} F_T - \left(\rho_w v_w g\right) A_{cooled} \tag{4-3}$$

where W_{fuel} is weight of the fuel, $W_{coolant}$ is weight of the coolant, and A_{cooled} is total area to be cooled by injection. The quantity C_j is the specific fuel consumption and may also be written as

$$I_{sp} = \frac{3600}{C_j} \tag{4-4}$$

where I_{sp} is the specific impulse, measured in thrust produced per unit weight of fuel consumed per unit time. Now Eq.(4-3) becomes

$$\frac{dW}{dt} = -\frac{F_T}{I_{sp}} - \dot{w}_{inj} A_{cooled} = -\frac{W}{I_{sp}\left(\frac{L}{D}\right)} - \dot{w}_{inj} A_{cooled} \tag{4-5}$$

where $\dot{w}_{inj} = \rho_w v_w g$. Therefore, Eq.(4-5) is written as

$$dt = \frac{dW}{-\frac{W}{I_{sp}\left(L/D\right)} - \dot{w}_{inj} A_{cooled}} \tag{4-6}$$

The horizontal speed $V_\infty = dx/dt$, so Eq.(4-6) may be integrated from the initial time to the final time, assuming that I_{sp}, L/D, and $\dot{w}_{inj}A_{cooled}$ are constant, as follows:

$$\int_{t_{initial}}^{t_{final}} dt = \int_0^R \frac{dx}{V_\infty} = \frac{R}{V_\infty} = \int_{W_{initial}}^{W_{final}} \frac{dW}{-\frac{W}{I_{sp}(L/D)} - \dot{w}_{inj}A_{cooled}} = -\frac{\ln\left(\dot{w}_{inj}A_{cooled} + \frac{W}{I_{sp}(L/D)}\right)}{\left(\frac{1}{I_{sp}(L/D)}\right)}\Bigg|_{W_{initial}}^{W_{final}} \tag{4-7}$$

$$V_\infty t_{final} = R = -I_{sp}\left(\frac{L}{D}\right)V_\infty \ln\left(\frac{\dot{w}_{inj}A_{cooled} + \frac{W_{final}}{I_{sp}(L/D)}}{\dot{w}_{inj}A_{cooled} + \frac{W_{initial}}{I_{sp}(L/D)}}\right) \tag{4-8}$$

where t_{final} is the flight time and R is the flight range. Here W_{final} is the weight of the vehicle after all the fuel and coolant are consumed during the cruising flight and $W_{initial}$ is the weight of the vehicle when carrying full fuel and full coolant. Eq.(4-8) is used to find the flight range starting with a constant initial amount of fuel and coolant. Note that no matter how much coolant is used for surface injection, Eq.(4-8) gives the flight range or flight time with the given amount of fuel and coolant being completely consumed. As already discussed, L/D depends on both $\dot{w}_{inj}$ and A_{cooled}. The weight of the vehicle before it uses any fuel and coolant, $W_{initial}$ is set to:

$$W_{initial} = L_0 = C_{L,0}\frac{1}{2}\gamma P_\infty {M_\infty}^2 S_{ref} \tag{4-9}$$

where $C_{L,0}$ is the lift coefficient without mass injection. The total weight of the fuel and the coolant is:

$$W_{initial} - W_{final} = W_{fuel} + W_{coolant} \tag{4-10}$$

Comparison of Range, L /D, and Heat Transfer Variance

Since we have already established methods to find the aerodynamic forces including skin friction, heat transfer, effects of mass injection, and flight range for hypersonic flow. We are now

ready to show the advantages and penalties arising from injecting mass through the surface of a hypersonic cruising vehicle. In this study, the characteristic Reynolds number is used, and it is defined as: $\mathrm{Re}_{char.} = \rho_\infty V_\infty / \mu_\infty$. Our theories were applied to an X-24C vehicle assumed to be flying at Mach number of 6.00 with angle of attack of 6º at 30000m ($\mathrm{Re}_{char.} = 2.21\times10^6 / m$ and $q = 29500Pa$) altitude and 35000m ($\mathrm{Re}_{char.} = 9.95\times10^5 / m$ and $q = 14000Pa$) altitude. The surface temperature is assumed to be constant with or without injection. For the case at which we do not use mass injection, some unspecified internal cooling system may accomplish constant surface temperature assumption by absorbing the thermal power. For injected surface, surface temperature may remain constant temperature of coolant that is being injected through the panel surface. The fuel used here is assumed to be hydrocarbon, with $I_{sp} = 1000s$. The material injected through the surface of vehicle is not specified, but it is at least a gas. W_{final} is the weight of the vehicle after 100s of cruising flight with zero mass injection. Therefore, Eq.(4-8) provides the flight time for which the aircraft can cruise when the total amount of the fuel and coolant is the same as $W_{initial} - W_{final}$ whether or not we use mass injection.

Injecting on the windward panels on the forward section of the fuselage, where $q_{c,w}$ is relatively high, reduces heat transfer with a small rate of mass injection. However, each of these individual panels has a relatively high angle of attack with respect to the free stream direction, therefore the effect of mass injection may be relatively smaller than other panels in reducing drag. Cooling aft panels, where $q_{c,w}$ is relatively low, reduces heat transfer less effectively, and the required mass injection rate is relatively large. However, each of these panel's angle of attack is lower than the ones of the forward panels, and each has a large area, so it is not simple to choose the panels to be cooled so that the mass is used most effectively.

Therefore, the panels to be cooled were determined by two aspects of the heat transfer experienced by each panel: the thermal power transfer, $\dot{Q}_{c,w}(W)$, and the heat flux, $q_{c,w}(W/m^2)$. Around the nose of body and the leading edge of the wings, each panel has relatively high $q_{c,w}$, but low $\dot{Q}_{c,w}$ because of the small area of the individual panels. On the other hand, the aft panels have low $q_{c,w}$, but high $\dot{Q}_{c,w}$ since each panel has relatively large area.

Figure 4-1 shows the distribution of panel area as a function of the panel's heat transfer, $q_{c,w}(W/m^2)$ at $M=6$ and 30000*m* altitude for no injection. The total area of panels whose uncooled heat transfer $q_{c,w}$ is between 0 and $5000\,W/m^2$ is slightly larger than $25\,m^2$, and there is $35\,m^2$ area of panels that have uncooled $q_{c,w}$ between 5000 and $10000\,W/m^2$. Figure 16 is slightly different as it shows the distribution of the number of panel as a function of the panel's uncooled heat transfer, $q_{c,w}(W/m^2)$ at $M=6$ and 30000*m* altitude. For example, there are 14 panels whose uncooled heat transfer $q_{c,w}$ is between 0 and $5000\,W/m^2$, while there are over 50 panels with uncooled $q_{c,w} > 100000 W/m^2$.

Figures 4-3 to 4-5 show the effect of mass injection on the flight time, and therefore the range, L/D, and reduction of heat power (normalized heat power), respectively at $M=6$ and 30000*m* altitude. Normalized heat power is defined as $\dot{Q}/\dot{Q}_0$. $\dot{Q}_0$ is heat power absorbed by all panels when we do not use any mass injection, and $\dot{Q}$ is heat power absorbed by all panels with mass injection. When $\dot{m}_{inj}$ is zero, $\dot{Q}=\dot{Q}_0$. The panels to be cooled are determined by the heat transfer, $q_{c,w}$ experienced by the individual panels without injection. As we increase the rate of mass injection, the flight range decreases and this effect is more pronounced for cases in which there is a larger area to be cooled. As discussed in the previous chapter, mass injection reduces

skin friction and heat transfer. Therefore, we get an advantage in reducing the heat transfer but with the penalty of decreasing flight range.

For instance, consider the case of a mass injection rate of $0.005\,kg/m^2s$ at $M=6$ and 30000m altitude. Figure 4-3 shows that the flight time is 85.16s when all panels are cooled by the selected mass injection rate of $0.005\,kg/m^2s$. The flight time is 89.97s when only the panels that have $q_{c,w}$ of more than $10000\,W/m^2$ are cooled by the selected mass injection rate of $0.005\,kg/m^2s$. Figure 4-4 shows that L/D is 1.245 when all panels are cooled by the mass injection rate of $0.005\,kg/m^2s$ and L/D is 1.236 when only the panels that have $q_{c,w}$ of more than $30000\,W/m^2$ are cooled by the mass injection rate of $0.005\,kg/m^2s$. Figure 4-5 shows that normalized heat power is 0.8936 when all panels are cooled by the mass injection rate of $0.005\,kg/m^2s$. However, the normalized heat power is 0.9517 when the panels that have $q_{c,w}$ of more than $50000\,W/m^2$ are cooled by the mass injection rate of $0.005\,kg/m^2s$.

Before Figures 4-6 to 4-11 are discussed, two normalized parameters: normalized injected mass and normalized reduction of heat power are introduced. Normalized injected mass is defined as:

$$\tilde{m} = \frac{\dot{m}_{inj} A_{cooled}}{\dot{m}_{inj} A_{total}} = \frac{A_{cooled}}{A_{total}} \tag{4-11}$$

where $\dot{m}_{inj}$ is the rate of mass injected and A_{total} is the total surface area of the vehicle. Normalized reduction of heat power is defined as:

$$\tilde{\dot{Q}} = \frac{\dot{Q}_0 - \dot{Q}}{\dot{Q}_0} \tag{4-12}$$

In order to show more details of the advantages and penalties associated with mass injection, Figures 4-6 to 4-11 show the relation between injected mass, reduction of heating, and

reduction of flight range at the relatively low mass injection rate of $0.001\ kg/m^2s$. Figures 4-12 to 4-17 show results at mass injection rate an order of magnitude greater, i.e. $0.01\ kg/m^2s$.

For instance, the case of mass injection rate of $0.001\ kg/m^2s$ at $M = 6$ and $30000m$ altitude is introduced. Figure 4-6 shows that normalized reduction of heat power is 0.02091 when normalized injected mass is 0.8482 and when the panels that have $q_{c,w}$ of more than $5000\,W/m^2$ are cooled by the mass injection rate of $0.001\ kg/m^2s$. Figure 4-7 shows that the flight time is $97.14s$ when normalized injected mass is 0.8482 and when the panels that have $q_{c,w}$ of more than $5000\,W/m^2$ are cooled by the mass injection rate of $0.001\ kg/m^2s$. Figure 4-8 shows that the flight time is $97.14s$ when normalized reduction of heat power is 0.02091 and when the panels that have $q_{c,w}$ of more than $5000\,W/m^2$ are cooled by the mass injection rate of $0.001\ kg/m^2s$. Figure 4-9 shows that reduction of heat power is $71.46kW$ when injected mass rate is $0.1427\ kg/s$ and when the panels that have $q_{c,w}$ of more than $5000\,W/m^2$ are cooled by the mass injection rate of $0.001\ kg/m^2s$. Figure 4-10 shows that reduction of heat energy is $16700kJ$ when injected mass is $13.86\,kg$ and when the panels that have $q_{c,w}$ of more than $5000\,W/m^2$ are cooled by the mass injection rate of $0.001\ kg/m^2s$. Figure 4-11 shows that flight time is $97.14s$ when reduction of heat energy is $16700kJ$ and when the panels that have $q_{c,w}$ of more than $5000\,W/m^2$ are cooled by the mass injection rate of $0.001\ kg/m^2s$.

Figures 4-6 and 4-12 show the effect of normalized injected mass on normalized reduction of heat power. Figures 4-7 and 4-13 show the effect of normalized injected mass on the flight time, while Figures 4-8 and 4-14 show the effect of normalized reduction of heat power on the flight time. Figures 4-9 and 4-15 show the effect of injected mass rate on the reduction of heat

power. The effect of injected mass on the reduction of heat energy is shown in Figures 4-10 and 4-16. Finally, Figure 4-11 and 4-17 show the advantage (reduction of heat transfer) we have for the penalty (reduction of flight time). The data labels attached to the lines are the allowable heat transfer, $q_{c,w}$ of the panels. For instance, a data label, "$50000\,W/m^2$" means that all panels whose heat transfer is above $50000\,W/m^2$ (the allowable heat transfer) are cooled by mass injection.

Figures 4-6 and 4-12 show a concave downward curve, which indicates that choosing the highly heated panels for cooling reduces total heat transfer more effectively than cooling all panels. Figures 4-7 and 4-13 show an almost linear relation between normalized mass and flight time. Therefore, as more mass injected, the more the flight time (therefore flight range) decreases at an almost constant rate. Figures 4-8 and 4-14 show a concave downward curve relation between reduction of heat power and flight time. Thus, injecting on too many panels is increasingly apt to decrease flight time. Figures 4-9 and 4-15 show the actual reduction in heat power with various mass injection rates (kg/s). Figures 4-10 and 4-16 show the reduction in heat energy with various mass injection (kg). These figures give a slightly concave downward relation between the reduction of heat energy and the mass injection. Figures 4-11 and 4-17 give a slightly concave downward curve. Again, this indicates that cooling only panels whose heat transfer, $q_{c,w}$ is relatively high gives the most effective advantage (reduction of heat transfer) by mass injection.

Figures 4-18 to 4-20 show the X-24C vehicle whose panels are to be cooled if the heat transfer $q_{c,w}$ is more than $50000\,W/m^2$ at $M=6$ and $30000m$ altitude. The shadings (red panels) are the panels cooled. In general, the nose, the vertical fin, and the leading edges of the wing and the off-center fin have high $q_{c,w}$. The bottom panels also have high heat absorption. The panels

around the nose of the fuselage and the leading edge of the wings have the highest heat absorption.

Figures 4-21 to 4-40 are analogous to Figures 4-1 to 4-20. They are at the exactly same conditions except for that the panels to be cooled are determined by heat power, $\dot{Q}_{c,w}$ of individual panels in Figures 4-21 to 4-40. Also, Figure 4-41 to 4-60 are analogous to Figures 4-1 to 4-20. Figures 4-41 to 4-60 are results for $M = 6$ and 35000m altitude, and the cooled panels are chosen by heat transfer, $q_{c,w}$ of the individual panel. Finally, Figures 4-61 to 4-80 are also analogous to Figures 4-1 to 4-20. Figures 4-41 to 4-80 are results for $M = 6$ and 35000m altitude, and the panels to be cooled are determined by the heat power, $\dot{Q}_{c,w}$ of individual panels.

Figures 4-23 to 37 show results very similar to those of Figures 4-3 to 4-17. There is little of difference to note within difference between Figures 4-3 to 4-17 and 4-23 to 4-37. Therefore, whether we choose the panels to be cooled by taking those with the higher heat transfer, $q_{c,w}$ or the higher heat power, $\dot{Q}_{c,w}$ of individual panels, cooling only some of the highest heated panels gives the most effective advantage. Figures 4-38 to 4-40 show that we have panels whose heat power is higher in the aft portion of the vehicle although the front panels have the higher heat transfer, $q_{c,w}$. This is because aft panels have larger area, and the heat power $\dot{Q}_{c,w}$ absorbed by each panel is defined as $q_{c,w} \cdot dA$.

Figures 4-43 to 4-57 also show the same behavior as Figures 4-3 to 4-17. Comparing Figures 4-6, 4-7, 4-12 and 4-13 and Figures 4-46, 4-47, 4-52, and 4-53, we find that the effect of mass injection (normalized injected mass) on reducing flight time and heat transfer (normalized reduction of heat power) is larger for the case at 35000m altitude. Comparing Figure 4-48 and 4-54 with Figure 4-8 and 4-14, the same value of normalized reduction of heat power gives slightly

greater reduction in flight time at 35000*m* altitude. This is because of lower dynamic pressure at higher altitude. Comparing Figures 4-9, 4-10, 4-15, and 4-16 and Figures 4-49, 4-50, 4-55, and 4-56, it is found that the effect of mass injection on the reduction of heat power and heat energy is slightly larger at 35000*m* altitude or almost same for both cases. This argument may look as if it conflicts with the comparison of Figures 4-6 and 4-46, but this is because the total heat power and heat energy absorbed by the vehicle are larger for the case at 30000*m* than the case at 35000*m*.

Figures 4-63 to 4-77 show similar behavior and values to those of Figures 4-43 to 4-57 and we have seen that Figures 4-23 to 4-37 do not change much from Figures 4-3 to 4-17. In general, choosing the panels to be cooled by taking the higher heat transfer, $q_{c,w}$ gives more effective advantage (reduction of heat transfer) than by taking the higher heat power, $\dot{Q}_{c,w}$ of individual panel. However, the X-24C has the highest $q_{c,w}$ around the nose where the panels have relatively high angle of attack. On the other hand, the aft panels have the highest $\dot{Q}_{c,w}$ and relatively low angle of attack. The effect of mass injection on reducing heat transfer and increasing *L/D* is generally more pronounced at the panels whose angle of attack is low as we have seen in figure 3-1 and 3-2. Therefore, for the X-24C there is not much difference results between choosing the panels by using $q_{c,w}$ and $\dot{Q}_{c,w}$ of each panels. As a brief result, table 4-1 shows comparison of critical flight parameters for various mass onjection rate.

One of the parameters that indicates how the vehicle is slender (or "fat") is the volume-surface parameter, τ defined as:

$$\tau = \frac{V^{2/3}}{S} \tag{4-13}$$

where V is the volume of the vehicle, and S is the spanwise area of the vehicle. The smaller τ is, the more slender the vehicle. For the X-24C aircraft, $\tau = 0.18$ which is relatively high value for a cruising vehicle. A more slender body would have a higher *L/D* than the X-24C because the contribution of aerodynamic forces caused by pressure is smaller. The fraction of skin friction force in total drag force is therefore larger, which makes the increment of *L/D* by mass injection larger. Therefore, it is expected that reduction of flight range by mass injection decreases or that flight time increases for a more slender body.

Table 4-1. Comparison of flight parameters at $M = 6$ and 30000*m* altitude. The panels that have $q_{c,w}$ of more than $50000\,W/m^2$ are cooled by the mass injection rate: 0, 0.001, $0.01\,(kg/m^2 s)$.

Parameter	$0\,(kg/m^2 s)$	$0.001\,(kg/m^2 s)$	$0.010\,(kg/m^2 s)$
t_{final} (s)	100.0	96.65	74.04
L/D	1.225	1.227	1.240
$\dot{Q}/\dot{Q}_0$	1.000	0.990	0.906
A_{cooled}/A_{total}	0.000	0.3634	0.3634
reduction of heat power (kW)	0.000	33.76	320.8
reduction of heat energy (MJ)	0.000	7.585	66.851
W_{fuel} (kg)	442.7	436.8	397.4
$W_{coolant}$ (kg)	0.000	5.909	45.27

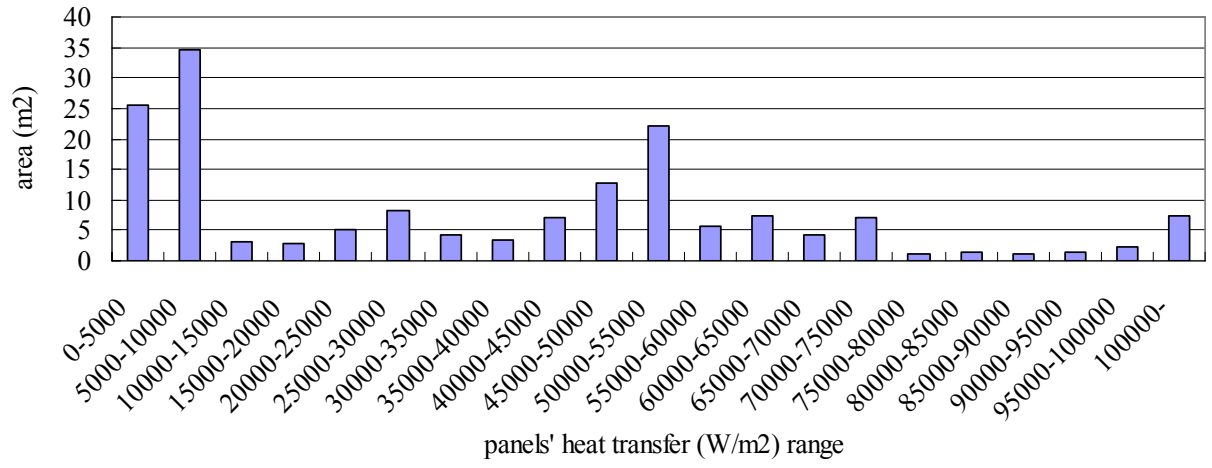

Figure 4-1. Distribution of panel area with a given heat transfer, $q_{c,w}\,(W/m^2)$ at $M = 6$ and 30000*m* altitude.

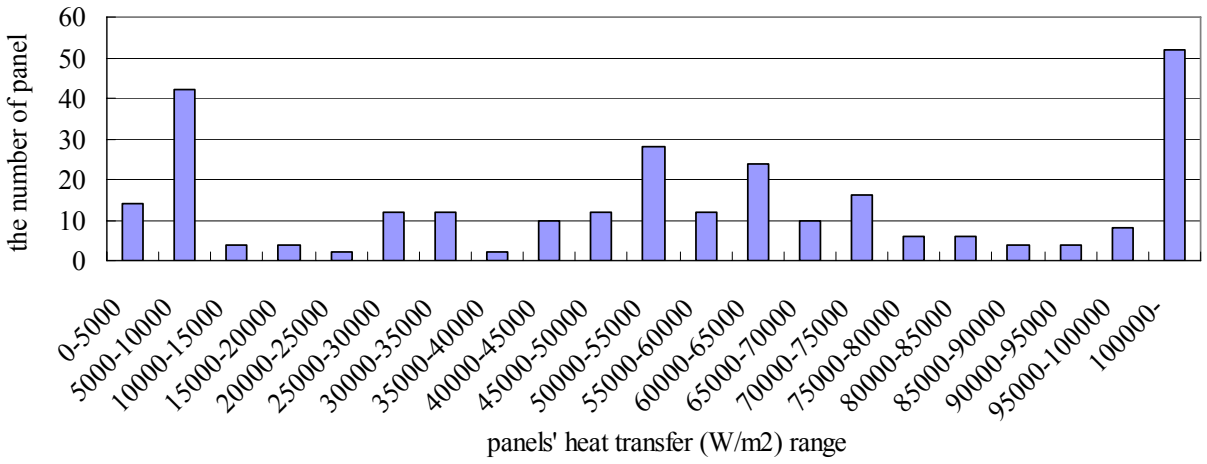

Figure 4-2. Distribution of the number of panels with a given heat transfer, $q_{c,w}\,(W/m^2)$ at $M = 6$ and 30000*m* altitude.

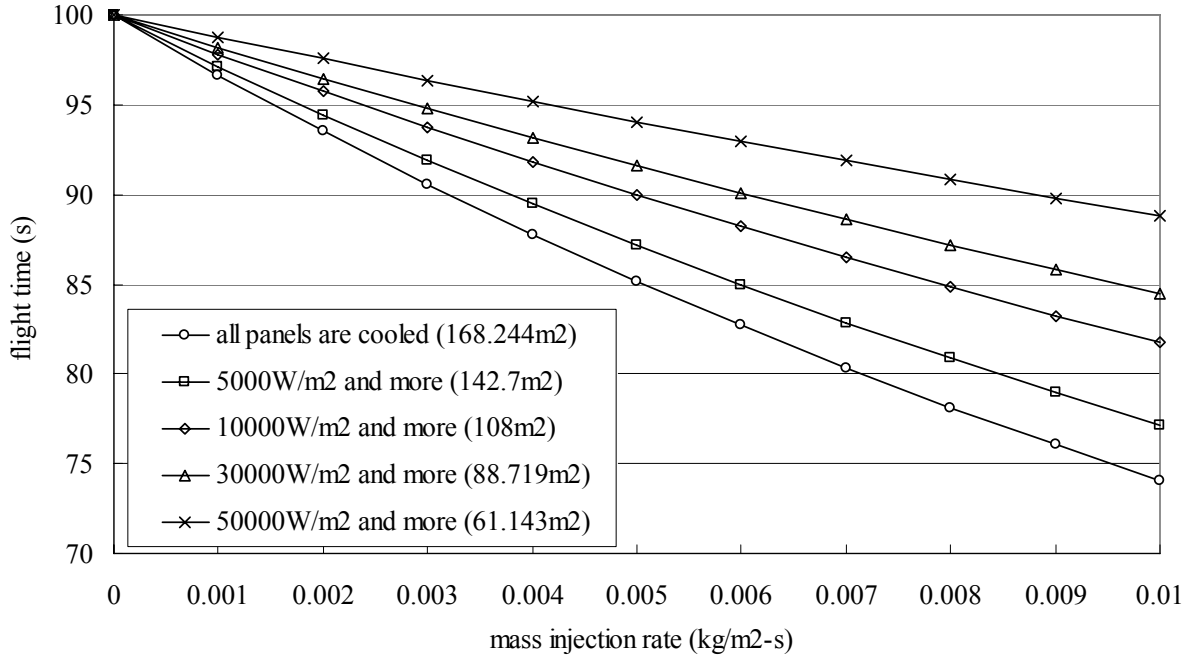

Figure 4-3. Mass injection effect on the flight time at $M = 6$ and 30000*m* altitude. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

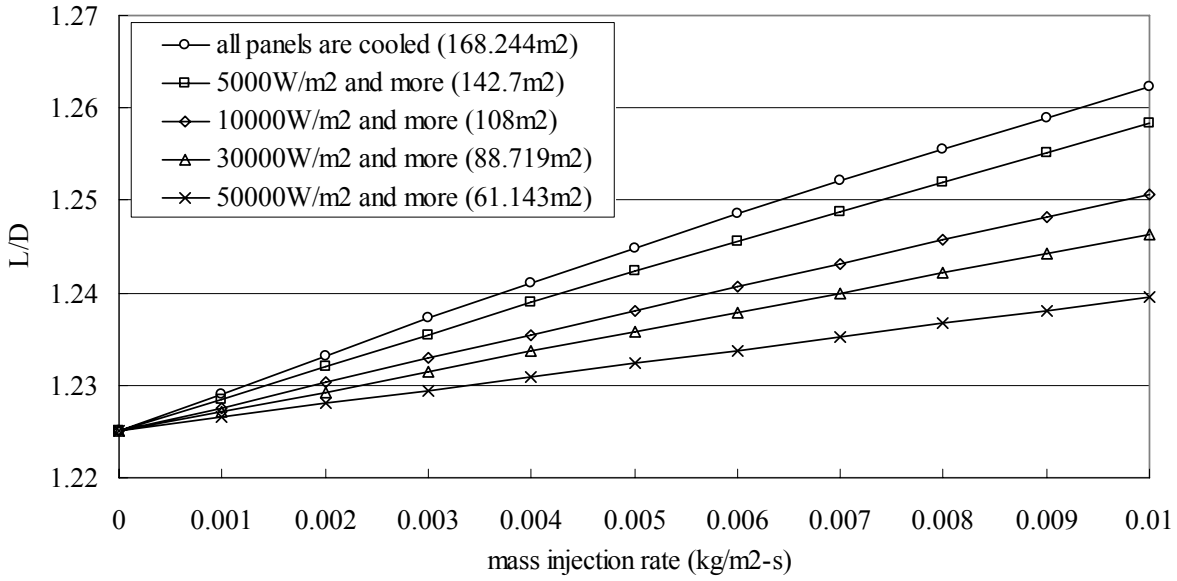

Figure 4-4. Mass injection effect on L/D at $M = 6$ and 30000m altitude. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

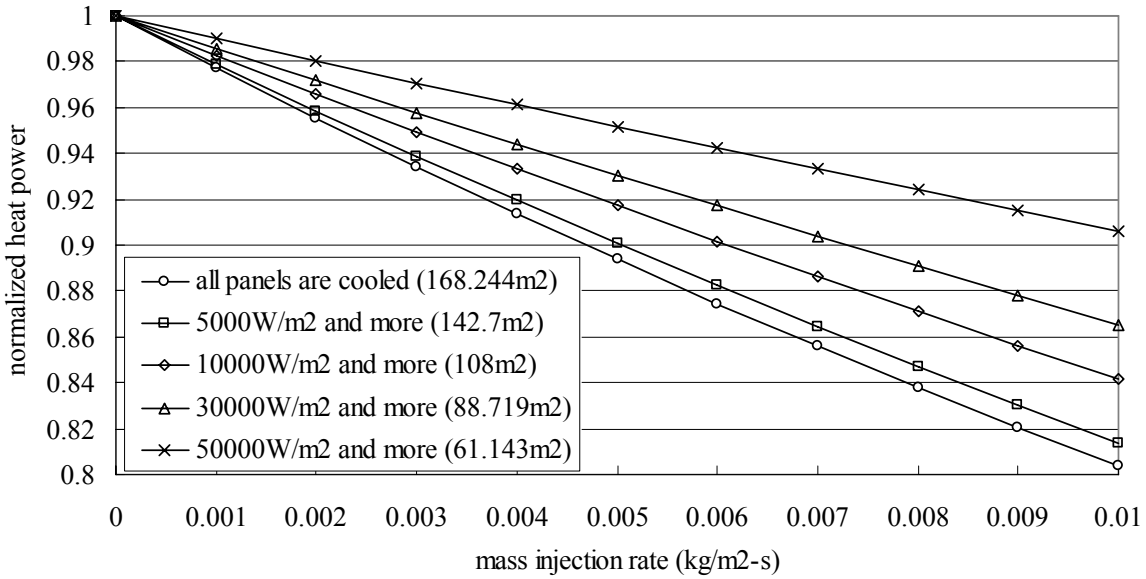

Figure 4-5. Mass injection effect on reduction of heat power at $M = 6$ and 30000m altitude. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

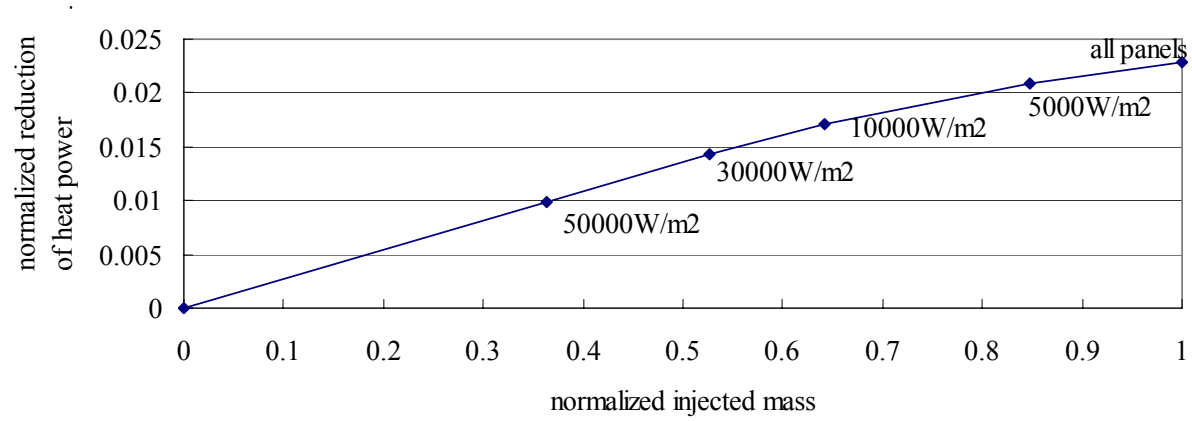

Figure 4-6. Normalized reduction of heat power v.s. normalized injected mass at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

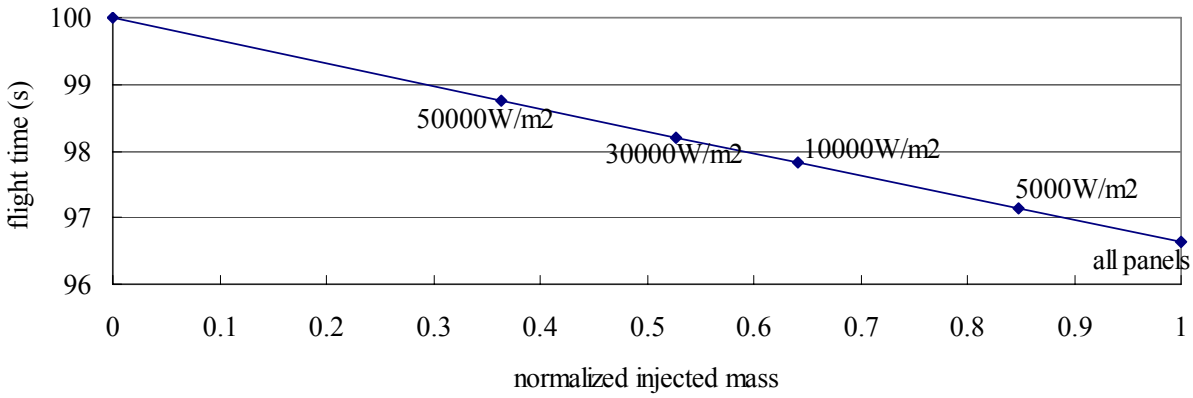

Figure 4-7. Flight time v.s. normalized injected mass at at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

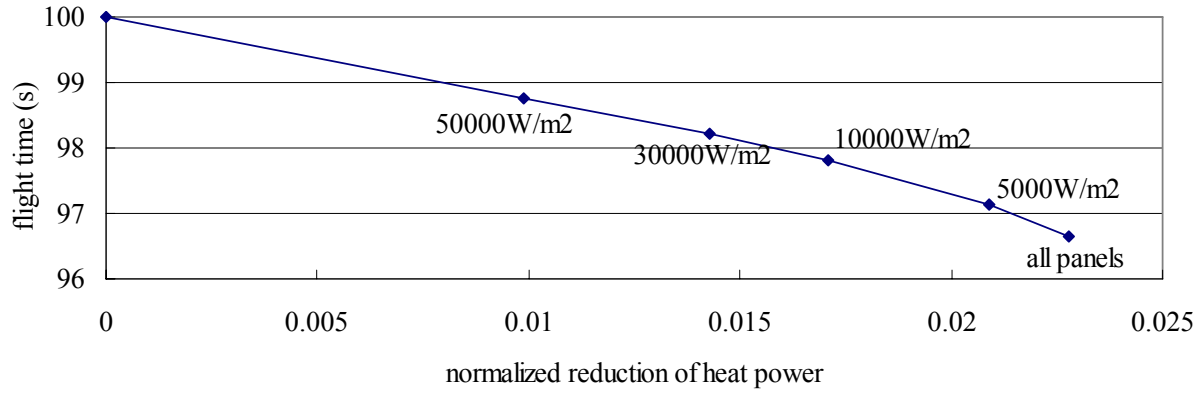

Figure 4-8. Flight time v.s. normalized reduction of heat power at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

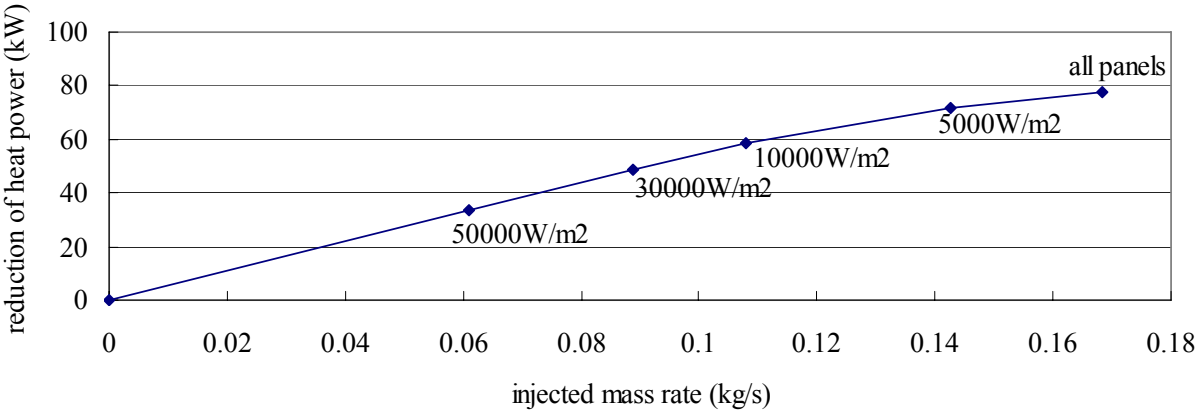

Figure 4-9. Reduction of heat power (kW) v.s. injected mass rate (kg/s) at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

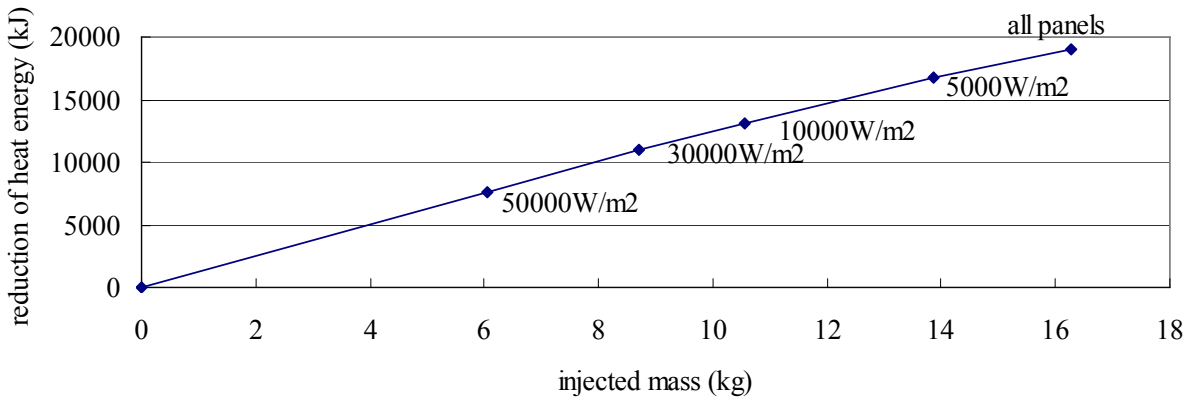

Figure 4-10. Reduction of heat energy (kJ) v.s. injected mass (kg) at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

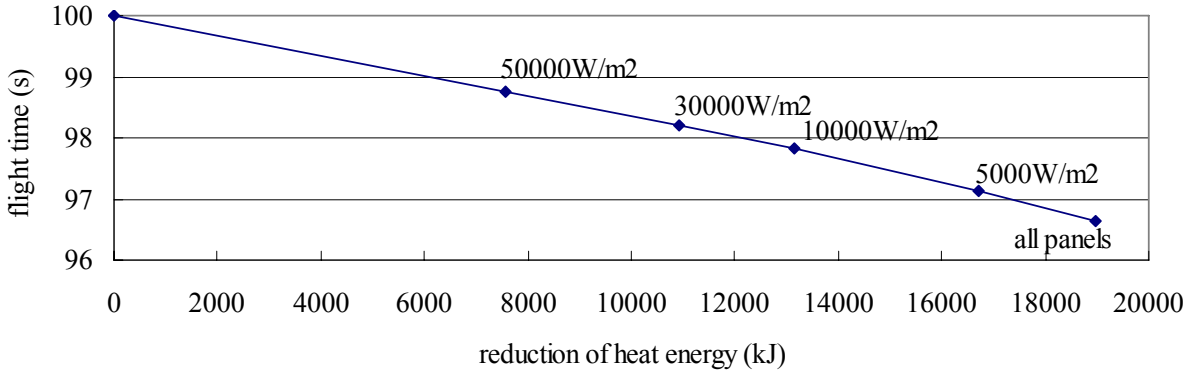

Figure 4-11. Flight time v.s. reduction of heat energy (kJ) at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

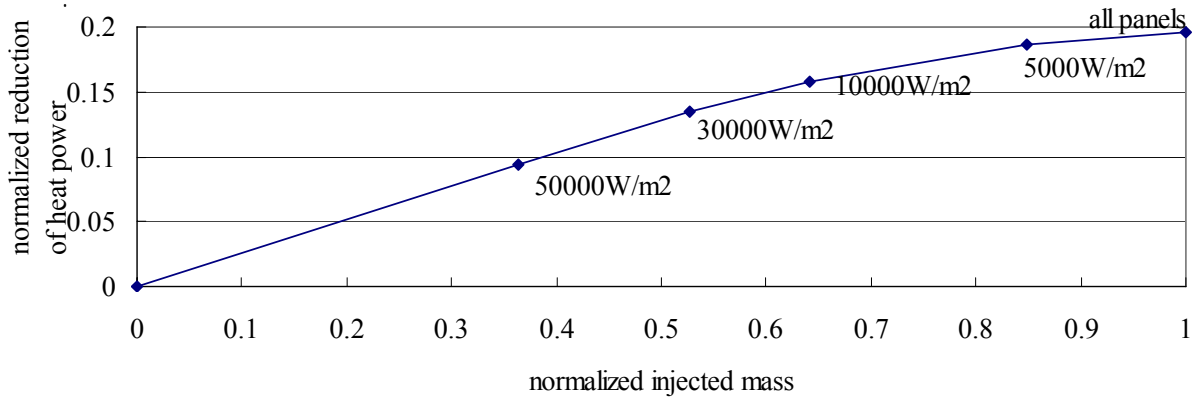

Figure 4-12. Normalized reduction of heat power v.s. normalized injected mass at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

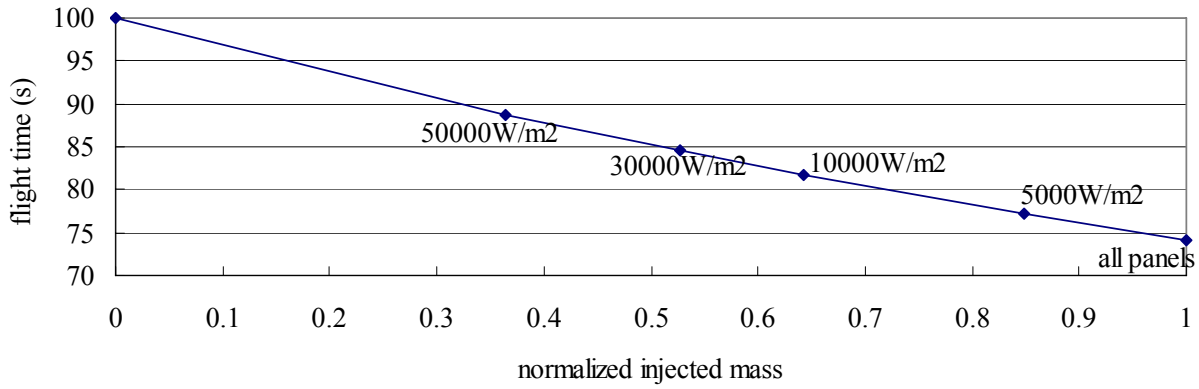

Figure 4-13. Flight time v.s. normalized injected mass at at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

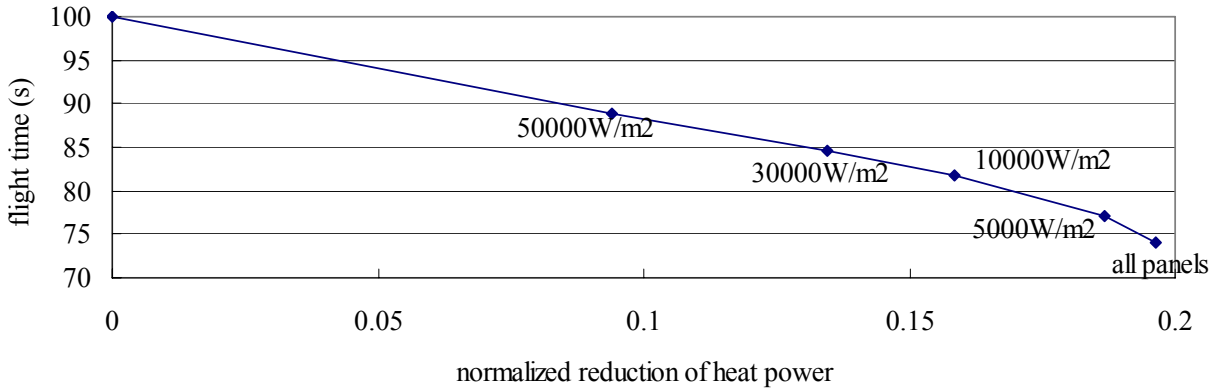

Figure 4-14. Flight time v.s. normalized reduction of heat power at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

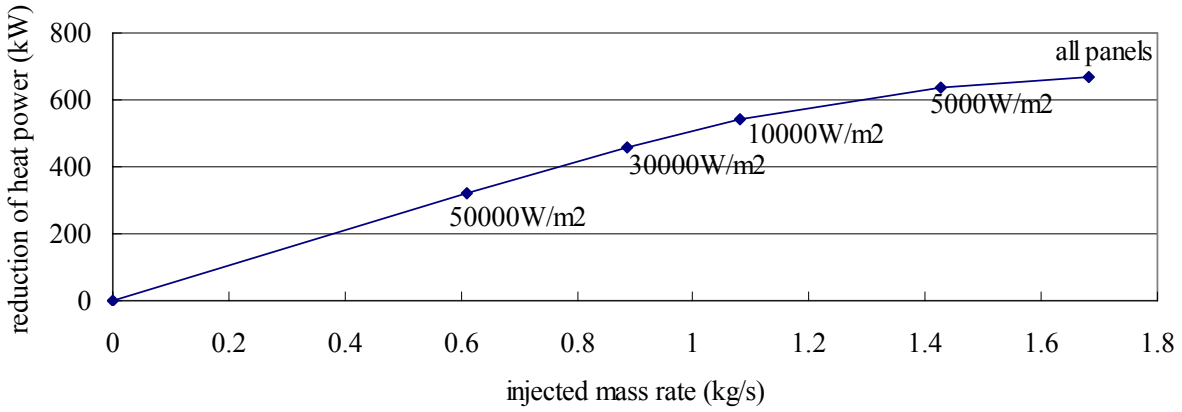

Figure 4-15. Reduction of heat power (kW) v.s. injected mass rate (kg/s) at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

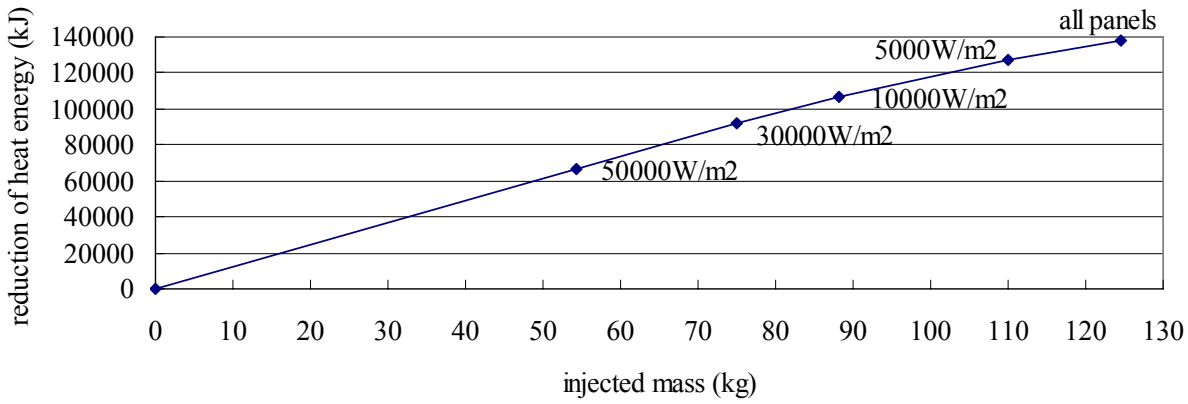

Figure 4-16. Reduction of heat energy (kJ) v.s. injected mass (kg) at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

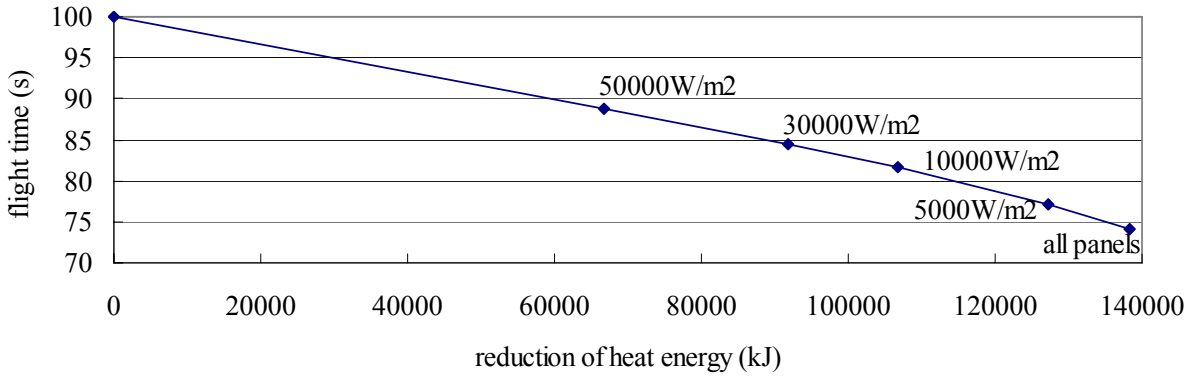

Figure 4-17. Flight time v.s. reduction of heat energy (kJ) at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

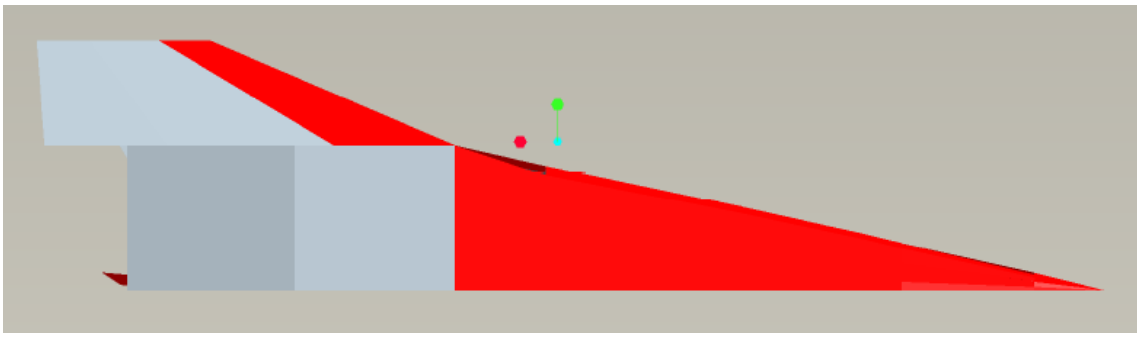

Figure 4-18. Bottom view of X24C with $50000\,W/m^2$ of allowable $q_{c,w}$ at $30000m$ altitude.

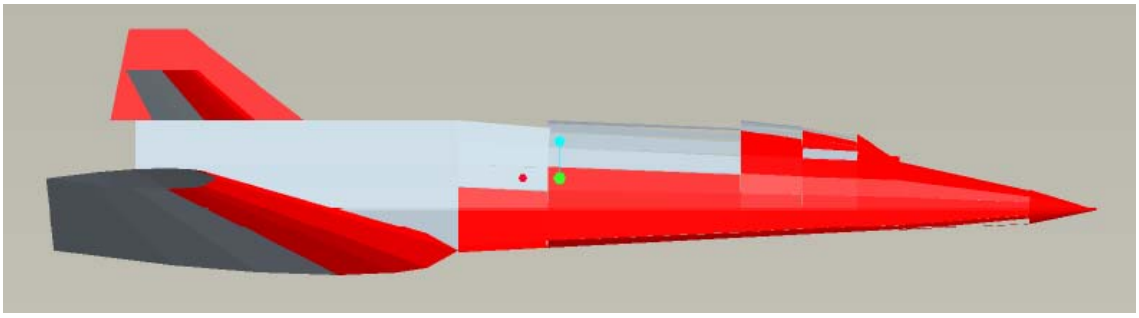

Figure 4-19. Side view of X24C with $50000\,W/m^2$ of allowable $q_{c,w}$ at $30000m$ altitude.

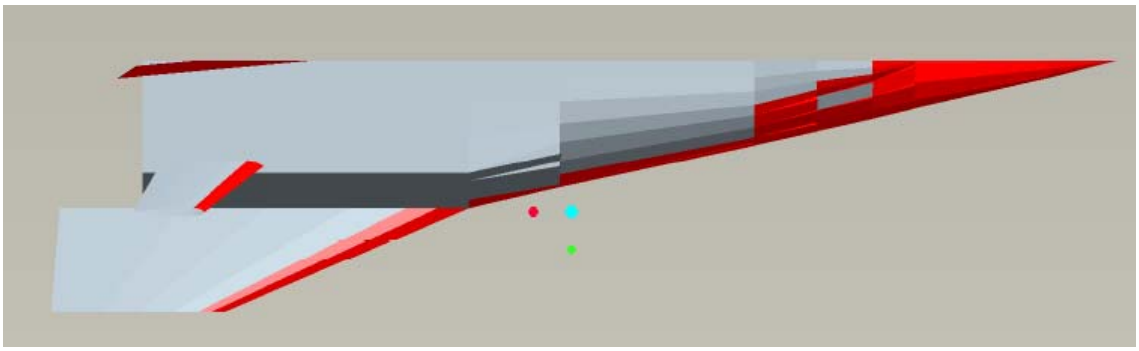

Figure 4-20. Top view of X24C with $50000\,W/m^2$ of allowable $q_{c,w}$ at $30000m$ altitude.

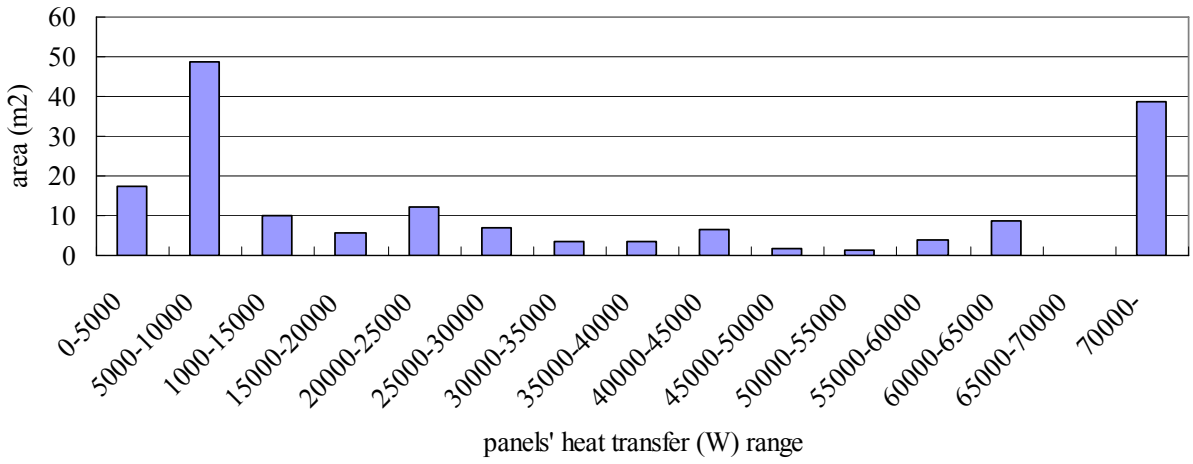

Figure 4-21. Distribution of panel area with a given heat transfer, $\dot{Q}_{c,w}$ (W) at $M = 6$ and 30000m altitude.

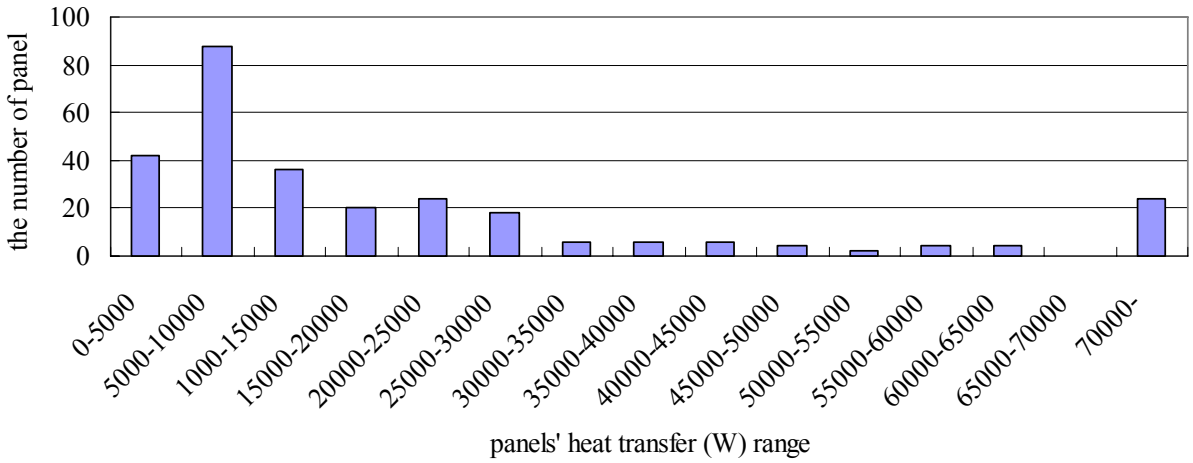

Figure 4-22. Distribution of the number of panels with a given heat transfer, $\dot{Q}_{c,w}$ (W) at $M = 6$ and 30000m altitude.

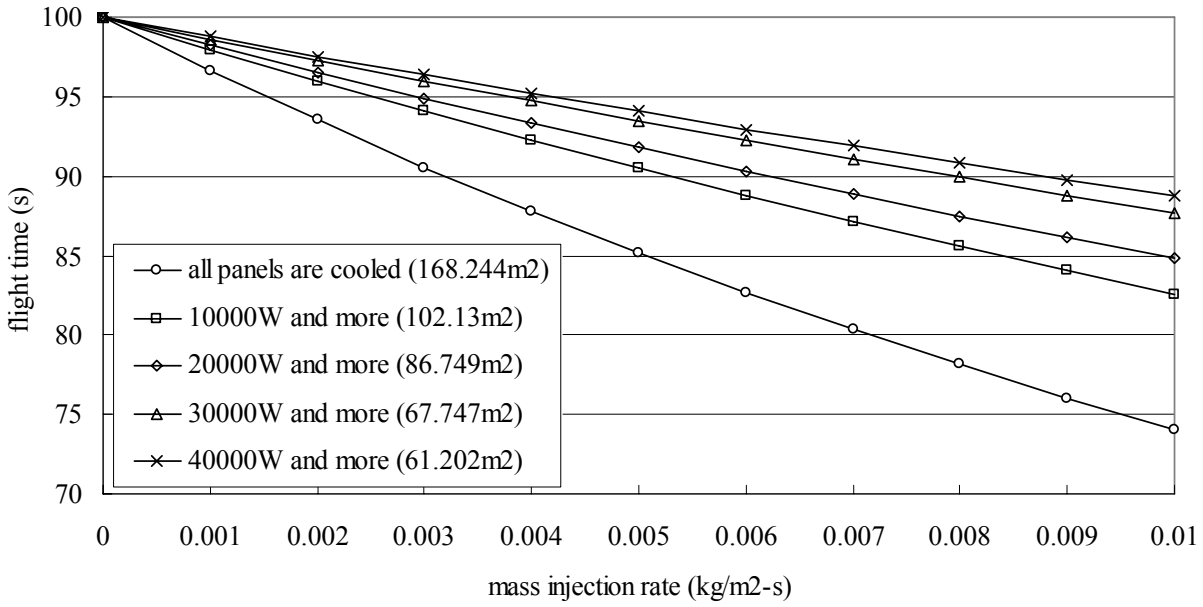

Figure 4-23. Mass injection effect on the flight time at $M = 6$ and 30000*m* altitude. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

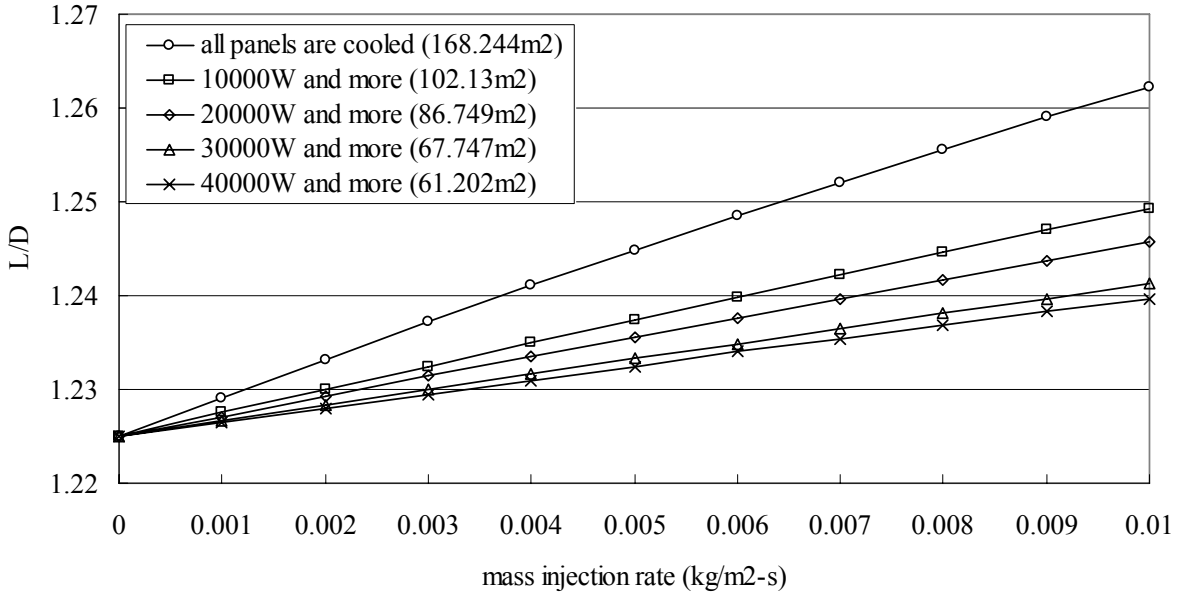

Figure 4-24. Mass injection effect on L/D at $M = 6$ and 30000*m* altitude. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

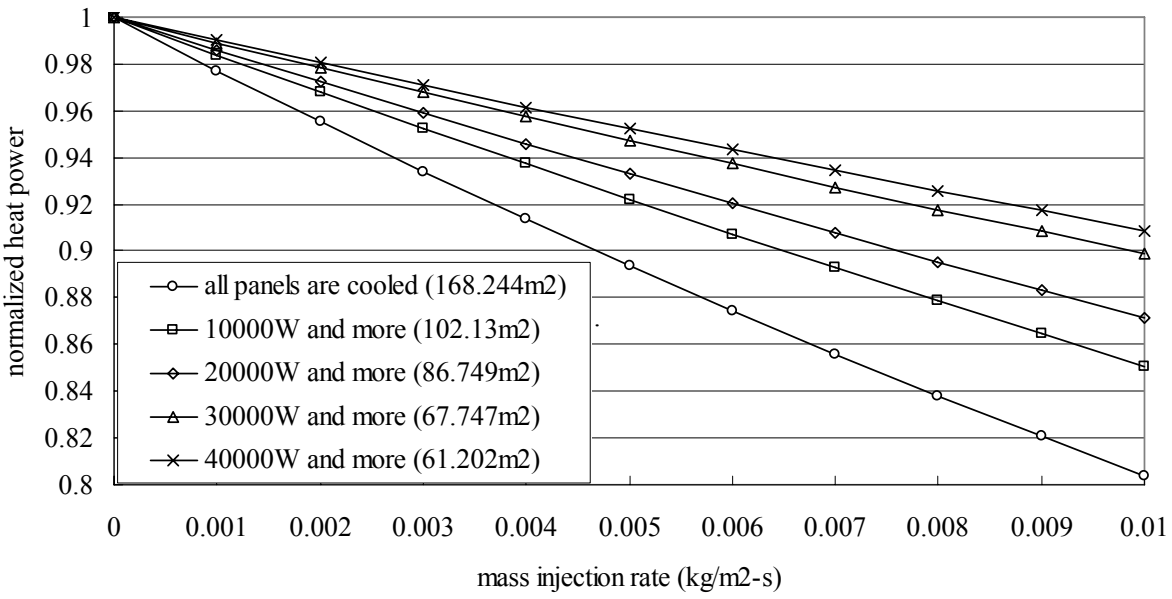

Figure 4-25. Mass injection effect on reduction of heat power at $M = 6$ and 30000m altitude. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

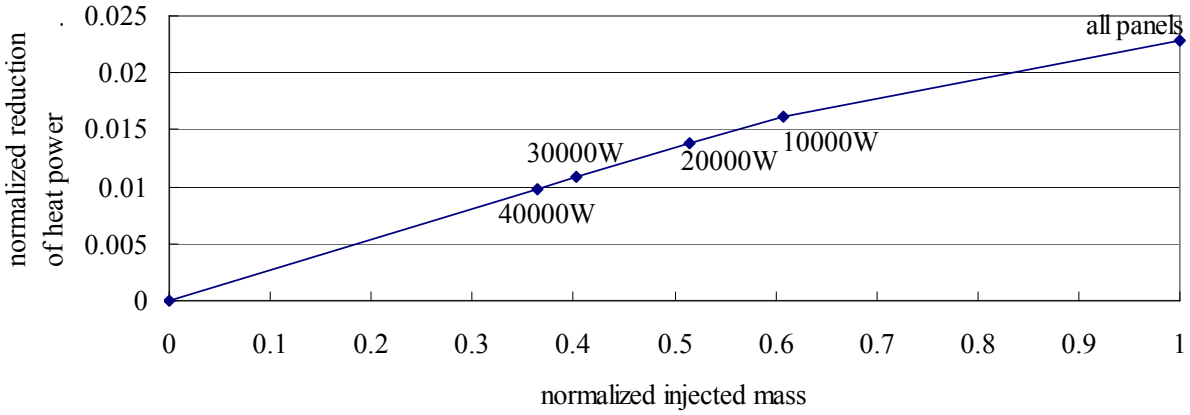

Figure 4-26. Normalized reduction of heat power v.s. normalized injected mass at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

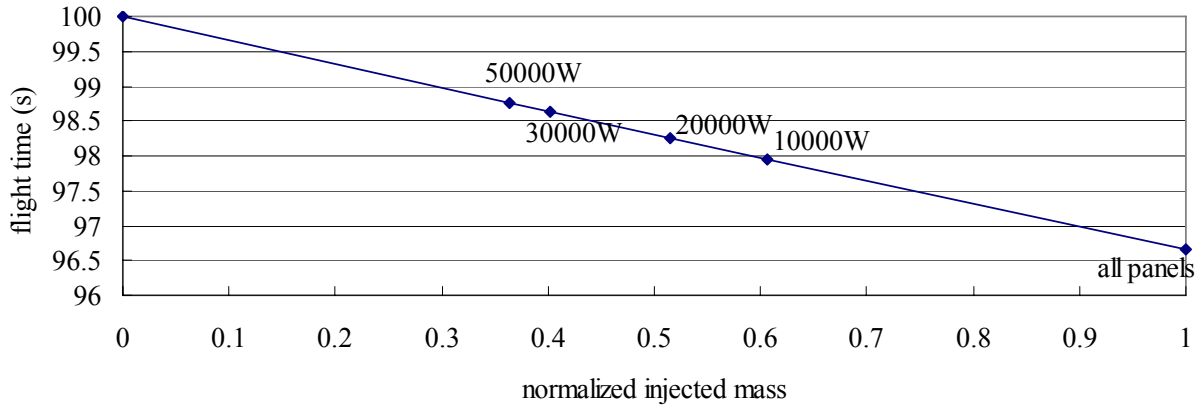

Figure 4-27. Flight time v.s. normalized injected mass at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

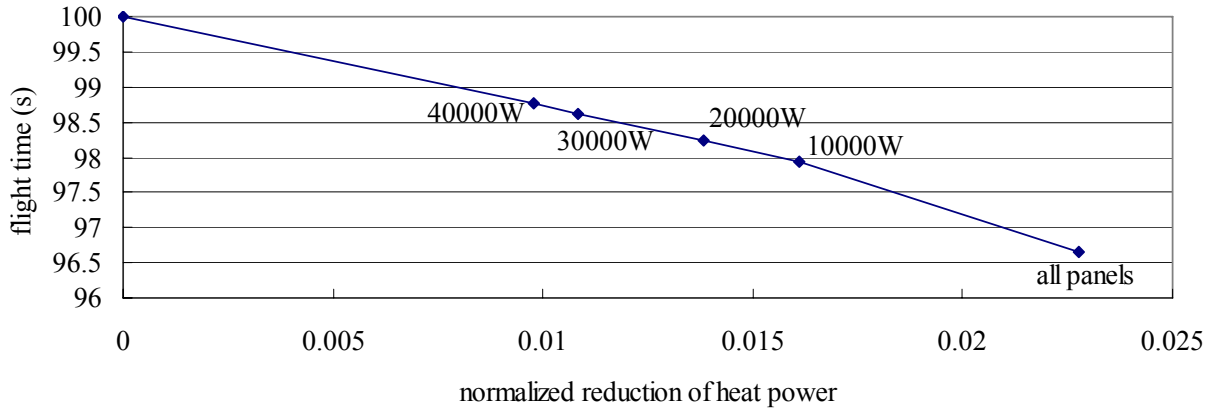

Figure 4-28. Flight time v.s. normalized reduction of heat power at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

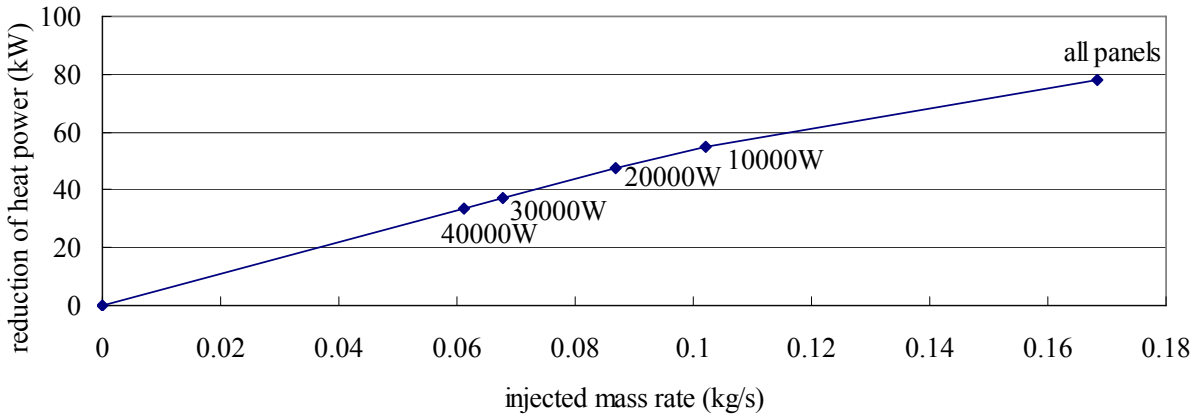

Figure 4-29. Reduction of heat power (kW) v.s. injected mass rate (kg/s) at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

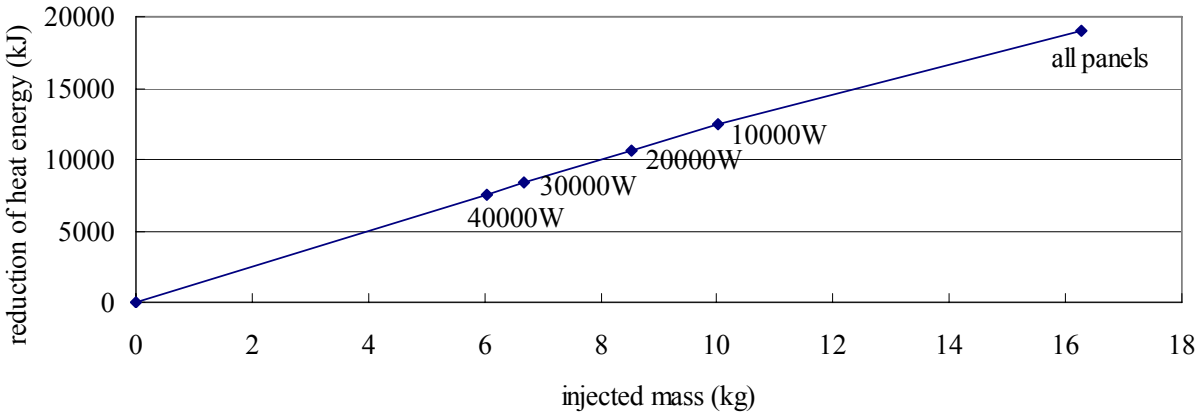

Figure 4-30. Reduction of heat energy (kJ) v.s. injected mass (kg) at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

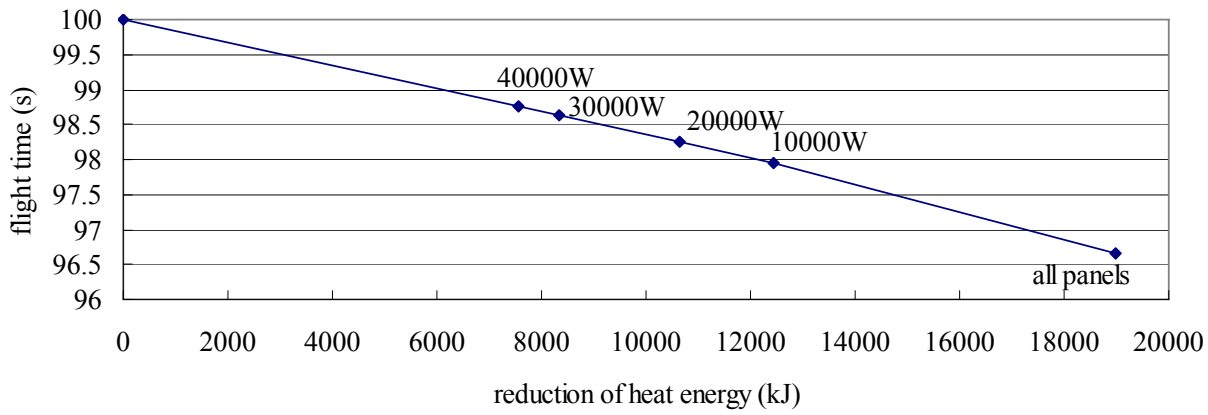

Figure 4-31. Flight time v.s. reduction of heat energy (kJ) at $M = 6$ and 30000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

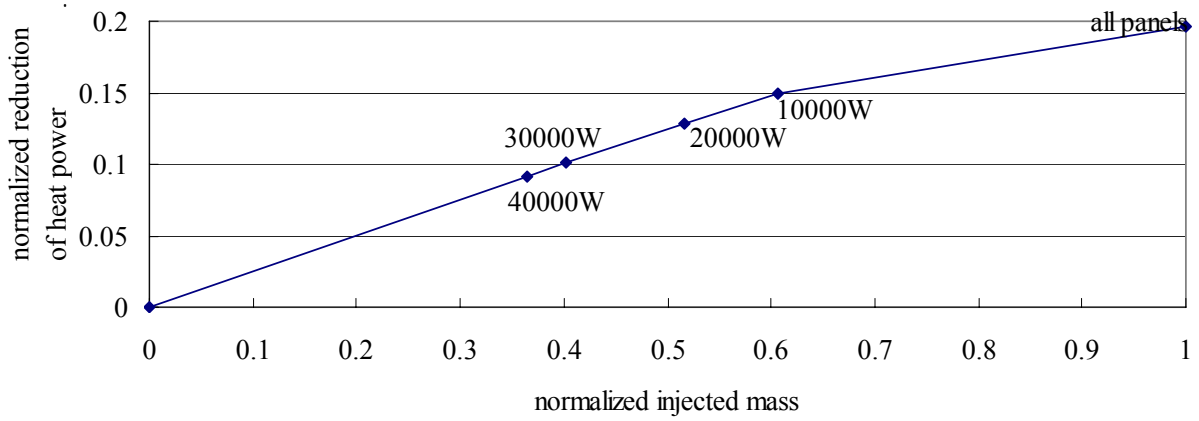

Figure 4-32. Normalized reduction of heat power v.s. normalized injected mass at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

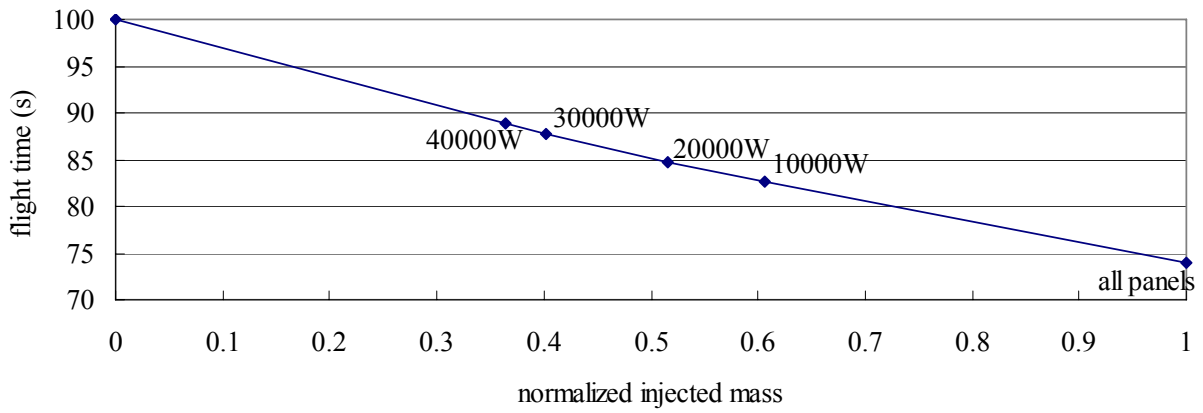

Figure 4-33. Flight time v.s. normalized injected mass at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

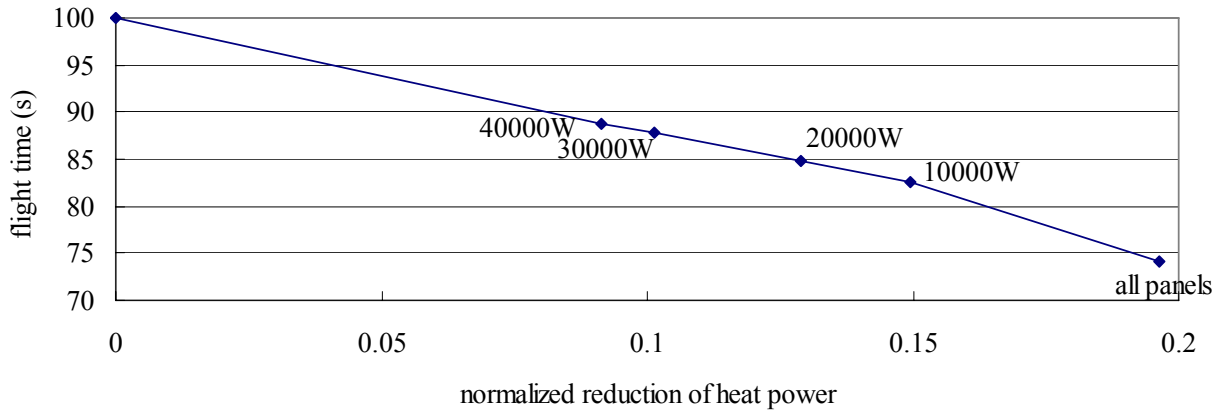

Figure 4-34. Flight time v.s. normalized reduction of heat power at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

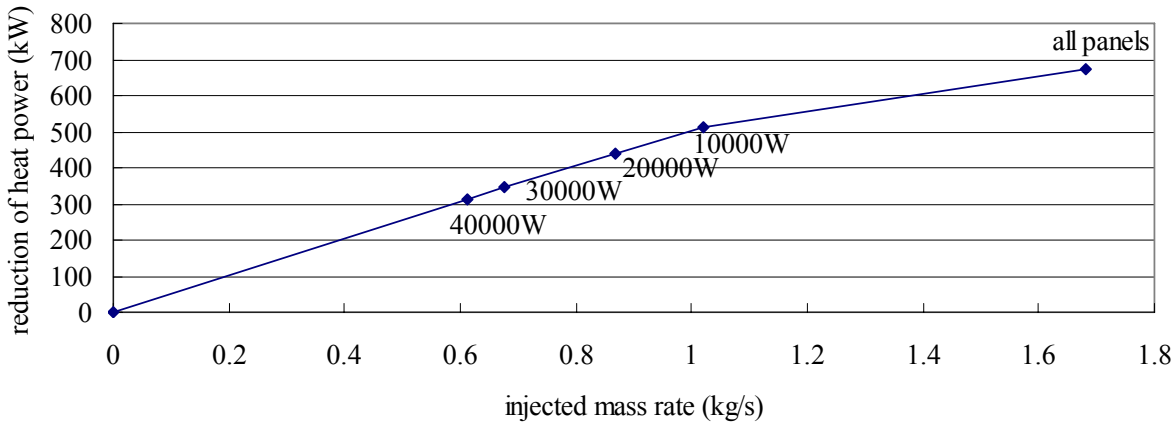

Figure 4-35. Reduction of heat power (kW) v.s. injected mass rate (kg/s) at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

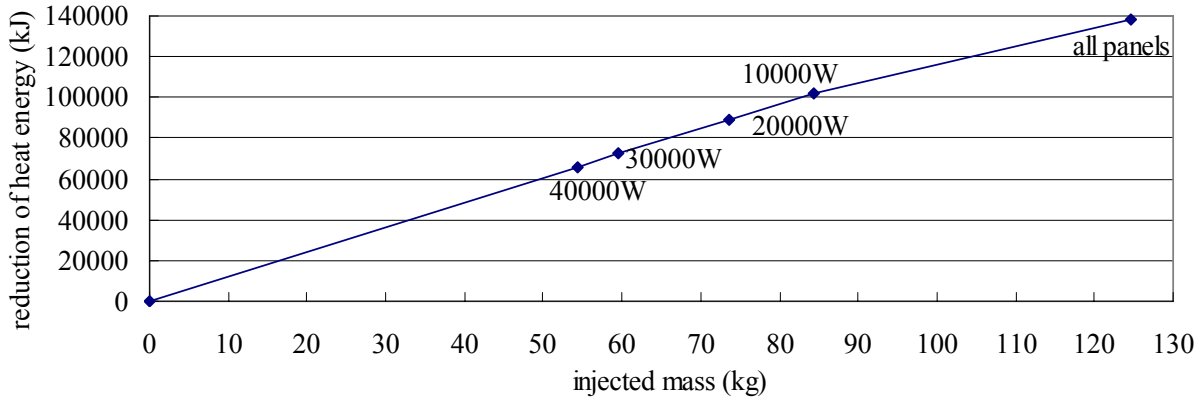

Figure 4-36. Reduction of heat energy (kJ) v.s. injected mass (kg) at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

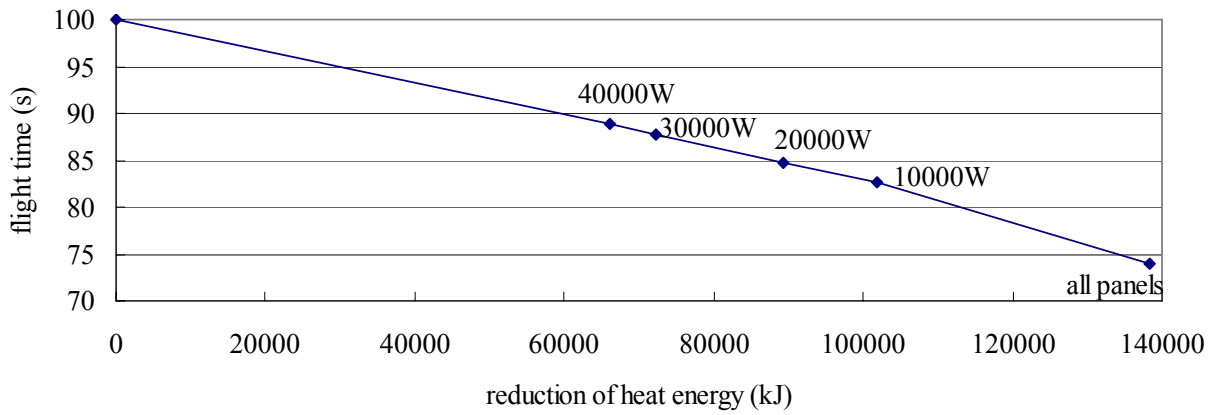

Figure 4-37. Flight time v.s. reduction of heat energy (kJ) at $M = 6$ and 30000m altitude with $0.01\ kg/m^2$ s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

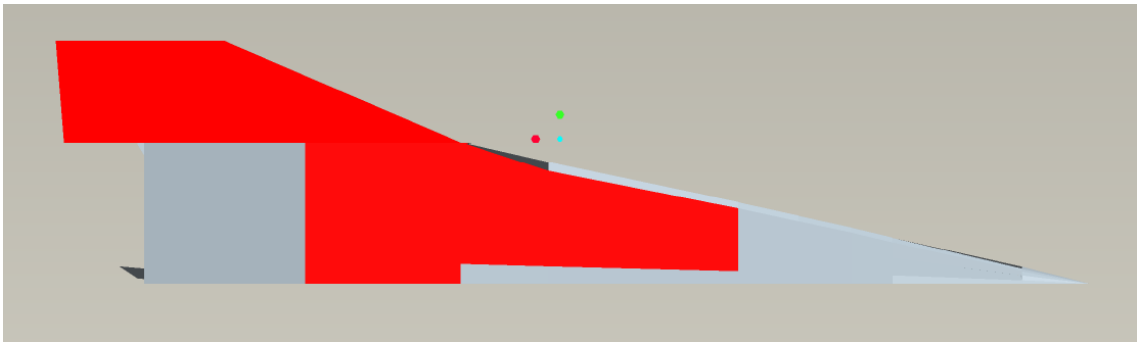

Figure 4-38. Bottom view of X-24C with $40000\,W$ of allowable $\dot{Q}_{c,w}$ at $M = 6$ and 30000m altitude.

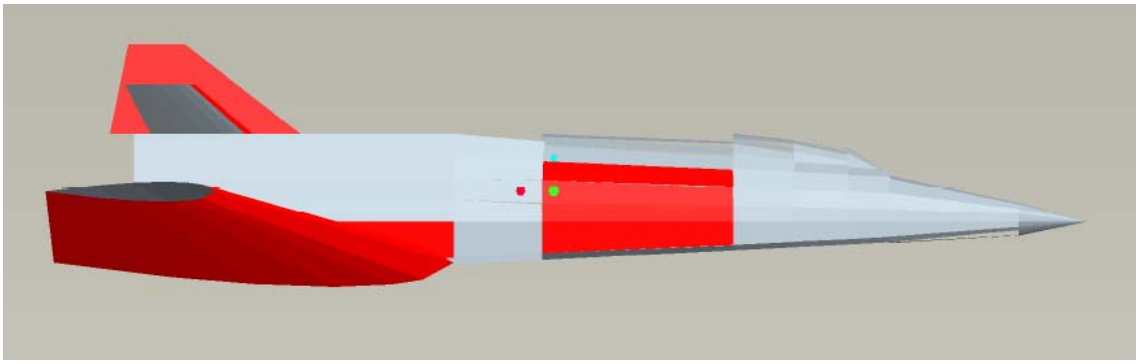

Figure 4-39. Side view of X-24C with $40000\,W$ of allowable $\dot{Q}_{c,w}$ at $M = 6$ and 30000m altitude.

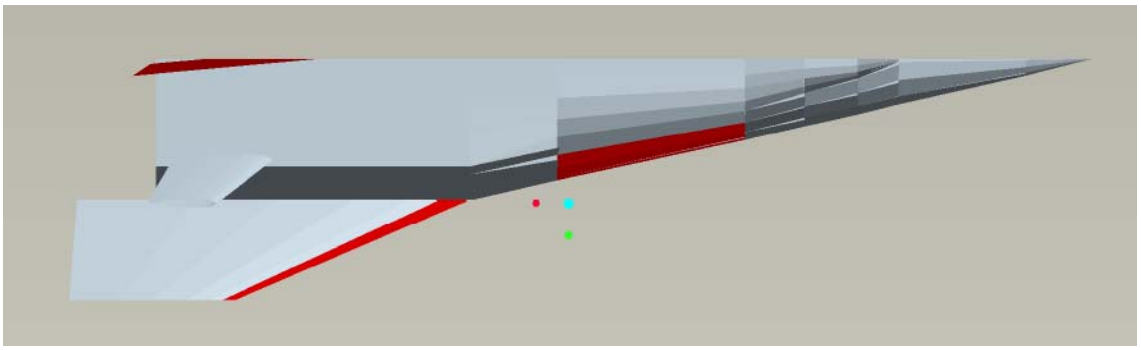

Figure 4-40. Top view of X-24C with 40000W of allowable $\dot{Q}_{c,w}$ at $M = 6$ and 30000m altitude.

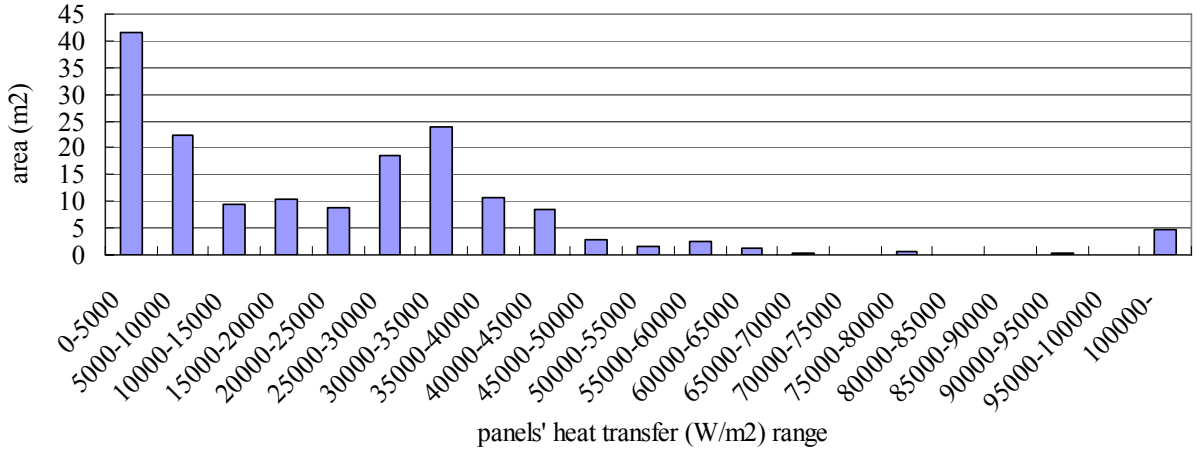

Figure 4-41. Distribution of panel area with a given heat transfer, $q_{c,w}\,(W/m^2)$ at $M = 6$ and 35000m altitude.

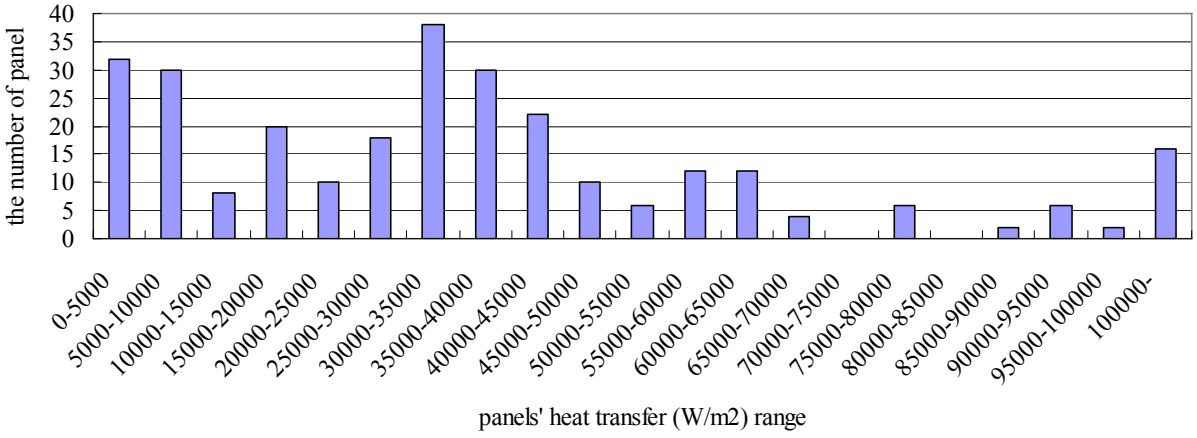

Figure 4-42. Distribution of the number of panels with a given heat transfer, $q_{c,w}\,(W/m^2)$ at $M = 6$ and 35000m altitude.

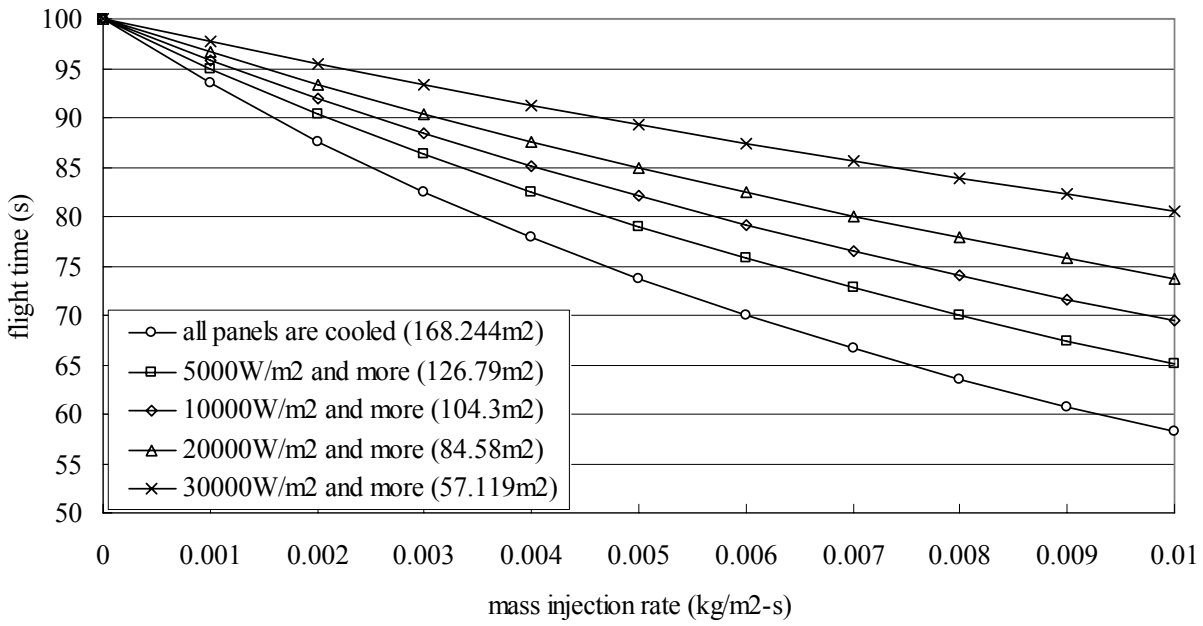

Figure 4-43. Mass injection effect on the flight time at $M = 6$ and 35000m altitude. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

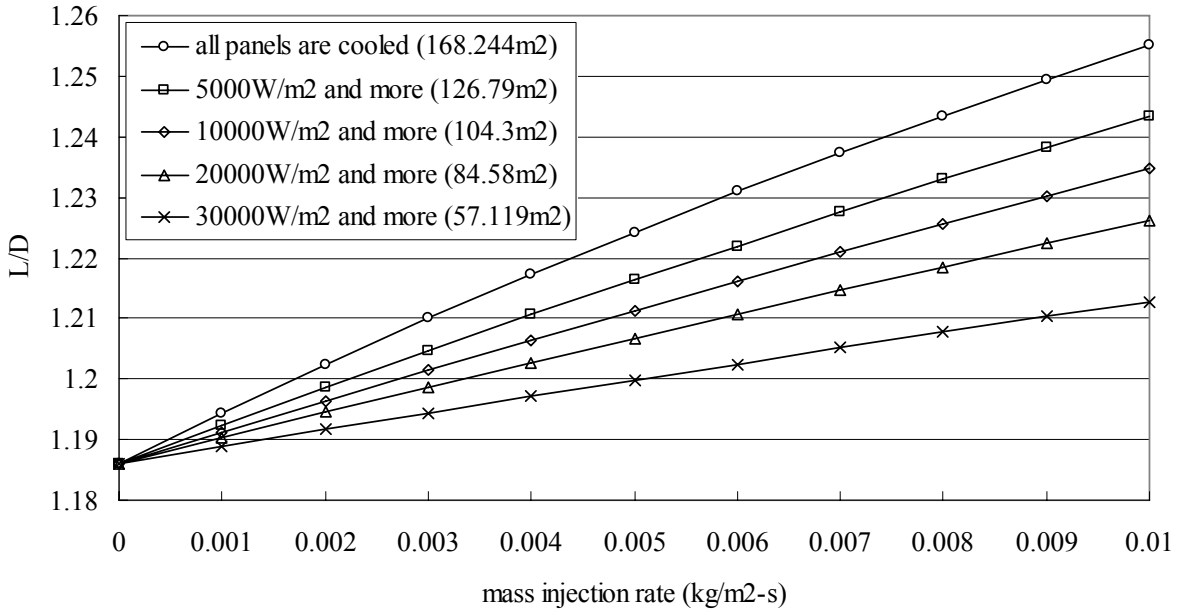

Figure 4-44. Mass injection effect on L/D at $M = 6$ and 35000m altitude. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

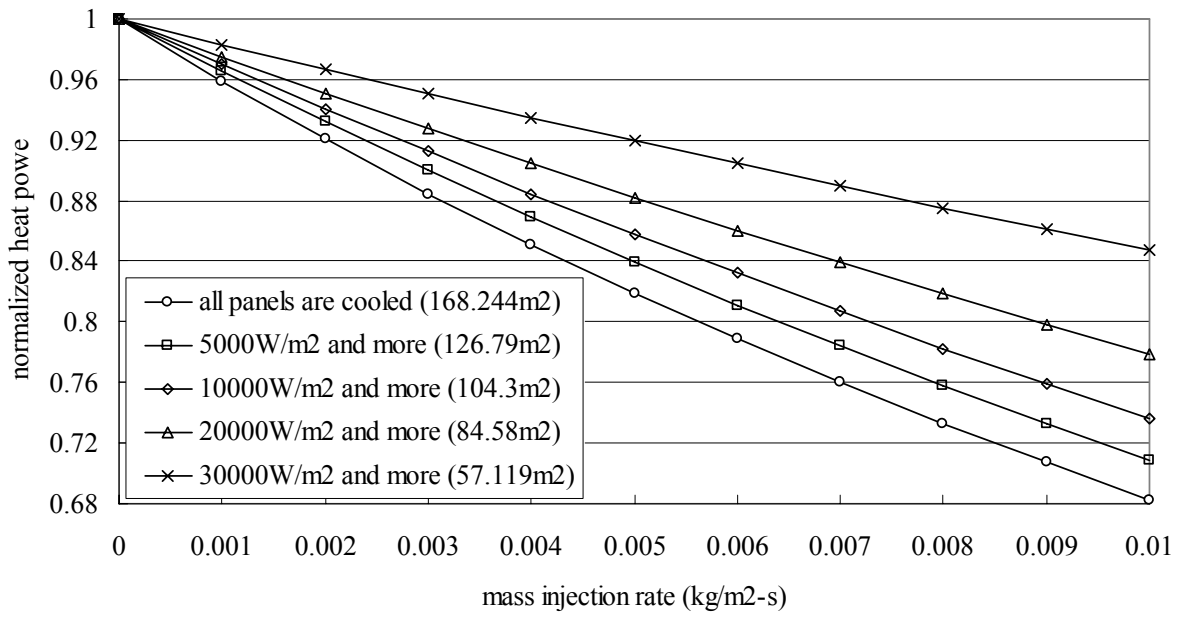

Figure 4-45. Mass injection effect on reduction of heat power at $M = 6$ and 35000m altitude. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

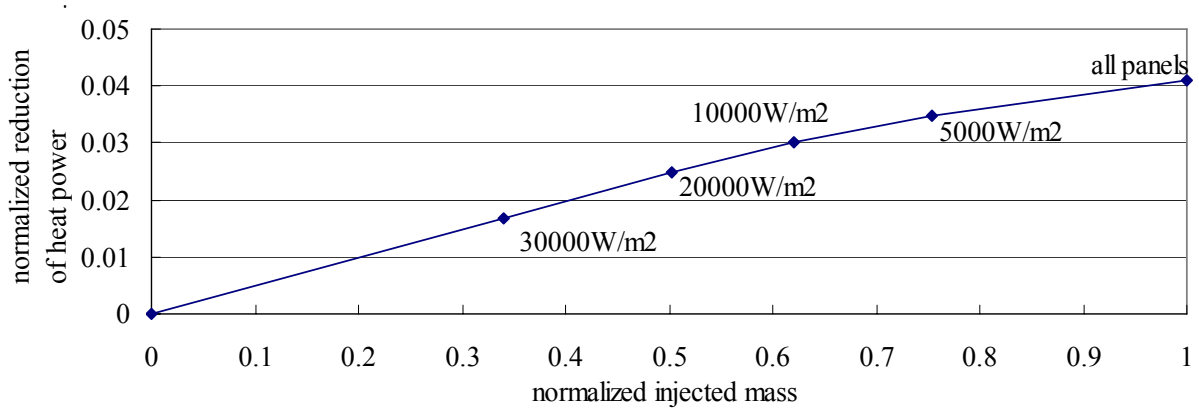

Figure 4-46. Normalized reduction of heat power v.s. normalized injected mass at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

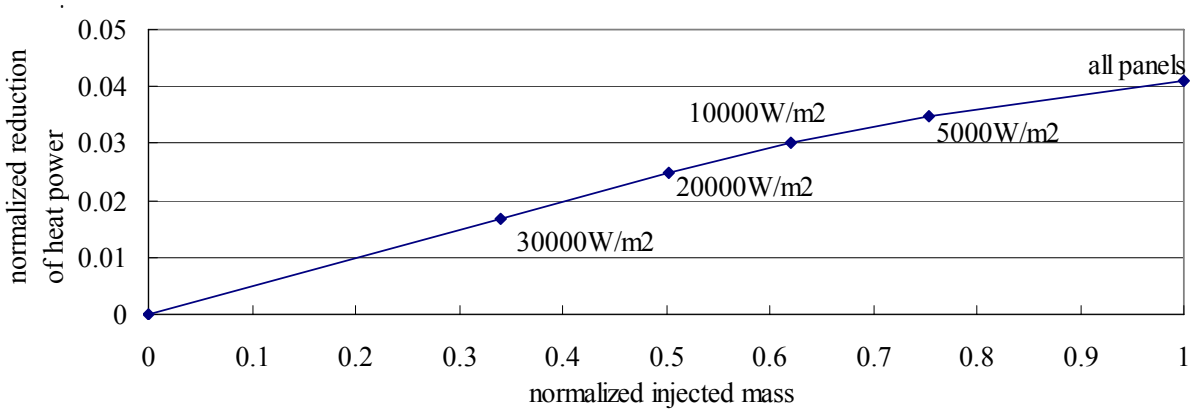

Figure 4-47. Flight time v.s. normalized injected mass at at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}(W/m^2)$.

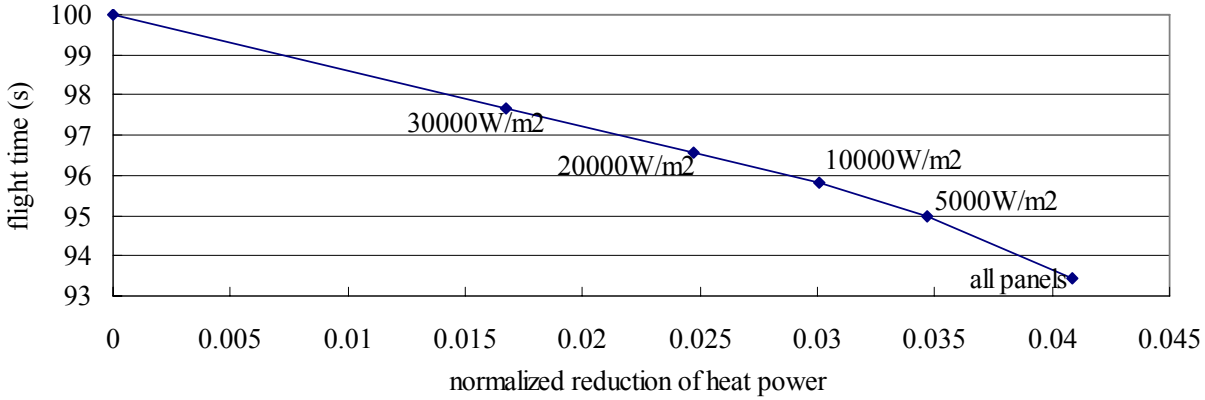

Figure 4-48. Flight time v.s. normalized reduction of heat power at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}(W/m^2)$.

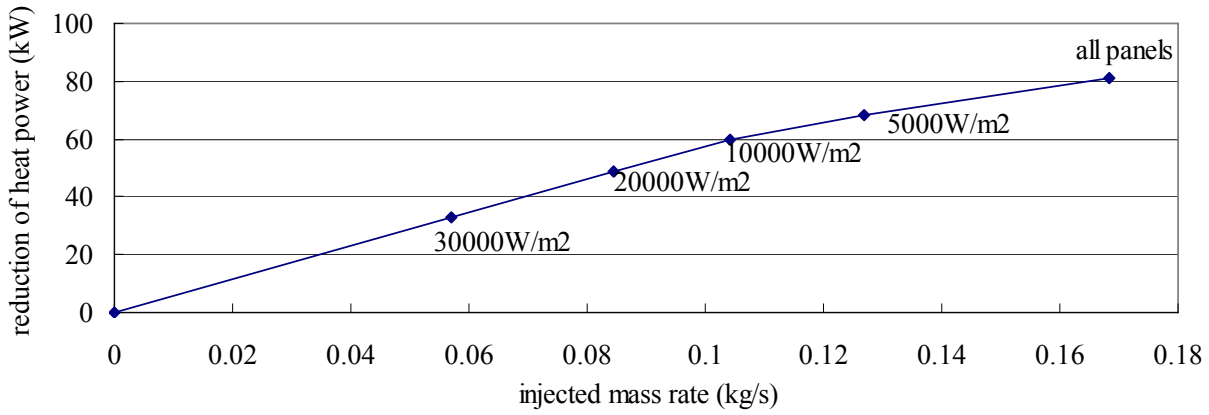

Figure 4-49. Reduction of heat power (kW) v.s. injected mass rate (kg/s) at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

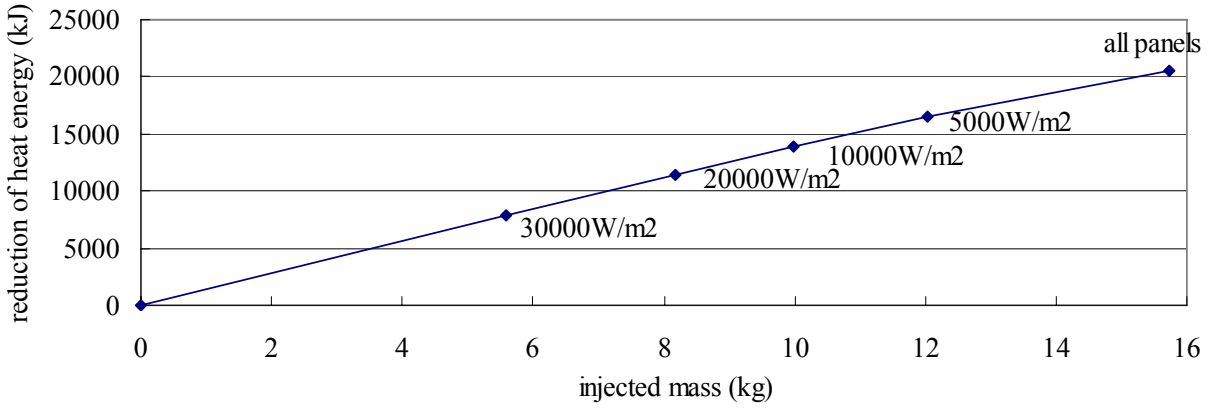

Figure 4-50. Reduction of heat energy (kJ) v.s. injected mass (kg) at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

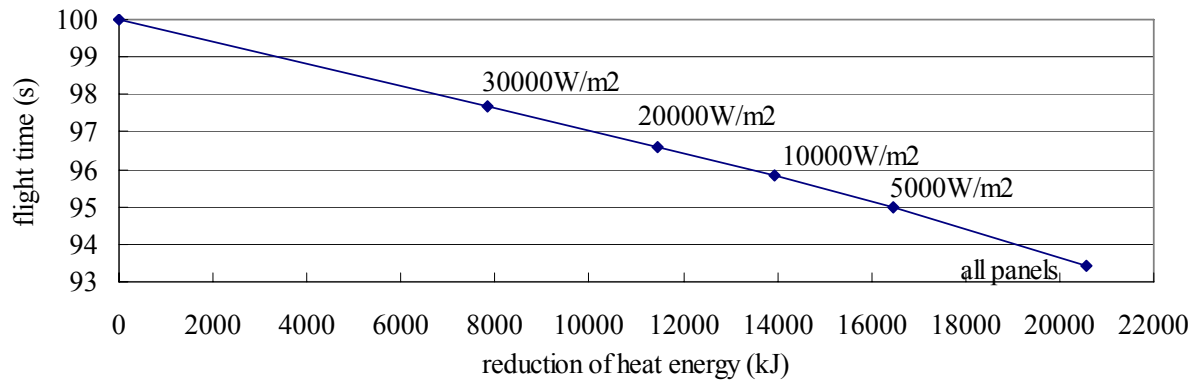

Figure 4-51. Flight time v.s. reduction of heat energy (kJ) at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

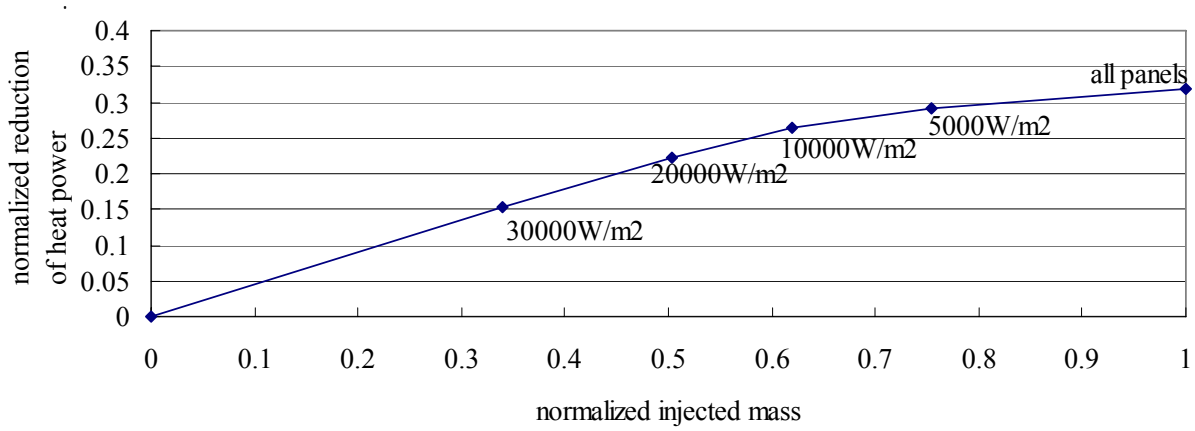

Figure 4-52. Normalized reduction of heat power v.s. normalized injected mass at $M = 6$ and 35000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

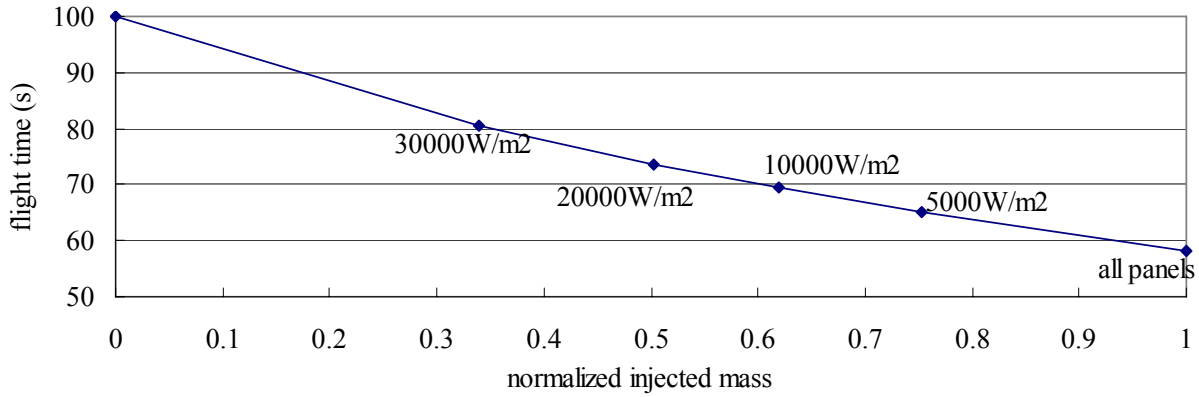

Figure 4-53. Flight time v.s. normalized injected mass at at $M = 6$ and 35000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

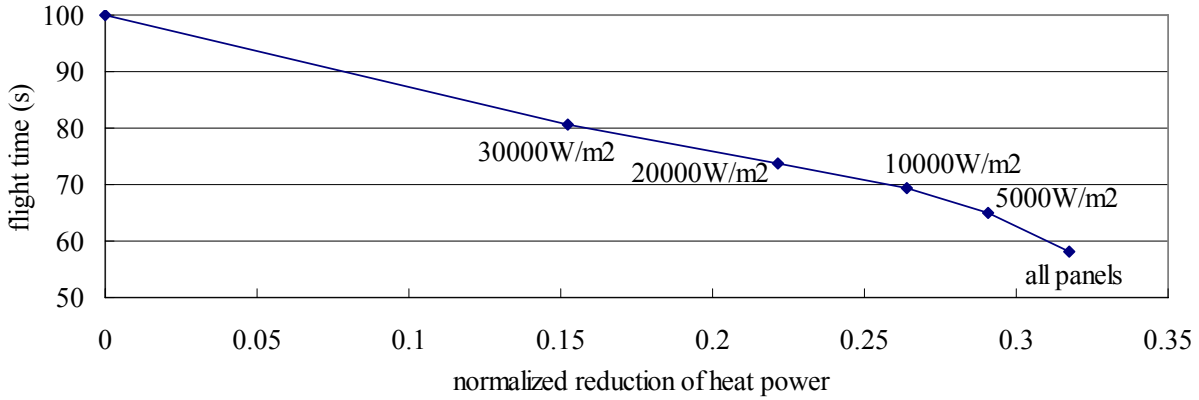

Figure 4-54. Flight time v.s. normalized reduction of heat power at $M = 6$ and 35000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}\,(W/m^2)$.

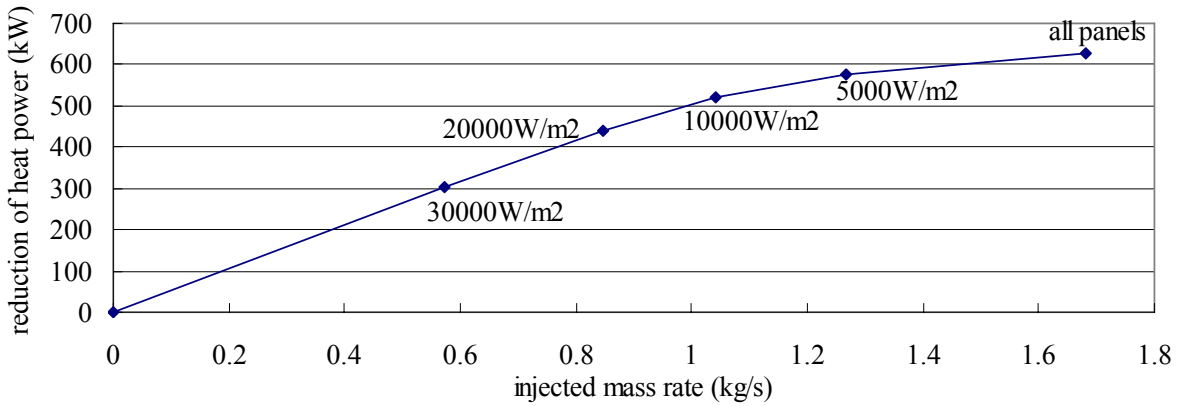

Figure 4-55. Reduction of heat power (kW) v.s. injected mass rate (kg/s) at $M = 6$ and 35000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}(W/m^2)$.

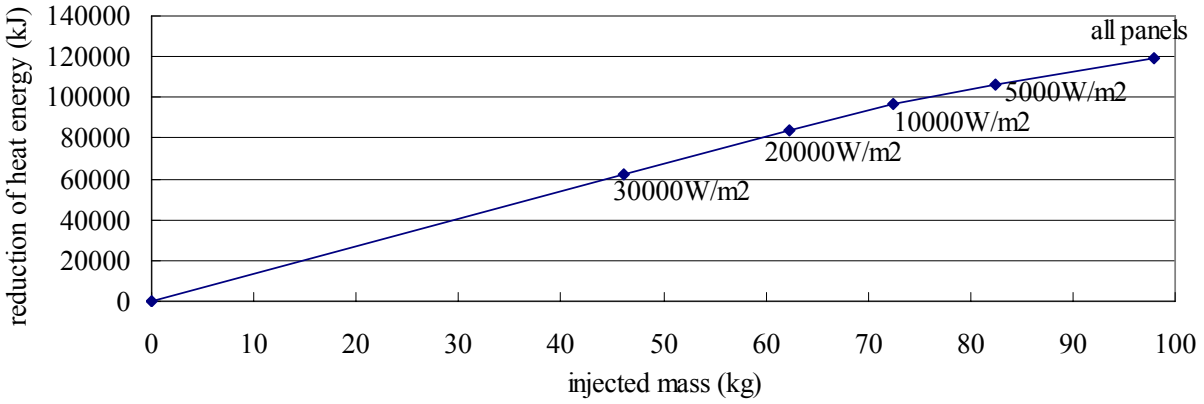

Figure 4-56. Reduction of heat energy (kJ) v.s. injected mass (kg) at $M = 6$ and 35000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}(W/m^2)$.

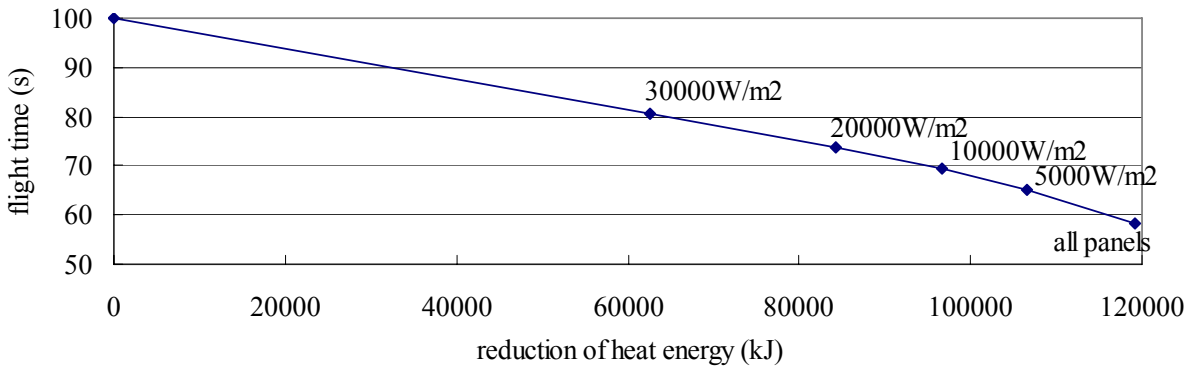

Figure 4-57. Flight time v.s. reduction of heat energy (kJ) at $M = 6$ and 35000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $q_{c,w}$ (W/m^2).

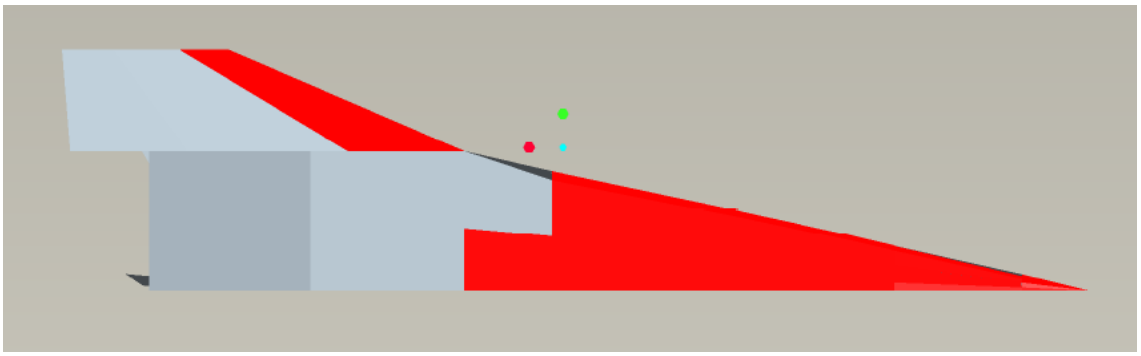

Figure 4-58. Bottom view of X-24C with 30000 W/m^2 of allowable $q_{c,w}$ at 35000m altitude.

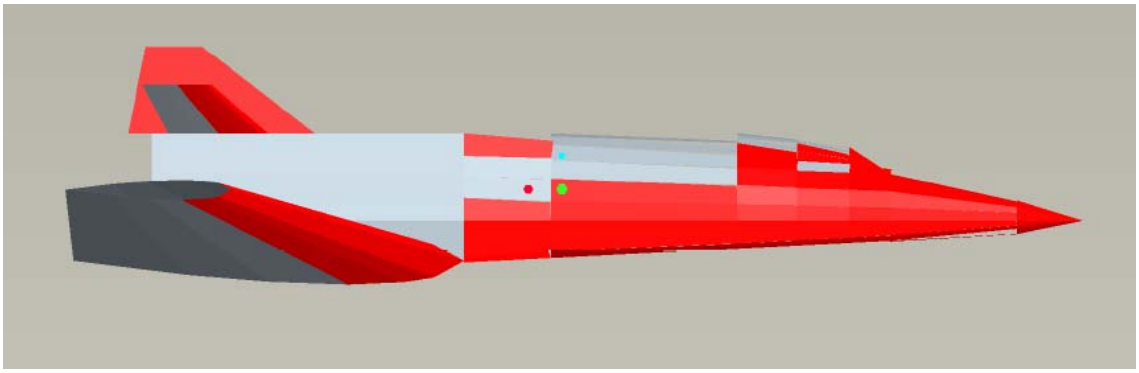

Figure 4-59. Side view of X-24C with 30000 W/m^2 of allowable $q_{c,w}$ at 35000m altitude.

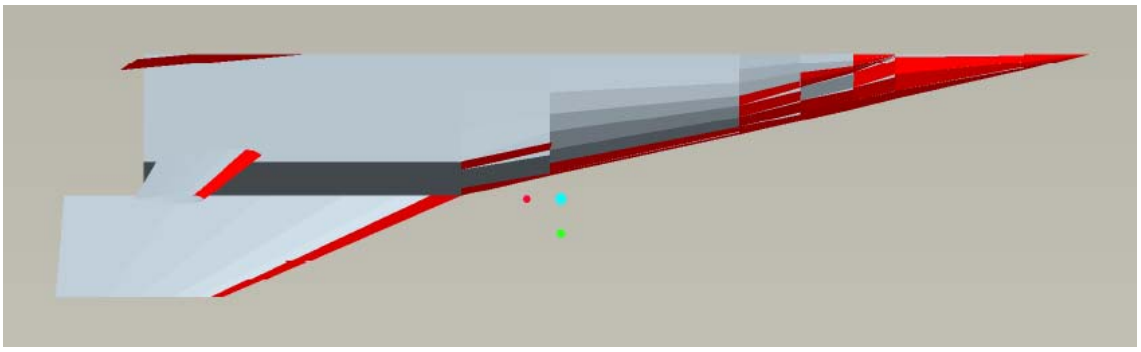

Figure 4-60. Top view of X-24C with 30000W/m^2 of allowable $q_{c,w}$ at 35000m altitude.

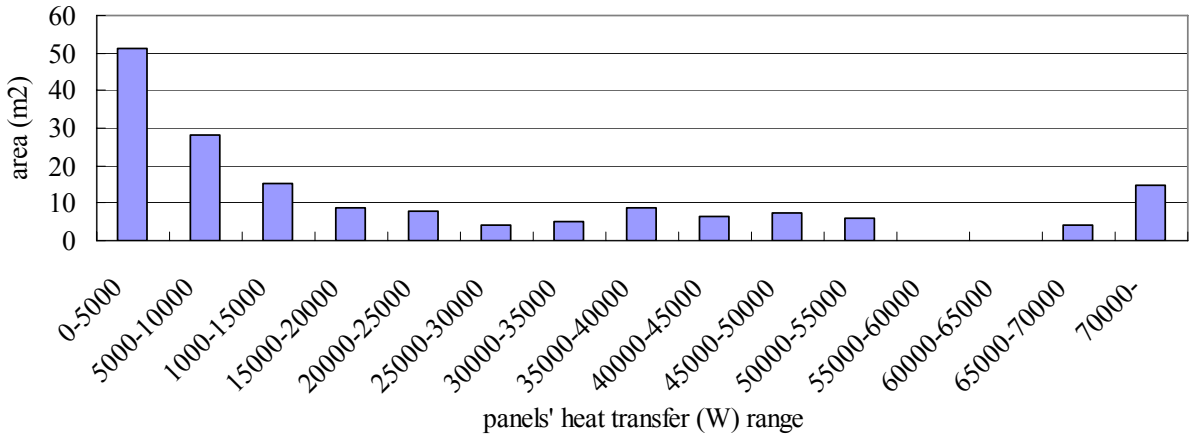

Figure 4-61. Distribution of panel area with a given heat transfer, $\dot{Q}_{c,w}(W)$ at $M=6$ and 35000m altitude.

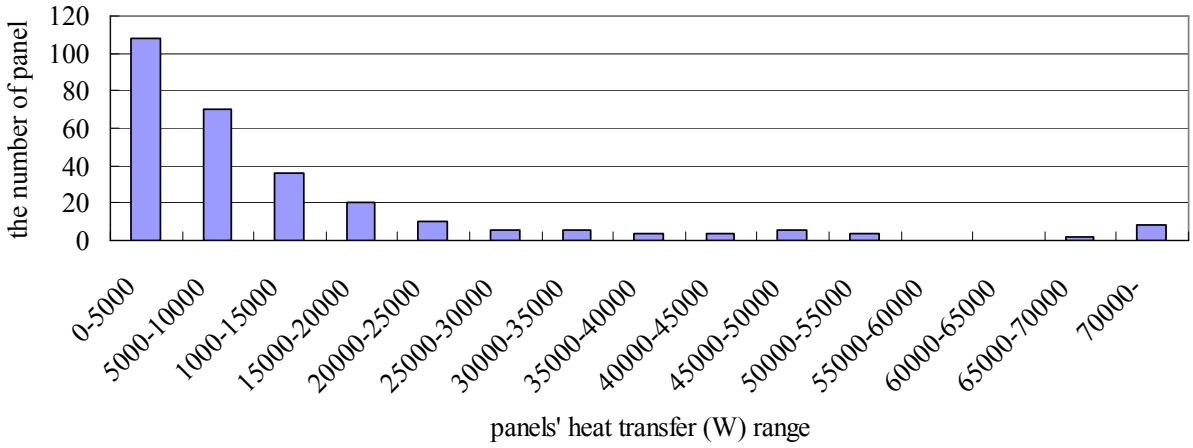

Figure 4-62. Distribution of the number of panels with a given heat transfer, $\dot{Q}_{c,w}(W)$ at $M=6$ and 35000m altitude.

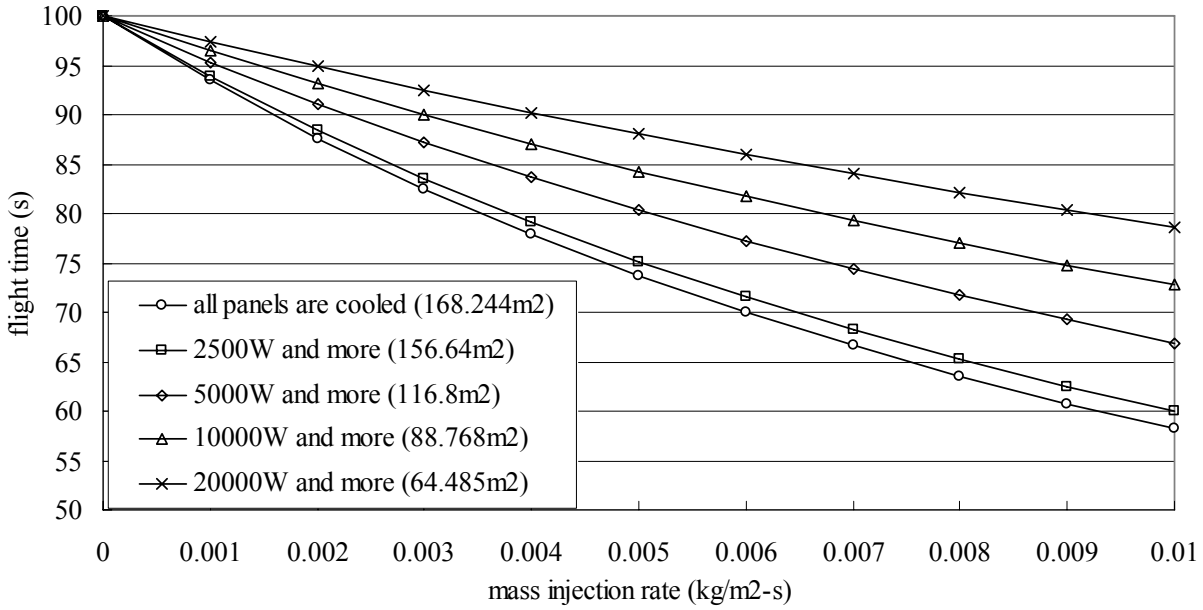

Figure 4-63. Mass injection effect on the flight time at $M = 6$ and 35000*m* altitude. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

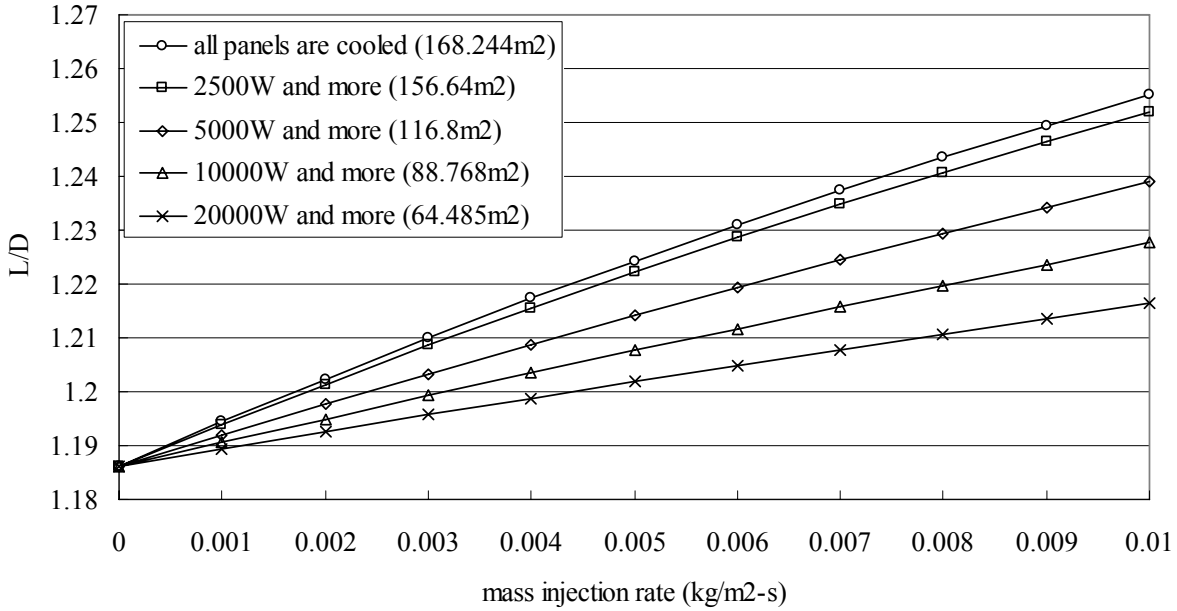

Figure 4-64. Mass injection effect on L/D at $M = 6$ and 35000*m* altitude. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

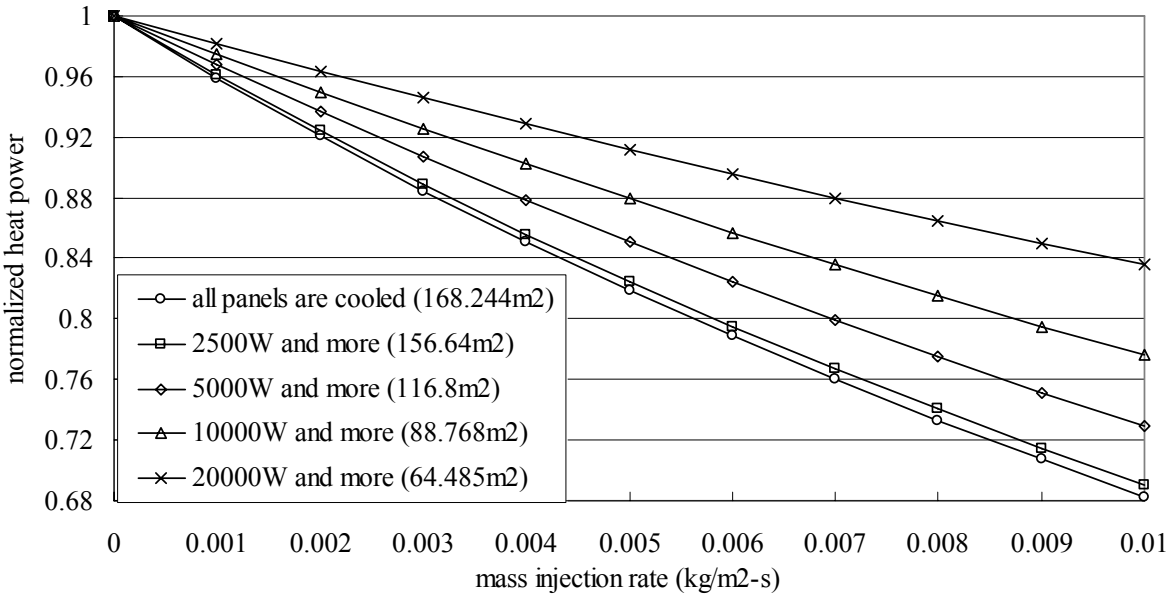

Figure 4-65. Mass injection effect on reduction of heat power at $M = 6$ and 35000m altitude. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

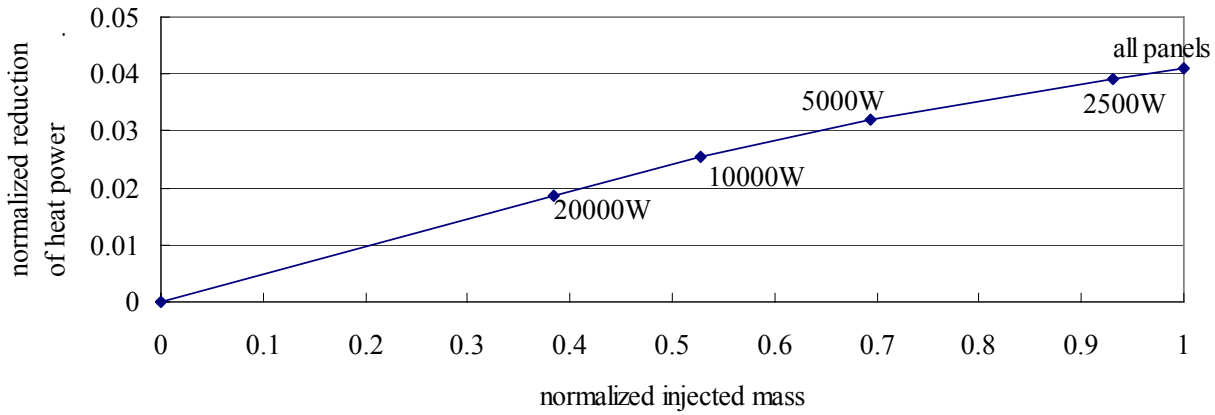

Figure 4-66. Normalized reduction of heat power v.s. normalized injected mass at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

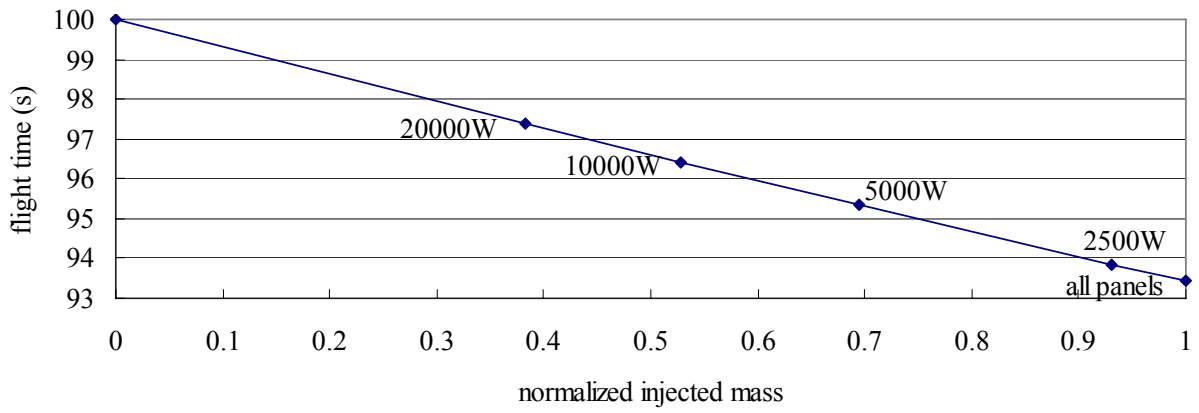

Figure 4-67. Flight time v.s. normalized injected mass at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

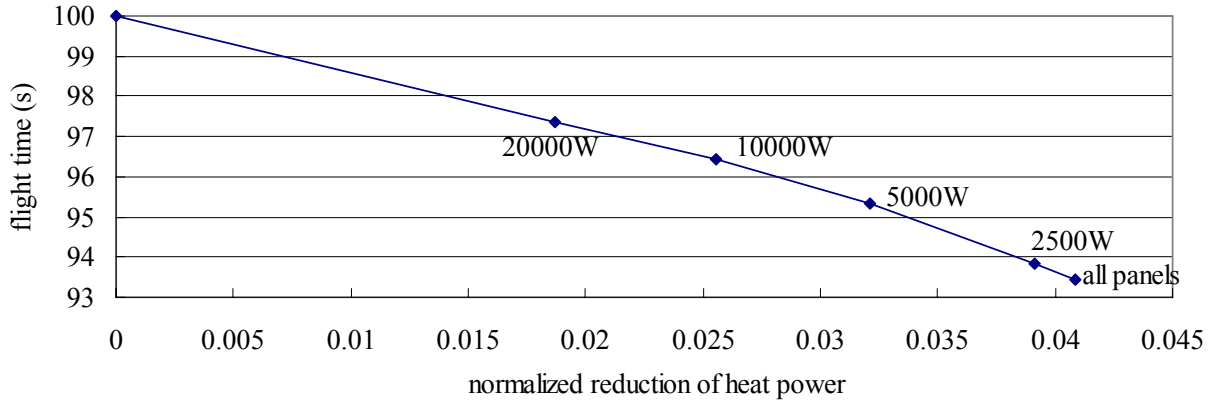

Figure 4-68. Flight time v.s. normalized reduction of heat power at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

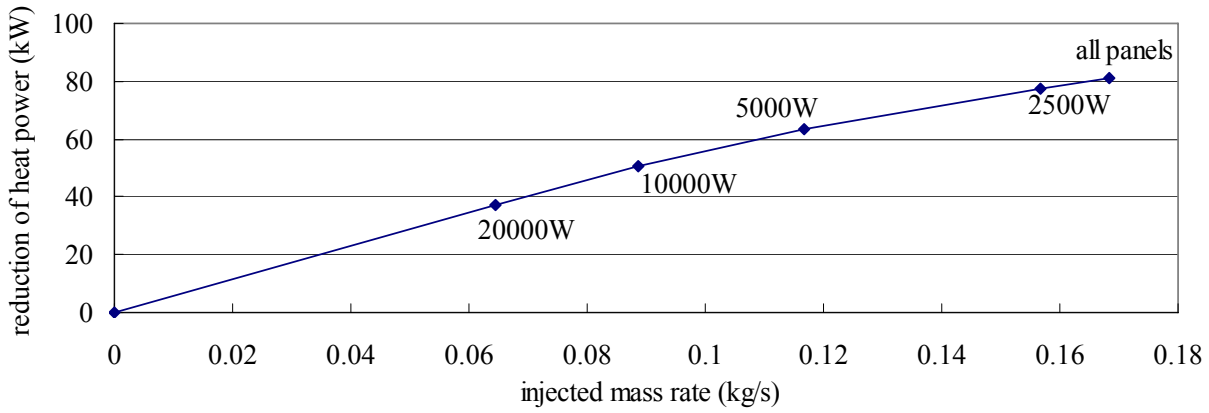

Figure 4-69. Reduction of heat power (kW) v.s. injected mass rate (kg/s) at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

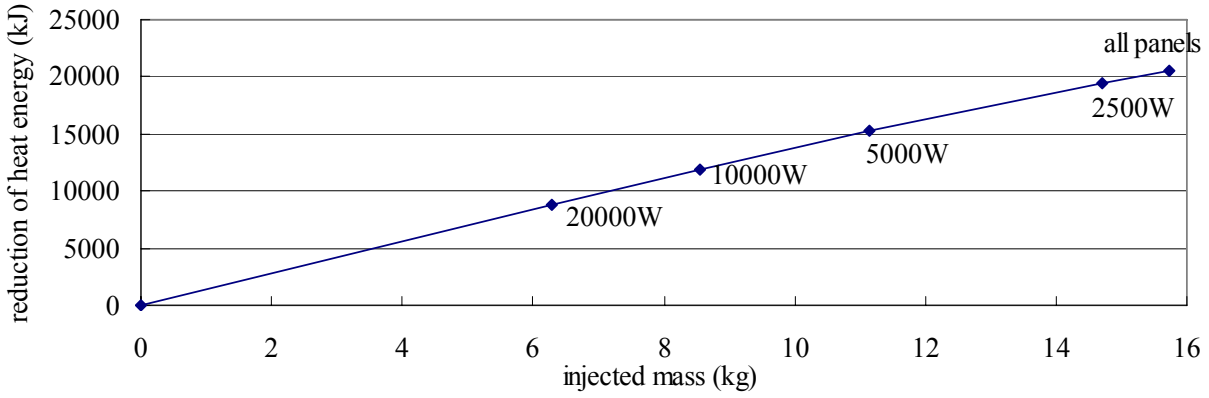

Figure 4-70. Reduction of heat energy (kJ) v.s. injected mass (kg) at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

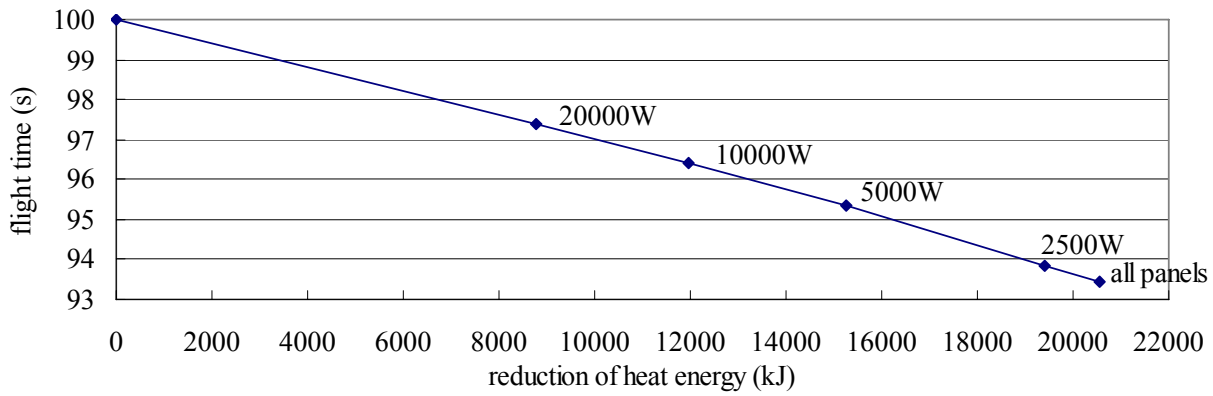

Figure 4-71. Flight time v.s. reduction of heat energy (kJ) at $M = 6$ and 35000m altitude with 0.001 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

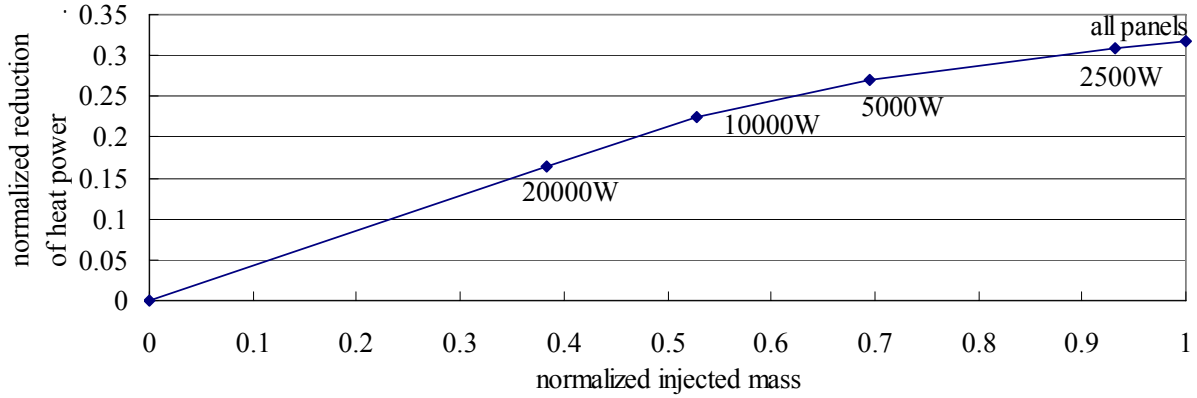

Figure 4-72. Normalized reduction of heat power v.s. normalized injected mass at $M = 6$ and 35000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

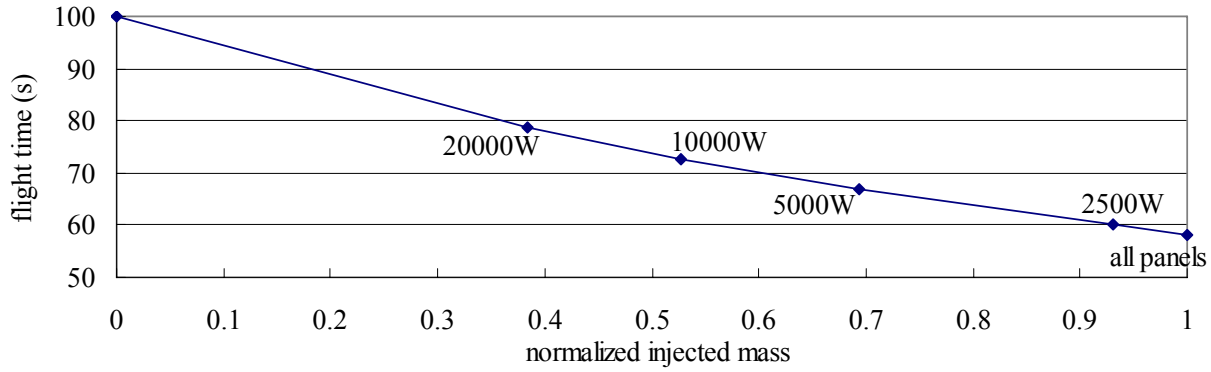

Figure 4-73. Flight time v.s. normalized injected mass at $M = 6$ and 35000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

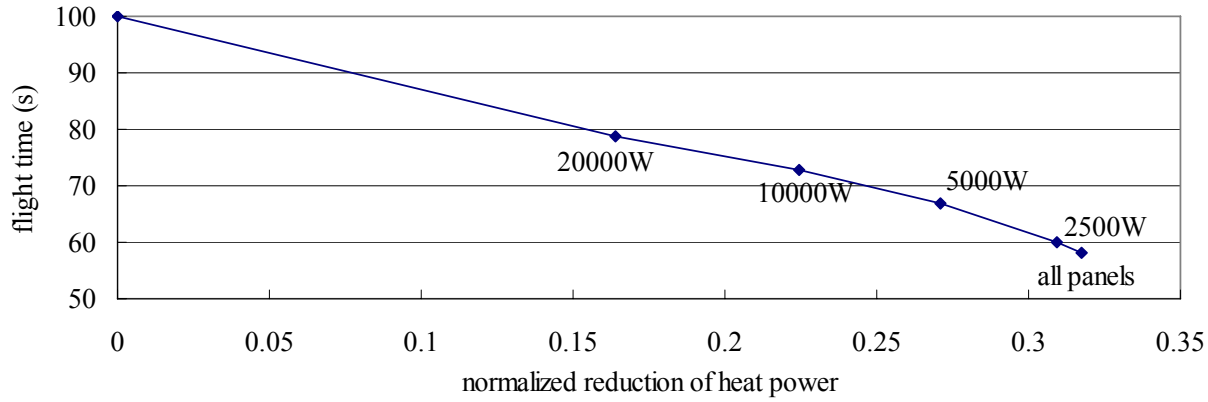

Figure 4-74. Flight time v.s. normalized reduction of heat power at $M = 6$ and 35000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

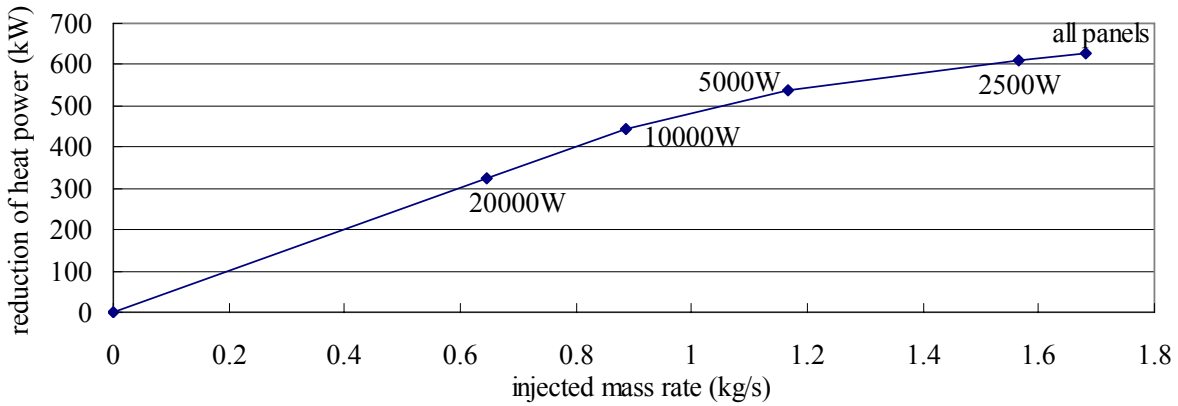

Figure 4-75. Reduction of heat power (kW) v.s. injected mass rate (kg/s) at $M = 6$ and 30000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

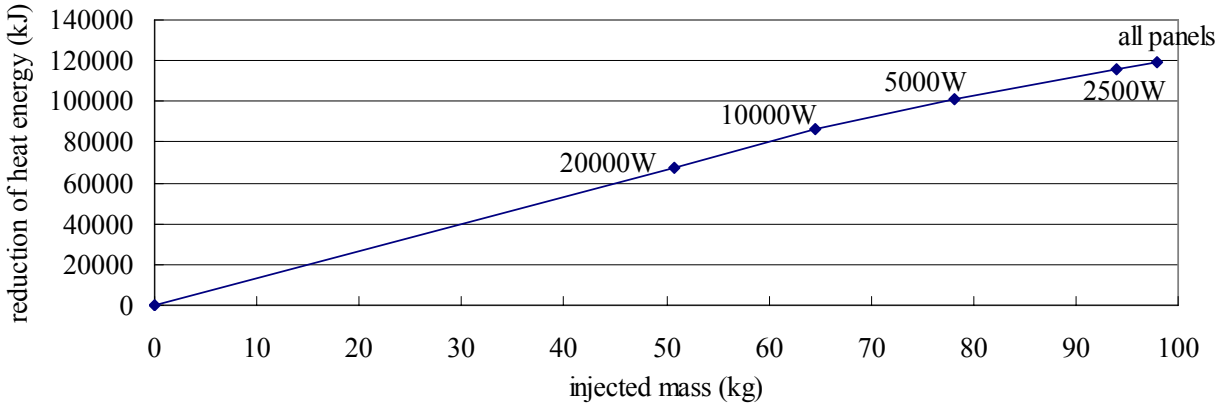

Figure 4-76. Reduction of heat energy (kJ) v.s. injected mass (kg) at $M = 6$ and 35000m altitude with 0.01 kg/m^2 s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

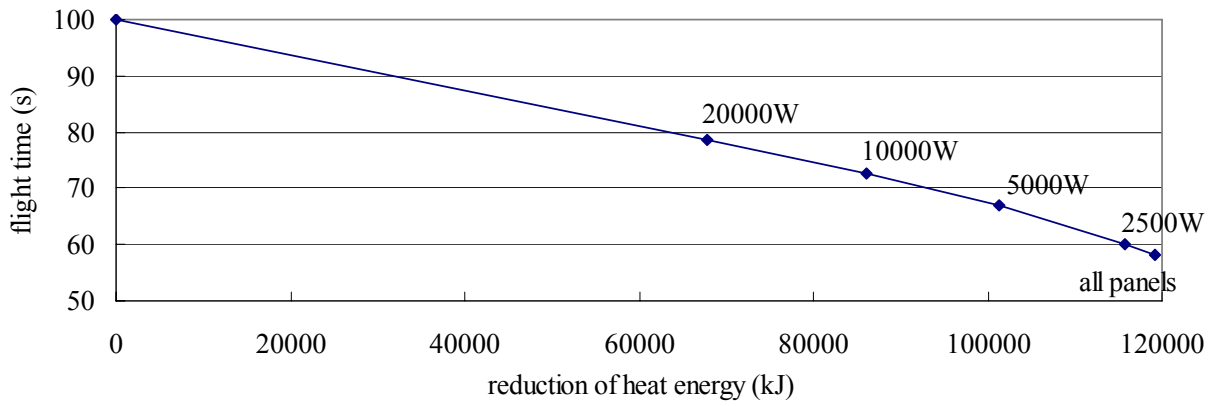

Figure 4-77. Flight time v.s. reduction of heat energy (kJ) at $M = 6$ and 35000m altitude with $0.01\ kg/m^2$ s of mass injection. The panels to be cooled are determined by $\dot{Q}_{c,w}$ (W).

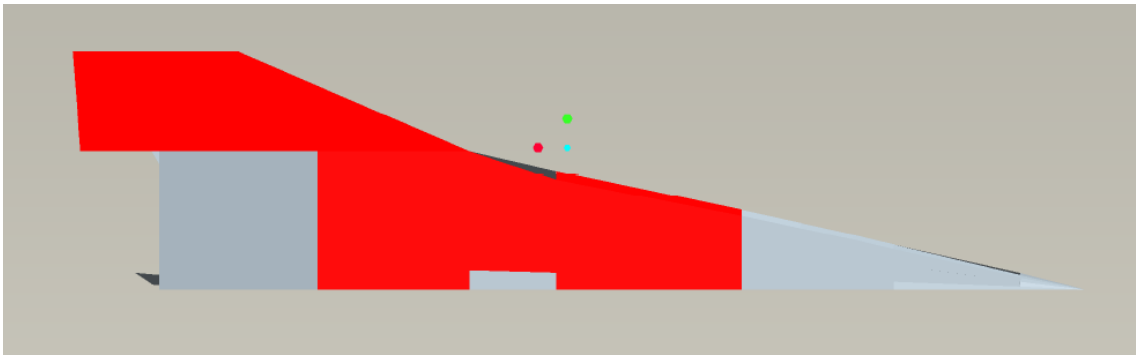

Figure 4-78. Bottom view of X-24C with $20000\,W$ of allowable $\dot{Q}_{c,w}$ at 35000m altitude.

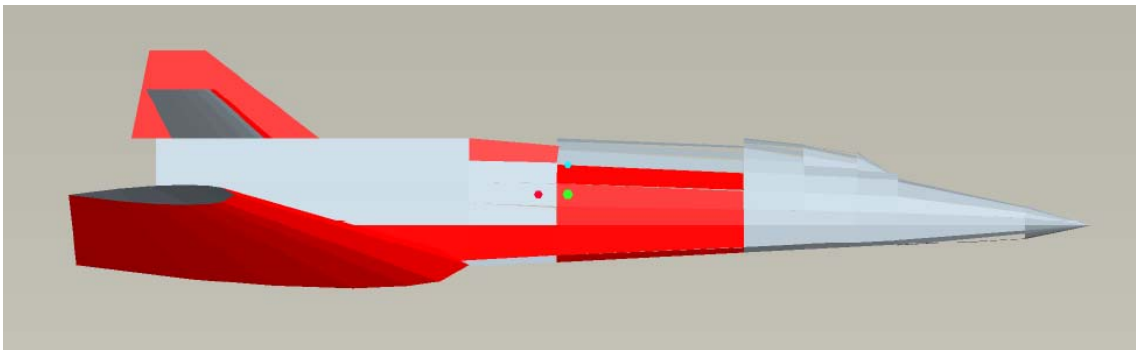

Figure 4-79. Side view of X-24C with $20000\,W$ of allowable $\dot{Q}_{c,w}$ at 35000m altitude.

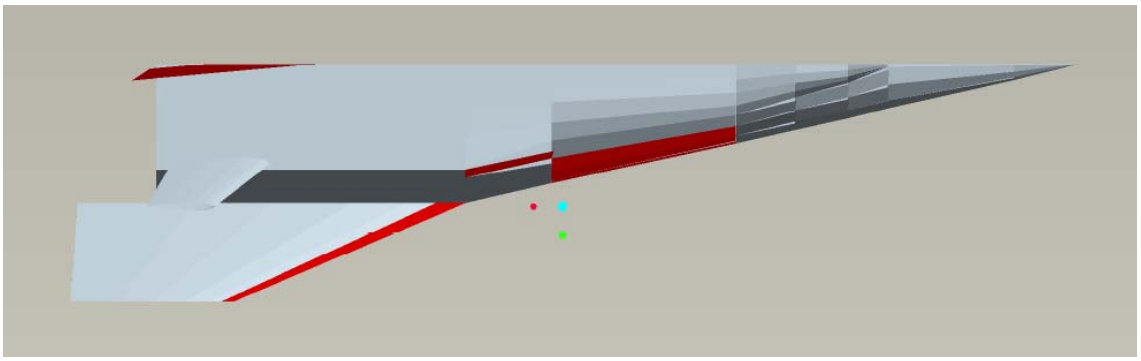

Figure 4-80. Top view of X-24C with $20000 W$ of allowable $\dot{Q}_{c,w}$ at $35000 m$ altitude.

CHAPTER 5
CONCLUSIONS

Conclusions of this Study

The local surface inclination method (Newtonian theory) was shown to reasonably predict the pressure around a flight vehicle like the X-24C. The flat plate reference enthalpy method for laminar and turbulent flow was utilized, along with a classical approximation of thermodynamic properties for considering high temperature effects, to calculate heat transfer over a hypersonic vehicle. This method provided good agreement with available heat transfer measurements. However, around a stagnation point or line, the flat plate reference enthalpy method gave extremely high values, much higher than experimental data. Therefore, around stagnation points or lines, the blunt body heat transfer method was used since it gave reasonable solutions. For surfaces on which the two methods give the same or close heat transfer, the flat plate reference enthalpy method was preferred. These simple methods are based on many approximations, but for preliminary design considerations, they are sufficiently accurate tools to compute the flow properties including aerodynamic forces, moments, and heating.

In order to reduce skin friction and heat transfer on a vehicle surface, we used an existing simple algebraic Couette flow solution since it has the virtue of fitting the experimental data. The effect of mass injection is relatively large for the lower α. Also, the more mass is injected, the less the effect of blowing grows. Therefore, even a small amount of injected mass is effective in improving flight performance (*L/D*) and thermal protection (less heating). Even though the amount of fuel is reduced in favor of carrying the coolant for surface injection, the fuel still provides the necessary thrust so that flight time is not reduced much, because blowing reduces the total drag. Reducing flight range is the penalty of mass injection, while reducing heat transfer is the advantage of mass injection. It has been shown that choosing the panels to be cooled by

taking the panels whose heat transfer, $q_{c,w}$ or $\dot{Q}_{c,w}$, are the highest gives the most advantage for the least penalty. This illustrates the conclusion that simply providing injection on all panels is not an efficient cooling method.

The X-24C vehicle has relatively high pressure forces because of the fuselage shape. Using mass injection on a more slender cruising vehicle which has a higher fraction of viscous drag than the X-24C will have less penalty (reduced flight range) or even zero penalty (constant or increasing flight range), in addition to reducing heat transfer.

Future Work

In this study, we considered the effects of mass injection only on the boundary layer over the plates through which injection occurs. However, in reality, the injected mass effects a change in flow structures downstream of the injected plates. The injected gas has a normal component of profile velocity, which makes the boundary layer profile less full and therefore the gradient of tangential velocity in normal direction becomes smaller than the boundary layer with zero injection. In general, the shear stress is linearly dependent on this gradient, and the skin friction would be the integrated value of local shear stress. This is the physics describing why injected gas reduces skin friction. The "less full" boundary layer profile cannot rapidly return to the original equilibrium profile just downstream of the injected plate. The injected gas, therefore, will reduce the skin friction on regions downstream of the injected plate.

Eventually, the "less full" boundary layer returns to the equilibrium profile at a certain point, after which the injected gas has no effect. Establishing the method that predicts how much skin friction and heat transfer are reduced is a logical extension to this study. The more accurate analysis of viscous effect reduction given by this method will make the results more reliable.

Furthermore these downstream effects will show that the current results are actually conservative and that even better performance should be possible.

APPENDIX A
MATLAB CODE TO COMPUTE FLIGHT PERFORMANCE OF X24C

```
%*************************************************************************
%code name: x24c0012
%written by: Yoshifumi Nozaki
%date: 3/20/2007
%This code computes the x-24c cruising vehicle flight performance
%center off fin & wing has NACA0012 airfoil.
%Inviscid and viscous hypersonic flow field can be solved by using
%the modified Newtonian method and Reynolds' analogy approximation
%This code works with other code
%*************************************************************************

%****** Flow Properties ***************************************************
% ro:       air density (kg/m3)
% Pinf:     atmospheric pressuer (N/m2)
% Rair:     gas constant of air (287J/kg-K)
% Tinf:     free stream temperature (K)
% cpinf:    specific heat constant pressure of free stream (J/g-K or kJ/kg-K)
% M:        Mach number
% gamma:    specific heat ratio
% Vinf:     free stream velocity (m/s)
% a:        speed of sound (m/s)
% qinf:     dynamic pressure (kN/m2)
%*************************************************************************

%***** Orbiter Characteristics ********************************************
% aoa:      angle of attack (deg.)
% Sref:     wing planform area (m2)
% chord:    chord length (m)
% b:        wing span (m)
% ep:       density ration across shock wave for hypersonic flow
% d1_el:    left elavon 1 deflection (deg.)
% d2_el:    left elavon 2 deflection (deg.)
% d1_er:    right elavon 1 deflection (deg.)
% d2_er:    right elavon 2 deflection (deg.)
% dr:       rudder deflection (deg.)
% ref:      reference point e.g. center of gravity
% refa:     reference point accounting angle of attack
%*************************************************************************

%***** Flight Performance *************************************************
% Lift:     lift (N)
% Drag:     drag (N)
% L_D:      L/D ratio (inviscid)
```

```
% L_D_vis   L/D ratio (viscous without cooling)
% L_D_cool  L/D ratio (viscous with cooling)
% rolling:  rolling moment (N-m)
% pitching: pitching moment (N-m)
% yawing:   yawing moment (N-m)
% CL:       lift coefficient
% CD:       drag coefficient
% C_m:      pitching moment coefficient
% C_l:      rolling moment coefficient
% C_n:      yawing moment coefficient
% CN:       normal force coefficient
% CA:       axial force coefficient
%*********************************************************************

%***** Computing Arrays ************************************************
% nod_o:  original point coordinate node (left side)
% nod:    point corrdinate at some angle of attack (left side)
% nodr_o:  original point coordinate node (right side)
% nodr:    point corrdinate at some angle of attack (right side)
% Pvec:    P vector of each panel
% Qvec:    Q vector of each panel
% Nvec:    N vector of each panel
% nunit:   n (N unit) vector of each panel
% Area:    area of each panel (m2)
% cp:      local pressure coefficient of each panel
% dF:      differential force (pressure force) acting on each panel
% dL:      differential lift force acting on each panel
% dD:      differential drag force acting on each panel
% cent:    panel center point (centroid)
% radi:    radius vector w.r.t. reference
% d_l:     differential rolling moment on each panel
% d_m:     differential rolling moment on each panel
% d_n:     differential rolling moment on each panel
% Pvecr:    P vector of each panel (right side)
% Qvecr:    Q vector of each panel (right side)
% Nvecr:    N vector of each panel (right side)
% nunitr:   n (N unit) vector of each panel (right side)
% Arear:    area of each panel (m2) (right side)
% cpr:      local pressure coefficient of each panel (right side)
% dFr:      differential force (pressure force) acting on each panel (right side)
% dLr:      differential lift force acting on each panel (right side)
% dDr:      differential drag force acting on each panel (right side)
% centr:    panel center point (centroid) (right side)
% radir:    radius vector w.r.t. reference (right side)
% d_lr:     differential rolling moment on each panel (right side)
% d_mr:     differential rolling moment on each panel (right side)
```

```
% d_nr:     differential rolling moment on each panel (right side)
%**************************************************************************

% input data
ro = 0.014283                % air density (kg/m3)  Z=36000m or 31500m
Pinf = 935.425354              % atmospheric pressuer (N/m2)
Rair = 287                   % gas constant of air (287J/kg-K)
Tinf = 228.15                % free stream temperature (K)
cpinf= 1006                  % specific heat constant pressure of free stream (J/kg-K)
M = 6.00                     % Mach number
gamma = 1.4                  % specific heat ratio
ref = [9.0932 0 0]           % reference point e.g. center of gravity

% flow properties computaion
a = (gamma*Rair*Tinf)^0.5        % speed of sound (m/s)
Vinf = M*a                   % free stream velocity (m/s)
qinf = 0.5*ro*Vinf^2             % dynamic pressure (N/m2)

% flight condition (input)
aoa =  6.0                   % angle of attack (deg.)
Sref = 57.20                 % vehicle planform area (m2)
chord = 14.7066                % chord length (m)
b = 7.37616                  % wing span (m)
ep = (gamma-1)/(gamma+1)         % density ration across shock wave for hypersonic flow

% reference point accounting angle of attack
refa = [((ref(1,1)^2+ref(1,2)^2)^0.5)*cos(atan(ref(1,2)/ref(1,1))-aoa*3.14159265359/180) 0
((ref(1,1)^2+ref(1,2)^2)^0.5)*sin(atan(ref(1,2)/ref(1,1))-aoa*3.14159265359/180)]

% original point coordinate system left side
nod_o =0.0254*[0 0 0;
   41 0 -8.3;
   41 2.0 -8.1;
   41 5.0 -7.5;
   41 7.5 -5.0;
   41 9.0 -3.0;
   41 10.3 0;
   41 10.3 5.0;
   41 8.5 8.0;
   41 5.0 11.0;
   41 1.0 12.3;
   41 0.75 12.4;
   41 0.50 12.4;
   41 0.25 12.4;
   41 0 12.4;
   120 0 -12.4;
```

120 5.0 -12.4;
120 15.5 -12.4;
120 25.0 -12.4;
120 27.9 -9.0;
120 27.9 0;
120 27.0 8.0;
120 23.0 18.0;
120 13.0 27.0;
120 2.0 29.0;
120 1.5 29.0;
120 1.0 29.0;
120 0.5 29.0;
120 0 29.0;
145 0 -13.7;
145 5.5 -13.7;
145 18.0 -13.7;
145 31.0 -13.7
145 34.1 -10.5;
145 34.1 0;
145 33.0 8.5;
145 29.0 19.0;
145 21.0 29.0;
145 12.0 34.0;
145 10.0 38.0;
145 8.0 42.0;
145 4.0 44.5;
145 0 45.5;
178 0 -15.4;
178 6.5 -15.4;
178 20.5 -15.4;
178 38.0 -15.4;
178 41.4 -12.0;
178 41.4 0;
178 40.0 9.5;
178 36.0 20.0;
178 29.0 31.0;
178 21.0 37.0;
178 17.0 43.5;
178 12.0 48.0;
178 6.0 49.5;
178 0 49.7;
215 0 -17.3;
215 7.5 -17.3
215 24.0 -17.3;
215 46.0 -17.3;
215 49.7 -14.0;

215 49.7 0;
215 48.0 10.0;
215 45.0 22.0;
215 39.0 33.0;
215 32.0 42.0;
215 26.0 48.0;
215 18.0 51.0;
215 10.0 53.0;
215 0 54.0;
331 0 -23.3;
331 10.5 -23.3;
331 34.0 -23.3;
331 69.0 -23.3;
331 74.5 -19.0;
331 74.5 0;
331 73.0 12.0;
331 67.0 26.0;
331 59.0 37.5;
331 48.0 47.0;
331 37.0 52.0;
331 24.0 54.0;
331 11.0 54.0;
331 0 54.0;
385 0 -26.1;
385 12.0 -26.1;
385 39.0 -26.1;
385 86.9 -26.1;
385 86.9 -22.0;
385 86.9 0;
385 82.0 14.0;
385 76.0 27.0;
385 71.0 40.0;
385 66.0 54.0;
385 46.0 54.0;
385 24.0 54.0;
385 11.0 54.0;
385 0 54.0;
480 0 -31.0;
480 14.5 -31.0;
480 46.0 -31.0;
480 86.9 -31.0;
480 86.9 -28.0;
480 86.9 0;
480 82.0 14.0;
480 76.0 27.0;
480 71.0 40.0;

480 66.0 54.0;
480 46.0 54.0;
480 24.0 54.0;
480 11.0 54.0;
480 0.0 54.0;
579 0 -4.1;
579 14.5 -4.1;
579 46.0 -4.1;
579 86.9 -4.1;
579 86.9 -2.0;
579 86.9 0;
579 82.0 14.0;
579 76.0 27.0;
579 71.0 40.0;
579 66.0 54.0;
579 46.0 54.0;
579 24.0 54.0;
579 11.0 54.0;
579 0 54.0;
478 0 54;
594 10.5 54;
548 0 108;
583 3.15 108;
507 62 54;
511.776418 64.631838 54;
516.584952 65.342358 54;
525.361858 65.720558 54;
537.945254 65.285752 54;
550.710744 64.241238 54;
569 62 54;
511.776418 59.368162 54;
516.584952 58.657642 54;
525.361858 58.279442 54;
537.945254 58.714248 54;
550.710744 59.758762 54;
542 89 84;
545.312677 90.825307 84;
548.647628 91.318087 84;
554.734837 91.580387 84;
563.462031 91.278828 84;
572.315516 90.554407 84;
585 89 84;
545.312677 87.174693 84;
548.647628 86.681913 84;
554.734837 86.419613 84;
563.462031 86.721172 84;

```
    572.315516 87.445593 84;
    385 86.9 -25;
    403.720477 86.9 -14.684893;
    422.566828 86.9 -11.900113;
    456.966637 86.9 -10.417813;
    506.285431 86.9 -12.121972;
    556.317916 86.9 -16.215793;
    628 86.9 -25;
    403.720477 86.9 -35.315107;
    422.566828 86.9 -38.099887;
    456.966637 86.9 -39.582187;
    506.285431 86.9 -37.878028;
    556.317916 86.9 -33.784207;
    530 149 19;
    537.935017 149 23.372247;
    545.923388 149 24.552627;
    560.504377 149 25.180927;
    581.409051 149 24.458588;
    602.616236 149 22.723347;
    633 149 19;
    537.935017 149 14.627753;
    545.923388 149 13.447373;
    560.504377 149 12.819073;
    581.409051 149 13.541412;
    602.616236 149 15.276653]

num_nod = 178+1 %the number of nod
num_panel = 142

for i=1:num_nod %To make y-component of each nod to the negative
   nod_o(i,2) = -1.0*nod_o(i,2)
end

% point corrdinate at some angle of attack left & right side
  nod(1,1)=0; nod(1,2)=0; nod(1,3)=0

 for i = 2:num_nod
  nod(i,1) = ((nod_o(i,1)^2+nod_o(i,3)^2)^0.5)*cos(atan(nod_o(i,3)/nod_o(i,1))-
aoa*3.14159265359/180)
  nod(i,2) = nod_o(i,2)
  nod(i,3) = ((nod_o(i,1)^2+nod_o(i,3)^2)^0.5)*sin(atan(nod_o(i,3)/nod_o(i,1))-
aoa*3.14159265359/180)
end

% Pvec:    P vector of each panel
% Pvec of body panels (panel 1-117)
```

```
for i = 1:13
   Pvec(i,1)=nod(i+1,1)-nod(1,1); Pvec(i,2)=nod(i+1,2)-nod(1,2); Pvec(i,3)=nod(i+1,3)-nod(1,3)
end
for i = 14:26
   Pvec(i,1)=nod(i+2,1)-nod(i-11,1); Pvec(i,2)=nod(i+2,2)-nod(i-11,2); Pvec(i,3)=nod(i+2,3)-
nod(i-11,3)
end
for i = 27:39
   Pvec(i,1)=nod(i+3,1)-nod(i-10,1); Pvec(i,2)=nod(i+3,2)-nod(i-10,2); Pvec(i,3)=nod(i+3,3)-
nod(i-10,3)
end
for i = 40:52
   Pvec(i,1)=nod(i+4,1)-nod(i-9,1); Pvec(i,2)=nod(i+4,2)-nod(i-9,2); Pvec(i,3)=nod(i+4,3)-nod(i-
9,3)
end
for i = 53:65
   Pvec(i,1)=nod(i+5,1)-nod(i-8,1); Pvec(i,2)=nod(i+5,2)-nod(i-8,2); Pvec(i,3)=nod(i+5,3)-nod(i-
8,3)
end
for i = 66:78
   Pvec(i,1)=nod(i+6,1)-nod(i-7,1); Pvec(i,2)=nod(i+6,2)-nod(i-7,2); Pvec(i,3)=nod(i+6,3)-nod(i-
7,3)
end
for i = 79:91
   Pvec(i,1)=nod(i+7,1)-nod(i-6,1); Pvec(i,2)=nod(i+7,2)-nod(i-6,2); Pvec(i,3)=nod(i+7,3)-nod(i-
6,3)
end
for i = 92:104
   Pvec(i,1)=nod(i+8,1)-nod(i-5,1); Pvec(i,2)=nod(i+8,2)-nod(i-5,2); Pvec(i,3)=nod(i+8,3)-nod(i-
5,3)
end
for i = 105:117
   Pvec(i,1)=nod(i+9,1)-nod(i-4,1); Pvec(i,2)=nod(i+9,2)-nod(i-4,2); Pvec(i,3)=nod(i+9,3)-nod(i-
4,3)
end
   %Pvec center fin panels(panel 118)
Pvec(118,1)=nod(129,1)-nod(130,1); Pvec(118,2)=nod(129,2)-nod(130,2);
Pvec(118,3)=nod(129,3)-nod(130,3)
   %Pvec off center fin panels(panel 119-130)
Pvec(119,1)=nod(145,1)-nod(132,1); Pvec(119,2)=nod(145,2)-nod(132,2);
Pvec(119,3)=nod(145,3)-nod(132,3)
for i= 120:123
   Pvec(i,1)=nod(i+26,1)-nod(i+13,1); Pvec(i,2)=nod(i+26,2)-nod(i+13,2);
Pvec(i,3)=nod(i+26,3)-nod(i+13,3)
end
```

```
Pvec(124,1)=nod(150,1)-nod(137,1); Pvec(124,2)=nod(150,2)-nod(137,2);
Pvec(124,3)=nod(150,3)-nod(137,3)
Pvec(125,1)=nod(144,1)-nod(139,1); Pvec(125,2)=nod(144,2)-nod(139,2);
Pvec(125,3)=nod(144,3)-nod(139,3)
for i= 126:129
   Pvec(i,1)=nod(i+25,1)-nod(i+14,1); Pvec(i,2)=nod(i+25,2)-nod(i+14,2);
Pvec(i,3)=nod(i+25,3)-nod(i+14,3)
end
Pvec(130,1)=nod(155,1)-nod(138,1);  Pvec(130,2)=nod(155,2)-nod(138,2);
Pvec(130,3)=nod(155,3)-nod(138,3)
   %Pvec wing panels(panel 131-142)
Pvec(131,1)=nod(168,1)-nod(157,1); Pvec(131,2)=nod(168,2)-nod(157,2);
Pvec(131,3)=nod(168,3)-nod(157,3)
for i= 132:135
   Pvec(i,1)=nod(i+37,1)-nod(i+26,1); Pvec(i,2)=nod(i+37,2)-nod(i+26,2);
Pvec(i,3)=nod(i+37,3)-nod(i+26,3)
end
Pvec(136,1)=nod(173,1)-nod(162,1); Pvec(136,2)=nod(173,2)-nod(162,2);
Pvec(136,3)=nod(173,3)-nod(162,3)
Pvec(137,1)=nod(175,1)-nod(156,1); Pvec(137,2)=nod(175,2)-nod(156,2);
Pvec(137,3)=nod(175,3)-nod(156,3)
for i= 138:141
   Pvec(i,1)=nod(i+38,1)-nod(i+25,1); Pvec(i,2)=nod(i+38,2)-nod(i+25,2);
Pvec(i,3)=nod(i+38,3)-nod(i+25,3)
end
Pvec(142,1)=nod(174,1)-nod(167,1);  Pvec(142,2)=nod(174,2)-nod(167,2);
Pvec(142,3)=nod(174,3)-nod(167,3)

% Qvec of body panels (panel 1-117)
for i = 1:13
   Qvec(i,1)=nod(i+2,1)-nod(1,1); Qvec(i,2)=nod(i+2,2)-nod(1,2); Qvec(i,3)=nod(i+2,3)-
nod(1,3)
end
for i = 14:26
   Qvec(i,1)=nod(i+3,1)-nod(i-12,1); Qvec(i,2)=nod(i+3,2)-nod(i-12,2); Qvec(i,3)=nod(i+3,3)-
nod(i-12,3)
end
for i = 27:39
   Qvec(i,1)=nod(i+4,1)-nod(i-11,1); Qvec(i,2)=nod(i+4,2)-nod(i-11,2); Qvec(i,3)=nod(i+4,3)-
nod(i-11,3)
end
for i = 40:52
   Qvec(i,1)=nod(i+5,1)-nod(i-10,1); Qvec(i,2)=nod(i+5,2)-nod(i-10,2); Qvec(i,3)=nod(i+5,3)-
nod(i-10,3)
end
for i = 53:65
```

```
   Qvec(i,1)=nod(i+6,1)-nod(i-9,1); Qvec(i,2)=nod(i+6,2)-nod(i-9,2); Qvec(i,3)=nod(i+6,3)-
nod(i-9,3)
end
for i = 66:78
   Qvec(i,1)=nod(i+7,1)-nod(i-8,1); Qvec(i,2)=nod(i+7,2)-nod(i-8,2); Qvec(i,3)=nod(i+7,3)-
nod(i-8,3)
end
for i = 79:91
   Qvec(i,1)=nod(i+8,1)-nod(i-7,1); Qvec(i,2)=nod(i+8,2)-nod(i-7,2); Qvec(i,3)=nod(i+8,3)-
nod(i-7,3)
end
for i = 92:104
   Qvec(i,1)=nod(i+9,1)-nod(i-6,1); Qvec(i,2)=nod(i+9,2)-nod(i-6,2); Qvec(i,3)=nod(i+9,3)-
nod(i-6,3)
end
for i = 105:117
   Qvec(i,1)=nod(i+10,1)-nod(i-5,1); Qvec(i,2)=nod(i+10,2)-nod(i-5,2); Qvec(i,3)=nod(i+10,3)-
nod(i-5,3)
end
   %Qvec center fin panels(panel 118)
Qvec(118,1)=nod(131,1)-nod(128,1); Qvec(118,2)=nod(131,2)-nod(128,2);
Qvec(118,3)=nod(131,3)-nod(128,3)

 %Pvec off center fin panels(panel 119-130)
Qvec(119,1)=nod(144,1)-nod(133,1); Qvec(119,2)=nod(144,2)-nod(133,2);
Qvec(119,3)=nod(144,3)-nod(133,3)
for i= 120:123
   Qvec(i,1)=nod(i+25,1)-nod(i+14,1); Qvec(i,2)=nod(i+25,2)-nod(i+14,2);
Qvec(i,3)=nod(i+25,3)-nod(i+14,3)
end
Qvec(124,1)=nod(149,1)-nod(138,1);  Qvec(124,2)=nod(149,2)-nod(138,2);
Qvec(124,3)=nod(149,3)-nod(138,3)

Qvec(125,1)=nod(151,1)-nod(132,1); Qvec(125,2)=nod(151,2)-nod(132,2);
Qvec(125,3)=nod(151,3)-nod(132,3)
for i= 126:129
   Qvec(i,1)=nod(i+26,1)-nod(i+13,1); Qvec(i,2)=nod(i+26,2)-nod(i+13,2);
Qvec(i,3)=nod(i+26,3)-nod(i+13,3)
end
Qvec(130,1)=nod(150,1)-nod(143,1); Qvec(130,2)=nod(150,2)-nod(143,2);
Qvec(130,3)=nod(150,3)-nod(143,3)

   %Pvec wing panels(panel 131-142)
Qvec(131,1)=nod(169,1)-nod(156,1); Qvec(131,2)=nod(169,2)-nod(156,2);
Qvec(131,3)=nod(169,3)-nod(156,3)
for i= 132:135
```

```
   Qvec(i,1)=nod(i+38,1)-nod(i+25,1); Qvec(i,2)=nod(i+38,2)-nod(i+25,2);
Qvec(i,3)=nod(i+38,3)-nod(i+25,3)
end
Qvec(136,1)=nod(174,1)-nod(161,1); Qvec(136,2)=nod(174,2)-nod(161,2);
Qvec(136,3)=nod(174,3)-nod(161,3)

Qvec(137,1)=nod(168,1)-nod(163,1); Qvec(137,2)=nod(168,2)-nod(163,2);
Qvec(137,3)=nod(168,3)-nod(163,3)
for i= 138:141
   Qvec(i,1)=nod(i+37,1)-nod(i+26,1); Qvec(i,2)=nod(i+37,2)-nod(i+26,2);
Qvec(i,3)=nod(i+37,3)-nod(i+26,3)
end
Qvec(142,1)=nod(179,1)-nod(162,1);  Qvec(142,2)=nod(179,2)-nod(162,2);
Qvec(142,3)=nod(179,3)-nod(162,3)

% Nvec:    N vector of each panel

for i=1: num_panel
   Nvec(i,1) = Pvec(i,2)*Qvec(i,3)-Pvec(i,3)*Qvec(i,2)
   Nvec(i,2) = Pvec(i,3)*Qvec(i,1)-Pvec(i,1)*Qvec(i,3)
   Nvec(i,3) = Pvec(i,1)*Qvec(i,2)-Pvec(i,2)*Qvec(i,1)
end

% nunit:   n (N unit) vector of each panel
for i=1:num_panel
   nunit(i,1)=Nvec(i,1)/(Nvec(i,1)^2+Nvec(i,2)^2+Nvec(i,3)^2)^0.5
   nunit(i,2)=Nvec(i,2)/(Nvec(i,1)^2+Nvec(i,2)^2+Nvec(i,3)^2)^0.5
   nunit(i,3)=Nvec(i,3)/(Nvec(i,1)^2+Nvec(i,2)^2+Nvec(i,3)^2)^0.5
end

% Area:    area of each panel (m2)
for i=1:num_panel
   Area(i,1) = 0.5*(Nvec(i,1)^2+Nvec(i,2)^2+Nvec(i,3)^2)^0.5
end

% cp:     local pressure coefficient of each panel
for i=1:num_panel
   if nunit(i,1)<0
   cp(i,1) = (2.0-0.0)*nunit(i,1)^2
   else
   cp(i,1) = 0
   end
end

% dF:      differential force (pressure force) acting on each panel
for i=1:num_panel
```

```
   dF(i,1) = -cp(i,1)*qinf*Area(i,1)*nunit(i,1)
   dF(i,2) = -cp(i,1)*qinf*Area(i,1)*nunit(i,2)
   dF(i,3) = -cp(i,1)*qinf*Area(i,1)*nunit(i,3)
end

% dL:      differential lift force acting on each panel
% dD:      differential drag force acting on each panel
% dLr:     differential lift force acting on each panel (right side)
% dDr:     differential drag force acting on each panel (right side)
for i=1:num_panel
   dL(i,1) = dF(i,3)
   dD(i,1) = dF(i,1)
end

% cent:    panel center point (centroid)
% centr:   panel center point (centroid) (right side)

for i= 1:13
   cent(i,1)=(nod(i+1,1)+nod(i+2,1))/3; cent(i,2)=(nod(i+1,2)+nod(i+2,2))/3;
cent(i,3)=(nod(i+1,3)+nod(i+2,3))/3
end
for i= 14:26
   cent(i,1)=(nod(i+2,1)+nod(i+3,1)+nod(i-12,1)+nod(i-11,1))/4
   cent(i,2)=(nod(i+2,2)+nod(i+3,2)+nod(i-12,2)+nod(i-11,2))/4
   cent(i,3)=(nod(i+2,3)+nod(i+3,3)+nod(i-12,3)+nod(i-11,3))/4
end
for i= 27:39
   cent(i,1)=(nod(i+3,1)+nod(i+4,1)+nod(i-11,1)+nod(i-10,1))/4
   cent(i,2)=(nod(i+3,2)+nod(i+4,2)+nod(i-11,2)+nod(i-10,2))/4
   cent(i,3)=(nod(i+3,3)+nod(i+4,3)+nod(i-11,3)+nod(i-10,3))/4
end
for i= 40:52
   cent(i,1)=(nod(i+4,1)+nod(i+5,1)+nod(i-10,1)+nod(i-9,1))/4
   cent(i,2)=(nod(i+4,2)+nod(i+5,2)+nod(i-10,2)+nod(i-9,2))/4
   cent(i,3)=(nod(i+4,3)+nod(i+5,3)+nod(i-10,3)+nod(i-9,3))/4
end
for i= 53:65
   cent(i,1)=(nod(i+5,1)+nod(i+6,1)+nod(i-9,1)+nod(i-8,1))/4
   cent(i,2)=(nod(i+5,2)+nod(i+6,2)+nod(i-9,2)+nod(i-8,2))/4
   cent(i,3)=(nod(i+5,3)+nod(i+6,3)+nod(i-9,3)+nod(i-8,3))/4
end
for i= 66:78
   cent(i,1)=(nod(i+6,1)+nod(i+7,1)+nod(i-8,1)+nod(i-7,1))/4
   cent(i,2)=(nod(i+6,2)+nod(i+7,2)+nod(i-8,2)+nod(i-7,2))/4
   cent(i,3)=(nod(i+6,3)+nod(i+7,3)+nod(i-8,3)+nod(i-7,3))/4
end
```

```
for i= 79:91
   cent(i,1)=(nod(i+7,1)+nod(i+8,1)+nod(i-7,1)+nod(i-6,1))/4
   cent(i,2)=(nod(i+7,2)+nod(i+8,2)+nod(i-7,2)+nod(i-6,2))/4
   cent(i,3)=(nod(i+7,3)+nod(i+8,3)+nod(i-7,3)+nod(i-6,3))/4
end
for i= 92:104
   cent(i,1)=(nod(i+8,1)+nod(i+9,1)+nod(i-6,1)+nod(i-5,1))/4
   cent(i,2)=(nod(i+8,2)+nod(i+9,2)+nod(i-6,2)+nod(i-5,2))/4
   cent(i,3)=(nod(i+8,3)+nod(i+9,3)+nod(i-6,3)+nod(i-5,3))/4
end
for i= 105:117
   cent(i,1)=(nod(i+9,1)+nod(i+10,1)+nod(i-5,1)+nod(i-4,1))/4
   cent(i,2)=(nod(i+9,2)+nod(i+10,2)+nod(i-5,2)+nod(i-4,2))/4
   cent(i,3)=(nod(i+9,3)+nod(i+10,3)+nod(i-5,3)+nod(i-4,3))/4
end
%cent center fin panels(panel 118)
for j=1:3
   cent(118,j)=(nod(128,j)+nod(129,j)+nod(130,j)+nod(131,j))/4
end
%cent off center fin panels(panel 119-130)
cent(119,1)=(nod(132,1)+nod(133,1)+nod(144,1)+nod(145,1))/4
cent(119,2)=(nod(132,2)+nod(133,2)+nod(144,2)+nod(145,2))/4
cent(119,3)=(nod(132,3)+nod(133,3)+nod(144,3)+nod(145,3))/4
for i= 120:123
   cent(i,1)=(nod(i+13,1)+nod(i+14,1)+nod(i+25,1)+nod(i+26,1))/4
   cent(i,2)=(nod(i+13,2)+nod(i+14,2)+nod(i+25,2)+nod(i+26,2))/4
   cent(i,3)=(nod(i+13,3)+nod(i+14,3)+nod(i+25,3)+nod(i+26,3))/4
end
cent(124,1)=(nod(137,1)+nod(138,1)+nod(149,1)+nod(150,1))/4
cent(124,2)=(nod(137,2)+nod(138,2)+nod(149,2)+nod(150,2))/4
cent(124,3)=(nod(137,3)+nod(138,3)+nod(149,3)+nod(150,3))/4
cent(125,1)=(nod(132,1)+nod(139,1)+nod(144,1)+nod(151,1))/4
cent(125,2)=(nod(132,2)+nod(139,2)+nod(144,2)+nod(151,2))/4
cent(125,3)=(nod(132,3)+nod(139,3)+nod(144,3)+nod(151,3))/4
for i= 126:129
   cent(i,1)=(nod(i+25,1)+nod(i+26,1)+nod(i+13,1)+nod(i+14,1))/4
   cent(i,2)=(nod(i+25,2)+nod(i+26,2)+nod(i+13,2)+nod(i+14,2))/4
   cent(i,3)=(nod(i+25,3)+nod(i+26,3)+nod(i+13,3)+nod(i+14,3))/4
end
cent(130,1)=(nod(143,1)+nod(138,1)+nod(155,1)+nod(150,1))/4
cent(130,2)=(nod(143,2)+nod(138,2)+nod(155,2)+nod(150,2))/4
cent(130,3)=(nod(143,3)+nod(138,3)+nod(155,3)+nod(150,3))/4

%cent wing panels(panel 131-142)
cent(131,1)=(nod(156,1)+nod(157,1)+nod(168,1)+nod(169,1))/4
cent(131,2)=(nod(156,2)+nod(157,2)+nod(168,2)+nod(169,2))/4
```

```
cent(131,3)=(nod(156,3)+nod(157,3)+nod(168,3)+nod(169,3))/4
for i= 132:135
   cent(i,1)=(nod(i+25,1)+nod(i+26,1)+nod(i+37,1)+nod(i+38,1))/4
   cent(i,2)=(nod(i+25,2)+nod(i+26,2)+nod(i+37,2)+nod(i+38,2))/4
   cent(i,3)=(nod(i+25,3)+nod(i+26,3)+nod(i+37,3)+nod(i+38,3))/4
end
cent(136,1)=(nod(173,1)+nod(174,1)+nod(161,1)+nod(162,1))/4
cent(136,2)=(nod(173,2)+nod(174,2)+nod(161,2)+nod(162,2))/4
cent(136,3)=(nod(173,3)+nod(174,3)+nod(161,3)+nod(162,3))/4
cent(137,1)=(nod(156,1)+nod(163,1)+nod(168,1)+nod(169,1))/4
cent(137,2)=(nod(156,2)+nod(163,2)+nod(168,2)+nod(169,2))/4
cent(137,3)=(nod(156,3)+nod(163,3)+nod(168,3)+nod(169,3))/4
for i= 138:141
   cent(i,1)=(nod(i+25,1)+nod(i+26,1)+nod(i+37,1)+nod(i+38,1))/4
   cent(i,2)=(nod(i+25,2)+nod(i+26,2)+nod(i+37,2)+nod(i+38,2))/4
   cent(i,3)=(nod(i+25,3)+nod(i+26,3)+nod(i+37,3)+nod(i+38,3))/4
end
cent(142,1)=(nod(167,1)+nod(162,1)+nod(179,1)+nod(174,1))/4
cent(142,2)=(nod(167,2)+nod(162,2)+nod(179,2)+nod(174,2))/4
cent(142,3)=(nod(167,3)+nod(162,3)+nod(179,3)+nod(174,3))/4

%Arc length (only body)
arc =0
arc(1,1)=-cent(1,2)
for i= 2:13
   arc(i,1)=arc(i-1)+((cent(i,2)-cent(i-1,2))^2+(cent(i,3)-cent(i-1,3))^2)^0.5
end
totalarc1=arc(13,1)-cent(13,2)
arc(14,1)=-cent(14,2)
for i= 15:26
   arc(i,1)=arc(i-1)+((cent(i,2)-cent(i-1,2))^2+(cent(i,3)-cent(i-1,3))^2)^0.5
end
totalarc2=arc(26,1)-cent(26,2)
arc(27,1)=-cent(27,2)
for i= 28:39
   arc(i,1)=arc(i-1)+((cent(i,2)-cent(i-1,2))^2+(cent(i,3)-cent(i-1,3))^2)^0.5
end
totalarc3=arc(39,1)-cent(39,2)
arc(40,1)=-cent(40,2)
for i= 41:52
   arc(i,1)=arc(i-1)+((cent(i,2)-cent(i-1,2))^2+(cent(i,3)-cent(i-1,3))^2)^0.5
end
totalarc4=arc(52,1)-cent(52,2)
arc(53,1)=-cent(53,2)
for i= 54:65
   arc(i,1)=arc(i-1)+((cent(i,2)-cent(i-1,2))^2+(cent(i,3)-cent(i-1,3))^2)^0.5
```

```
end
totalarc5=arc(65,1)-cent(65,2)
arc(66,1)=-cent(66,2)
for i= 67:78
   arc(i,1)=arc(i-1)+((cent(i,2)-cent(i-1,2))^2+(cent(i,3)-cent(i-1,3))^2)^0.5
end
totalarc6=arc(78,1)-cent(78,2)
arc(79,1)=-cent(79,2)
for i= 80:91
   arc(i,1)=arc(i-1)+((cent(i,2)-cent(i-1,2))^2+(cent(i,3)-cent(i-1,3))^2)^0.5
end
totalarc7=arc(91,1)-cent(91,2)
arc(92,1)=-cent(92,2)
for i= 93:104
   arc(i,1)=arc(i-1)+((cent(i,2)-cent(i-1,2))^2+(cent(i,3)-cent(i-1,3))^2)^0.5
end
totalarc8=arc(104,1)-cent(104,2)
arc(105,1)=-cent(105,2)
for i= 106:117
   arc(i,1)=arc(i-1)+((cent(i,2)-cent(i-1,2))^2+(cent(i,3)-cent(i-1,3))^2)^0.5
end
totalarc9=arc(117,1)-cent(117,2)

% radi:    radius vector w.r.t. reference
for i=1:num_panel
   radi(i,1) = (cent(i,1)*cos(aoa*3.14159265359/180)-cent(i,3)*sin(aoa*3.14159265359/180))-
ref(1,1)
   radi(i,2) = cent(i,2)-ref(1,2)
   radi(i,3) = (cent(i,3)*cos(aoa*3.14159265359/180)+cent(i,1)*sin(aoa*3.14159265359/180))-
ref(1,3)
end

for i=1:num_panel
   d_m(i,1) = -
(radi(i,1)*(dF(i,3)*cos(aoa*3.14159265359/180)+dF(i,1)*sin(aoa*3.14159265359/180))-
radi(i,3)*(dF(i,1)*cos(aoa*3.14159265359/180)-dF(i,3)*sin(aoa*3.14159265359/180)))
end
pitching = sum(d_m)*2
C_m = pitching/qinf/Sref/chord

%****** Viscous consideration ****************************************
%From here viscous effect is accounted into flow field
%In order to find the skin friction and heat transfer, local Reynolds'
%analogy is used. (Approximate analysis)
%u_e     : tangential component of velocity on each panel (m/s)
%u_er    : tangential component of velocity on each panel (m/s) right side
```

```
%u_eunit : unit vector of tangential component of velocity on each panel
%u_eunitr: unit vector of tangential component of velocity on each panel right side
%PMang   : Prandtl-Meyer expansion angle (rad)
%psi     : deflection angle (rad)
%h_e     : enthalpy at edge of boundary layer (this is h_known in
%          thermodynamic property code (ThermPropAir.m)
%          unit: J/kg, change unit to J/g so that h_e can be used in
%          ThermPropAir.m
%h_er    : enthalpy at edge of boundary layer right side
%s       : the distance along the surface of the vehicle measured from
%        relevant stagnationpoint for body (panel 1 - 40 & 75)
%        for wing section 1 (panel 43 - 53)
%        for wing section 2 (panel 54 - 64)
%        for wing section 2 (panel 65 - 74)
%        for rudder (panel 41 & 42)
%l_s     :the distance between two centroids
%sr      : the distance along the surface of the vehicle measured from
%        relevant stagnationpoint for right body (panel 1 - 40 & 75)
%        for right wing section 1 (panel 43 - 53)
%        for right wing section 2 (panel 54 - 64)
%        for right wing section 2 (panel 65 - 74)
%        for right rudder (panel 41 & 42)
%l_s r   :the distance between two centroids for right side
%*************************************************************************

%tangential component of velocity on each panel
for i=1:num_panel

       u_e(i,1) = Vinf - (Vinf*nunit(i,1))*nunit(i,1)
       u_e(i,2) = 0 - (Vinf*nunit(i,1))*nunit(i,2)
       u_e(i,3) = 0 - (Vinf*nunit(i,1))*nunit(i,3)
end
u_e(num_panel+1,1)=Vinf; u_e(num_panel+1,2)=0; u_e(num_panel+1,3)=0
for i=1:num_panel+1
       %u_eunit : unit vector of tangential component of velocity on each panel
       u_eunit(i,1) = u_e(i,1)/(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5
       u_eunit(i,2) = u_e(i,2)/(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5
       u_eunit(i,3) = u_e(i,3)/(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5
end

%enthalpy at edge of boundary layer
%computed by using energy equation
h_inf = Tinf*1006              %cp = 1006 J/kg-K at T = 250K
for i=1:num_panel
   h_e(i,1) = 0.5*Vinf^2 + h_inf - 0.5*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2) %unit: J/kg or m2/s2
end
```

```
%s   : the distance along the surface of the vehicle measured from
%      relevant stagnationpoint for body (panel 1 - 40 & 75)
%      for wing section 1 (panel 43 - 53)
%      for wing section 2 (panel 54 - 64)
%      for wing section 2 (panel 65 - 74)
%      for rudder (panel 41 & 42)
%l_s : the distance between two centroids

nose_vector = -1.0*cos(aoa*3.14159265359/180)

for i=1:117
   for j=i:117
   l_s(i,j)=((cent(i,1)-cent(j,1))^2 + (cent(i,2)-cent(j,2))^2 + (cent(i,3)-cent(j,3))^2)^0.5
   end
end
for i=119:130
   for j=i:130
   l_s(i,j)=((cent(i,1)-cent(j,1))^2 + (cent(i,2)-cent(j,2))^2 + (cent(i,3)-cent(j,3))^2)^0.5
   end
end
for i=131:142
   for j=i:142
   l_s(i,j)=((cent(i,1)-cent(j,1))^2 + (cent(i,2)-cent(j,2))^2 + (cent(i,3)-cent(j,3))^2)^0.5
   end
end

% for body  (panel 1 - 117)
   for i= 1:13
      s(i,1) = (cent(i,1)^2+cent(i,2)^2+cent(i,3)^2)^0.5 - 0.205 + 0.10110667
   end
   for i= 14:117
      s(i,1) = s(i-13,1) + l_s(i-13,i)
   end
   s(118,1) = ((cent(118,1)-(nod(128,1)+nod(130,1))/2)^2+(cent(118,3)-
(nod(128,3)+nod(130,3))/2)^2)^0.5
   s(119,1) = ((cent(119,1)-(nod(132,1)+nod(144,1))/2)^2+(cent(119,3)-
(nod(132,3)+nod(144,3))/2)^2)^0.5
   s(125,1) = ((cent(125,1)-(nod(132,1)+nod(144,1))/2)^2+(cent(125,3)-
(nod(132,3)+nod(144,3))/2)^2)^0.5
   for i= 120:124
      s(i,1) = s(i-1,1) + l_s(i-1,i)
   end
   for i= 126:130
      s(i,1) = s(i-1,1) + l_s(i-1,i)
   end
```

```
   s(131,1) = ((cent(131,1)-(nod(156,1)+nod(168,1))/2)^2+(cent(131,3)-
(nod(156,3)+nod(168,3))/2)^2)^0.5
   s(137,1) = ((cent(137,1)-(nod(156,1)+nod(168,1))/2)^2+(cent(137,3)-
(nod(156,3)+nod(168,3))/2)^2)^0.5
   for i= 132:136
      s(i,1) = s(i-1,1) + l_s(i-1,i)
   end
   for i= 138:142
      s(i,1) = s(i-1,1) + l_s(i-1,i)
   end

   s(num_panel+1,1)=0
%***** Thermodynamic properties *********************************************
%Now u_e, h_e, and s are known.
%Pin = input P (atm)
%h_known = input h (J/g or kJ/kg)
%T_out = input T (K)
%*********************************************
%T_e is found by temperature (Pin, h_known)
%ro  is found by density (Pin, Tout)
%mu  is found by viscosity (Pin, Tout)
%Pr  is found by prandtl (Pin, Tout)
%*****************************************************************************

%***** Reynolds number and skin friction
*********************************************
%The flat plate reference enthalpy method is used in each panel, using
%Reynonds' analogy with heat transfer to calculate slin friction for both
%laminar and turbulent cases
%
%Res      : Reynolds number based on the local tangential velocity (u_e),
%          temperature (T_e), and distance (s) from the stagnation point
%Res      = ro_e * u_e * s / mu_e
%
%cf       : local skin friction coefficient
%cf       = 2AA/(Pr_e^(1/3)) * (ro_ref/ro_e)^sa * (mu_ref/mu_e)^sb * Res^(sc-1) * (3^sj)^0.5
%AA  = 0.332*Pr^(1/3) -------------laminar
%      0.0296*Pr^(1/3) -------------turbulent
%sa  = 0.5 ------------------------laminar
%      0.8 ------------------------turbulent
%sb  = 0.5 ------------------------laminar
%      0.2 ------------------------turbulent
%sc  = 0.5 ------------------------laminar
%      0.8 ------------------------turbulent
%sj  = 0   ------------------------flat plate
%      1   ------------------------axisymmetric
```

```
%
%Pin      : the pressure computed by inviscid analysis for each panel (input for function)
%h_known  : enthalpy at edge of boundary layer for each panel (input for function)
%Tout     : temperature at edge of boundary layer for each panel (input for function)
%h_e      : enthalpy at edge of boundary layer for each panel
%T_e      : temperature at edge of boundary layer for each panel
%ro_e     : air density at edge of boundary layer for each panel
%mu_e     : viscosity at edge of boundary layer for each panel
%Pr_e     : Prandtl Number at edge of boundary layer for each panel
%h_ref    : enthalpy at edge of boundary layer for each panel
%T_ref    : reference temperature at edge of boundary layer for each panel
%ro_ref   : reference air density at edge of boundary layer for each panel
%mu_ref   : reference viscosity at edge of boundary layer for each panel
%Pr_ref   : reference Prandtl Number at edge of boundary layer for each panel
%h_w      : enthalpy at wall
%h_aw     : adiabatic wall enthalpy
%*********************************************************************

%****** Flow Properties (again) ***********************************
% ro:       air density (kg/m3)
% Pinf:     atmospheric pressuer (N/m2)
% Rair:     gas constant of air (287J/kg-K)
% Tinf:     free stream temperature (K)
% cpinf:    specific heat constant pressure of free stream (J/g-K or kJ/kg-K)
% M:        Mach number
% gamma:    specific heat ratio
% Vinf:     free stream velocity (m/s)
% a:        speed of sound (m/s)
% qinf:     dynamic pressure (kN/m2)
%*********************************************************************


% Reynolds Number and skin friction for left side
T_e(num_panel+1,1)=Tinf
P_e(num_panel+1,1)=Pinf/101325
cp(num_panel+1,1)=0.0

for i=1:num_panel+1
   Pin = (cp(i,1)*qinf + Pinf)/101325   % must be in unit of atm
   P_e(i,1) = Pin
   h_known = (0.5*Vinf^2 + Tinf*cpinf - 0.5*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2))/1000  %unit
of J/g

   h_e(i,1) = h_known
   Tout = temperature (Pin, h_known)
   if i == num_panel+1
```

```
      Tout=Tinf
   end
   T_e(i,1) = Tout

   ro_out   = density (Pin, Tout)
   ro_e(i,1) = ro_out

   mu_out   = viscosity (Pin, Tout)
   mu_e(i,1) = mu_out

   Pr_out   = prandtl (Pin, Tout)
   Pr_e(i,1)= Pr_out

end

%tangential component of velocity on "shadow" panel is found by P-M expansion

for i= 1:13
   jj(i,1) = num_panel+1
end
for i= 14:117
   jj(i,1) = i-13
end
   jj(118,1) = 104
   jj(119,1) = num_panel+1
   jj(125,1) = num_panel+1
for i= 120:124
   jj(i,1) = i-1
end
for i= 126:130
   jj(i,1) = i-1
end
   jj(131,1) = num_panel+1
   jj(137,1) = num_panel+1
for i= 132:136
   jj(i,1) = i-1
end
for i= 138:142
   jj(i,1) = i-1
end

for i = 1:num_panel
   if nunit(i,1) > 0
      j=jj(i,1)
      M1 = ((u_e(j,1)^2+u_e(j,2)^2+u_e(j,3)^2)/(gamma*Rair*T_e(j,1)))^0.5 %M of j panel
      if M1<1
```

```
        M2 =M1
      else
        M2 = shadowM (i,jj,u_eunit, T_e, u_e, gamma, ep, Rair) %M of i panel
        T_e(i,1) =T_e(j,1)*(1+((gamma-1)*M1^2)/2)/(1+((gamma-1)*M2^2)/2)
        P_e(i,1) =P_e(j,1)*(T_e(i,1)/T_e(j,1))^(gamma/(gamma-1))
        ro_e(i,1) = ro_e(j,1)*(T_e(j,1)/T_e(i,1))^(1/(gamma-1))
        Pin = P_e(i,1); Tout=T_e(i,1)
        h_e(i,1) =enthalpy (Pin, Tout)
        u_e(i,1) = (M2*(gamma*T_e(i,1)*Rair)^0.5)*u_eunit(i,1)
        u_e(i,2) = (M2*(gamma*T_e(i,1)*Rair)^0.5)*u_eunit(i,2)
        u_e(i,3) = (M2*(gamma*T_e(i,1)*Rair)^0.5)*u_eunit(i,3)
      end
   end
end

for i=1:num_panel
   dF(i,1) = -P_e(i,1)*101325*Area(i,1)*nunit(i,1)
   dF(i,2) = -P_e(i,1)*101325*Area(i,1)*nunit(i,2)
   dF(i,3) = -P_e(i,1)*101325*Area(i,1)*nunit(i,3)
end

for i=1:num_panel
   dL(i,1) = dF(i,3)
   dD(i,1) = dF(i,1)
end

% Lift:     lift (N)
% Drag:     drag (N)
% L_D:      L/D ratio
Lift_inv = sum(dL)*2
Drag_inv = sum(dD)*2
L_D_inv = Lift_inv/Drag_inv %inviscid
CLinv = Lift_inv/qinf/Sref
CDinv = Drag_inv/qinf/Sref
CNinv = CLinv*cos(aoa*3.14159265359/180)+CDinv*sin(aoa*3.14159265359/180)
CAinv = CDinv*cos(aoa*3.14159265359/180)-CLinv*sin(aoa*3.14159265359/180)
for i=1:num_panel
   d_m(i,1) = -
(radi(i,1)*(dF(i,3)*cos(aoa*3.14159265359/180)+dF(i,1)*sin(aoa*3.14159265359/180))-
radi(i,3)*(dF(i,1)*cos(aoa*3.14159265359/180)-dF(i,3)*sin(aoa*3.14159265359/180)))
end
pitching = sum(d_m)*2
C_minv = pitching/qinf/Sref/chord

for i=1:num_panel+1
   if s(i,1) == 0
```

```
      s(i,1) = 0.000001 %zero Reynolds Number provides NaN heat transfer, so s is set to very
small number like 0.000001
   end
   Res(i,1) = ro_e(i,1)*((u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5)*s(i,1)/mu_e(i,1) % Reynolds
Number based on s
end

%to make jj array has the same index
jj(num_panel+1) = 0

% local skin friction
for i=1:num_panel+1
%h_w      : enthalpy at wall
%h_aw     : adiabatic wall enthalpy
%h_ref    : reference enthalpy
   h_aw(i,1)  = h_e(i,1) + (Pr_e(i,1)^0.5)*0.5*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)/1000
   h_awtub(i,1)  = h_e(i,1) + (Pr_e(i,1)^(1/3))*0.5*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)/1000
   % here, wall temperature is set to adiabatic wall temperature. h_w can
   % be also set to the cold case (0 K)
   h_w(i,1)   = 319.5 %h_aw(i,1)  (Tw = 314.5K)

   h_ref(i,1) = 0.28*h_e(i,1) + 0.5*h_w(i,1) + 0.22*h_aw(i,1)
   h_reftub(i,1) = 0.28*h_e(i,1) + 0.5*h_w(i,1) + 0.22*h_awtub(i,1)
%T_ref    : reference temperature at edge of boundary layer for each panel
%ro_ref   : reference air density at edge of boundary layer for each panel
%mu_ref   : reference viscosity at edge of boundary layer for each panel
%Pr_ref   : reference Prandtl Number at edge of boundary layer for each panel
   Pin = P_e(i,1)
   %Laminar
   h_known = h_ref(i,1)
   Tout = temperature (Pin, h_known)
   T_ref(i,1) = Tout
   ro_out   = density (Pin, Tout)
   ro_ref(i,1) = ro_out
   mu_out   = viscosity (Pin, Tout)
   mu_ref(i,1) = mu_out
   Pr_out   = prandtl (Pin, Tout)
   Pr_ref(i,1)= Pr_out
   %Turbulent
   h_known = h_reftub(i,1)
   Tout = temperature (Pin, h_known)
   T_reftub(i,1) = Tout
   ro_out   = density (Pin, Tout)
   ro_reftub(i,1) = ro_out
   mu_out   = viscosity (Pin, Tout)
   mu_reftub(i,1) = mu_out
```

```
   Pr_out   = prandtl (Pin, Tout)
   Pr_reftub(i,1)= Pr_out

%******* skin friction ************************************************
%cf : local skin friction coefficient (laminar)
%cf_tub : local skin friction coefficient (turbulent)
%*********************************************************************
   AA  = 0.332*Pr_e(i,1)^(1/3) %-------------laminar
   AAtub = 0.0296*Pr_e(i,1)^(1/3) %-------------turbulent
   sa  = 0.5 %------------------------laminar
   satub = 0.8 %------------------------turbulent
   sb  = 0.5 %------------------------laminar
   sbtub = 0.2 %------------------------turbulent
   sc  = 0.5 %------------------------laminar
   sctub = 0.8 %------------------------turbulent
   sj  = 0  % ------------------------flat plate

   cf(i,1) = 2*AA/(Pr_e(i,1)^(1/3)) * (ro_ref(i,1)/ro_e(i,1))^sa * (mu_ref(i,1)/mu_e(i,1))^sb *
Res(i,1)^(sc-1) * (3^sj)^0.5
   cf_tub(i,1) = 2*AAtub/(Pr_e(i,1)^(1/3)) * (ro_reftub(i,1)/ro_e(i,1))^satub *
(mu_reftub(i,1)/mu_e(i,1))^sbtub * Res(i,1)^(sctub-1) * (3^sj)^0.5
end

%******** heat transfer (ref. enthalpy) ********************************
%q_cw_en: Local Heat Transfer by using flat plate reference enthalpy methods  (J/s-m2)
%Nu = q_cw*s/(k_e *(T_w - T_aw)) --> q_cw_en = cf*Pr_e^(1/3)*Res*k_e*(T_w -
T_aw)/(2*s)
%*********************************************************************

for i=1:num_panel+1
   Pin = P_e(i,1)
   Tout = T_e(i,1)
   k_con = conductivity (Pin, Tout)
   k_e(i,1) = k_con  % (J/m-sec-K)
end
for i=1:num_panel+1
   h_known = h_w(i,1)
   Pin = P_e(i,1)
   T_w(i,1) = temperature (Pin, h_known)
   h_known = h_aw(i,1)
   T_aw(i,1) = temperature (Pin, h_known)
   h_known = h_awtub(i,1)
   T_awtub(i,1) = temperature (Pin, h_known)
   q_cw_en(i,1) =  -cf(i,1)*(Pr_e(i,1)^(1/3))*Res(i,1)*k_e(i,1)*(T_w(i,1) - T_aw(i,1))/(2*s(i,1))
% W/m2 J/s-m2,
```

```
   q_cw_en_tub(i,1) =  -cf_tub(i,1)*(Pr_e(i,1)^(1/3))*Res(i,1)*k_e(i,1)*(T_w(i,1) -
T_awtub(i,1))/(2*s(i,1))  % W/m2 J/s-m2,
end

%********** heat transfer (blunt body)
************************************************************
%q_cw_bl : Local Heat Transfer by using blunt body method (J/s-m2)
%q_cw_bl(s=0) = (0.9038/(ep^0.25)) * (C_w/Pr_e)^0.1 * (ro*Vinf*mu_e/R_b/Pr_e)^0.5 * (h_e
- h_w) % at stagnation point, so e : se here
%C_w = (ro_w*mu_w/(ro_e*mu_e)) % e : se here
%C_ws = (ro_e*mu_e/(ro_se*mu_se))
%q_cw_bl(s) = q_cw_bl(s=0) * C_ws * u_e*rbody^jq / (2^(jq+1) * (u_e(i)-u_e(i-1))/(s(i)-s(i-
1)))^0.5 * SUM(i=1-->i,C_ws(i)*u_e(i)*rbody^(2*jq)*(s(i)-s(i-1))
%rbody : radius of cross section of bodies of revolution
%jq    : 1 for bodies, 0 for 2-D
%************************************************************************

   for i=1:num_panel

z_original=((cent(i,1)^2+cent(i,3)^2)^0.5)*sin(aoa*3.14159265359/180+atan(cent(i,3)/cent(i,1))
)
      rbody(i,1)=(cent(i,2)^2+z_original^2)^0.5
   end
      rbody(num_panel+1,1)=0.001

%Find stagnation M, u_e, T_e for equilibrium condition

   ro1ro2 = 0.1 % step 1: first guess of ro_1/ro_2

   press_1 = Pinf
   temp_1 = Tinf
   ro_1 = ro
   vel_1 = Vinf
   enth_1 = enthalpy (press_1/101325, temp_1)
   error_equi = 1
   while error_equi > 0.0001
   press_2 = press_1 + ro_1*(vel_1)^2*(1-ro1ro2) % step 2: obtain p2
   enth_2 = enth_1 + (0.5/1000)*(vel_1)^2*(1-ro1ro2^2) % step 2: obtain h2 (J/g)
   P_stag = press_2/101325
   temp_2 = temperature (P_stag, enth_2) % step 3: obtain T2
   ro_2 = density (P_stag, enth_2) % step 3: obtain ro_2
   ro1ro2new = ro_1/ro_2  % step 4: obtain  new ro1ro2
   error_equi = abs(ro1ro2 - ro1ro2new)
   ro1ro2 = ro1ro2new
   end
   vel_2 = ro1ro2 * vel_1
```

```
   M_stag = vel_2/((gamma*temp_2*Rair)^0.5) %Mach number at stagnation point
   P_stag = press_2/101325 %Pressure at stagnation point (atm)
   T_stag = temp_2 % Temperature at stagnation point (K)
   h_stag = enth_2 % enthalpy at stagnation point

   u_stag_e = vel_2

   P_stag; T_stag; T_sw=temperature (P_stag, h_w(1,1)); Pin = P_stag
   ro_sw = density (Pin, T_sw); mu_sw = viscosity (Pin, T_sw)
   ro_se = ro/ro1ro2; mu_se = viscosity (Pin, T_stag)
   Pr_se = prandtl (Pin, T_stag)
   h_se = enthalpy (Pin, T_stag); h_sw= enthalpy (Pin, T_sw)
   C_w01 = (ro_sw*mu_sw)/ro_se/mu_se
   R_b01 =  0.3048%0.10110667%((1+((nod(8,3)-nod(1,3))/(nod(8,1)-
nod(1,1)))^2)^1.5)/abs((nod(8,3)-2*nod(1,3)+nod(2,3))/(nod(8,1)-nod(1,1))/(nod(2,1)-nod(1,1)))
   q_cw_bl01 = 1000*(0.9038/(ep^0.25)) * (C_w01/Pr_se)^0.1 *
(ro*Vinf*mu_se/R_b01/Pr_se)^0.5 * (h_se - h_sw) % at stagnation point(body) W/m2 J/s-m2, so
e : se here
   q_cw_bl_nose= q_cw_bl01

   s(num_panel+1,1)=0
   u_e(num_panel+1,1)=u_stag_e; u_e(num_panel+1,2)=0; u_e(num_panel+1,3)=0
   rbody(num_panel+1,1)=0.0001 %radius of cross section of body
   ro_e(num_panel+1,1)=ro_se
   mu_e(num_panel+1,1)=mu_se
   C_ws(num_panel+1,1)=1.0

   for k= 1:13
   SUMq = 0
   for i =k:13:117
      j=jj(i,1)
      C_ws(i,1) = (ro_e(i,1)*mu_e(i,1))/(ro_se*mu_se)
      f_s_1(i,1) = (C_ws(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5)*(rbody(i,1)^2)
      f_s_0(i,1) = (C_ws(j,1)*(u_e(j,1)^2+u_e(j,2)^2+u_e(j,3)^2)^0.5)*(rbody(j,1)^2)
      SUMq = SUMq + (f_s_0(i,1) + f_s_1(i,1))*(s(i,1)-s(j,1))/2
      q_cw_bl(i,1) = (SUMq^-
0.5)*q_cw_bl01*C_ws(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5*rbody(i,1)/((4*(((u_e(k,1)^
2+u_e(k,2)^2+u_e(k,3)^2)^0.5-u_stag_e)/(s(k,1))))^0.5)
   end
   end

   for i=119:130
      rbody(i,1)=1.0 %for two dimensional
   end
   for i=131:142
      rbody(i,1)=1.0 %for two dimensional
```

```
    end

    P_stag; T_stag; T_sw=temperature (P_stag, h_w(119,1)); Pin = P_stag
    ro_sw = density (Pin, T_sw); mu_sw = viscosity (Pin, T_sw)
    ro_se = density (Pin, T_stag); mu_se = viscosity (Pin, T_stag)
    Pr_se = prandtl (Pin, T_stag)
    h_se = enthalpy (Pin, T_stag); h_sw= enthalpy (Pin, T_sw)
    C_w01 = (ro_sw*mu_sw)/ro_se/mu_se
    R_b01 = 0.018548985 %15.17 % ((1+((nod(8,3)-nod(1,3))/(nod(8,1)-
nod(1,1)))^2)^1.5)/abs((nod(8,3)-2*nod(1,3)+nod(2,3))/(nod(8,1)-nod(1,1))/(nod(2,1)-nod(1,1)))
    q_cw_bl011 = 1000*(0.9038/(ep^0.25)) * (C_w01/Pr_se)^0.1 *
(ro*Vinf*mu_se/R_b01/Pr_se)^0.5 * (h_se - h_sw) % at stagnation point(wing) W/m2 J/s-m2, so
e : se here

    s(num_panel+1,1)=0
    u_e(num_panel+1,1)=Vinf; u_e(num_panel+1,2)=0; u_e(num_panel+1,3)=0
    %u_e(121,1)=u_stag_e; u_e(121,2)=0; u_e(121,1)=0
    %q_cw_bl(53,1) = q_cw_bl01

    SUMq = 0
    for i =119:124
        j=jj(i,1)
        C_ws(i,1) = (ro_e(i,1)*mu_e(i,1))/(ro_se*mu_se)
        f_s_1(i,1) = C_ws(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5
        f_s_0(i,1) = C_ws(j,1)*(u_e(j,1)^2+u_e(j,2)^2+u_e(j,3)^2)^0.5
        SUMq = SUMq + (f_s_0(i,1) + f_s_1(i,1))*(s(i,1)-s(j,1))/2
        q_cw_bl(i,1) = (SUMq^-
0.5)*q_cw_bl011*C_ws(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5/((2*(((u_e(121,1)^2+u_e(1
21,2)^2+u_e(121,3)^2)^0.5-u_stag_e)/(s(121,1))))^0.5)
    end
    SUMq = 0
    for i =125:130
        j=jj(i,1)
        C_ws(i,1) = (ro_e(i,1)*mu_e(i,1))/(ro_se*mu_se)
        f_s_1(i,1) = C_ws(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5
        f_s_0(i,1) = C_ws(j,1)*(u_e(j,1)^2+u_e(j,2)^2+u_e(j,3)^2)^0.5
        SUMq = SUMq + (f_s_0(i,1) + f_s_1(i,1))*(s(i,1)-s(j,1))/2
        q_cw_bl(i,1) = (SUMq^-
0.5)*q_cw_bl011*C_ws(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5/((2*(((u_e(127,1)^2+u_e(1
27,2)^2+u_e(127,3)^2)^0.5-u_stag_e)/(s(127,1))))^0.5)
    end

    P_stag; T_stag; T_sw=temperature (P_stag, h_w(137,1)); Pin = P_stag
    ro_sw = density (Pin, T_sw); mu_sw = viscosity (Pin, T_sw)
    ro_se = density (Pin, T_stag); mu_se = viscosity (Pin, T_stag)
    Pr_se = prandtl (Pin, T_stag)
```

```
   h_se = enthalpy (Pin, T_stag); h_sw= enthalpy (Pin, T_sw)
   C_w01 = (ro_sw*mu_sw)/ro_se/mu_se
   R_b01 = 0.061123322 % ((1+((nod(8,3)-nod(1,3))/(nod(8,1)-
nod(1,1)))^2)^1.5)/abs((nod(8,3)-2*nod(1,3)+nod(2,3))/(nod(8,1)-nod(1,1))/(nod(2,1)-nod(1,1)))
   q_cw_bl012 = 1000*(0.9038/(ep^0.25)) * (C_w01/Pr_se)^0.1 *
(ro*Vinf*mu_se/R_b01/Pr_se)^0.5 * (h_se - h_sw) % at stagnation point(wing) W/m2 J/s-m2, so
e : se here

   s(num_panel+1,1)=0
   u_e(num_panel+1,1)=Vinf; u_e(num_panel+1,2)=0; u_e(num_panel+1,3)=0

   SUMq = 0
   for i =131:136
      j=jj(i,1)
      C_ws(i,1) = (ro_e(i,1)*mu_e(i,1))/(ro_se*mu_se)
      f_s_1(i,1) = C_ws(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5
      f_s_0(i,1) = C_ws(j,1)*(u_e(j,1)^2+u_e(j,2)^2+u_e(j,3)^2)^0.5
      SUMq = SUMq + (f_s_0(i,1) + f_s_1(i,1))*(s(i,1)-s(j,1))/2
      q_cw_bl(i,1) = (SUMq^-
0.5)*q_cw_bl012*C_ws(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5/((2*(((u_e(121,1)^2+u_e(1
21,2)^2+u_e(121,3)^2)^0.5-u_stag_e)/(s(121,1))))^0.5)
  end
  SUMq = 0
  for i =137:142
      j=jj(i,1)
      C_ws(i,1) = (ro_e(i,1)*mu_e(i,1))/(ro_se*mu_se)
      f_s_1(i,1) = C_ws(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5
      f_s_0(i,1) = C_ws(j,1)*(u_e(j,1)^2+u_e(j,2)^2+u_e(j,3)^2)^0.5
      SUMq = SUMq + (f_s_0(i,1) + f_s_1(i,1))*(s(i,1)-s(j,1))/2
      q_cw_bl(i,1) = (SUMq^-
0.5)*q_cw_bl012*C_ws(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5/((2*(((u_e(127,1)^2+u_e(1
27,2)^2+u_e(127,3)^2)^0.5-u_stag_e)/(s(127,1))))^0.5)
  end

%***** heat transfer choice
**********************************************************
%
% q_cw : heat transfer at each panel (J/m-s2)
%
% q_cw_en: Local Heat Transfer by using flat plate reference enthalpy methods  (J/s-m2)
% q_cw_bl : Local Heat Transfer by using blunt body method (J/s-m2)
%
%   Around the stagnation points, the flat plate reference enthalpy method provides
%   extremely high value of the heat transfer as discussed before, so the blunt bodies
%   solutions should be used for the panels around the stagnation points. For far
%   panels from the stagnation points, the both methods agree closely, and at least
```

```
%   both have the similar behavior. Therefore, the flat plate reference enthalpy method
%   is used to find the heat transfer for far surface of the body and wings.
%
% *around the stagnation point or leading edge: stagnation point(panel) next panel
%  otherwise, far from the stagnation points
%
% tangential component of velocity on "shadow" panel is found by P-M expansion
% s_b  =[nose_vector; nunit(6,1); nunit(12,1); nunit(18,1); nunit(24,1); nunit(30,1)]
% s_w1 =[nunit(53,1); nunit(52,1); nunit(51,1)]
% s_w2 =[nunit(64,1); nunit(63,1); nunit(62,1)]
% s_w3 =[nunit(74,1); nunit(73,1)]
%**************************************************************************
**********

q_cw = q_cw_en_tub %set q_cw to q_cw_en here, and q_cw_bl will be applied to around the
stagnation later

for i=1:num_panel
   St2(i,1) = q_cw(i,1)/(ro*Vinf)/(h_se - h_w(i,1))/1000
end
Pin=Pinf/101325; Tout=Tinf
mu_e(num_panel+1,1) = viscosity (Pin, Tout)
Re_inf = ro*Vinf/mu_e(num_panel+1,1)  %Reynolds number divided by certain characteristic
length
%Use blunt body scheme is applied only for the stagnation points(body and
%wing) and the the most downstream panel of each stream.

%***** cooling analysis
**********************************************************************
%
% cf : local skin friction coefficient (laminar)
% cf_tub : local skin friction coefficient (turbulent)
% cf_cool : local skin friction coefficient of cooled condition. This array
%          includes cf at non-cooled panels also.
%
% q_cw : Local Heat Transfer (J/s-m2); reference enthalpy method or blun body method
% q_cw_cool : local heat transfer (J/s-m2) of cooled condition. This array
%           includes q_cw at non-cooled panels also.
%
% nunit:  n (N unit) vector of each panel
% Area:   area of each panel (m2)
% ro_e    : air density at edge of boundary layer for each panel (kg/m3)
% u_e     : tangential component of velocity on each panel (m/s)
% v_w     : normal velocity at the wall of each panel (m/s)
% ro_w    : injected gas density at the wall (kg/m3)
%
```

```
% Lift_vis:     lift (N) (viscous without cooling)
% Lift_viscool: lift (N) (viscous with cooling)
% Drag_vis:     drag (N) (viscous without cooling)
% Drag_viscool: drag (N) (viscous with cooling)
% L_D_vis:      L/D ratio (viscous without cooling)
% L_D_viscool:  L/D ratio (viscous with cooling)
% dF_f:         differential friction force (N) acting on each panel (without cooling)
% dF_fcool:     differential friction force (N) acting on each panel (with cooling)
% dLr:     differential lift force acting on each panel (right side)
% dDr:     differential drag force acting on each panel (right side)
%**************************************************************************
***************

%*** Total L/D ratio of viscous case without cooling *********

for i=1:num_panel
   qinf_e(i,1) = ro_e(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)/2.0
   dF_fx(i,1) =
cf_tub(i,1)*qinf_e(i,1)*Area(i,1)*u_e(i,1)/((u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5)
   dF_fy(i,1) =
cf_tub(i,1)*qinf_e(i,1)*Area(i,1)*u_e(i,2)/((u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5)
   dF_fz(i,1) =
cf_tub(i,1)*qinf_e(i,1)*Area(i,1)*u_e(i,3)/((u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5)

end

Lift_vis = (sum(dL)+sum(dF_fz(:,1)))*2
Drag_vis = (sum(dD)+sum(dF_fx(:,1)))*2
L_D_vis = Lift_vis/Drag_vis
CLvis = Lift_vis/qinf/Sref
CDvis = Drag_vis/qinf/Sref
CNvis = CLvis*cos(aoa*3.14159265359/180)+CDvis*sin(aoa*3.14159265359/180)
CAvis = CDvis*cos(aoa*3.14159265359/180)-CLvis*sin(aoa*3.14159265359/180)

for i=1:num_panel
   d_mvis(i,1) = -
(radi(i,1)*((dF(i,3)+dF_fz(i,1))*cos(aoa*3.14159265359/180)+(dF(i,1)+dF_fx(i,1))*sin(aoa*3.1
4159265359/180))-radi(i,3)*((dF(i,1)+dF_fx(i,1))*cos(aoa*3.14159265359/180)-
(dF(i,3)+dF_fz(i,1))*sin(aoa*3.14159265359/180)))
end
pitching_vis = sum(d_mvis)*2
C_mvis = pitching_vis/qinf/Sref/chord

%*** Determine the panel cooled **********************************
% panel around the bottom noze: [8,9,10,11,12,13,14] (not stagnation point)
%
```

```
% Injected Gas properties:
%  H2: 0.08078kg/m3 (T=300K)
%*****************************************************************

int_bf = 11 %the number of step of 0.1 in bf. (int_bf - 1)*0.1 = max of bf
mass_rate_step = 0.001 %kg/m2-s

%partial bottom panels are cooled
for i=1:num_panel
 for j=1:int_bf
    cf_cool(i,j) = cf_tub(i,1)
 end
end

for i=1:num_panel
    Nu(i,1) = -q_cw(i,1)*s(i,1)/(k_e(i,1)*(T_w(i,1)-T_aw(i,1)))
end

for i=1:num_panel
    St(i,1) = Pr_e(i,1)*Nu(i,1)/Res(i,1)
end

for i=1:num_panel
 for j=1:int_bf
    q_cw_cool(i,j) = q_cw(i,1)
 end
end

A_cool2 =0
for i=1:1:num_panel
if q_cw(i,1)>0 %Choose panels to be cooled here by setting the minimum value of allowable
heat trans.
    cooled_panel2(i,1)=1.0
    A_cool2 = A_cool2+Area(i,1)
    %1:num_panel%all panels
 for j = 1:int_bf
    m_inj(i,j)=mass_rate_step*(j-1)
    bf2(i,j) = m_inj(i,j)/(0.5*cf_tub(i,1)*ro_e(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5)
    bh2(i,j) = m_inj(i,j)/(St(i,1)*ro_e(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5)
    if j==1
    cf_cool(i,j) = cf_tub(i,1)
    q_cw_cool(i,j) = q_cw(i,1)
    else
    cf_cool(i,j) = cf_tub(i,1)*(bf2(i,j)/(exp(bf2(i,j))-1))
    q_cw_cool(i,j) = q_cw(i,1)*(bh2(i,j)/(exp(bh2(i,j))-1))
    end
```

```
   end
else
end
end

for i=1:num_panel
   qinf_e(i,1) = ro_e(i,1)*(u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)/2.
   for j=1:int_bf
   dF_fxcool(i,j) =
cf_cool(i,j)*qinf_e(i,1)*Area(i,1)*u_e(i,1)/((u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5)
   dF_fycool(i,j) =
cf_cool(i,j)*qinf_e(i,1)*Area(i,1)*u_e(i,2)/((u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5)
   dF_fzcool(i,j) =
cf_cool(i,j)*qinf_e(i,1)*Area(i,1)*u_e(i,3)/((u_e(i,1)^2+u_e(i,2)^2+u_e(i,3)^2)^0.5)
   end
end

Q_total=0.0
for i=1:num_panel
   Q_total = Q_total + q_cw(i,1)*Area(i,1)
end

for j=1:int_bf
   Lift_viscool2(j,1) = (sum(dL)+sum(dF_fzcool(:,j)))*2
   Drag_viscool2(j,1) = (sum(dD)+sum(dF_fxcool(:,j)))*2
   L_D_viscool2(j,1) = Lift_viscool2(j,1)/Drag_viscool2(j,1)
   L_Dimprovement2(j,1)=L_D_viscool2(j,1)/L_D_vis
   Q_total_cool=0.0
   for i=1:num_panel
      Q_total_cool = Q_total_cool + q_cw_cool(i,j)*Area(i,1)
   end
   q_cwimprovement2(j,1)=Q_total_cool/Q_total
end
```

APPENDIX B
MATLAB CODE (FUNCTION) TO COMPUTE TEMPERATURE

```
%*************************************************************************
%code thermo_property (temperature)
%This code compute thermodynamic property of air as function of T an P
%Thermodynamic property includes:
%compressibility Z(T, P)
%temperature T(P, h)
%density ro(T, P)
%viscocity mu(T,P)
%Prandtl number Pr
%Inputs are: entahlpy and pressure, so h and P must be known
%output of this code is accurate but not exact solution
%*************************************************************************

%****** Input ************************************************************
%Pin     : Pressure input, this value must be known  unit:atm
%h_known : enthalpy input, this value must be known  unit:J/g or kJ/kg
%*************************************************************************
function [Tout] = temperature (Pin, h_known)

%Pin = 0.01              %Pressure input, this value must be known  unit:atm
%h_known = 11054.6       %enthalpy input, this value must be known  unit:J/g or kJ/kg


%****** Compressibility **************************************************
%computed by using interp2 function with known Hansen's data (T and P)
%inter2 is function of 2-D data interpolation
%T       : temperature
%P       : pressure
%Tin      : input temperature
%Pin      : input pressure
%ZTP      : compressibility table
%Z       : compressibility of air as f(Tin, Pin)
%*************************************************************************

T = 0:500:15000   %Temperature range 0K --> 15000K
P = [100 10 1 0.1 0.01 0.001 0.0001 0]  %Pressure range 0atm --> 0.0001atm --> 100atm
ZTP =[1.000 1.000 1.000 1.000 1.000 1.000 1.003 1.012 1.033 1.071 1.118 1.159 ...
   1.189 1.214 1.243 1.284 1.341 1.418 1.512 1.616 1.718 1.807 1.876 1.927 1.965 ...
   1.993 2.017 2.039 2.062 2.086 2.113
   1.000 1.000 1.000 1.000 1.000 1.001 1.009 1.035 1.089 1.149 1.186 1.208 ...
   1.235 1.279 1.351 1.457 1.590 1.727 1.838 1.914 1.962 1.993 2.018 2.042 2.067 ...
   2.098 2.135 2.180 2.233 2.297 2.372
   1.000 1.000 1.000 1.000 1.000 1.004 1.026 1.092 1.165 1.196 1.214 1.248 ...
```

```
   1.316 1.437 1.607 1.778 1.896 1.959 1.993 2.018 2.042 2.071 2.111 2.163 2.232 ...
   2.318 2.426 2.553 2.700 2.861 3.028
   1.000 1.000 1.000 1.000 1.001 1.011 1.072 1.167 1.198 1.213 1.252 1.348 ...
   1.529 1.752 1.904 1.971 2.001 2.023 2.050 2.090 2.149 2.234 2.351 2.505 2.694 ...
   2.910 3.135 3.347 3.527 3.667 3.769
   1.000 1.000 1.000 1.000 1.002 1.033 1.149 1.197 1.208 1.245 1.359 1.599 ...
   1.849 1.961 1.997 2.017 2.044 2.090 2.166 2.286 2.462 2.700 2.983 3.272 3.520 ...
   3.700 3.818 3.889 3.932 3.957 3.973
   1.000 1.000 1.000 1.000 1.005 1.088 1.192 1.203 1.228 1.337 1.622 1.898 ...
   1.983 2.006 2.027 2.067 2.144 2.284 2.510 2.832 3.202 3.526 3.745 3.867 3.931 ...
   3.963 3.979 3.988 3.993 3.996 3.997
   1.000 1.000 1.000 1.000 1.016 1.163 1.200 1.211 1.287 1.577 1.910 1.990 ...
   2.008 2.032 2.088 2.210 2.446 2.826 3.282 3.645 3.843 3.932 3.969 3.985 3.993 ...
   3.996 3.998 3.999 3.999 4.000 4.000
   1.000 1.000 1.000 1.000 1.016 1.163 1.200 1.211 1.287 1.577 1.910 1.990 ...
   2.008 2.032 2.088 2.210 2.446 2.826 3.282 3.645 3.843 3.932 3.969 3.985 3.993 ...
   3.996 3.998 3.999 3.999 4.000 4.000]


%****** Temperature ***************************************************
%Z function is known, T and h relation is computed by using interp2 function with
%known Hansen's data (h, P, and Z)
%Now, P and h are known. Set P known and assume T and iterate to match h
%given by T and h relation
%T       : temperature (K)
%P       : pressure (atm)
%h       : enthalpy (J/g)
%h_known  : enthalpy known (input enthalpy) (J/g)
%Z       : compressibility
%R       : universal gas constant (J/mol-K)
%MWair    : molecular weight (g/mol)
%ZH_RT    : dimensionless enthalpy (Zh/RT) table givenin Hansen's
%ZHRT     : dimensionless enthalpy (Zh/RT) as function of (T, P)
%Tin      : input temperature (guessed temperature)
%Pin      : input pressure
%ZTP      : compressibility table
%Z       : compressibility of air as f(Tin, Pin)
%**********************************************************************

T = 0:500:15000   %Temperature range 0K --> 15000K
P = [100 10 1 0.1 0.01 0.001 0.0001 0]  %Pressure range 0atm --> 0.0001atm --> 100atm
ZH_RT = [3.52 3.52 3.65 3.80 3.92 4.01 4.13 4.34 4.70 5.20 5.73 6.13 6.38 6.62 6.95 7.44 8.16
...
      9.10 10.20 11.36 12.42 13.23 13.77 14.08 14.22 14.28 14.30 14.31 14.34 14.40 14.49
   3.52 3.52 3.65 3.80 3.92 4.03 4.25 4.75 5.56 6.29 6.62 6.80 7.11 7.72 8.76 10.24 11.99 13.63
...
```

```
      14.79 15.40 15.61 15.64 15.60 15.58 15.62 15.74 15.96 16.28 16.71 17.26 17.92
   3.52 3.52 3.65 3.80 3.92 4.09 4.61 5.75 6.74 6.98 7.10 7.58 8.70 10.64 13.20 15.48 16.73
17.09 17.04 ...
      16.91 16.84 16.90 17.13 17.57 18.24 19.16 20.32 21.72 23.29 24.98 26.66
   3.52 3.52 3.65 3.80 3.93 4.27 5.55 7.08 7.28 7.33 7.96 9.73 12.93 16.46 18.34 18.66 18.43
18.17 ...
      18.09 18.29 18.85 19.84 21.31 23.28 25.69 28.36 30.99 33.27 34.97 36.02 36.53
   3.52 3.52 3.65 3.80 3.97 4.81 7.13 7.62 7.53 8.14 10.48 15.14 19.30 20.35 20.01 19.54 19.34
...
      19.60 20.49 22.17 24.78 28.28 32.31 36.13 39.01 40.66 41.26 41.17 40.69 40.01 39.24
   3.52 3.52 3.65 3.80 4.07 6.16 8.02 7.77 8.09 10.55 16.68 21.58 21.97 21.24 20.69 20.72 21.65
...
      23.85 27.66 33.00 38.79 43.28 45.57 46.09 45.64 44.74 43.69 42.61 41.55 40.53 39.57
   3.52 3.52 3.65 3.80 4.41 8.02 8.19 8.03 9.82 16.80 23.46 23.58 22.54 21.93 22.29 24.26 28.65
...
      35.75 43.74 49.15 50.96 50.64 49.48 48.07 46.64 45.26 43.97 42.76 41.64 40.58 39.60
   3.52 3.52 3.65 3.80 4.41 8.02 8.19 8.03 9.82 16.80 23.46 23.58 22.54 21.93 22.29 24.26 28.65
...
      35.75 43.74 49.15 50.96 50.64 49.48 48.07 46.64 45.26 43.97 42.76 41.64 40.58 39.60]

%P and h are known. Set P known and assume T and iterate to match h
%given by T and h relation above
R = 8.3144      %universal gas constant  unit:J/K-mol
MWair = 29
cp_0 = 1.0    % specific heat constant pressure at T = 0K
Tin = h_known/cp_0            %guess temperature , unit:K
dh = -1         % initialize difference between hi and h_known
while dh < 0
   Z = interp2(T, P, ZTP,Tin, Pin)
   ZHRT = interp2(T, P, ZH_RT,Tin, Pin)
   hi = ZHRT*R*Tin/Z/MWair  %enthalpy i  unit:J/g
   dh = h_known - hi
   Tin = Tin - 5
end
Tout = Tin + 5 %unit:K
```

APPENDIX C
MATLAB CODE (FUNCTION) TO COMPUTE DENSITY

```
%*************************************************************************
%code thermo_property
%This code compute thermodynamic property of air as function of T an P
%Thermodynamic property includes:
%compressibility Z(T, P)
%temperature T(P, h)
%density ro(T, P)
%viscocity mu(T,P)
%Prandtl number Pr
%Inputs are: entahlpy and pressure, so h and P must be known
%output of this code is accurate but not exact solution
%*************************************************************************

%****** Input ***************************************************************
%Pin     : Pressure input, this value must be known  unit:atm
%h_known : enthalpy input, this value must be known  unit:J/g or kJ/kg
%*************************************************************************
function [ro] = density (Pin, Tout)


T = 0:500:15000   %Temperature range 0K --> 15000K
P = [100 10 1 0.1 0.01 0.001 0.0001 0]  %Pressure range 0atm --> 0.0001atm --> 100atm
ZTP =[1.000 1.000 1.000 1.000 1.000 1.000 1.003 1.012 1.033 1.071 1.118 1.159 ...
   1.189 1.214 1.243 1.284 1.341 1.418 1.512 1.616 1.718 1.807 1.876 1.927 1.965 ...
   1.993 2.017 2.039 2.062 2.086 2.113
   1.000 1.000 1.000 1.000 1.000 1.001 1.009 1.035 1.089 1.149 1.186 1.208 ...
   1.235 1.279 1.351 1.457 1.590 1.727 1.838 1.914 1.962 1.993 2.018 2.042 2.067 ...
   2.098 2.135 2.180 2.233 2.297 2.372
   1.000 1.000 1.000 1.000 1.000 1.004 1.026 1.092 1.165 1.196 1.214 1.248 ...
   1.316 1.437 1.607 1.778 1.896 1.959 1.993 2.018 2.042 2.071 2.111 2.163 2.232 ...
   2.318 2.426 2.553 2.700 2.861 3.028
   1.000 1.000 1.000 1.000 1.001 1.011 1.072 1.167 1.198 1.213 1.252 1.348 ...
   1.529 1.752 1.904 1.971 2.001 2.023 2.050 2.090 2.149 2.234 2.351 2.505 2.694 ...
   2.910 3.135 3.347 3.527 3.667 3.769
   1.000 1.000 1.000 1.000 1.002 1.033 1.149 1.197 1.208 1.245 1.359 1.599 ...
   1.849 1.961 1.997 2.017 2.044 2.090 2.166 2.286 2.462 2.700 2.983 3.272 3.520 ...
   3.700 3.818 3.889 3.932 3.957 3.973
   1.000 1.000 1.000 1.000 1.005 1.088 1.192 1.203 1.228 1.337 1.622 1.898 ...
   1.983 2.006 2.027 2.067 2.144 2.284 2.510 2.832 3.202 3.526 3.745 3.867 3.931 ...
   3.963 3.979 3.988 3.993 3.996 3.997
   1.000 1.000 1.000 1.000 1.016 1.163 1.200 1.211 1.287 1.577 1.910 1.990 ...
   2.008 2.032 2.088 2.210 2.446 2.826 3.282 3.645 3.843 3.932 3.969 3.985 3.993 ...
   3.996 3.998 3.999 3.999 4.000 4.000
   1.000 1.000 1.000 1.000 1.016 1.163 1.200 1.211 1.287 1.577 1.910 1.990 ...
   2.008 2.032 2.088 2.210 2.446 2.826 3.282 3.645 3.843 3.932 3.969 3.985 3.993 ...
   3.996 3.998 3.999 3.999 4.000 4.000]
R = 8.3144      %universal gas constant  unit:J/K-mol
```

```
MWair = 29

%****** density ************************************************************
%density is found by using simple equation of state with compressibility
%ro = P*MWair/(Z*R*T)
%ro    : density (kg/m3)
%Z     : compressibility
%MWair : molecular weight (g/mol)
%R     : universal gas constant (J/mol-K)
%T     : temperature computed above (K)
%***************************************************************************

Z = interp2(T, P, ZTP,Tout, Pin)
ro = Pin*101325*MWair/R/Tout/1000
```

APPENDIX D
MATLAB CODE (FUNCTION) TO COMPUTE VOSCOSITY

```
%*********************************************************************
%code thermo_property (viscosiy)
%This code compute thermodynamic property of air as function of T an P
%Thermodynamic property includes:
%compressibility Z(T, P)
%temperature T(P, h)
%density ro(T, P)
%viscocity mu(T,P)
%Prandtl number Pr
%Inputs are: entahlpy and pressure, so h and P must be known
%output of this code is accurate but not exact solution
%*********************************************************************

%****** Input ****************************************************
%Pin     : Pressure input, this value must be known  unit:atm
%h_known : enthalpy input, this value must be known  unit:J/g or kJ/kg
%*********************************************************************
function [mu] = viscosity (Pin, Tout)

%Pin = 0.01              %Pressure input, this value must be known  unit:atm
%h_known = 11054.6       %enthalpy input, this value must be known  unit:J/g or kJ/kg


%****** Compressibility *********************************************
%computed by using interp2 function with known Hansen's data (T and P)
%inter2 is function of 2-D data interpolation
%T       : temperature
%P       : pressure
%Tin      : input temperature
%Pin      : input pressure
%ZTP      : compressibility table
%Z       : compressibility of air as f(Tin, Pin)
%*********************************************************************

R = 8.3144      %universal gas constant  unit:J/K-mol
MWair = 29

%****** Viscosity ***************************************************
%computed by using interp2 function with known Hansen's data (T and P)
%inter2 is function of 2-D data interpolation
%T       : temperature
%P       : pressure
%Tout     : input temperature (computed above)
%Pin      : input pressure
%Cmu      : viscousity ratio (mu/nu0)  given in Hansen's
%mu0      : reference viscosity
%mu       : viscosity (gm/cm-s)
```

```
%**************************************************************************

T = 0:500:15000   %Temperature range 0K --> 15000K
P = [100 10 1 0.1 0.01 0.001 0.0001 0]  %Pressure range 0atm --> 0.0001atm --> 100atm
Cmu = [1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.003 1.010 1.022 1.036 1.050 1.072 1.089
1.112 ...
       1.143 1.185 1.238 1.298 1.361 1.418 1.467 1.509 1.549 1.577 1.581 1.594 1.599 1.601 1.604
   1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.001 1.008 1.022 1.036 1.052 1.067 1.090 1.124 1.175
1.238 ...
       1.307 1.368 1.418 1.468 1.496 1.501 1.511 1.520 1.516 1.508 1.492 1.468 1.415 1.387
   1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.003 1.016 1.029 1.043 1.060 1.090 1.139 1.208 1.283
1.342 ...
       1.386 1.425 1.438 1.445 1.448 1.442 1.424 1.394 1.342 1.274 1.187 1.082 0.940 0.828
   1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.006 1.020 1.033 1.051 1.086 1.148 1.229 1.294 1.332
1.371 ...
       1.386 1.396 1.393 1.375 1.335 1.267 1.168 1.040 0.881 0.711 0.547 0.408 0.268 0.212
   1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.010 1.022 1.033 1.051 1.086 1.148 1.229 1.294 1.332
1.347 ...
       1.343 1.314 1.251 1.143 0.983 0.782 0.571 0.387 0.249 0.158 0.100 0.067 0.042 0.016
   1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.010 1.024 1.055 1.128 1.209 1.257 1.286 1.303 1.307
1.280 ...
       1.207 1.068 0.853 0.595 0.361 0.200 0.108 0.063 0.036 0.024 0.018 0.015 0.013 0.012
   1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.011 1.032 1.096 1.181 1.227 1.256 1.271 1.264 1.210
1.072 ...
       0.826 0.517 0.261 0.118 0.055 0.029 0.018 0.012 0.009 0.008 0.007 0.007 0.008 0.008
   1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.011 1.032 1.096 1.181 1.227 1.256 1.271 1.264 1.210
1.072 ...
       0.826 0.517 0.261 0.118 0.055 0.029 0.018 0.012 0.009 0.008 0.007 0.007 0.008 0.008]

mu0 = 1.462*((Tout)^0.5)/(1+112/Tout)*10^-5
mu = interp2(T, P, Cmu, Tout, Pin)*mu0/10
```

APPENDIX E
MATLAB CODE (FUNCTION) TO COMPUTE PRANDTL NUMBER

```
%*************************************************************************
%code thermo_property (Prandtl number)
%This code compute thermodynamic property of air as function of T an P
%Thermodynamic property includes:
%compressibility Z(T, P)
%temperature T(P, h)
%density ro(T, P)
%viscocity mu(T,P)
%Prandtl number Pr
%Inputs are: entahlpy and pressure, so h and P must be known
%output of this code is accurate but not exact solution
%*************************************************************************

%****** Input ************************************************************
%Pin     : Pressure input, this value must be known  unit:atm
%h_known : enthalpy input, this value must be known  unit:J/g or kJ/kg
%*************************************************************************
function [Pr] = prandtl (Pin, Tout)

%Pin = 0.01              %Pressure input, this value must be known  unit:atm
%h_known = 11054.6       %enthalpy input, this value must be known  unit:J/g or kJ/kg


%****** Compressibility *************************************************
%computed by using interp2 function with known Hansen's data (T and P)
%inter2 is function of 2-D data interpolation
%T       : temperature
%P       : pressure
%Tin      : input temperature
%Pin      : input pressure
%ZTP      : compressibility table
%Z       : compressibility of air as f(Tin, Pin)
%*************************************************************************


%****** Prandtl number ***************************************************
%computed by using interp2 function with known Hansen's data (T and P)
%inter2 is function of 2-D data interpolation
%T       : temperature
%P       : pressure
%Tout     : input temperature (computed above)
%Pin      : input pressure
%Pr_table : Prandtl number given in Hansen's
%Pr       : Prandtl number
%*************************************************************************

T = 0:500:15000   %Temperature range 0K --> 15000K
```

```
P = [100 10 1 0.1 0.01 0.001 0.0001 0]  %Pressure range 0 atm --> 0.0001atm --> 100atm
Pr_table =[0.738 0.738 0.756 0.767 0.773 0.762 0.740 0.678 0.640 0.654 0.702 0.748 0.763 0.610 0.593
0.595 ...
       0.620 0.666 0.730 0.806 0.886 0.937 0.955 0.947 0.908 0.728 0.525 0.438 0.421 0.401 0.394
    0.738 0.738 0.756 0.767 0.773 0.751 0.680 0.631 0.662 0.743 0.767 0.620 0.592 0.592 0.620 0.688
0.788 ...
       0.891 0.961 0.966 0.872 0.532 0.463 0.434 0.412 0.396 0.383 0.369 0.360 0.349 0.341
    0.738 0.738 0.756 0.767 0.773 0.696 0.627 0.660 0.762 0.752 0.611 0.583 0.602 0.673 0.796 0.927
0.983 ...
       0.943 0.807 0.497 0.429 0.404 0.382 0.369 0.355 0.343 0.333 0.319 0.302 0.277 0.253
    0.738 0.738 0.756 0.767 0.766 0.645 0.636 0.744 0.759 0.610 0.581 0.617 0.738 0.906 0.986 0.969
0.648 ...
       0.411 0.382 0.364 0.348 0.339 0.327 0.312 0.292 0.263 0.227 0.185 0.144 0.0986 0.0819
    0.738 0.738 0.756 0.767 0.724 0.611 0.740 0.737 0.619 0.578 0.624 0.785 0.969 0.955 0.830 0.424
0.387 ...
       0.363 0.348 0.336 0.319 0.295 0.254 0.201 0.146 0.101 0.0688 0.0470 0.0345 0.0245 0.0129
    0.738 0.738 0.756 0.767 0.668 0.654 0.745 0.658 0.580 0.611 0.799 0.989 0.891 0.464 0.404 0.371
0.351 ...
       0.335 0.316 0.279 0.216 0.145 0.0877 0.0524 0.0346 0.0238 0.0190 0.0162 0.0149 0.0130 0.0120
    0.738 0.738 0.756 0.767 0.614 0.771 0.714 0.606 0.587 0.764 0.993 0.871 0.455 0.392 0.361 0.342
0.322 ...
       0.279 0.200 0.114 0.0576 0.0314 0.0213 0.0167 0.0143 0.0129 0.0121 0.0110 0.0108 0.0109 0.0110
    0.738 0.738 0.756 0.767 0.614 0.771 0.714 0.606 0.587 0.764 0.993 0.871 0.455 0.392 0.361 0.342
0.322 ...
       0.279 0.200 0.114 0.0576 0.0314 0.0213 0.0167 0.0143 0.0129 0.0121 0.0110 0.0108 0.0109
0.0110]

        Pr = interp2(T, P, Pr_table, Tout, Pin)
```

APPENDIX F
MATLAB CODE (FUNCTION) TO COMPUTE ENTHALPY

```
%*************************************************************************
%code enthalpy
%This code compute thermodynamic property of air as function of T an P
%
%Inputs are: temperature and pressure, so T and P must be known
%output of this code is accurate but not exact solution
%*************************************************************************

%****** Input ************************************************************
%Pin     : Pressure input, this value must be known  unit:atm
%Tout    : Temperature input, this value must be known  unit:K
%*************************************************************************
function [h_known] = enthalpy (Pin, Tout)


%****** Compressibility *************************************************
%computed by using interp2 function with known Hansen's data (T and P)
%inter2 is function of 2-D data interpolation
%T       : temperature
%P       : pressure
%Tin      : input temperature
%Pin      : input pressure
%ZTP      : compressibility table
%Z       : compressibility of air as f(Tin, Pin)
%*************************************************************************

T = 0:500:15000   %Temperature range 0K --> 15000K
P = [100 10 1 0.1 0.01 0.001 0.0001 0]  %Pressure range 0atm --> 0.0001atm --> 100atm
ZTP =[1.000 1.000 1.000 1.000 1.000 1.000 1.003 1.012 1.033 1.071 1.118 1.159 ...
   1.189 1.214 1.243 1.284 1.341 1.418 1.512 1.616 1.718 1.807 1.876 1.927 1.965 ...
   1.993 2.017 2.039 2.062 2.086 2.113
   1.000 1.000 1.000 1.000 1.000 1.001 1.009 1.035 1.089 1.149 1.186 1.208 ...
   1.235 1.279 1.351 1.457 1.590 1.727 1.838 1.914 1.962 1.993 2.018 2.042 2.067 ...
   2.098 2.135 2.180 2.233 2.297 2.372
   1.000 1.000 1.000 1.000 1.000 1.004 1.026 1.092 1.165 1.196 1.214 1.248 ...
   1.316 1.437 1.607 1.778 1.896 1.959 1.993 2.018 2.042 2.071 2.111 2.163 2.232 ...
   2.318 2.426 2.553 2.700 2.861 3.028
   1.000 1.000 1.000 1.000 1.001 1.011 1.072 1.167 1.198 1.213 1.252 1.348 ...
   1.529 1.752 1.904 1.971 2.001 2.023 2.050 2.090 2.149 2.234 2.351 2.505 2.694 ...
   2.910 3.135 3.347 3.527 3.667 3.769
   1.000 1.000 1.000 1.000 1.002 1.033 1.149 1.197 1.208 1.245 1.359 1.599 ...
   1.849 1.961 1.997 2.017 2.044 2.090 2.166 2.286 2.462 2.700 2.983 3.272 3.520 ...
   3.700 3.818 3.889 3.932 3.957 3.973
   1.000 1.000 1.000 1.000 1.005 1.088 1.192 1.203 1.228 1.337 1.622 1.898 ...
   1.983 2.006 2.027 2.067 2.144 2.284 2.510 2.832 3.202 3.526 3.745 3.867 3.931 ...
   3.963 3.979 3.988 3.993 3.996 3.997
   1.000 1.000 1.000 1.000 1.016 1.163 1.200 1.211 1.287 1.577 1.910 1.990 ...
```

```
   2.008 2.032 2.088 2.210 2.446 2.826 3.282 3.645 3.843 3.932 3.969 3.985 3.993 ...
   3.996 3.998 3.999 3.999 4.000 4.000
   1.000 1.000 1.000 1.000 1.016 1.163 1.200 1.211 1.287 1.577 1.910 1.990 ...
   2.008 2.032 2.088 2.210 2.446 2.826 3.282 3.645 3.843 3.932 3.969 3.985 3.993 ...
   3.996 3.998 3.999 3.999 4.000 4.000]


%****** Temperature ***************************************************
%Z function is known, T and h relation is computed by using interp2 function with
%known Hansen's data (h, P, and Z)
%Now, P and h are known. Set P known and assume T and iterate to match h
%given by T and h relation
%T       : temperature (K)
%P       : pressure (atm)
%h       : enthalpy (J/g)
%h_known  : enthalpy known (input enthalpy) (J/g)
%Z       : compressibility
%R       : universal gas constant (J/mol-K)
%MWair    : molecular weight (g/mol)
%ZH_RT    : dimensionless enthalpy (Zh/RT) table givenin Hansen's
%ZHRT     : dimensionless enthalpy (Zh/RT) as function of (T, P)
%Tin      : input temperature (guessed temperature)
%Pin      : input pressure
%ZTP      : compressibility table
%Z       : compressibility of air as f(Tin, Pin)
%**********************************************************************

T = 0:500:15000   %Temperature range 0K --> 15000K
P = [100 10 1 0.1 0.01 0.001 0.0001 0]  %Pressure range 0atm --> 0.0001atm --> 100atm
ZH_RT = [3.52 3.52 3.65 3.80 3.92 4.01 4.13 4.34 4.70 5.20 5.73 6.13 6.38 6.62 6.95 7.44 8.16 ...
       9.10 10.20 11.36 12.42 13.23 13.77 14.08 14.22 14.28 14.30 14.31 14.34 14.40 14.49
   3.52 3.52 3.65 3.80 3.92 4.03 4.25 4.75 5.56 6.29 6.62 6.80 7.11 7.72 8.76 10.24 11.99 13.63 ...
       14.79 15.40 15.61 15.64 15.60 15.58 15.62 15.74 15.96 16.28 16.71 17.26 17.92
   3.52 3.52 3.65 3.80 3.92 4.09 4.61 5.75 6.74 6.98 7.10 7.58 8.70 10.64 13.20 15.48 16.73 17.09 17.04
...
       16.91 16.84 16.90 17.13 17.57 18.24 19.16 20.32 21.72 23.29 24.98 26.66
   3.52 3.52 3.65 3.80 3.93 4.27 5.55 7.08 7.28 7.33 7.96 9.73 12.93 16.46 18.34 18.66 18.43 18.17 ...
       18.09 18.29 18.85 19.84 21.31 23.28 25.69 28.36 30.99 33.27 34.97 36.02 36.53
   3.52 3.52 3.65 3.80 3.97 4.81 7.13 7.62 7.53 8.14 10.48 15.14 19.30 20.35 20.01 19.54 19.34 ...
       19.60 20.49 22.17 24.78 28.28 32.31 36.13 39.01 40.66 41.26 41.17 40.69 40.01 39.24
   3.52 3.52 3.65 3.80 4.07 6.16 8.02 7.77 8.09 10.55 16.68 21.58 21.97 21.24 20.69 20.72 21.65 ...
       23.85 27.66 33.00 38.79 43.28 45.57 46.09 45.64 44.74 43.69 42.61 41.55 40.53 39.57
   3.52 3.52 3.65 3.80 4.41 8.02 8.19 8.03 9.82 16.80 23.46 23.58 22.54 21.93 22.29 24.26 28.65 ...
       35.75 43.74 49.15 50.96 50.64 49.48 48.07 46.64 45.26 43.97 42.76 41.64 40.58 39.60
   3.52 3.52 3.65 3.80 4.41 8.02 8.19 8.03 9.82 16.80 23.46 23.58 22.54 21.93 22.29 24.26 28.65 ...
       35.75 43.74 49.15 50.96 50.64 49.48 48.07 46.64 45.26 43.97 42.76 41.64 40.58 39.60]

%P and h are known. Set P known and assume T and iterate to match h
%given by T and h relation above
R = 8.3144      %universal gas constant  unit:J/K-mol
MWair = 29
```

```
%cp_0 = 1.0    % specific heat constant pressure at T = 0K
%hin = cp_0*Tout            %guess temperature , unit:K
%dT = 1         % initialize difference between hi and h_known
%while dh < 0
   Z = interp2(T, P, ZTP,Tout, Pin)
   ZHRT = interp2(T, P, ZH_RT,Tout, Pin)
   h_known = ZHRT*R*Tout/Z/MWair  %enthalpy i  unit:J/g
   %dh = h_known - hi
   %Tin = Tin - 10
   %end
%Tout = Tin + 10 %unit:K
```

APPENDIX G
MATLAB CODE (FUNCTION) TO COMPUTE MACH NUMBER FOR PRANDTL-MAYER EXPANSION

```
function [M2] = shadowM (i,jj,u_eunit, T_e, u_e, gamma, ep, Rair)

%function [T_e, i] = temp(i,jj,nunit, T_e, u_e, gamma, ep, Rair)
%This function will compute the Mach number (M2) of shadow panel
%from the data of the panel in front of shadow panel

      j = jj(i,1)
          PMang_guess =0
          aaa= [u_eunit(j,1) u_eunit(j,2) u_eunit(j,3)]; bbb= [u_eunit(i,1) u_eunit(i,2)
u_eunit(i,3)]
          ccc = cross(aaa,bbb)  %a:unit normal vector of jth panel; b:unit vector of ith panel
          cc = (ccc(1)^2+ccc(2)^2+ccc(3)^2)^0.5
          aa = (aaa(1)^2+aaa(2)^2+aaa(3)^2)^0.5
          bb = (bbb(1)^2+bbb(2)^2+bbb(3)^2)^0.5
          psi = asin(cc/(aa*bb))  %deflection angle (rad)
          M1 = ((u_e(j,1)^2+u_e(j,2)^2+u_e(j,3)^2)/(gamma*Rair*T_e(j,1)))^0.5 %Mach
number of j panel
          PMang2 = psi + (ep^(-0.5))*atan((ep*(M1^2-1))^0.5)-atan((M1^2-1)^0.5)
          maxPMang = (((gamma+1)/(gamma-1))^0.5-1)*3.14159265359/2
       if PMang2 >= maxPMang
          M2 = 1000
          else
          M2 = M1-0.01  %M2 is Mach number of i panel and this is initial guess
          while PMang2 > PMang_guess
             PMang_guess = (ep^(-0.5))*atan((ep*(M2^2-1))^0.5)-atan((M2^2-1)^0.5)
             M2 = M2 + 0.01

             %aaa
             %bbb
             %ccc
             %aa
             %bb
             %cc
             %M1
             %PMang2
          end
       end
  M2 =M2-0.01
```

LIST OF REFERENCES

1. Liepmann, H. W., Roshko, A.: *Elements of Gasdynamics*, Wiley, New York, 1957

2. Anderson, J. D.: *Hypersonic and High Temperature Gas Dynamics*, McGraw-Hill, New York, 1989

3. Oosthuizen, P. H., Carscallen, W. E.: *Compressible Fluid Flow*, McGraw-Hill, New York, 1997

4. Hansen, C. F.: *Approximations for the Thermodynamic and Transport Properties of High-Temperature Air*, NASA Technical Report R-50, 1959

5. Schetz, J. A.: *Boundary Layer Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1993

6. Fay, J. A. and Riddell, F. R.: "Theory of Stagnation Point Heat Transfer in Dissociated Air", Journal of the Aeronautical Sciences, vol.25, no.2, February, 1958, pp. 73-85

7. Lees, L.: "Laminar Heat Transfer over Blunt-Nosed Bodies at Hypersonic Flight Speeds", Jet Propulsion, vol.26, 1956, pp. 259-269, 274

8. Phillips, W. F.: *Mechanics of Flight*, John Wiley & Sons, Inc., Hoboken, NJ, 2004

9. Wannernwetsch, G. D.: "Pressure Tests of the AFFDL X-24C-10D Model at Mach Number of 1.5, 3.0, 5.0 and 6.0," von Karman Gas Dynamics Facility, Arnold Engineering Development Center, TN, AEDC-DR-76-92, Nov. 1976.

10. Neumann, R. D., Patterson, J. L., and Sliski, N. J.: "Aerodynamic Heating to the Hypersonic Research Aircraft X24C," AIAA Paper 78-37, 1978.

11. Shang, J. S., Scherr, S. J.: "Navier-Stokes Solution for a Complete Re-Entry Configuration," Journal of Aircraft, vol.23, No.12, Dec. 1986, pp. 881-888

12. Hamilton J. T., Wallace R. O., Dill C.C., "Launch vehicle aerodynamic data base development comparison with flight data," NASA CP-2283, Pt. 1, 1983, pp. 19-36

13. Hwang, D. P.: "A Proof of Concept Experiment for Reducing Skin Friction By Using a Micro-Blowing Technique," 35th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 6-9, 1997.

14. Kays, W. M., Crawford, M. E.: *Convective Heat and Mass Transfer*, McGraw-Hill, New York, 1980

BIOGRAPHICAL SKETCH

Yoshifumi Nozaki was born in Kochi, Japan, in 1981. He graduated from Tosa High School in 2000. He received his Bachelor of Science in Mechanical Engineering from Utah State University in May of 2004. He entered the graduate program at the University of Florida in August of 2005 under Dr. Pasquale M. Sforza in the Department of Mechanical and Aerospace Engineering to obtain his Master of Science.